User's Manual

SuperH™ファミリ マルチコア マイコン用 E10A-USB エミュレータ

ユーザーズマニュアル

SuperH™ファミリ E10A-USB
HS0005KCU04HJ

本資料に記載の全ての情報は本資料発行時点のものであり、ルネサス エレクトロニクスは、
予告なしに、本資料に記載した製品または仕様を変更することがあります。
ルネサス エレクトロニクスのホームページなどにより公開される最新情報をご確認ください。

ルネサス エレクトロニクス
www.renesas.com

Rev.5.01 2016.03

ご注意書き

1. 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・システムの設計において、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て、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。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。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。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等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。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、
   各品質水準は、以下に示す用途に製品が使用されることを意図しております。

   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、産業用ロボット等

   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、各種安全装置等

   当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図し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損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の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。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、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、社会的損害等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、お客様の機器・システムとしての出荷保証を行ってください。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、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 本資料に記載されている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・システムに使用することはできません。また、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に使用しないでください。当社製品または技術を輸出する場合は、「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。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
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禁じます。

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

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

規制に関する情報

● European Union regulatory notices

This product complies with the following EU Directives. (These directives are only valid in the European Union.)

CE Certifications:

• Electromagnetic Compatibility (EMC) Directive 2014/30/EU

EN 55022:2010 Class A

WARNING: This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

EN 55024:2010

• Information for traceability
  • Authorised representative
    Name: Renesas Electronics Corporation
    Address: Toyosu Foresia, 3-2-24, Toyosu, Koto-ku, Tokyo 135-0061, Japan
  • Manufacturer
    Name: Renesas System Design Co.,Ltd.
    Address: 5-20-1, Josuihon-cho, Kodaira-shi, Tokyo 187-8588, Japan
  • Person responsible for placing on the market
    Name: Renesas Electronics Europe GmbH
    Address: Arcadiastrasse 10, 40472 Dusseldorf, Germany
  • Trademark and Type name
    Trademark: Renesas
    Product name: E10A-USB Emulator
    Type name: HS0005KCU04H / HS0005KCU14H

Environmental Compliance and Certifications:

• Restriction of the Use of Certain Hazardous Substances in Electrical and Electronic Equipment (RoHS) Directive 2002/95/EC

• Waste Electrical and Electronic Equipment (WEEE) Directive 2002/96/EC

● United States Regulatory notices

This product complies with the following EMC regulation. (This is only valid in the United States.)

FCC Certifications:

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

CAUTION: Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

重要事項

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

- エミュレータとは:
  ここで言うエミュレータとは、ルネサス エレクトロニクス株式会社(以下、「ルネサス」という)が製作した次の製品を指します。
  (1)エミュレータ、(2)ユーザインタフェースケーブル
  お客様のホストコンピュータ及びユーザシステムは含みません。

- エミュレータの使用目的:
  当エミュレータは、ルネサスマイクロコンピュータを使用したシステムの開発を支援する装置です。
  ソフトウェアとハードウェアの両面から、システム開発を支援します。
  この使用目的にしたがって、当エミュレータを正しくお使いください。
  この目的以外の当エミュレータの使用を堅くお断りします。

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

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

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

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

- エミュレータを使う人は:
  当エミュレータは、ユーザーズマニュアルをよく読み、理解した人のみが使ってください。
  特に、当エミュレータを初めて使う人は、当エミュレータをよく理解し、使い慣れている人から指導を受けることを
  お薦めします。

- 保証の範囲：
  ルネサスは、お客様が製品をご購入された日から1年間は、無償で故障品を交換いたします。ただし、
  （1）製品の誤用、濫用、またはその他異常な条件下での使用
  （2）ルネサス以外の者による改造、修理、保守、またはその他の行為
  （3）ユーザシステムの内容、または使用
  （4）火災、地震、またはその他の事故
  により、故障が生じた場合は、有償で交換を行います。
  また、日本国内で購入され、かつ、日本国内で使用されるものに限ります。

- その他の重要事項：

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

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

- 図について：
  このユーザーズマニュアルの一部の図は、実物と違っていることがあります。

- デバイス名について：
  このユーザーズマニュアルでは、例として、SHxxxxというデバイス名を使用しています。

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

安全事項

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

シグナル・ワードの定義

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

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

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

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

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

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

⚠ 警告

1. **感電、火災等の危険防止および品質保証のために、お客様ご自身による修理や改造は行わないでください。故障の際のアフターサービスにつきましては、ルネサス販売または特約店にお申し付けください。**

2. **ホストコンピュータまたはユーザシステムのパワーオン時、すべてのケーブル類の抜き差しを行わないでください。抜き差しを行った場合、エミュレータとユーザシステムの発煙、発火の可能性があります。また、デバッグ中のユーザプログラムを破壊する可能性があります。**

3. **ユーザインタフェースケーブルのユーザシステム上側のコネクタとユーザインタフェースケーブル側のコネクタの向きを確かめて正しく接続してください。接続を誤るとエミュレータとユーザシステムの発煙、発火の可能性があります。**

注意

廃棄に関して：

廃棄する時は必ず産業廃棄物として法令に従って処分してください。

European Union regulatory notices:

The WEEE (Waste Electrical and Electronic Equipment) regulations put responsibilities on producers for the collection and recycling or disposal of electrical and electronic waste. Return of WEEE under these regulations is applicable in the European Union only. This equipment (including all accessories) is not intended for household use. After use the equipment cannot be disposed of as household waste, and the WEEE must be treated, recycled and disposed of in an environmentally sound manner.
Renesas Electronics Europe GmbH can take back end of life equipment, register for this service at “ http://www.renesas.eu/weee ”.

エミュレータ使用時の注意事項

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

⚠ 警告

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

注意

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

このマニュアルについて

このマニュアルは、エミュレータ使用前の準備、エミュレータの機能、エミュレータ固有のデバッグ機能、チュートリアル、エミュレータのハード仕様、エミュレータのソフトウェア仕様を説明しています。

High-performance Embedded Workshop の基本的な使い方に関する情報、環境のカスタマイズ、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。

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

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

ユーザ登録について

ルネサスエレクトロニクスでは、ツール製品のユーザ登録をご購入されたお客様にお願いしています。ご登録いただくと、新製品のリリース、バージョンアップ、使用上の注意事項などをまとめたツールニュースを電子メールで受け取ることができます。

詳しくは、下記の「ツール製品のユーザ登録のご案内」をご覧ください。
[ツール製品のユーザ登録のご案内]　　http://japan.renesas.com/registertool_index

ご登録は、下記の My Renesas から登録してください。
[My Renesas]　　http://japan.renesas.com/myrenesas

ご登録いただいた内容は、アフターサービスの情報としてのみ利用させていただきます。
ご登録なき場合は、フィールドチェンジ、不具合情報の連絡など保守サービスが受けられなくなりますので、必ずご登録をお願いします。

目次

1.　はじめに

本システムは、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。

E10A-USB エミュレータの本体は、専用デバッグインタフェースを経由して、ユーザシステムに接続します。このため完成した製品に近い形態でデバッグを行うことができます。

また、USB1.1/2.0(Full-Speed)を搭載しているパーソナルコンピュータ（IBM PC 互換機）をホストコンピュータにして実験室、フィールドと場所を選ばずデバッグを行うことができます。

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

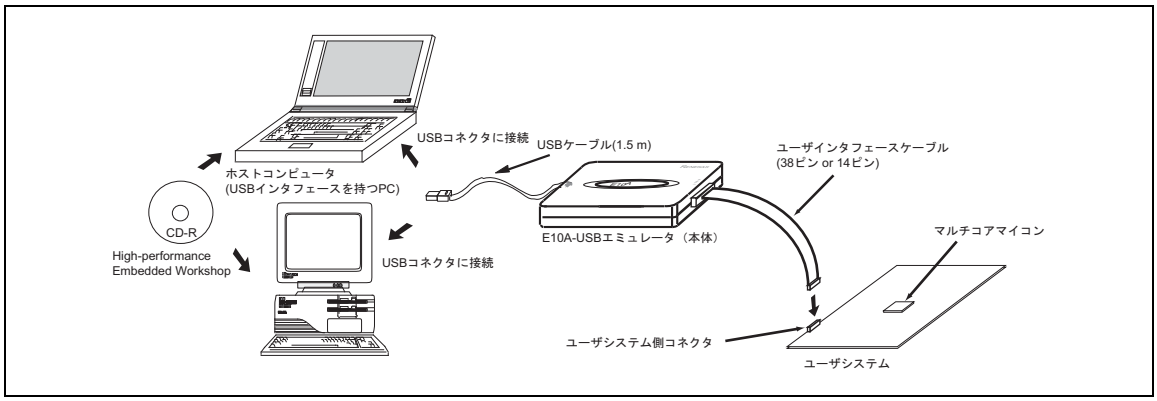


図 1.1　E10A-USB エミュレータを使用したシステム構成外観

E10A-USB エミュレータの特長は、以下の通りです。

(1) コストパフォーマンスに優れたエミュレータ
小型サイズ、USB 接続を実現しました。

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

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

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

(5) 製品形態でのユーザシステムのデバッグ
ユーザシステム完成時の製品形態に近い状態でユーザシステムのデバッグを行うことができます。

(6) コンパクトなデバッグ環境
ノート型パソコンをホストコンピュータとして使用でき、場所を選ばずデバッグ環境を作成することができます。

(7) AUDトレース機能【注】
AUD トレース機能により、大容量のリアルタイムトレースが可能です。

【注】 AUD とは、Advanced User Debugger の略です。機種によっては、サポートしていない製品があります。

1.1　使用上の注意事項

注意

E10A-USB エミュレータをお使いになる前に、以下の注意事項を必ず確認してください。
誤った使い方は、E10A-USB エミュレータ、ユーザプログラムおよびユーザシステムの破壊につながります。

(1) 製品を梱包箱から取り出し、納入品明細書に示されているものがそろっているか、確認してください。
(2) 製品に重量物を上積みするなどして、無理な力を加えないでください。
(3) 製品に過大な物理的衝撃を与えないでください。「1.2　使用環境条件」を参照してください。
(4) ホストコンピュータまたはユーザシステムの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。
(5) ケーブルを接続した後は、接続位置が正しいことを再度確認してください。接続方法については、「3　使用前の準備」を参照してください。
(6) すべてのケーブルを接続し終えてから、接続した各装置へ電源を投入してください。また、電源が入っているときにケーブルの接続および取り外しをしないでください。

1.2　使用環境条件

注意

E10A-USB エミュレータを使用する場合、表 1.1、および表 1.2 に示す条件を守ってください。
この条件を満たさない状態で E10A-USB エミュレータを使用した場合、E10A-USB エミュレータ、ユーザプログラムおよびユーザシステムが正常に動作しない場合があります。

表 1.1　使用環境条件

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

表 1.2　動作環境

項番	項目	動作環境	
1	オペレーティングシステム	Windows® XP　(32 ビット版)	Windows Vista®　(32 ビット版)、 Windows® 7　(32 ビット版/64 ビット版)
2	ホストコンピュータ	USB1.1/2.0(Full-Speed)を備えた IBM PC およびその互換機	
3	CPU	Core™ 2 Duo　2GHz 以上を推奨	Core™ 2 Duo　3.16GHz 以上を推奨
4	メモリ容量	推奨 1GB 以上（＋ロードモジュールのファイルサイズの 10 倍以上）	推奨 2GB 以上（＋ロードモジュールのファイルサイズの 10 倍以上）
5	ハードディスク容量	インストールディスク容量 600MB 以上 （スワップ領域を考慮してメモリ容量の 2 倍以上（推奨 4 倍以上）の空き容量をご用意ください。）	
6	マウスなどのポインティングデバイス	ホストコンピュータ本体に接続可能で Windows®XP および Windows Vista®、Windows®7 に対応している、マウスなどのポインティングデバイス	
7	ディスプレイ	モニタ解像度　1024 × 768 以上	
8	電源電圧	5.0±0.25V（USB バスパワータイプ）	
9	消費電流	500mA(max)	
10	CD-ROM ドライブ	E10A-USB エミュレータ用 High-performance Embedded Workshop をインストールするため、または E10A-USB エミュレータユーザーズマニュアルを参照するために必要	

1.3　梱包品の確認

梱包を解いた後、梱包品がそろっているか確認してください。E10A-USB エミュレータの梱包品は、別冊の「SHxxxx ご使用時の補足説明」の 1.1 章を参照してください。
確認した結果、梱包品に不足がありましたら、ルネサス エレクトロニクス販売または特約店、ルネサスエレクトロニクス コンタクトセンタ（csc@renesas.com）までご連絡ください。

2.　E10A-USB エミュレータ機能

本章では、E10A-USB エミュレータの機能を紹介します。
E10A-USB エミュレータがサポートするデバイスにより、多少機能が異なります。
各機能の使用方法は、「6　チュートリアル［SH-2A 編］」または「7　チュートリアル［SH-4A 編］」をご参照ください。

2.1　機能概要

E10A-USB エミュレータの機能概要を表 2.1 に示します。
製品ごとの機能については、オンラインヘルプを参照してください。

表 2.1　E10A-USB エミュレータの機能

項番	項目	機能
1	ユーザプログラム実行系機能	・デバイスが保証する範囲の動作周波数による、プログラム実行 ・リセットエミュレーション ・Step 機能 シングル Step 機能（1Step：1 命令） ソースレベル Step 機能（1Step：ソース 1 行） Step Over 機能（サブルーチン内はブレークしない） Step Out 機能（PC 実行中のサブルーチンの呼び出し元関数に戻るまで実行） ・同期機能 同期実行機能（一方の CPU の実行に同期して全 CPU 同時に実行） 同期ステップ機能（一方の CPU のステップに同期して全 CPU が実行） 同期ブレーク機能（一方の CPU のブレークにより同期して全 CPU がブレーク）
2	リセット機能	・ブレーク中、High-performance Embedded Workshop からデバイスへパワーオンリセット発行
3	トレース機能	・デバイス内蔵のトレース機能 ・AUD トレース 分岐トレース、メモリアクセストレース ・トレースデータのメモリ出力機能
4	ブレーク機能	・ハードウェアブレーク条件（条件、本数はデバイスにより異なる） ・PC ブレーク条件（255 箇所） ・強制ブレーク機能
5	パフォーマンス測定機能	マイコン内蔵のカウンタにより、Point to Point の経過サイクル数などを測定する機能

項番	項目	機能
6	メモリアクセス機能	・RAM へのダウンロード ・フラッシュメモリへのダウンロード ・1 行アセンブル ・逆アセンブル ・メモリリード ・メモリライト ・ユーザプログラム実行中の変数内容の表示自動更新 ・FILL ・サーチ ・ムーブコピー ・モニタ機能（物理アドレス）
7	汎用/制御レジスタアクセス機能	・汎用/制御レジスタのリード/ライト
8	内蔵 I/O レジスタアクセス機能	・内蔵 I/O レジスタのリード/ライト【注】
9	ソースレベルデバッグ機能	・豊富なソースレベルデバッグ機能。
10	コマンドライン機能	・コマンド入力をサポートしています。 各コマンドを入力順に羅列したファイルを作成すれば、バッチ処理を行うこともできます。
11	ヘルプ機能	・各機能の操作方法や、コマンドラインウィンドウから入力できるコマンドのシンタックスを記載しています。

【注】 [IO]ウィンドウは、”SHxxxx.io”に定義されている内容を表示しています。”SHxxxx.io”の内容を編集することにより、表示するレジスタを追加／削除することができます。
”SHxxxx.io”に記載すべき内容については、High-performance Embedded Workshop V.4.09 ユーザーズマニュアル「リファレンス　6　I/O ファイルフォーマット」を参照してください。
また、”SHxxxx.io”は以下ディレクトリ内にあります。
（xxxx はエミュレータデバイスグループ名を示します）
＜High-performance Embedded Workshop フォルダ＞
¥Tools¥Renesas¥DebugComp¥Platform¥E10A-USBM¥xxxx¥IOFiles

次の章から、E10A-USB エミュレータの特徴的な機能について説明します。

2.2　トレース機能

E10A-USB エミュレータには 2 種類のトレース機能があります。

2.2.1　内蔵トレース機能

分岐元／分岐先アドレスと、ニモニック、オペランド、ソース行を表示します。
デバイスに内蔵されているトレースバッファを使用して実現します。

【留意事項】

1. **トレース取得できる分岐命令の数、トレース表示内容は、製品によって異なります。各製品の仕様については、オンラインヘルプを参照してください。**
2. **製品によっては、内蔵トレース機能はサポートしておりません。各製品の仕様については、オンラインヘルプを参照してください。**
3. **製品によっては、内蔵トレース機能が拡張されています。各製品の仕様については、オンラインヘルプを参照してください。**

2.2.2　AUD トレース機能

デバイスの AUD 端子を E10A-USB エミュレータに接続している場合に有効な、大容量のトレース機能です。トレース取得するイベントが発生した場合、AUD 端子からリアルタイムにトレース情報が出力されます。
トレース取得できるイベントの数は、分岐元／分岐先の組を 1 個とすると最大 32,767 個です。

(1) トレース取得イベント
AUD トレース機能では、以下に示すイベントを取得することができます。

(a) 分岐発生情報
分岐元／分岐先アドレスを取得します。

(b) 指定範囲内メモリアクセス情報
指定した範囲内のメモリアクセスをトレース取得します。
メモリ範囲は 2 つまで指定できます。チャネル A、チャネル B にそれぞれ範囲を指定することができます。またそれぞれトレース取得するバスサイクルとして、リードサイクル、ライトサイクル、またはリードライトサイクルを選択できます。
本機能は、以降、「ウィンドウトレース機能」と呼びます。

(c) ソフトウェアトレース

特殊な命令を実行した場合に、実行時の PC 値と 1 つの汎用レジスタ内容をトレース取得します。
あらかじめ、C ソース上に Trace(x)関数（x は変数名）を記述し、コンパイル、
リンクしてください。詳細は SHC/C++コンパイラマニュアルを参照してください。
ロードモジュールを E10A-USB エミュレータにロードし、ソフトウェアトレース
機能を有効にして実行すると、Trace(x)関数を実行した PC 値と、x に対応する変数の値と、
ソースが表示されます。

【留意事項】

トレース取得できるイベントの種類は、製品によって異なります。
各製品の仕様についてはオンラインヘルプを参照してください。

(2)　トレース取得モード

AUD トレース機能では、トレースを取得する際の取得方法において以下のモードを持っています。

表 2.2 に、AUD トレースのトレース取得モードを示します。

表 2.2　AUD トレース取得モード

種別	モード	説明
トレース出力が連続して発生した場合の取得モード	Realtime trace モード	トレース情報の発生が集中し、AUD 端子からの出力が間に合わなくなると CPU はトレース情報の出力を一時的に停止します。このため、ユーザプログラムはリアルタイムに動作しますが、トレース情報が一部取得できないことがあります。
	Non realtime trace モード	トレース情報の発生が集中し、AUD 端子からの出力が間に合わなくなると CPU の動作を一時的に停止し、トレース情報の出力を優先します。このため、ユーザプログラムのリアルタイム性がなくなります。
E10A-USB エミュレータのトレースバッファがフルになった場合の取得モード	Trace continue モード	古い情報に新しい情報を上書きして、常に最新の情報を取得します。
	Trace stop モード	その後のトレースを取得しません。ユーザプログラムは継続して実行されます。

(3)　トレース表示内容

プログラムの実行停止後、[トレース]ウィンドウにトレース結果を表示します。
[トレース]ウィンドウへは、以下を表示します。

PTR　：トレースバッファ内ポインタ（最後に実行した命令が +0 となります）
IP　　：一番最近のトレース情報から遡って、いくつ前の情報であるかを示します。
　　分岐命令は、分岐元と分岐先で 1 つとカウントします。
Type　　：トレース取得情報の種類を表示します。
Address　：トレース取得アドレスを表示します。
Data　：トレース取得データを表示します。データのない情報に関しては、「********」を表示します。
Instruction、Source、Label：
　　トレース取得アドレスのニモニック、該当するソースと、ラベル情報を表示します。「Source」カラムをダブルクリックすると、[エディタ]ウィンドウの該当個所にカーソルが移動します。

また、Type 列、Address 列、Data 列は、選択されている AUD トレース種別によってそれぞれ以下の意味を持ちます。

表 2.3　[トレース]ウィンドウ表示内容

トレース種別	Type 列	Address 列	Data 列
分岐トレース	BRANCH	分岐元アドレス	表示なし
	DESTINATION	分岐先アドレス	表示なし
ウィンドウトレース【注 1】	MEMORY	メモリアクセスアドレス	メモリアクセスデータ
ソフトウェアトレース【注 1】	S_TRACE	Trace(x)関数実行アドレス	変数 x データ
データロスト【注 1】【注 2】	LOST	表示なし	表示なし
CPU ウエイト発生【注 1】【注 2】	CPU-WAIT	表示なし	表示なし

【注】 1.　内蔵トレースでは表示しません。
　　2.　デバッグ対象のデバイスによっては、[Lost]、[CPU-WAIT]が出力されません。
　　このため、トレースデータの出力が間に合わなかったことや、トレースデータ出力のためにCPU がウエイトを発生したことがわかりませんので、ご了承ください。

デバッグ対象デバイスによっては、以下の項目を表示するものがあります。
各製品の仕様については、別冊の「SHxxxx ご使用時の補足説明」、またはオンラインヘルプを参照してください。

PTR　　：トレースバッファ内ポインタ（最後に実行した命令が +0 となります）
IP　　：一番最近のトレース情報から遡って、いくつ前の情報であるかを示します。
分岐命令は、分岐元と分岐先で 1 つとカウントします。
Master　：アクセスを行ったバスマスタの種別
Type　　：トレース取得情報の種類を表示します。
Branch Type：分岐種別（分岐トレースの場合のみ表示されます。）
AUD トレースでは、PPC オプションを有効にした場合のみ表示されます。
Bus：どこのバスに対するアクセスであるかを表示
R/W：発生したデータアクセスが、リードアクセスかライトアクセスかを表示
Address　：トレース取得アドレスを表示します。
Data　　：トレース取得データを表示します。
PPC　　：パフォーマンスカウンタ出力
Instruction、Source、Label：
トレース取得アドレスのニモニック、該当するソースと、ラベル情報を表示します。「Source」カラムをダブルクリックすると、[エディタ]ウィンドウの該当個所にカーソルが移動します。

また、Type 列、BUS 列、R/W 列、Address 列、Data 列は、選択されているトレース種別によってそれぞれ以下の意味を持ちます。

表 2.4　[トレース]ウィンドウ表示内容

トレース種別	Type 列	BUS 列	R/W 列	Address 列	Data 列
分岐トレース	BRANCH【注 1】	表示なし	表示なし	分岐元アドレス【注 1】	表示なし
	DESTINATION	表示なし	表示なし	分岐先アドレス	表示なし
範囲内メモリアクセストレース	MEMORY	アクセス対象バス	READ/WRITE	メモリアクセスアドレス	メモリアクセスデータ【注 1】
ソフトウェアトレース	S_TRACE	表示なし	表示なし	Trace(x)関数実行アドレス	変数 x データ
システムバストレース	MEMORY	表示なし	READ/WRITE	メモリアクセスアドレス	メモリアクセスデータ (ライトのみ) 【注 1】
データロスト【注 2】	LOST	表示なし	表示なし	表示なし	表示なし
CPU ウエイト発生【注 1】	CPU-WAIT	表示なし	表示なし	表示なし	表示なし

【注】 1.　PPC オプション使用時は表示されません。
2.　デバッグ対象のデバイスによっては、[Lost]、[CPU-WAIT]が出力されません。
このため、トレースデータの出力が間に合わなかったことや、トレースデータ出力のために CPU がウエイトを発生したことがわかりませんので、ご了承ください。

2.2.3　トレースデータのメモリ出力機能

デバッグ対象のデバイスによっては、トレースデータを指定したメモリ範囲に書き出すことができます。トレースウィンドウには書き出したメモリ範囲よりデータを読み出し、結果を表示します。

【留意事項】

指定した範囲のメモリの上書きを行うため、プログラム領域を指定しないでください。

2.2.4　トレースウィンドウの便利な機能

トレースウィンドウでは、以下の便利な機能をサポートしています。

(1)指定データの検索

(2)指定データの抽出

(3)指定データをフィルタリングして再表示

(4)分岐先アドレスから、次の分岐元アドレスまでの情報補填

これらの機能の使用方法については、「5.7　トレース情報を見る」を参照してください。

(5)ユーザプログラム実行中のトレース設定内容変更

デバッグ対象のデバイスによっては、トレースの設定をユーザプログラム実行中に変更することができます。

各製品の仕様については、オンラインヘルプを参照してください。

2.3　ブレーク機能

E10A-USB エミュレータでは、以下の 3 種類のブレーク機能があります。

(1)　ハードウェアブレーク機能

デバイス内蔵のブレークコントローラを使用した機能です。
アクセスアドレス条件、命令フェッチアドレス条件、データ条件、バスサイクル条件などが設定できます。
アドレス条件はすべて論理アドレスです。
また、[エディタ]ウィンドウや[逆アセンブリ]ウィンドウの[Event]カラムからも設定できます。設定方法は、「5.3　プログラムをダウンロードする」を参照してください。

【注】　デバッグ対象のデバイスによっては、ハードウェアブレークの設定をユーザプログラム実行中に変更することができます。各製品の仕様については、オンラインヘルプを参照してください。

(2)　PCブレーク機能（BREAKPOINT）

指定アドレスの命令を専用命令に置きかえることでブレークする機能です。
メモリへのライトが発生するため RAM 領域および内蔵フラッシュメモリ以外の場所には設定できません。
本機能は、[Event]ダイアログボックスの[Breakpoint]ページで設定できます。
また、[エディタ]ウィンドウや[逆アセンブリ]ウィンドウ上で、設定したい行の[S/W ブレークポイント]カラムをダブルクリックすることによっても設定できます。

(3) 強制ブレーク機能

ユーザプログラムを強制的にブレークする機能です。

2.4　パフォーマンス測定機能

E10A-USB エミュレータには、パフォーマンスを測定する機能として、以下があります。

2.4.1　Point to Point の経過サイクルなどを測定する機能

本機能は、デバイス内蔵のカウンタにより、指定条件成立時から指定条件成立時までに要したサイクル数などを測定する機能です。

サポートデバイスによって、サイクル数だけでなく、キャッシュミスの回数や、TLB ミスの回数など、いろいろな項目を測定することもできます。

各製品の仕様は、オンラインヘルプを参照してください。

【留意事項】

測定できる項目は、製品によって異なります。また、製品によっては、本機能はサポートしておりません。各製品の仕様については、オンラインヘルプを参照してください。

2.5　メモリアクセス機能

E10A-USB エミュレータには以下のメモリアクセス機能があります。

(1) メモリリード/ライト機能

[メモリ]ウィンドウ：メモリ内容をウィンドウ表示します。

[メモリ]ウィンドウ OPEN 時に指定したサイズのみリードします。
エミュレータ内にキャッシュを持っていないため、常にリードサイクルが発生します。
また、[メモリ]ウィンドウからライトした場合は、ウィンドウの更新のために、 [メモリ]ウィンドウで表示されている範囲のリードが発生します。
[メモリ]ウィンドウを更新したくない場合、ポップアップの[表示固定]メニューで更新しない設定にすることができます。

me コマンド：　コマンドライン機能です。
指定アドレスを指定サイズでリード、ライトする機能です。

(2) ユーザプログラムのダウンロード機能

ワークスペース内に登録されたロードモジュールをダウンロードできます。
[デバッグ]メニューの[ダウンロード]で、ダウンロードするモジュールを選択できます。
また、ワークスペース内のロードモジュールを右クリックすることによって、ポップアップメニューが開きますが、このポップアップメニューからもダウンロードを行うことができます。
ダウンロード先は、RAM またはデバイス内蔵のフラッシュメモリです。
デバイス内蔵以外のフラッシュメモリへダウンロードする場合、[基本設定]メニューの[エミュレータ]を選択して[Configuration]ウィンドウを開き、 [Loading flash memory]ページで必要な設定を行ってください。
本機能では、デバッグ情報などソースレベルデバッグに必要な情報もダウンロードします。

(3) メモリデータのアップロード機能

指定アドレスから指定サイズ分、ファイルに保存することができます。

(4) メモリデータのダウンロード機能

ファイルに保存されているメモリ内容をダウンロードできます。
[メモリ]ウィンドウのポップアップメニューから[ロード]を選択してください。

(5) 変数内容表示

ユーザプログラムの指定した変数の内容を表示します。

(6) モニタ機能

デバッグ対象のデバイスによっては、ユーザプログラム実行中にメモリ内容をモニタすることができます。
各製品の仕様についてはオンラインヘルプを参照してください。

(7) そのほかのメモリ操作機能

その他、以下の機能があります。

- メモリフィル機能
- メモリコピー機能
- メモリセーブ機能
- メモリベリファイ機能
- メモリサーチ機能
- 内蔵 I/O 表示機能
- キャッシュテーブル表示、編集機能（キャッシュ内蔵デバイスのみ）
- TLB テーブル表示、編集機能（MMU 内蔵デバイスのみ）
- ラベル名、変数名とその内容を表示する機能

詳細につきましてはオンラインヘルプを参照してください。

【留意事項】

1. ユーザプログラム実行中のメモリアクセス

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

   参考値として、以下の環境でのユーザプログラムの停止時間を示します。
   環境
   ホスト PC：Core™2 CPU T7600 2.33 GHz
   SH7265：CPU クロック　66.6 MHz
   JTAG クロック：2.5 MHz
   コマインドラインウィンドウから 1 バイトメモリリードを行った場合、停止時間は約 70 ms になります。

2. ユーザプログラムブレーク中のメモリアクセス

   E10A-USB エミュレータは、フラッシュメモリ領域に対してもダウンロードすることができます。
   しかし他のメモリライト操作は RAM 領域および内蔵フラッシュメモリに対してのみ可能です。
   したがって、メモリライト、BREAKPOINT 等の設定は RAM 領域および内蔵フラッシュメモリのみに行ってください。
   また、MMU によりメモリ空間がリードのみ可能となっている場合にも、メモリライト、BREAKPOINT 設定、ダウンロード等の操作は行わないでください。

2.6　スタックトレース機能

E10A-USB エミュレータでは、スタック情報を用いて、現在の PC がある関数がどの関数からコールされているかを表示します。本機能は、Elf/Dwarf2 形式のデバッグ情報を持ったロードモジュールをロードした場合のみ使用できます。

本機能の使用方法については、「6.21、7.22　スタックトレース機能」を参照してください。

2.7　ブレーク中のユーザ割込開放機能

デバッグ対象のデバイスによっては、エミュレーション実行中の割り込みはすべてユーザに開放しています。ユーザプログラムブレーク中の場合、割り込み処理を実行するモードか、しないモードかを指定することができます。

2.8　オンラインヘルプ

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

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

3.　使用前の準備

3.1　E10A-USB エミュレータ使用フローチャート

E10A-USB エミュレータを使用するにあたって、梱包を解いた後下記の手順で準備を行ってください。

⚠ 警告

準備を行う前に図 3.1 中のアミのかかっている参照先をすべてよく読んで理解してください。誤った使い方は、E10A-USB エミュレータ、ユーザプログラムおよびユーザシステムの破壊につながります。

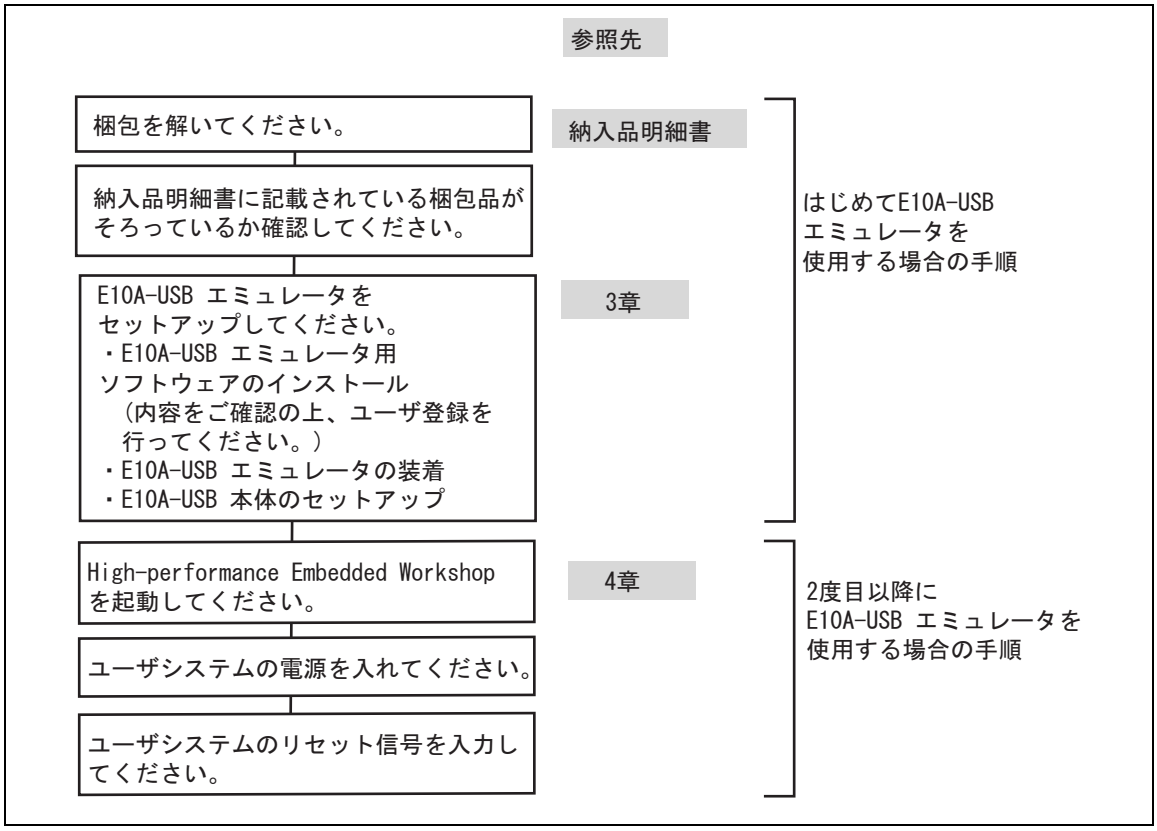


図 3.1　E10A-USB エミュレータ使用フローチャート

3.2　E10A-USB　エミュレータのハードウェア構成

E10A-USB　エミュレータは、図 3.2 に示すように E10A-USB エミュレータ本体、USB ケーブル、ユーザインタフェースケーブルで構成され、ホストコンピュータとは USB 1.1 で接続できます。また、USB2.0 準拠の USB ポートにも接続できます。

図 3.2　E10A-USB エミュレータのハードウェア構成（38 ピンケーブル時）

E10A-USB　エミュレータにおける各部の名称を下記に示します。

（1）上面部の構成

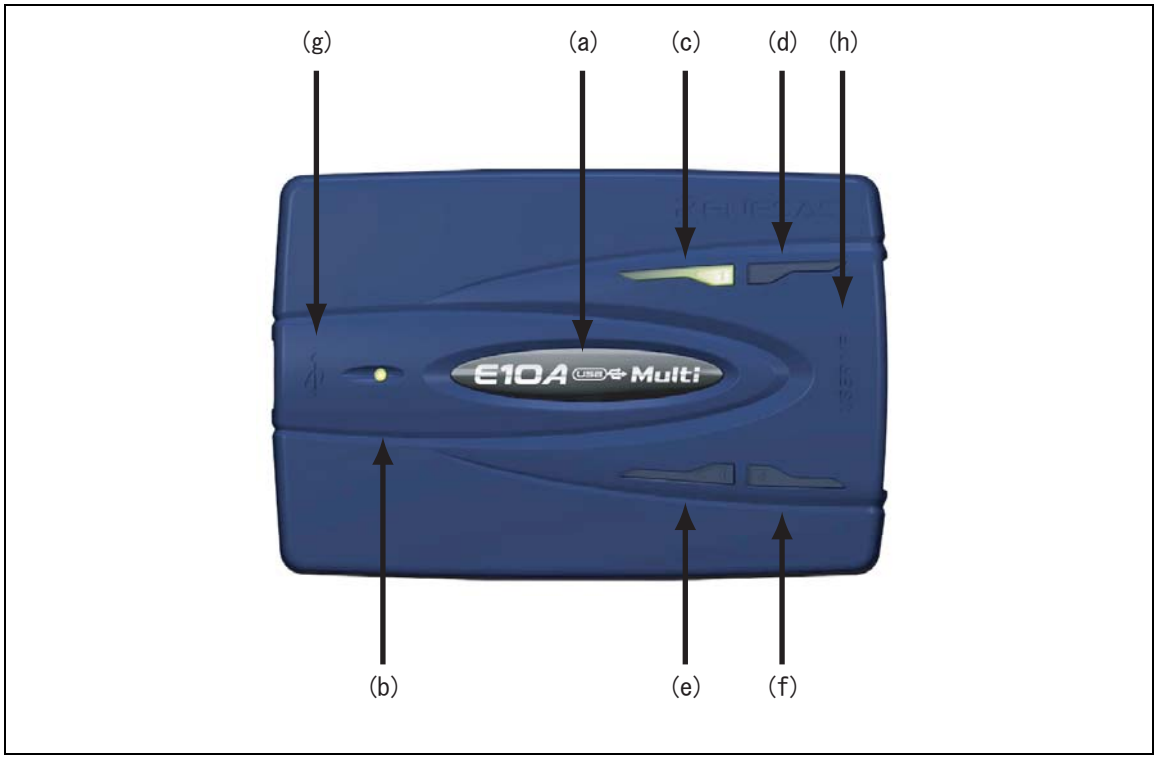


図 3.3　E10A-USB エミュレータの上面部

(a) E10A-USB ロゴプレート	：他の E シリーズエミュレータと簡単に識別できるように、黒色の E10A-USB エミュレータ専用プレートが張付けられています。
(b) ACTION LED 表示	：円形の LED です。点灯時は E10A-USB 制御ソフトウェアが動作していることを示しています。
(c) RUN LED 表示	：“1”と表示してある LED です。点灯時はユーザプログラムが実行状態にあることを示しています。
(d) ACT LED 表示	：“2”と表示してある LED です。点灯時は E10A-USB がマイコンと通信していることを示しています。
(e) コア切り替え LED 表示	：“3”と表示してある LED です。点灯時は E10A-USB が制御するマイコンを切り替えたことを示しています。
(f) UVCC LED 表示	：“4”と表示してある LED です。点灯時は E10A-USB にユーザ電源（UVCC）が供給されていることを示しています。
(g) ホスト側コネクタ仕様マーク	：“ ”と表示しています。ホストコンピュータ接続用コネクタが側面にあることを示しています。

(h) ユーザ側コネクタ仕様マーク : “USER/IF”と表示しています。ユーザインタフェースケーブル接続用コネクタが側面にあることを示しいます。

【注】LED が消えていても USB が未接続あるいは故障というわけではありません。

(2) ホスト側 側面の構成

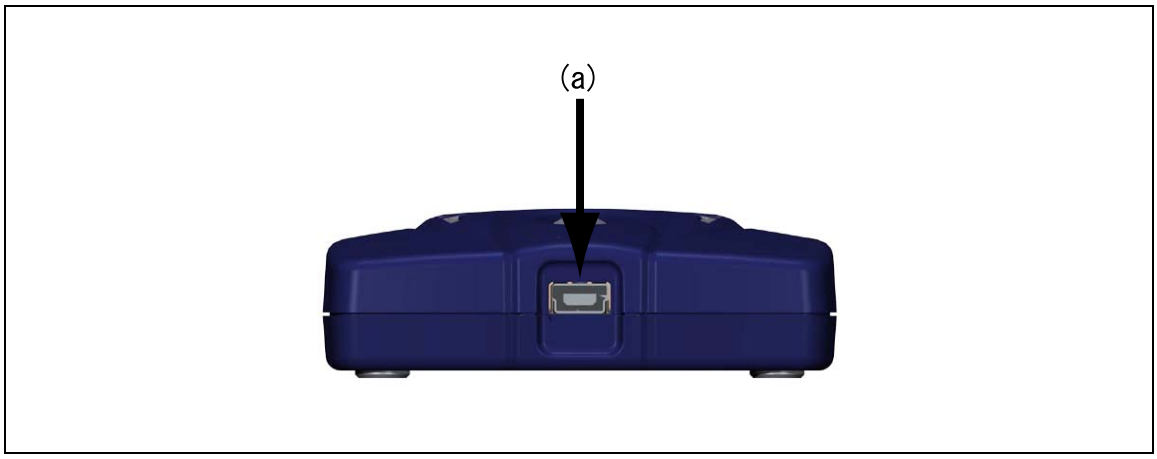


図 3.4 E10A-USB エミュレータのホスト側 側面

(a) ホスト側コネクタ : ホストコンピュータ接続用コネクタ(USB コネクタ)です。必ず付属品の USB ケーブルを接続してください。

(3) ユーザ側 側面の構成

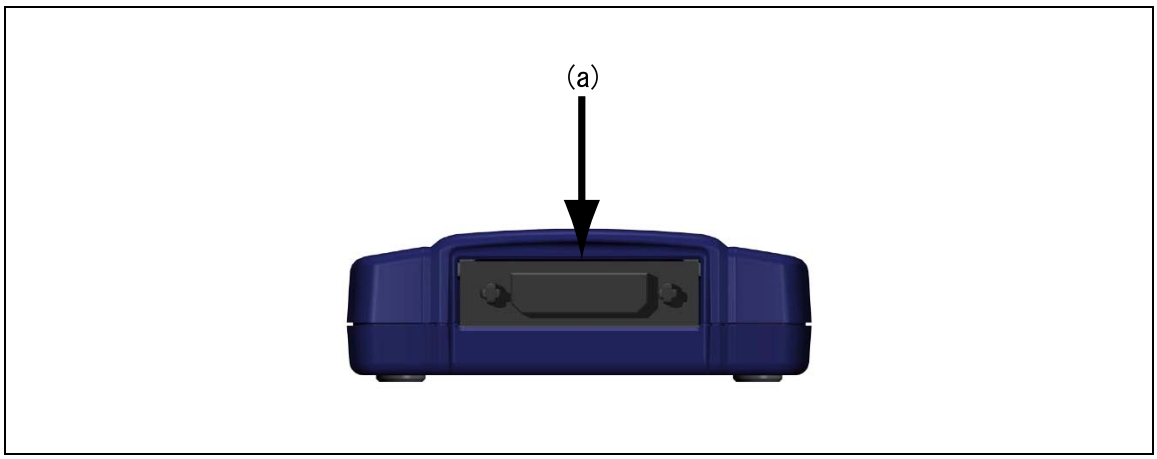


図 3.5 E10A-USB エミュレータのユーザ側 側面

(a) ユーザ側コネクタ : ユーザインタフェースケーブルを接続します。

（4）底面の構成

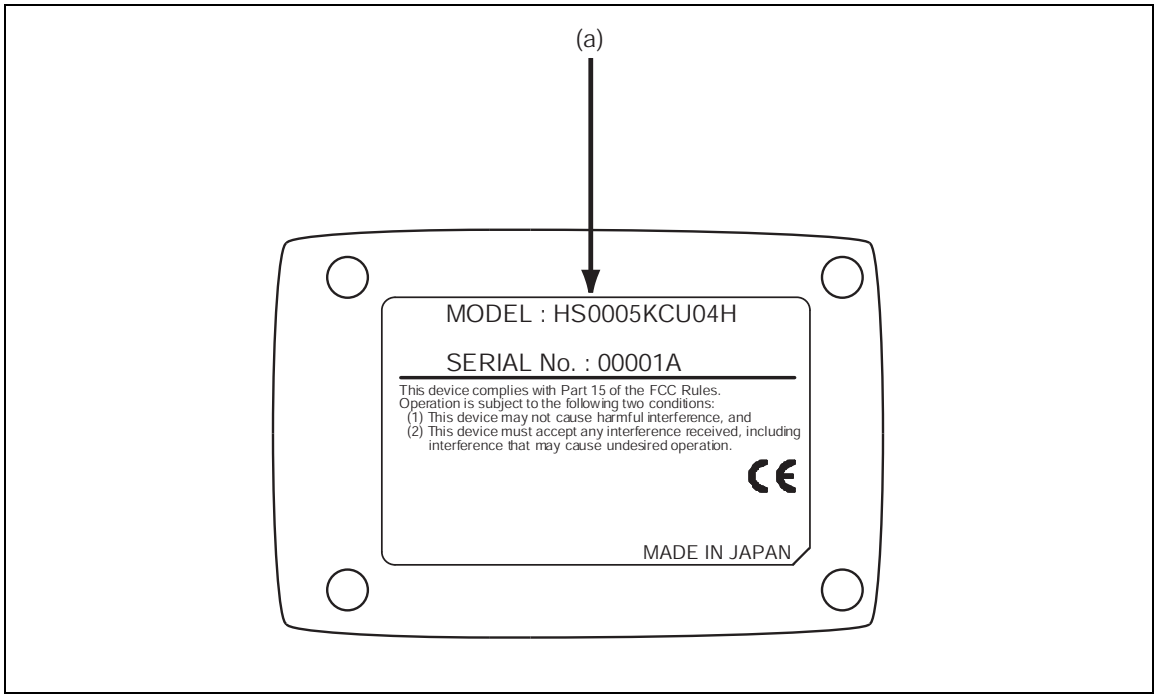


図 3.6 E10A-USB エミュレータの底面

(a) 製品管理シール ：E10A-USB エミュレータ個々のシリアル番号、レビジョン、安全規格などが書かれています。書かれている内容については購入時期により異なります。

3.3 CD-R について

CD-R のルートディレクトリには E10A-USB エミュレータソフトウェアインストール用プログラムが含まれています。

その他、各フォルダには下記に示すファイルおよびプログラムが含まれます。

表 3.1 CD-R フォルダ内容

フォルダ名	内容	備考
Dlls	Microsoft®ランタイムライブラリ	High-performance Embedded Workshop を動作させるために必要なランタイムライブラリです。インストール時にバージョンのチェックを行い、必要に応じてハードディスクにコピーされます。
Drivers	E10A-USB エミュレータ用ドライバ	E10A-USB エミュレータ用 USB ドライバです。
Help	E10A-USB エミュレータオンラインヘルプ	オンラインヘルプです。インストール時にハードディスクにコピーされます。
Manuals	E10A-USB エミュレータマニュアル	E10A-USB エミュレータユーザーズマニュアルです。 PDF 文書で提供しています。

3.4 エミュレータソフトウェアのインストール

CD-R のルートディレクトリから HewInstMan.exe を実行しインストールマネージャを起動してください。

インストールマネージャに従いインストールを行ってください。

【注】 Windows®XP をご使用の場合ドライバのインストール時に Windows®ロゴテストについての警告が表示される場合がありますが問題ありません。
[続行]を選択し、ドライバのインストールを進めてください。

USB ドライバは、以下の URL から”Renesas E-Series USB ドライバ”の最新バージョンをダウンロードし、インストールしてください。
http://japan.renesas.com/products/tools/emulation_debugging/onchip_debuggers/e10a_usb/downloads.jsp

3.5　ホストコンピュータと E10A-USB エミュレータとの接続

E10A-USB エミュレータとホストコンピュータを接続する方法を説明します。なお、E10A-USB エミュレータ本体における各コネクタの位置は、「3.2　E10A-USB エミュレータのハードウェア構成」を参照してください。

【注】　「新しいハードウェアの追加ウィザード」が表示された場合、[使用中のデバイスに最適なドライバを検索する（推奨）]を選択してください。

【留意事項】

E10A-USB エミュレータ装着前に、必ずエミュレータソフトウェアのインストールを行ってください。

⚠ 警告

ユーザシステムの電源投入時、USB インタフェースケーブルを除くケーブル類の抜き差しは、一切行わないでください。抜き差しを行った場合、E10A-USB 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。

E10A-USB エミュレータは、ホストコンピュータと USB 1.1 で接続できます。
また、USB2.0 準拠の USB ポートにも接続できます。システム構成を図 3.7 に示します。

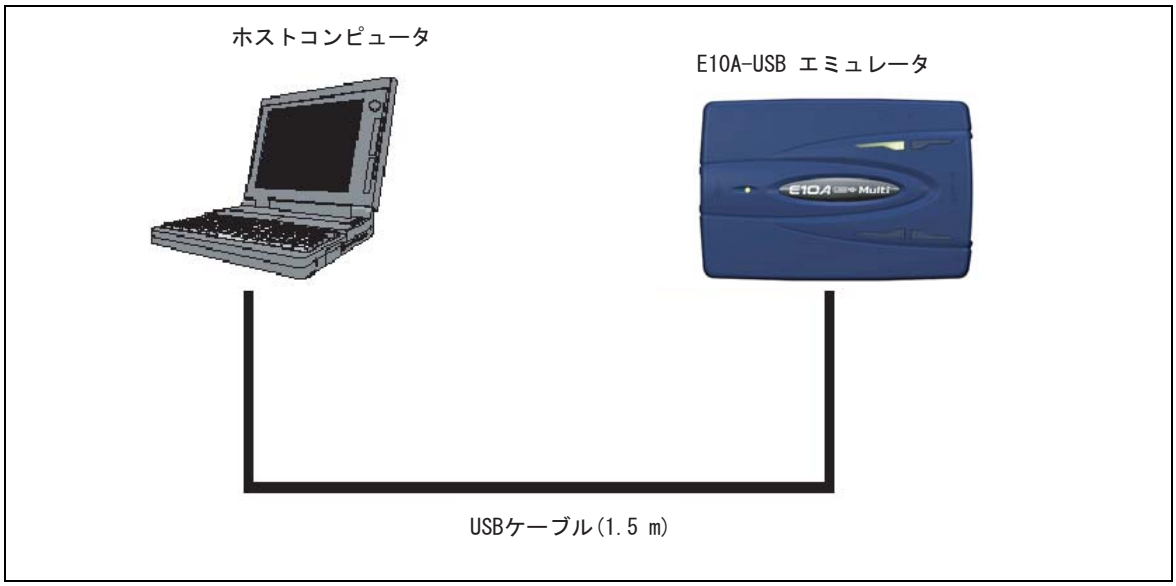


図 3.7　E10A-USB　エミュレータのホストコンピュータ接続システム構成

3.6　E10A-USB エミュレータとユーザシステムとの接続

以下に示す手順で E10A-USB エミュレータとユーザシステムをユーザインタフェースケーブルで接続してください。

また、装置の移動などのために E10A-USB エミュレータとユーザシステムを取り外したり、取り付ける場合も同様の手順で接続してください。

1. ホストコンピュータの電源がオフになっているか、E10A-USB エミュレータがUSBケーブルでホストコンピュータと接続されていないことを確認してください。
2. E10A-USB エミュレータのユーザ側 側面のコネクタにユーザインタフェースケーブルを接続します。
3. E10A-USB エミュレータのホスト側 側面のコネクタにUSB ケーブルを接続します。

図 3.8 にコネクタの配置を示します。

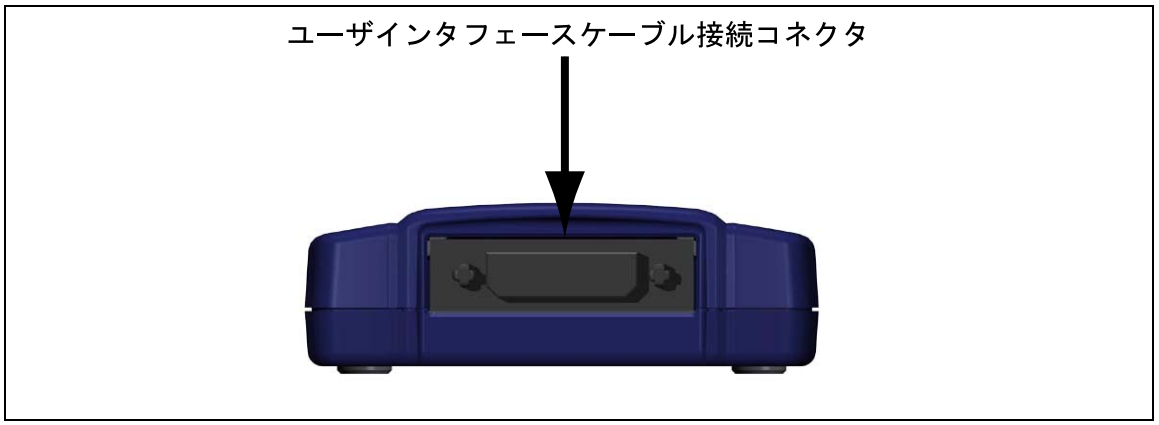


図 3.8　コネクタの配置

(1) コネクタをユーザシステム上に実装してください。E10A-USB　エミュレータが推奨するコネクタを表3.2に示します。

表 3.2 推奨コネクタ

	型 名	メーカ	仕様
14 ピン コネクタ	7614-6002	スリーエム株式会社	14 ピンストレートタイプ (国内推奨)
	2514-6002	3M Limited	14 ピンストレートタイプ (海外推奨)
38 ピン コネクタ	2-5767004-2	タイコエレクトロニクス ジャパン合同会社	38 ピン Mictor コネクタ

【注】 1. H-UDI ポートコネクタ実装時、14 ピンコネクタ使用時は、周囲 3 mm 四方に他の部品を実装しないでください。
2. H-UDI とは、JTAG (Joint Test Action Group)インタフェースとコンパチブルなインタフェース仕様です。

(2) コネクタのピン配置は、別冊の「SHxxxx ご使用時の補足説明」の2章に示すように配置されています。

(3) H-UDIポートコネクタの5ピンとH-UDIポートコネクタ中央に配置されているGNDバスリード(38ピンユーザインタフェースケーブル使用時)および9,10,12,13,14ピン(14ピンユーザインタフェースケーブル使用時)はPCB上でしっかりとGNDに接続してください。
電気的なGNDとして使用するほか、E10A-USB エミュレータがH-UDIポートコネクタの接続を監視するためにも使用しています。H-UDIポートコネクタのピン配置には注意してください。

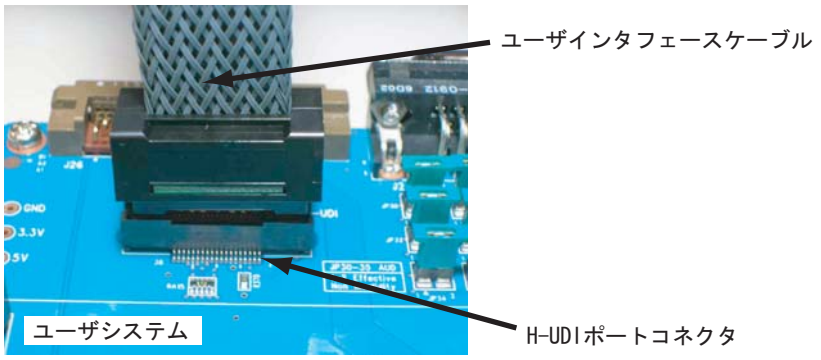


図 3.9　38 ピンタイプコネクタ使用時のユーザシステム側の
ユーザインタフェースケーブル接続方法

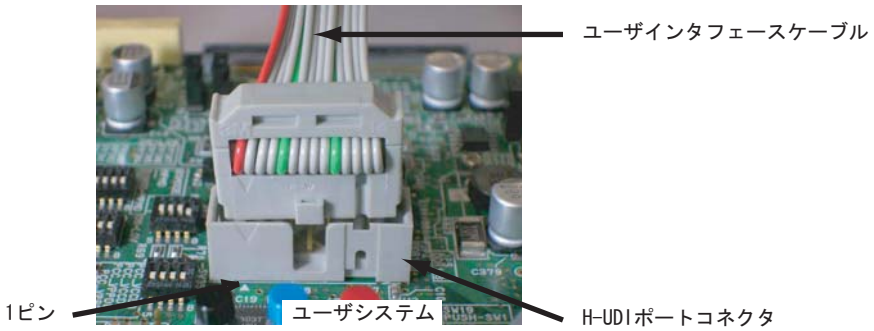


図 3.10　14 ピンストレートタイプコネクタ使用時のユーザシステム側の
ユーザインタフェースケーブル接続方法

注意

コネクタのピンの数え方は、コネクタ製造元のピン番号のふり方と異なりますので注意してください。

【留意事項】

1. コネクタの信号線の接続先は、パッケージによって異なります。デバイスのピン配置を参照してください。
2. エミュレータが動作する通信の範囲は、サポートするデバイスによって異なります。
3. ユーザシステムにコネクタを接続する際、信号の配線は、別冊の「SHxxxx ご使用時の補足説明」の 1 章を参照してください。
4. ユーザシステムを設計する際、バウンダリスキャン用ループにデバイスの TDI 信号、TDO 信号を接続しないでください。または、スイッチ等でデバイスを切り離すようにしてください。（図 3.11 参照）

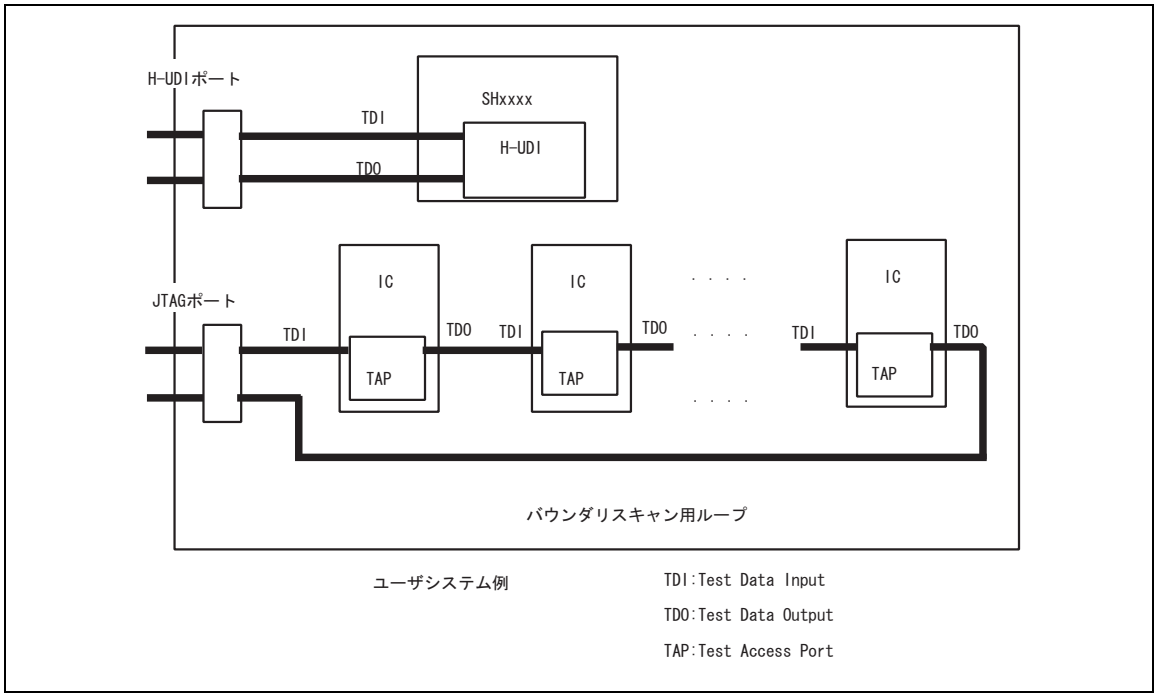


図 3.11 ユーザシステム設計時の注意

3.7　システムグランド系の接続

⚠ 警告

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、グランド電位の差により発煙、発火、感電の危険性があります。

エミュレータのシグナルグランドは、ユーザシステムのシグナルグランドに接続されます。
エミュレータ内部では、シグナルグランドとフレームグランドが接続されています。
ユーザシステムでは、シグナルグランドとフレームグランドを接続せず、フレームグランドだけを接地してください（図 3.12）。
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、ホストコンピュータの DC 電源入力（AC アダプタ）の GND 電位とユーザシステムのフレームグランドを同電位にしてください。
ホストコンピュータとターゲットシステムの GND に電位差がある場合、インピーダンスが低い GND ラインに過電流が流れ、細いラインの焼損などの危険性があります。

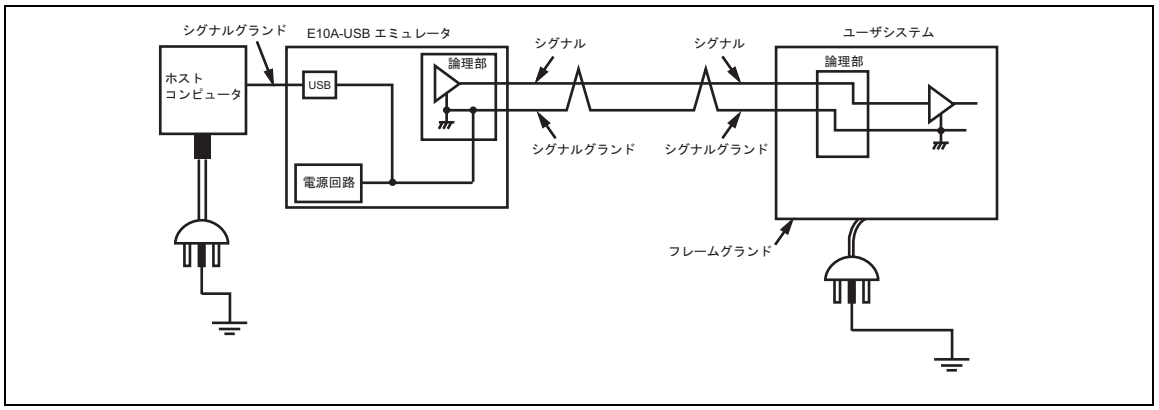


図 3.12　システムグランド系の接続

3.8　E10A-USB エミュレータ内インタフェース回路

図 3.13、図 3.14 にエミュレータ内インタフェース回路を示します。
プルアップ抵抗の値などを決めるときに参考にしてください。

【注】74LVC2G125 および 74LVC2T45 は H-UDI ポートコネクタからの VCC(1.8～5.0V)で駆動します。

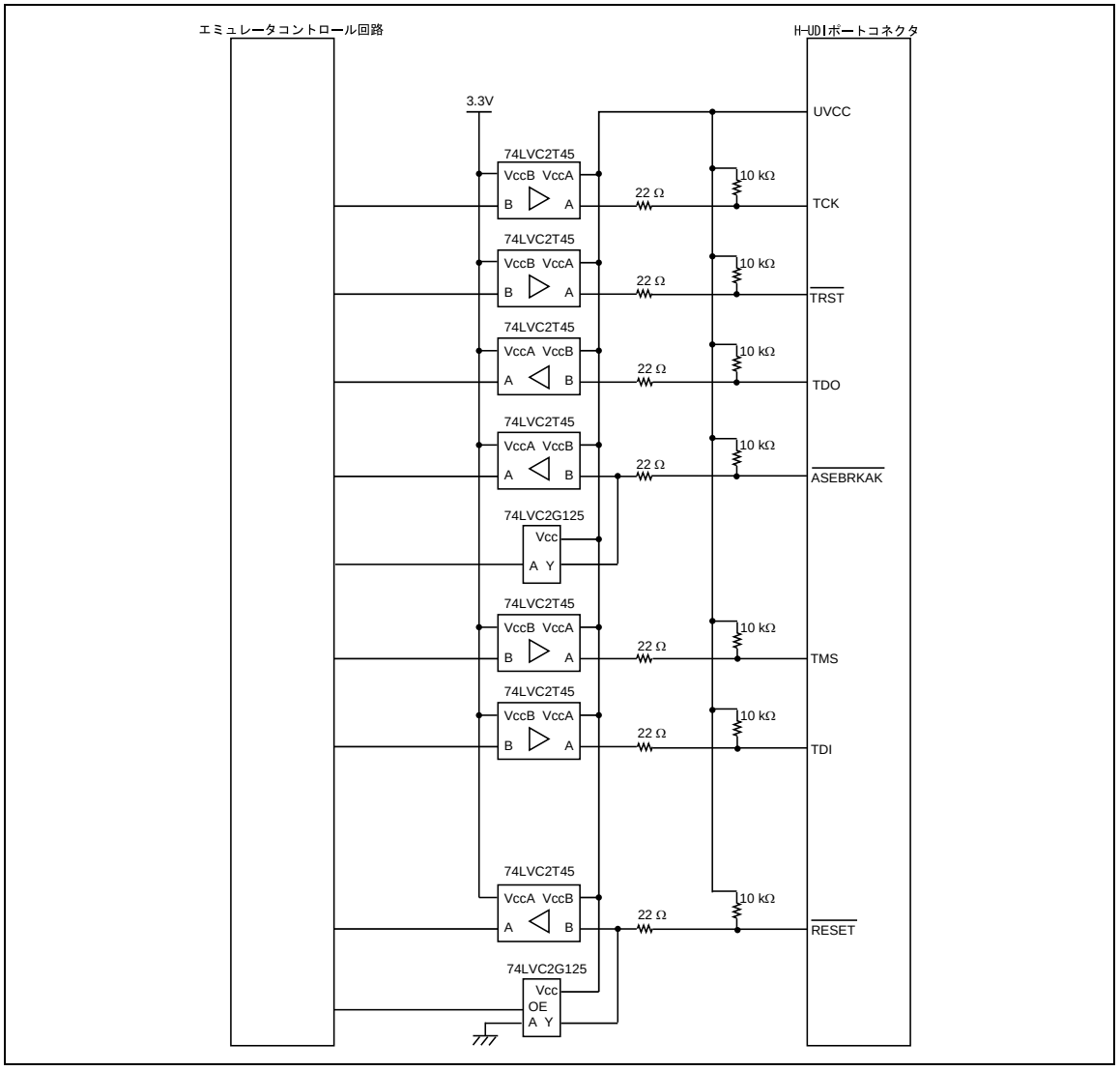


図 3.13　エミュレータ内インタフェース回路(H-UDI)

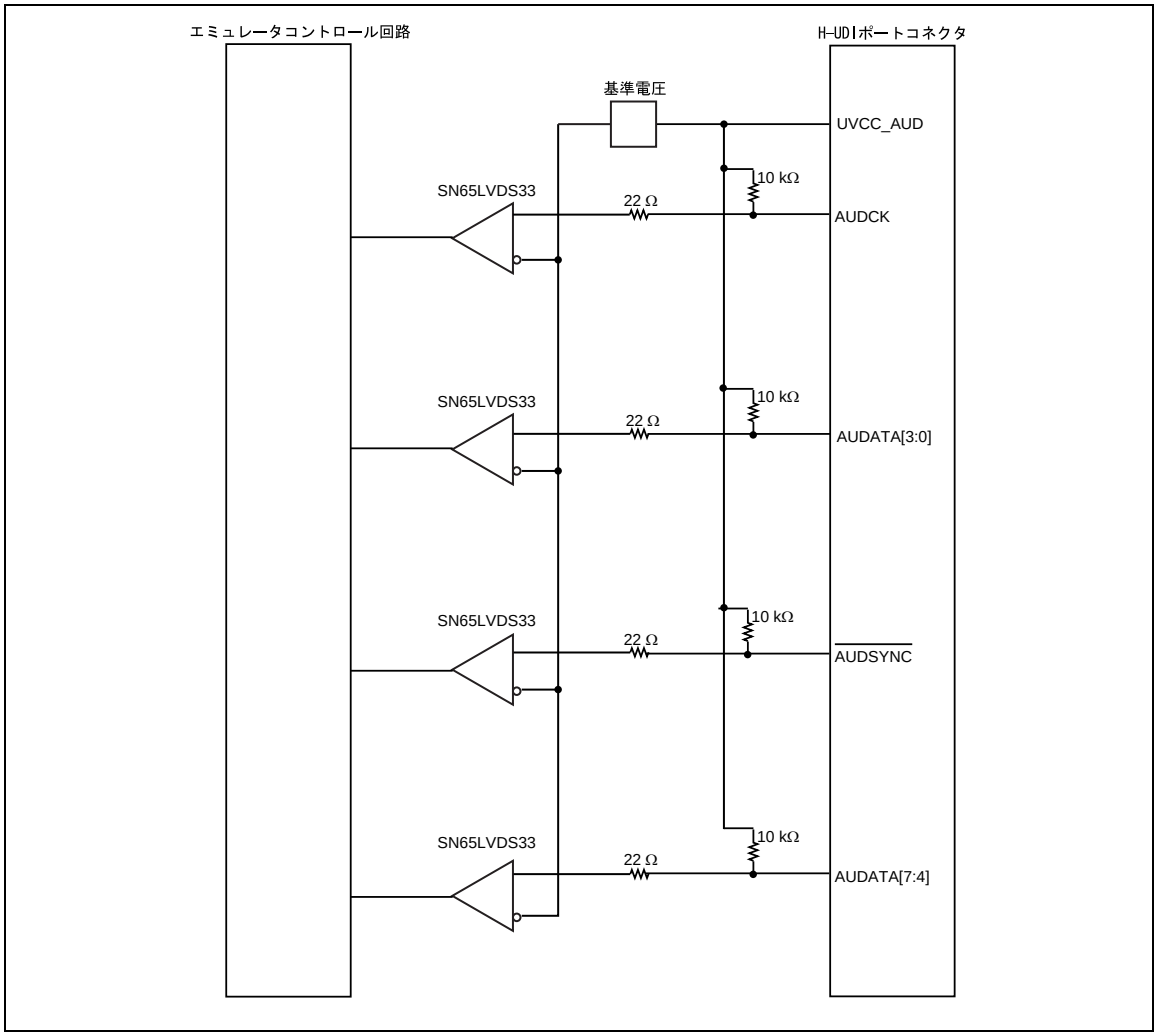


図 3.14 エミュレータ内インタフェース回路(AUD)

3.9　システムチェック

次に、ソフトウェアを実行し、E10A-USB エミュレータが正しく接続されていることをチェックします。

ここでは、製品に添付のチュートリアル用ワークスペースを使用して CPU0 のみ起動します。

新規にプロジェクトを作成して起動する方法や、既存のワークスペースを使用して起動する方法については、「4 デバッグの準備をする」を参照してください。

(1)ホストコンピュータとE10A-USB エミュレータを接続してください。

(2)E10A-USB エミュレータのコネクタとユーザインタフェースケーブルを接続します。

(3)ユーザシステム側のコネクタにユーザインタフェースケーブルを接続します。

(4)[スタート]メニューの[プログラム]から[Reneas]→[High-performance Embedded Workshop] → [High-performance Embedded Workshop]を選択してください。

(5)[ようこそ!]　ダイアログボックスが表示されます。

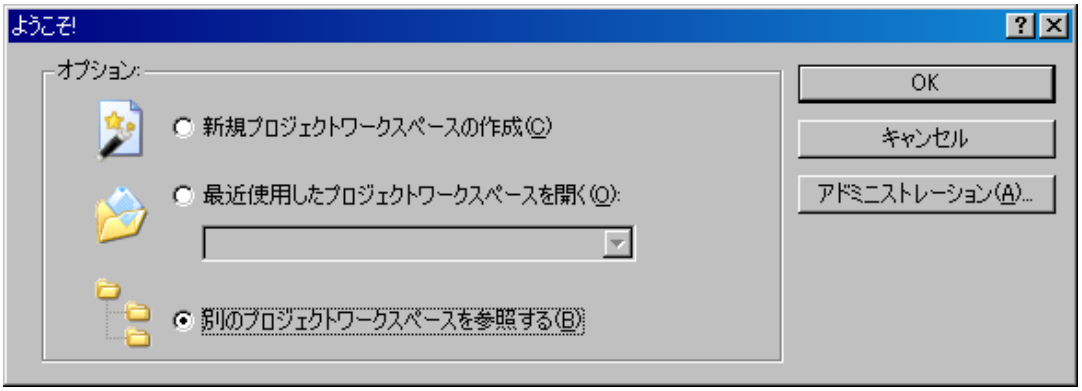


図 3.15　[ようこそ!]　ダイアログボックス

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

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

[ワークスペースを開く]ダイアログボックスが開きますので、以下のディレクトリを指定してください。

<OS がインストールされているドライブ>
¥WorkSpace¥Tutorial¥E10A-USBM¥xxxx¥xxxx¥Tutorial¥CPU0

xxxx は対象の製品グループを示します。
ディレクトリの指定後、以下のファイルを選択し[開く]ボタンを押してください。

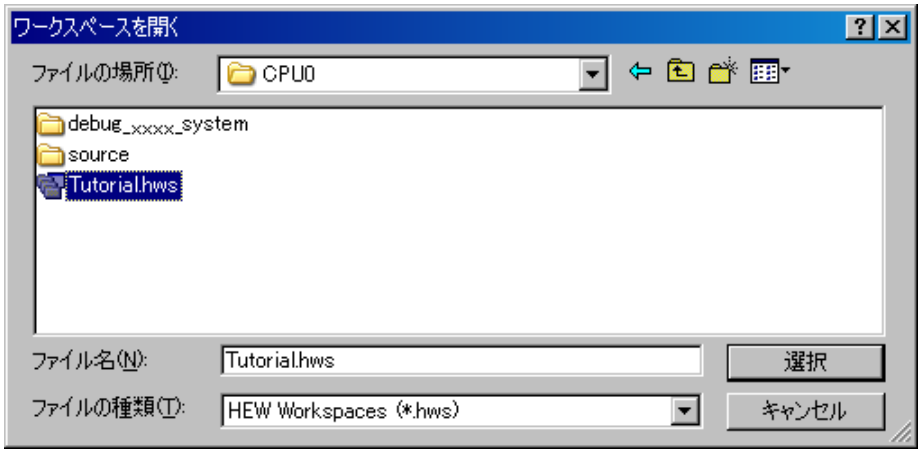


図 3.16　[ワークスペースを開く]ダイアログボックス

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

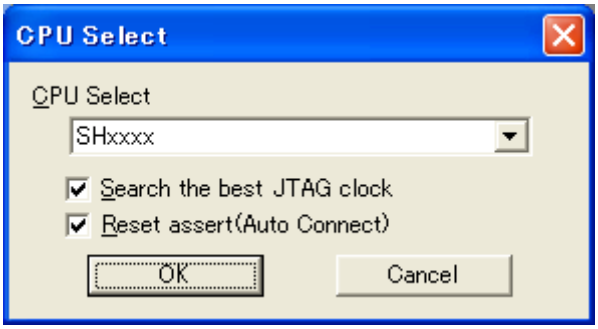


図 3.17　[CPU Select]ダイアログボックス

[CPU Select]ダイアログボックスには下記のオプションがあります。

- [Search the best JTAG clock]チェックボックス
  JTAG clock 値を検索し，使用可能な最速の値を初期値として起動します。

  [Reset assert(Auto Connect)]チェックボックス
  E10A-USB エミュレータからリセット信号を発行し、(12)、(14)の手順を省略します。

注意

別冊の SHxxxx ご使用時の補足説明「1.5　H-UDI ポートコネクタとチップ間の推奨接続例」の記載通りに結線されていない場合は、絶対に[Reset assert(Auto Connect)]オプションを使用しないでください。ユーザシステムの故障につながります。

ご使用のデバイス名を[CPU Select]ドロップダウンリストボックスより選択し、[OK]ボタンを押してください。

(7)　ご使用のデバイスによっては[Select Emulator mode]ダイアログボックスが表示されます。

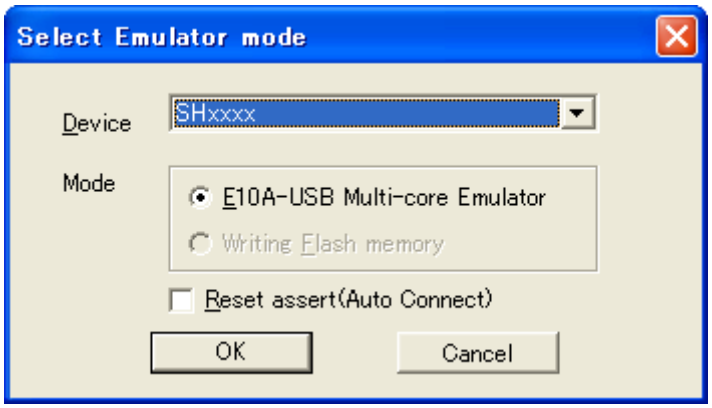


図 3.18　[Select Emulator mode]ダイアログボックス

[Select Emulator mode]ダイアログボックスには下記のオプションがあります。

[Reset assert(Auto Connect)]チェックボックス

E10A-USB エミュレータからリセット信号を発行し、(12)、(14)の手順を省略します。

注意

別冊の SHxxxx ご使用時の補足説明「1.5　H-UDI ポートコネクタとチップ間の推奨接続例」の記載通りに結線されていない場合は、絶対に[Reset assert(Auto Connect)]オプションを使用しないでください。ユーザシステムの故障につながります。

ご使用のデバイス名を[Device]ドロップダウンリストボックスより選択し、[OK]ボタンを押してください。

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

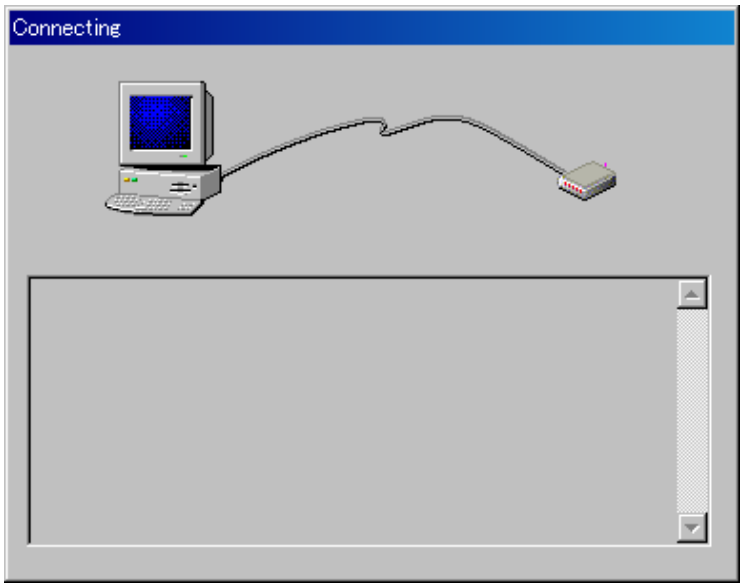


図 3.19　[Connecting]ダイアログボックス

(9) 製品グループがインストールされていない新規購入時または、異なるデバイスグループのエミュレータファームウェアがセットアップされていた場合は、図3.20、E10A-USB エミュレータ本体にセットアップされているエミュレータファームウェアのバージョンが古い場合は図3.21に示す確認ダイアログボックスが表示されます。

[OK]ボタンを押した場合、エミュレータファームウェアをセットアップします。

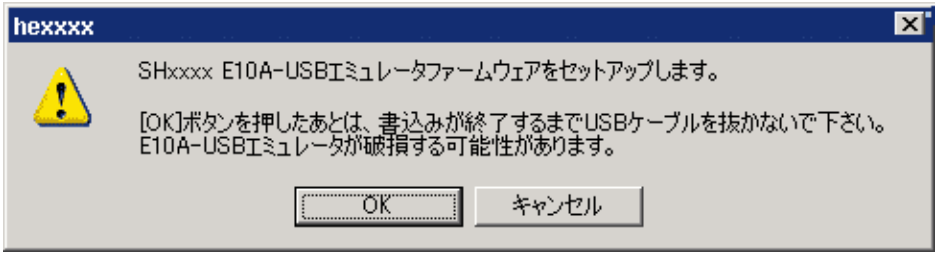


図 3.20　[エミュレータファームウェアセットアップ確認]ダイアログボックス

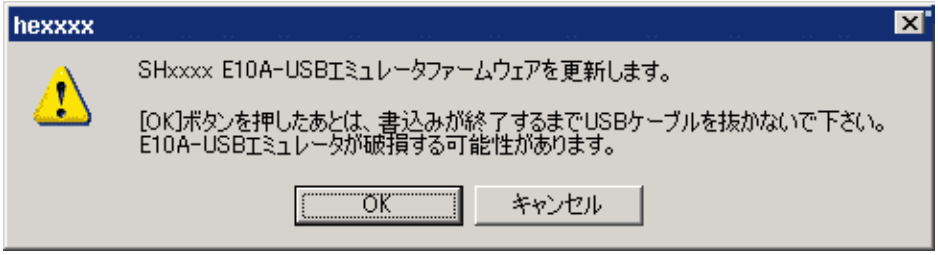


図 3.21　[SHxxxx エミュレータファームウェア更新確認]ダイアログボックス

注意

書き込みが終了するまで絶対に USB ケーブルを抜かないでください。
E10A-USB エミュレータの破壊につながります。

(10) [Reset assert(Auto Connect)]オプション使用時は、図3.22に示すダイアログが表示されます。
ユーザシステムの電源が入っている場合は、図3.22は表示されません。

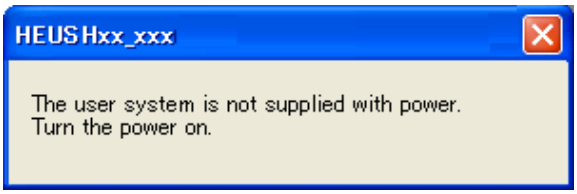


図 3.22　[電源投入要求]ダイアログボックス

(11)　ユーザシステムの電源を入れます。

(12)　図3.23に示すダイアログボックスが表示されます。

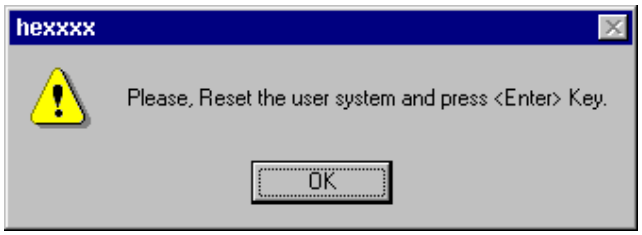


図 3.23　RESET 信号入力要求メッセージのダイアログボックス

(13)　ユーザシステムの電源を入れます。

(14)　ユーザシステムからRESET信号を入力し、[OK]ボタンをクリックします。

(15)　RESET信号が検出できなかった場合以下のダイアログボックスが表示されます。

図 3.24　[Can not find /RESET signal]ダイアログボックス

(16) High-performance Embedded Workshopの[Output]ウィンドウに”Connected”と表示されたら、CPU0用のE10A-USB エミュレータの起動は完了です。

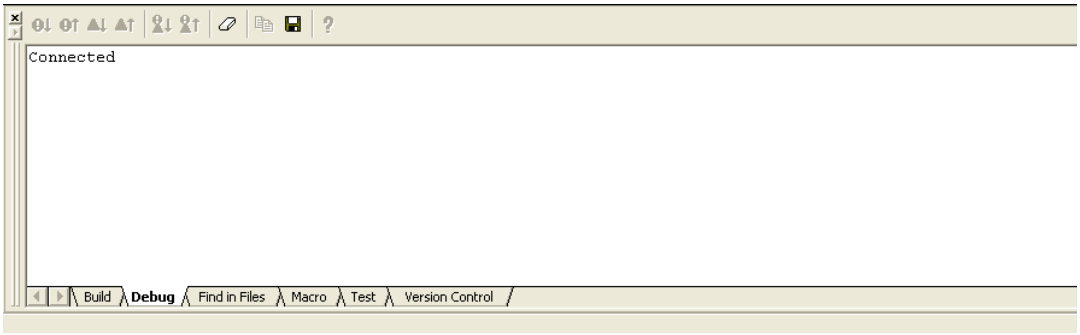


図 3.25 [Output]ウィンドウ

【留意事項】

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

(a) 以下のダイアログボックスが表示された場合で(12)の方法で起動できない場合、ユーザシステムの電源が入っていないか、RESET 信号がデバイスに入力されていない可能性があります。ユーザシステムの電源とリセット端子への入力回路を確認してください。

図 3.26 [Can not find /RESET signal]ダイアログボックス

(b) 以下のダイアログボックスが表示された場合、H-UDI ポートコネクタが正しく結線されていない可能性があります。H-UDI ポートコネクタとの結線を確認してください。

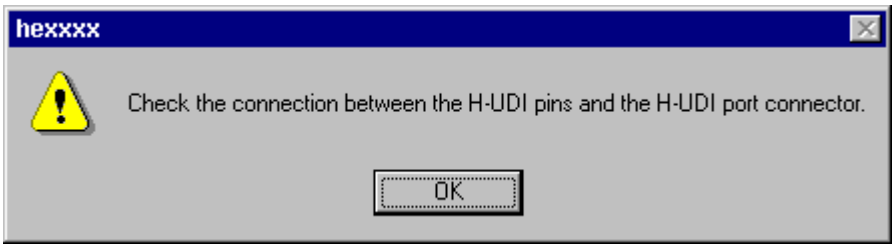


図 3.27 [Check the connection]ダイアログボックス

(c) 以下のダイアログボックスが表示された場合、デバイスが正常に動作していない可能性があります。デバイスが正常に動作できない要因がないかどうか確認してください。

図 3.28 [COMMUNICATION TIMEOUT ERROR]ダイアログボックス

図 3.29 [INVALID ASERAM FIRMWARE!]ダイアログボックス

図 3.30 [Error JTAG boot]ダイアログボックス

(d) 以下のダイアログボックスが表示された場合、MCU と E10A-USB エミュレータとの通信が取れません。一因として、MCU が正しく動作していない可能性があるため、設定を確認してください。

図 3.31 [Boot Failed!]ダイアログボックス

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

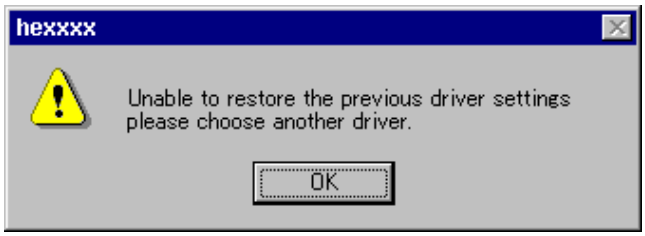


図 3.32 [Unable to restore the previous driver settings]ダイアログボックス

3. その他の要因で E10A-USB エミュレータが起動されない場合、状況に応じたメッセージボックスが表示されます。ボード上の結線などを確認する上で、メッセージの内容を参考にしてください。

4.　デバッグの準備をする

4.1　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E10A-USB エミュレータ、ユーザシステムを接続し、
ユーザシステムの電源を入れてください。

(2) [スタート]メニューの[プログラム]から[Renesas]→[High-performance Embedded Workshop]
→[High-performance Embedded Workshop]を選択してください。

(3) [ようこそ!]ダイアログボックスが表示されます。

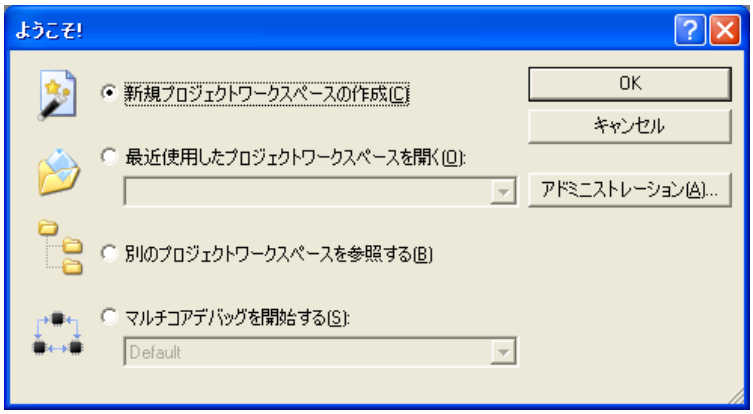


図 4.1　[ようこそ!]ダイアログボックス

- [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。
- [マルチコアデバッグを開始する]ラジオボタン
  マルチコアデバッグをする場合に選択します。
  マルチコアデバッグの履歴がある場合のみ表示されます。

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

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

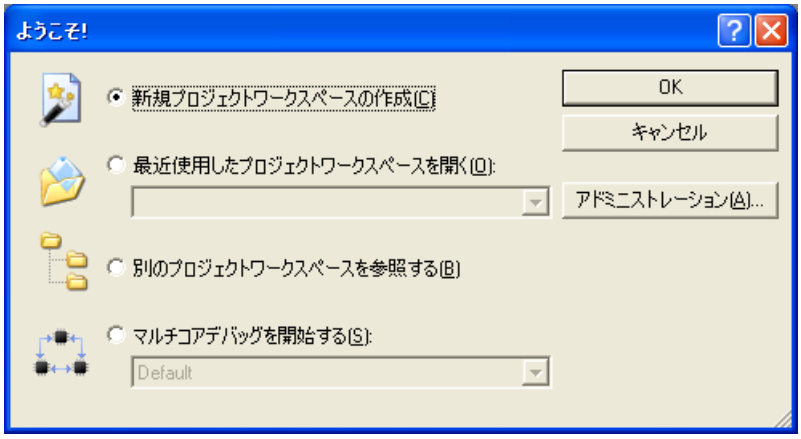


図 4.2 [ようこそ!]ダイアログボックス

(2)　Project Generatorが開始されます。
ここでは、ツールチェイン用の設定に関する説明は省略します。
ツールチェインをご購入されていない場合、以下の画面が開きます。

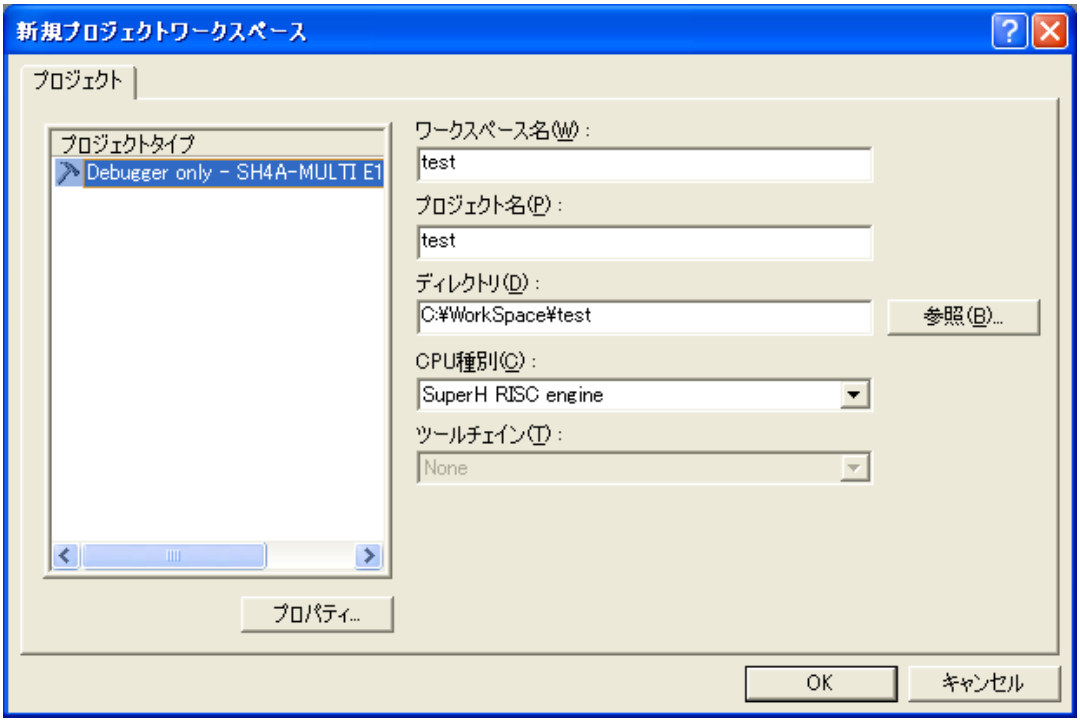


図 4.3　[新規プロジェクトワークスペース]ダイアログボックス

- [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。ここでは例として“test”と入力します。

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

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

(3)　次に、以下の画面が表示されます。

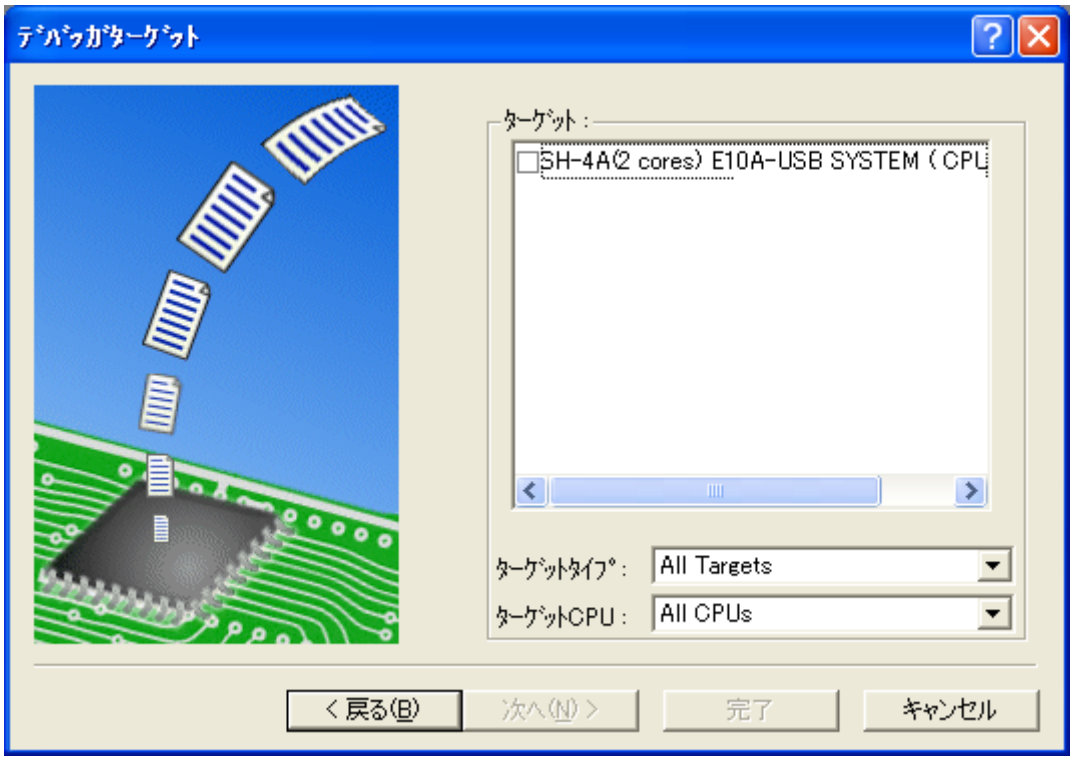


図 4.4　[デバッガターゲット]ダイアログボックス

ここで、該当する E10A-USB エミュレータにチェックし、[次へ]ボタンを押してください。

(4)　次に、コンフィグレーションファイル名を設定します。
コンフィグレーションとは、エミュレータ以外のHigh-performance Embedded Workshopの状態を保存するファイルです。
[コア]ドロップダウンリストボックスより、デバッグするCPUコア番号を選択してください。

図 4.5　[デバッガオプション]ダイアログボックス

これで E10A-USB エミュレータに関する設定は終了です。
[完了]ボタンを押し、Project Generator を終了してください。
High-performance Embedded Workshop が起動します。

(5)　High-performance Embedded Workshop起動後、自動的にE10A-USB エミュレータが接続されます。
接続中の操作については、「3.9　システムチェック」を参照してください。

4.1.2　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)

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

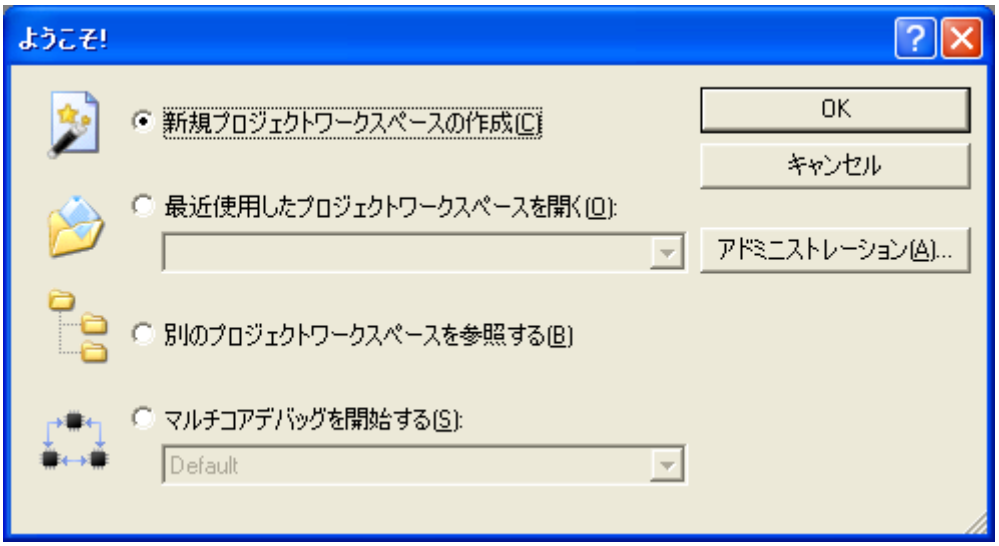


図 4.6　[ようこそ!]ダイアログボックス

(2) Project Generator が開始されます。
ツールチェインをご購入されている場合、以下の画面が開きます。

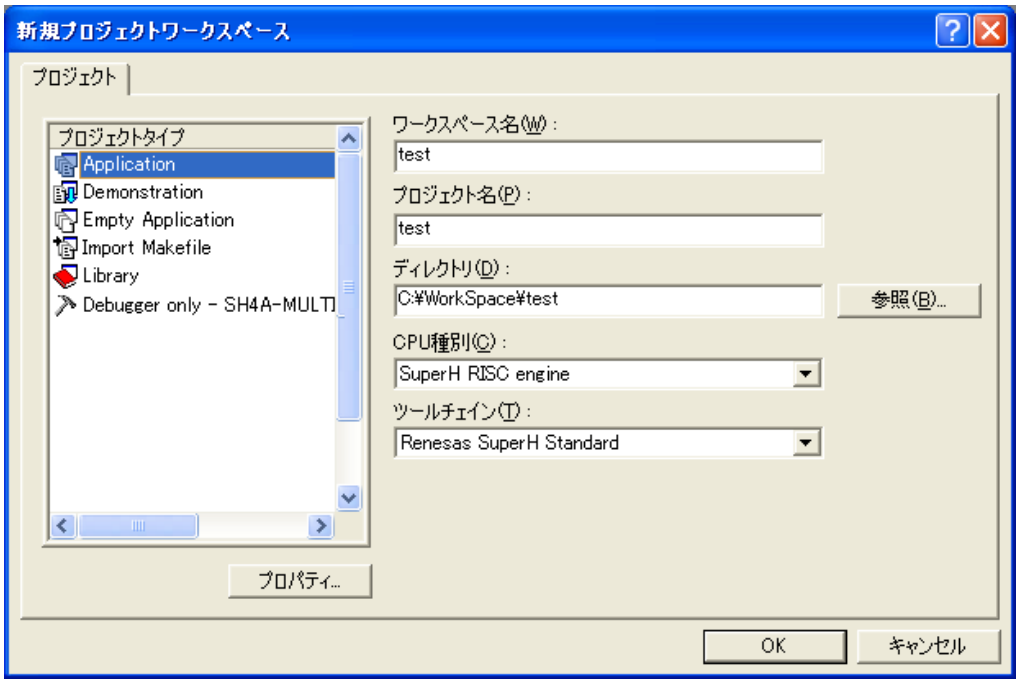


図 4.7　[新規プロジェクトワークスペース]ダイアログボックス

- [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。ここでは例として“test”と入力します。

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

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

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

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

【留意事項】

E10A-USB エミュレータの場合、[Demonstration]を選択した場合に以下の注意事項があります。

[Demonstration]は Simulator 用のプログラムです。生成されたプログラムをエミュレータで使用する場合、“Printf 文”を削除してください。

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

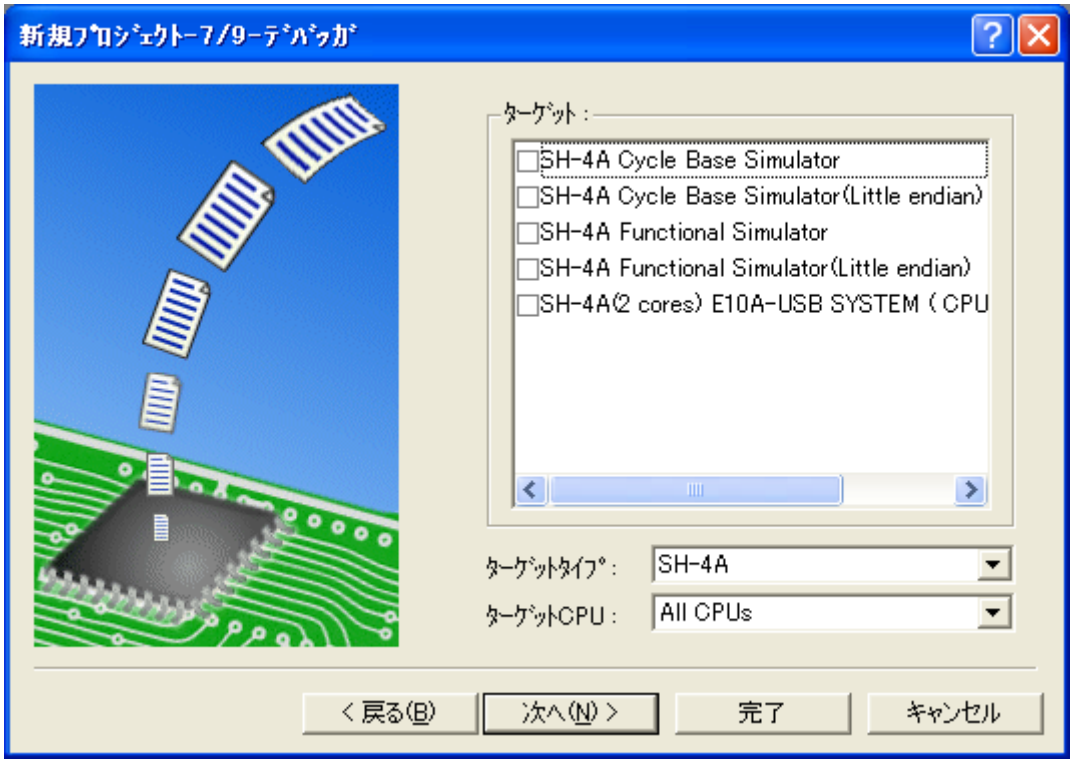


図 4.8　[新規プロジェクト-7/9-デバッガ]ダイアログボックス

ここで、該当する E10A-USB エミュレータにチェックし、[Next]ボタンを押してください。
必要であれば、他の製品にもチェックをしてください。

(4) 次に、コンフィグレーションファイル名を設定します。
　コンフィグレーションファイルとは、エミュレータ以外のHigh-performance Embedded Workshopの状態を保存するファイルです。
　［コア］ドロップダウンリストボックスより、デバッグするCPUコア番号を選択してください。

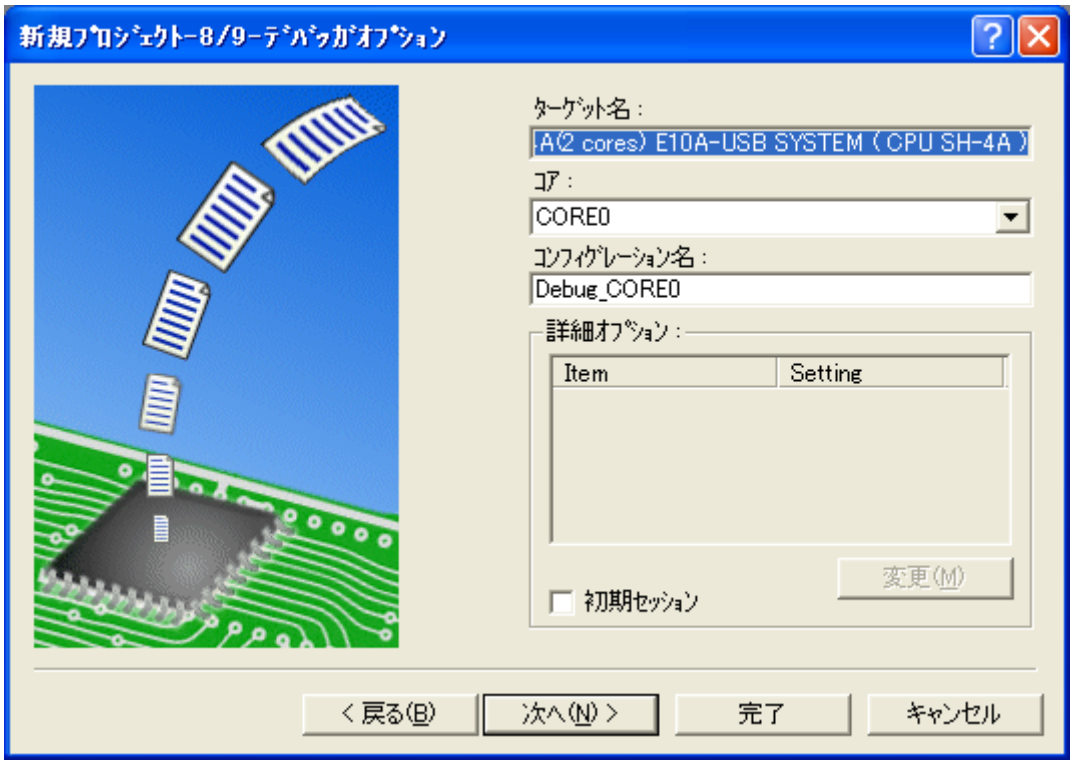


図 4.9　[新規プロジェクト-8/9-デバッガオプション]ダイアログボックス

　これで E10A-USB エミュレータに関する設定は終了です。
　画面の指示に従い、Project Generator を終了してください。High-performance Embedded Workshop が起動します。

(5) High-performance Embedded Workshop起動後、E10A-USB エミュレータを接続してください。
　E10A-USB 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。E10A-USB エミュレータを接続する場合は、以下のどちらかの操作をしてください。
　接続中の操作については、「3.9　システムチェック」を参照してください。

(a) E10A-USB 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]メニューの[デバッグの設定]を選択し、[デバッグの設定]ダイアログボックスを開いてください。ここで、ダウンロードモジュールや起動時に自動的に実行するコマンドチェインなどを登録することができます。
[デバッグの設定]ダイアログボックスの詳細については、「4.3　E10A-USB エミュレータ起動時の設定」を参照してください。
[デバッグの設定]ダイアログボックスの設定終了後、ダイアログボックスを閉じるとE10A-USB エミュレータが接続されます。

(b)　E10A-USB エミュレータ起動時の設定を行わずに簡単に接続する方法
E10A-USB エミュレータを使用する設定があらかじめ登録されているセッションファイルに切り替えることにより、E10A-USB エミュレータを簡単に接続できます。

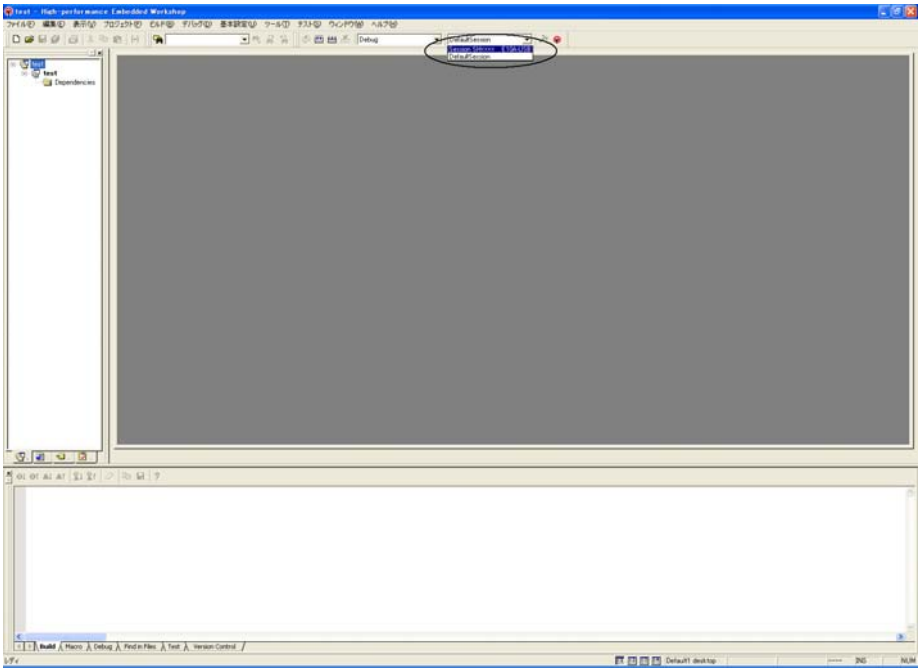


図 4.10　セッションファイルの選択

上記図中の、丸印の中にあるリストボックスから、「図 4.9　[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」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、E10A-USB エミュレータを使用する設定が登録されています。
選択終了後、E10A-USB エミュレータが自動的に接続されます。

4.1.3　既存のワークスペースを指定する場合

（1）High-performance Embedded Workshop起動時に表示される、
[ようこそ!]ダイアログボックスで、[別のプロジェクトワークスペースを参照する]
ラジオボタンを選択し、[OK]ボタンを押してください。

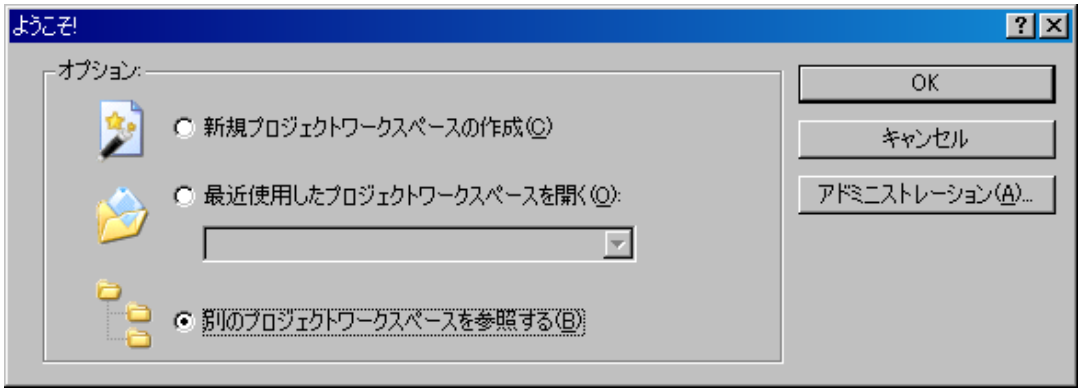


図 4.11　[ようこそ!]ダイアログボックス

(2)[ワークスペースを開く]ダイアログボックスが開きますので、ワークスペースが
作成されているディレクトリを指定してください。
ディレクトリの指定後、ワークスペースファイル（拡張子 .hws）を選択し
[開く]ボタンを押してください。

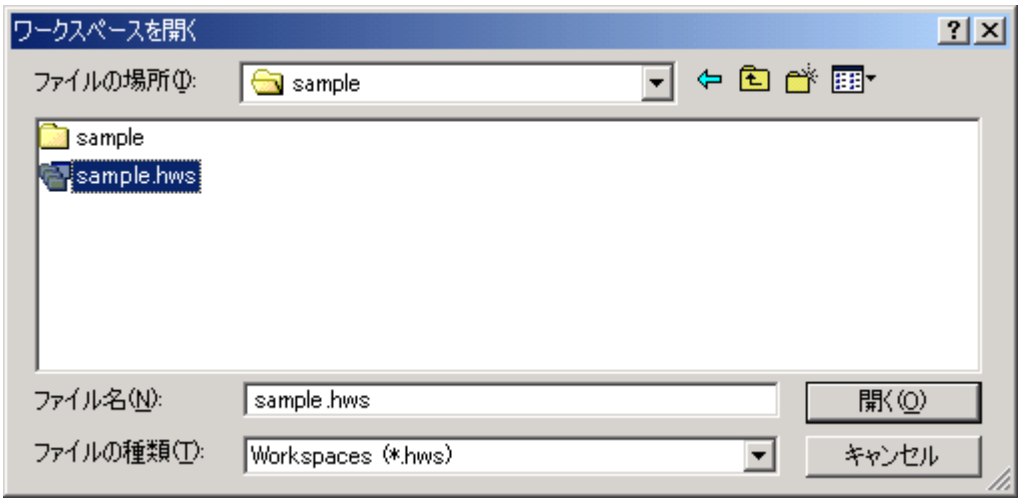


図 4.12　[ワークスペースを開く]ダイアログボックス

(3) 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、「4.5　エミュレータの接続」を参照してください。

4.2　同期デバッグ用プロジェクトを作成

同期デバッグを行う場合、ワークスペースにデバッグを行うコア数分のプロジェクトが含まれる必要があります。

4.2.1　新規プロジェクトを追加する場合

(1)「4.1.1 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）」、「4.1.2 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）」、「4.1.3 既存のワークスペースを指定する場合」で作成またはオープンしたワークスペースの［プロジェクト］メニューから［プロジェクトの挿入］を選択してください。［プロジェクトの挿入］ダイアログボックスが開きます。

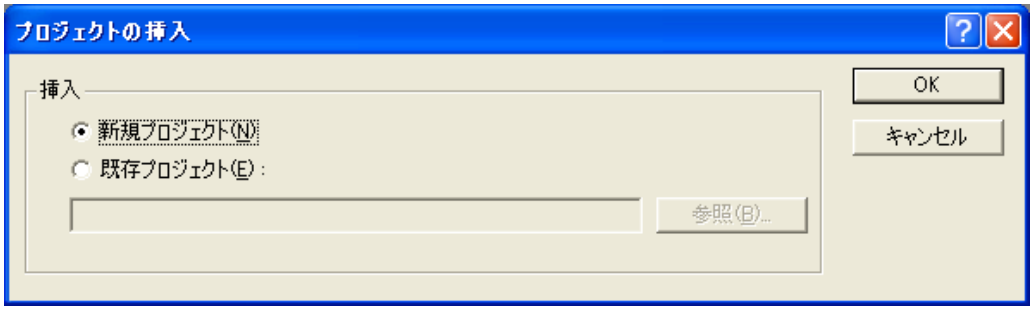


図 4.13　[プロジェクトの挿入]ダイアログボックス

(2)［新規プロジェクト］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。［新規プロジェクトの挿入］ダイアログボックスが開きます。

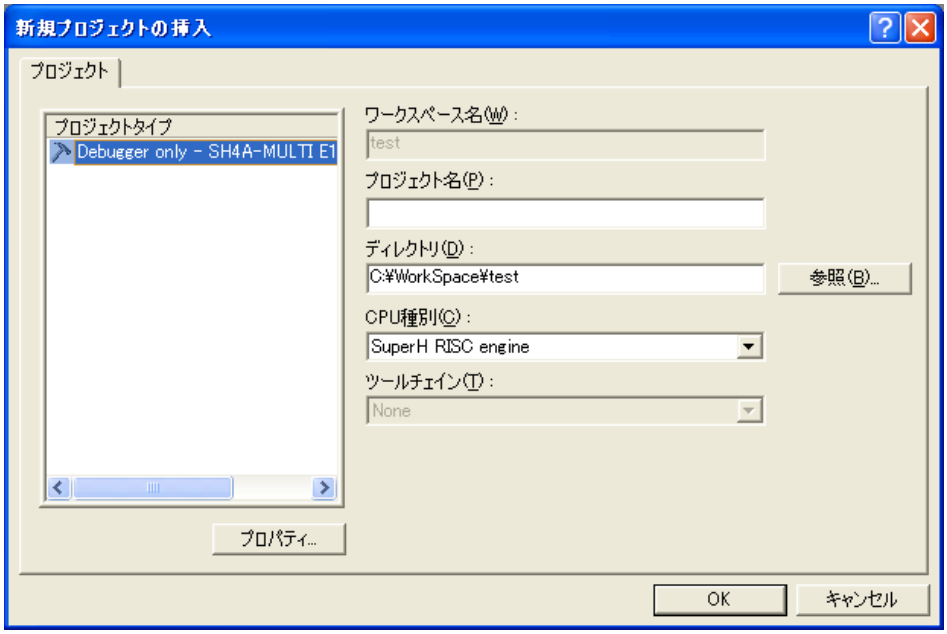


図 4.14　[新規プロジェクトの挿入]ダイアログボックス

(3)「4.1.1 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）」、「4.1.2 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）」と同様の手順でプロジェクトを作成してください。

4.2.2　既存プロジェクトを追加する場合

(1)「4.1.1 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）」、「4.1.2 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）」、「4.1.3 既存のワークスペースを指定する場合」で作成またはオープンしたワークスペースの［プロジェクト］メニューから［プロジェクトの挿入］を選択してください。［プロジェクトの挿入］ダイアログボックスが開きます。

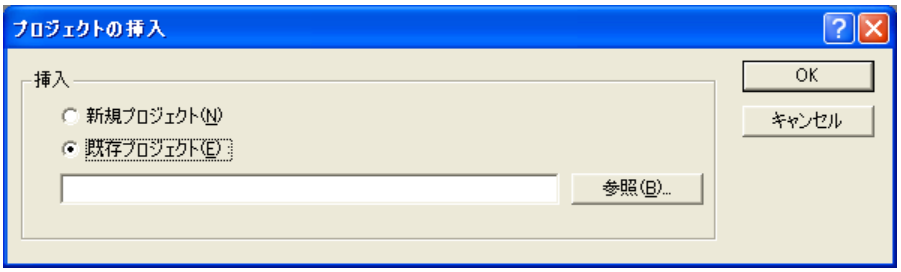


図 4.15　[プロジェクトの挿入]ダイアログボックス

(2)［既存プロジェクト］ラジオボタンを選択し、［参照］ボタンから追加するプロジェクトを選択し、［OK］ボタンをクリックしてください。

4.3　E10A-USB エミュレータ起動時の設定

4.3.1　エミュレータ起動時の設定

E10A-USB エミュレータの起動時、コマンドチェインの実行を自動的に行うことができます。
また、ダウンロードするロードモジュールを複数登録することができます。
登録したロードモジュールは、ワークスペースウインドウに表示されます。

[デバッグ]メニューから[デバッグの設定]を選択してください。
[デバッグの設定]ダイアログボックスが開きます。

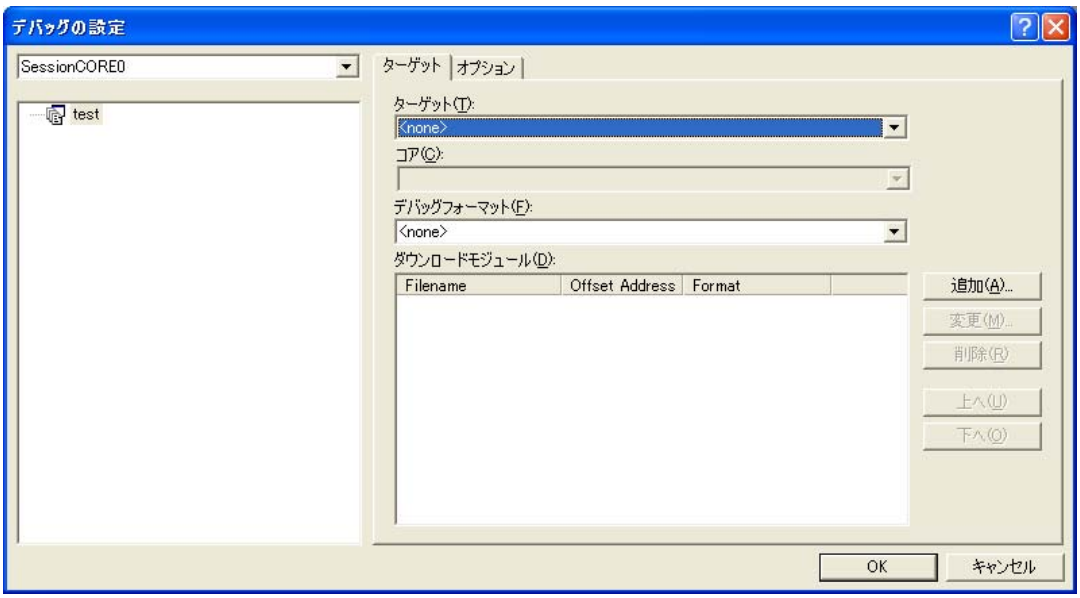


図 4.16　[デバッグの設定]ダイアログボックス（[ターゲット]ページ）

[ターゲット]ドロップダウンリストボックスで接続したい製品名を選択してください。
［コア］ドロップダウンリストボックスで接続したいコアを選択してください。
[デフォルトデバッグフォーマット] ドロップダウンリストボックスで、ダウンロードするロードモジュールの形式を選択し、それに対応するダウンロードモジュールを[ダウンロードモジュール]リストボックスに登録してください。

【注】　この時点ではプログラムのダウンロードはされていません。
ダウンロード方法については、「5.3　プログラムをダウンロードする」を参照してください。

次に、[オプション]ページをクリックしてください。

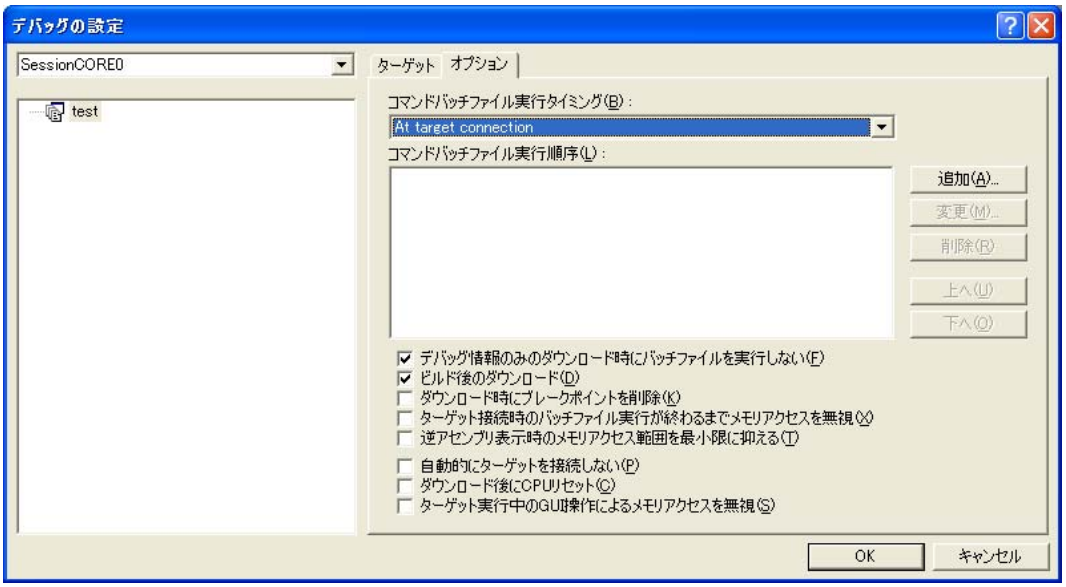


図 4.17　[デバッグの設定]ダイアログボックス（[オプション]ページ）

ここでは、指定したタイミングで自動的に実行するコマンドチェインを登録します。
指定できるタイミングは以下 4 点です。

- エミュレータ接続時
- リセット直後
- ダウンロード直前
- ダウンロード直後

[コマンドバッチファイル実行タイミング]ドロップダウンリストボックスで、コマンドチェインを実行するタイミングを指定してください。

また、[コマンドバッチファイル実行順序]リストボックスに、指定したタイミングで実行するコマンドチェインファイルを登録してください。

4.3.2　　プログラムのダウンロードについて

[Workspace]ウィンドウの[Download modules]にダウンロードモジュールが追加されます。
[Workspace]ウィンドウの[Download modules]のロードモジュールを右クリックで開き[ダウンロード]を選択するとダウンロードが開始します。

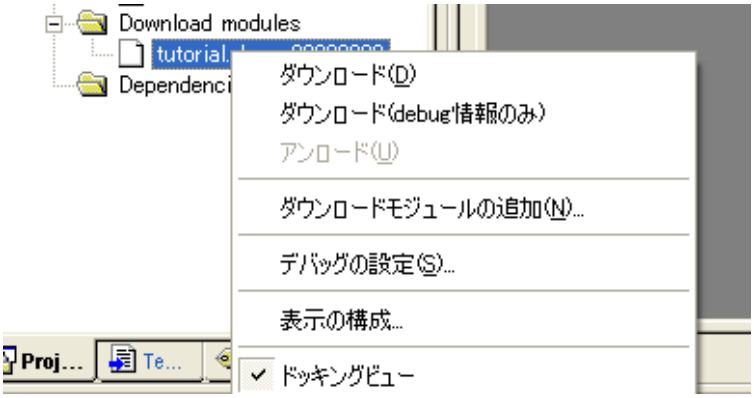


図 4.18　[Workspace]ウィンドウダウンロード画面([Project Files])

【留意事項】

1. 複数のロードモジュールをダウンロードする場合は、[デバッグ]メニューの[ダウンロード]→[All Download modules]を選択してください。
2. ソースレベルで同期デバッグを行うためには、各コア用プロジェクトでそれぞれ別にデバッグ情報ファイルをロードしてください。
   同一ファイル名のモジュールを登録している場合は、同期ダウンロードが可能です。
   同期ダウンロードに関しては、「5.1　同期デバッグを設定する」を参照してください。

4.4　　デバッグセッション

High-performance Embedded Workshop は、ビルダオプションをコンフィグレーションへ保存することができます。

同様に、High-performance Embedded Workshop は、デバッガオプションをセッションに保存することもできます。セッションには、デバッギングプラットフォーム、ダウンロードするプログラム、各デバッギングプラットフォームのオプションを保存することができます。

セッションは、コンフィグレーションとは直接関連がありません。

これは、複数のセッションが同じダウンロードモジュールを共有し、プログラムの不要なリビルドを避けられることを意味します。

各セッションのデータは、別々のファイルで High-performance Embedded Workshop プロジェクトに保存します。詳細については、以下で説明します。

4.4.1 セッションを選択する

セッション選択するには、次の 2 通りの方法があります。

- ツールバーから選択する

1. ツールバーのドロップダウンリストボックス（図4.19）からセッションを選んでください。

図 4.19 ツールバーの選択

- ダイアログボックスから選択する

1. [デバッグ->デバッグセッション...]を選んでください。
[デバッグセッション]ダイアログボックスを表示します（図 4.20）。

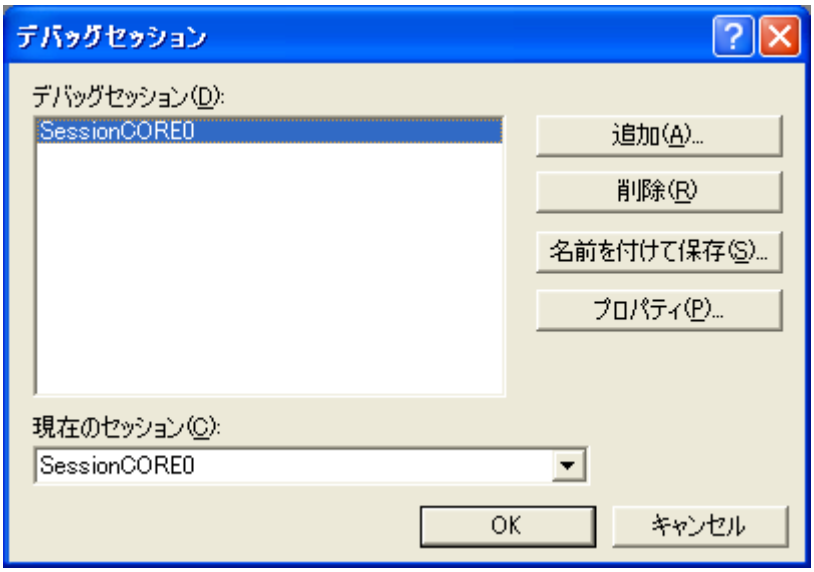


図 4.20 [デバッグセッション]ダイアログボックス

2. [現在のセッション]ドロップダウンリストボックスから使用したいセッションを選んでください。
3. [OK]ボタンをクリックして、セッションを設定してください。

4.4.2　セッションの追加と削除

別のセッションから設定をコピーしたり、セッションを削除したりして、新しいセッションを追加することができます。

- 新しい空のセッションを追加する

1. [デバッグ>デバッグセッション...]を選んでください。[デバッグセッション]ダイアログボックスを表示します（図 4.20）。
2. [追加...]ボタンをクリックしてください。[新規セッション追加]ダイアログボックスを表示します（図 4.21）。
3. [新規セッションの追加]ラジオボタンをチェックしてください。
4. セッションの名前を入力してください。
5. [OK]ボタンをクリックし、[デバッグセッション]ダイアログボックスを閉じてください。
6. 入力したセッション名のファイルを新しく作成します。ファイルが既に存在する場合は、エラーを表示します。

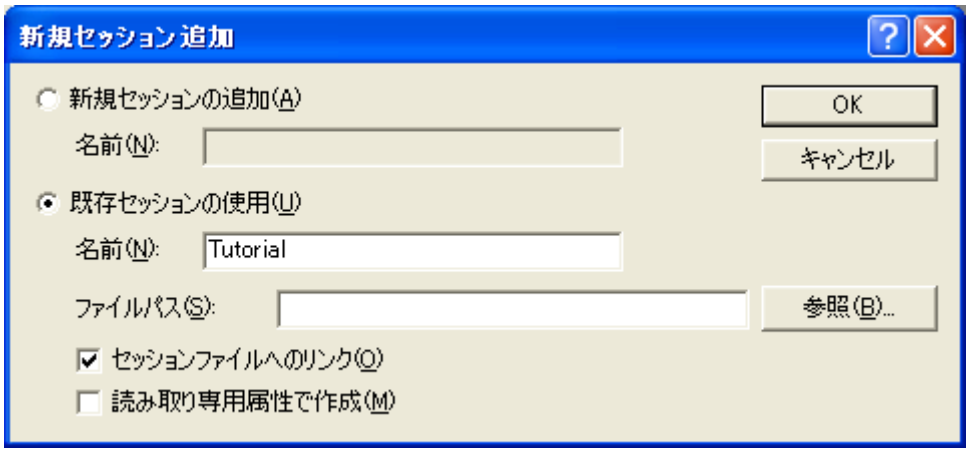


図 4.21　[新規セッション追加]ダイアログボックス

- 既存のセッションを新しいセッションファイルにインポートする
  1. [デバッグ>デバッグセッション...]を選んでください。[デバッグセッション]ダイアログボックスを表示します（図 4.20）。
  2. [追加...]ボタンをクリックしてください。[新規セッション追加]ダイアログボックスを表示します（図 4.21）。
  3. [既存セッションの使用]ラジオボタンをチェックしてください。
  4. セッションの名前を入力してください。
  5. 現在のプロジェクトにインポートしたい既存のセッションファイルを入力するか、[参照...]ボタンをクリックして選択してください。
     [セッションファイルへのリンク]チェックボックスをチェックしない場合、プロジェクトディレクトリにインポートした新しいセッションファイルを生成します。
     [セッションファイルへのリンク]チェックボックスをチェックした場合、プロジェクトディレクトリに新しいセッションファイルは生成せず、既存のセッションファイルにリンクします。
     [読み取り専用属性で作成]チェックボックスをチェックした場合、リンクしたセッションファイルをリードオンリーで使用します。
  6. [OK]ボタンをクリックし、[デバッグセッション]ダイアログボックスを閉じてください。

- セッションを削除する
  1. [デバッグ>デバッグセッション...]を選んでください。[デバッグセッション]ダイアログボックスを表示します（図 4.20）。
  2. 削除したいセッションを選んでください。
  3. [削除]ボタンをクリックしてください。現在のセッションを削除することはできません。
  4. [OK]ボタンをクリックし、[デバッグセッション]ダイアログボックスを閉じてください。

- セッションのプロパティを見る
  1. [デバッグ>デバッグセッション...]を選んでください。[デバッグセッション]ダイアログボックスを表示します（図 4.20）。
  2. 見たいプロパティのあるセッションを選んでください。
  3. [プロパティ]ボタンをクリックしてください。
     [セッションプロパティ]ダイアログボックスを表示します（図 4.22）。

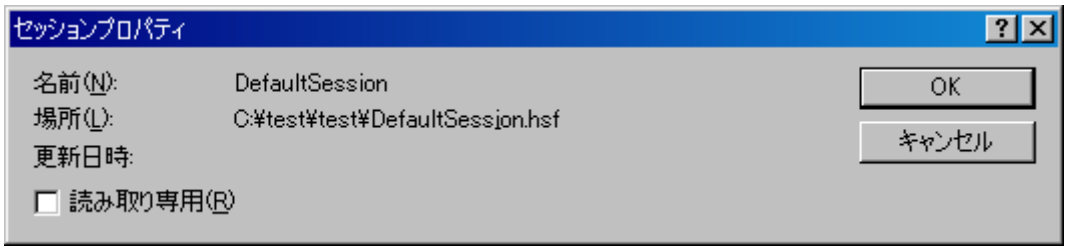


図 4.22　[セッションプロパティ]ダイアログボックス

- セッションをリードオンリーにする
  1. [デバッグ>デバッグセッション...]を選んでください。
     [デバッグセッション]ダイアログボックスを表示します（図 4.20）。
  2. リードオンリーにしたいセッションを選んでください。
  3. [プロパティ]ボタンをクリックしてください。
     [セッションプロパティ]ダイアログボックスを表示します（図 4.22）。
  4. [読み取り専用]チェックボックスをチェックしてください。
     リンクをリードオンリーにします。これは、デバッガ設定ファイルを共有する場合、およびデータを間違って修正したくない場合に便利です。
  5. [OK]ボタンをクリックしてください。

- セッションを別名で保存する
  1. [デバッグ>デバッグセッション...]を選んでください。
     [デバッグセッション]ダイアログボックスを表示します（図 4.20）。
  2. 保存したいセッションを選んでください。
  3. [名前を付けて保存]ボタンをクリックしてください。
     [セッションの保存]ダイアログボックスを表示します（図 4.23）。
  4. 新しいファイルを保存する場所を指定してください。
  5. セッションファイルを別の場所へエクスポートしたい場合は、[プロジェクトとのリンク]チェックボックスをチェックしないでください。現在のセッションの場所の代わりに、この場所を High-performance Embedded Workshop で使用したい場合は、[プロジェクトとのリンク]チェックボックスをチェックしてください。
  6. [保存]ボタンをクリックしてください。

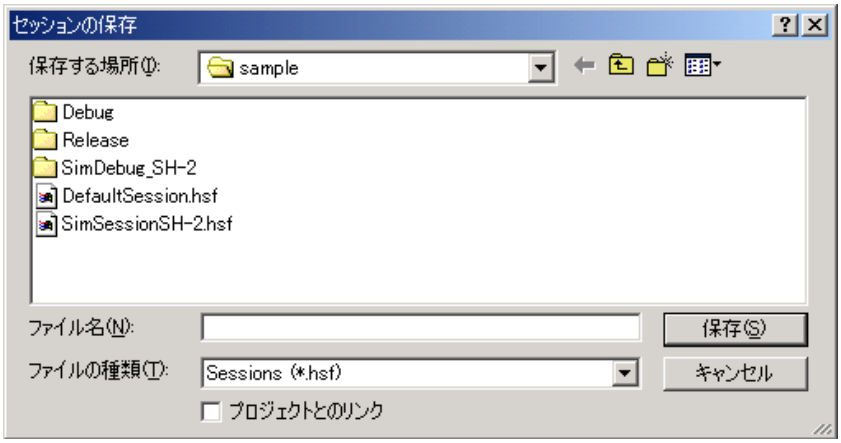


図 4.23 [セッションの保存]ダイアログボックス

4.4.3 セッション情報を保存する

➲セッションを保存するには

1. [ファイル->セッションの保存]を選んでください。

4.5　エミュレータの接続

エミュレータの接続には、以下の方法があります。

(1) E10A-USB 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]メニューの[デバッグの設定]を選択し、[デバッグの設定]ダイアログボックスを開いてください。ここで、ダウンロードモジュールや起動時に自動的に実行するコマンドチェインなどを登録することができます。

[デバッグの設定]ダイアログボックスの詳細については、「4.3　E10A-USB エミュレータ起動時の設定」を参照してください。

[デバッグの設定]ダイアログボックスの設定終了後、ダイアログボックスを閉じると、E10A-USB エミュレータが接続されます。

(2) E10A-USB エミュレータ起動時の設定を行わずに簡単に接続する方法

E10A-USB エミュレータを使用する設定があらかじめ登録されているセッションファイルに切り替えることにより、E10A-USB エミュレータを簡単に接続できます。

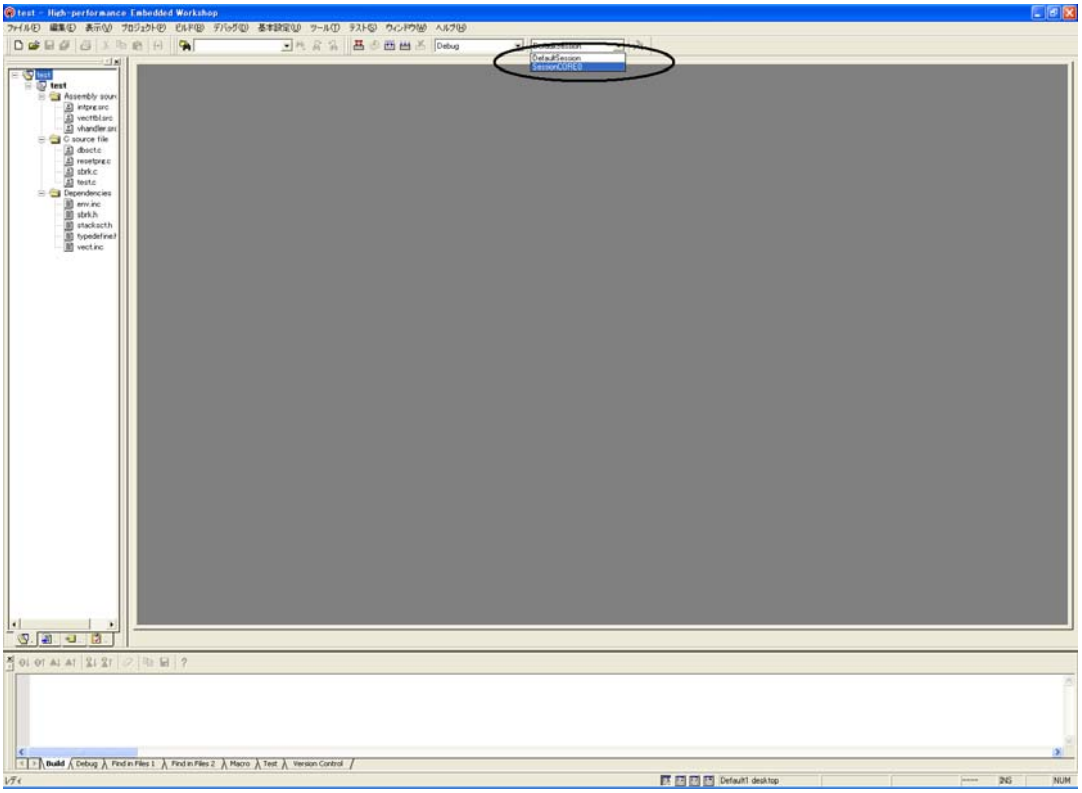


図 4.24　　セッションファイルの選択

上記図中の、丸印の中にあるリストボックスから、「図 4.9　[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」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、E10A-USB エミュレータを使用する設定が登録されています。
選択終了後、E10A-USB エミュレータが自動的に接続されます。
セッションファイルについての詳細は、「4.4　デバッグセッション」を参照してください。

4.6　エミュレータの再接続

エミュレータ切断状態時に以下の方法で再接続を行うことができます。

[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】　1. [デバッグ>デバッグの設定]から開く[デバッグの設定]ダイアログボックス（図 4.16　[デバッグの設定]ダイアログボックス（[ターゲット]ページ）参照)の[ターゲット]ドロップダウンリストボックスに E10A-USB エミュレータが選択されている必要があります。
2.　同期デバッグ時は、全てのコアを切断後に再接続を行ってください。

4.7　エミュレータの終了

ツールチェインをご使用の場合、エミュレータの終了方法は 2 通りあります。

- 起動中のエミュレータの接続を解除する方法
- 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]メニューから接続解除を選択するか、接続解除ツールバーボタン  をクリックしてください。

【注】　プログラム実行中や、ダイアログ表示中の CPU がある場合接続解除を選択しないでください。

(2) High-performance Embedded Workshop 自体を終了する方法
[ファイル]メニューから[アプリケーションの終了]を選択してください。

メッセージボックスが表示されます。必要なら、[はい]ボタンをクリックし、セッションをセーブしてください。セーブ後、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。

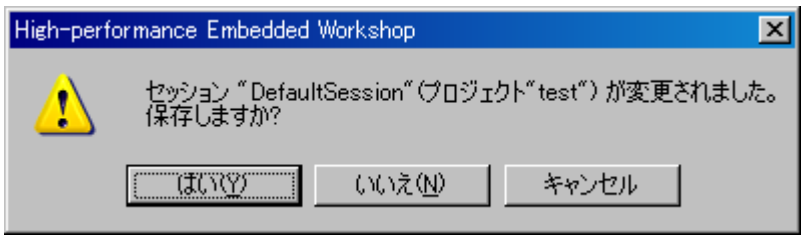


図 4.25　メッセージボックス

5. デバッグ

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

5.1 同期デバッグを設定する

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。

操作方法など同期デバッグ機能に関する詳細につきましては、High-performance Embedded Workshop V.4.09 ユーザーズマニュアル「18. 同期デバッグ機能」を参照してください。

5.1.1 [同期デバッグ]ダイアログボックスを開く

[デバッグ]メニューから[同期デバッグ]選択し、[同期デバッグ]ダイアログボックスを開きます。また、同期デバッグの履歴がある場合は、High-performance Embedded Workshop の[ようこそ！]ダイアログボックスから開くことも可能です。

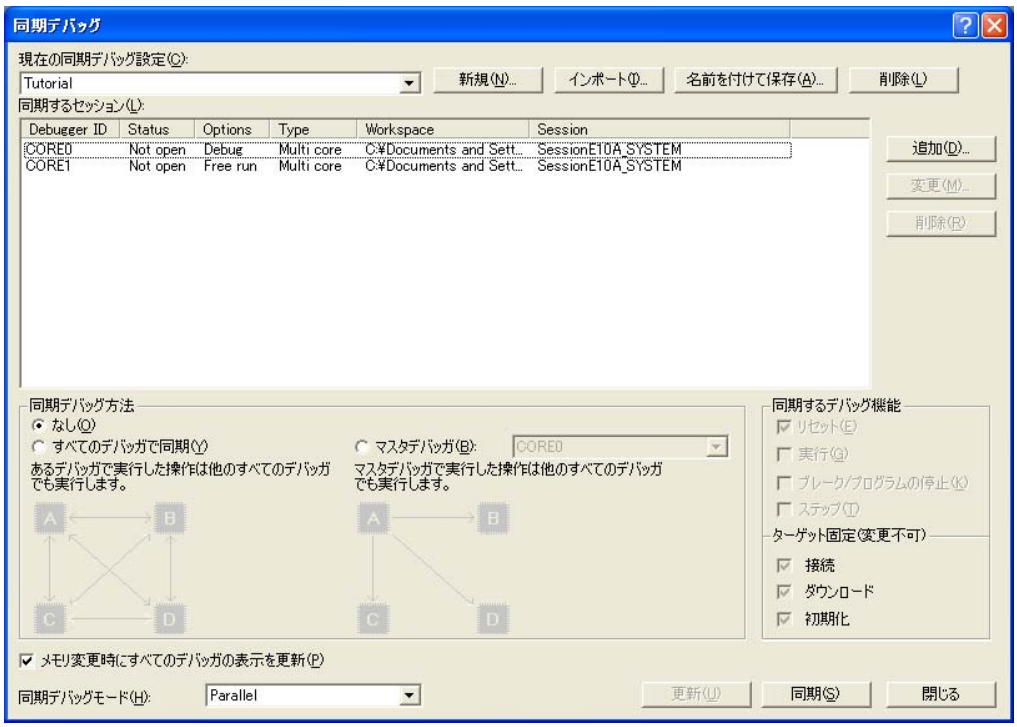


図 5.1 [同期デバッグ]ダイアログボックス

5.1.2 [同期するセッション]リストボックス

[同期するセッション]リストボックスでは、同期するセッションについての情報を設定します。設定できる項目は以下の通りです。

Debugger ID	独自の ID を設定します。デバッグセッションを特定できるものを設定してください。【注 1】	
Status	セッションの状態を表示	
	Not open:	セッションは現在 High-performance Embedded Workshop アプリケーションで開いていません。[同期]ボタンがクリックされたときに開きます。
	Not connect:	セッションは現在 E10A-USB エミュレータと接続していません。
	Break:	セッションは、E10A-USB エミュレータと接続しています。ユーザプログラムはブレーク中です。
	Running:	セッションは、E10A-USB エミュレータと接続しています。ユーザプログラムは実行中です。
Options	同期デバッグ開始時に、セッションがどのように扱うかを指定します。	
	Debug: セッションをデバッグ対象として扱います。	
	Freerun: セッションをデバッグ対象として扱いません。 ＣＰＵコアの動作は、エミュレータを接続しない場合と同様です。	
Type	セッションと関連するプラットフォームの型を表示します。【注 2】	
	Single core:	シングルコア用プラットフォーム
	Multi core:	マルチコア用プラットフォーム
Workspace	セッションを含むワークスペースファイルの絶対パスファイル名を表示します。	
Session	ワークスペース内のセッション名を表示します。	

【注】 1. 異なるデバイス用のセッションを混在させないでください。
2. シングルコア用プラットフォームとマルチコア用プラットフォームを混在させないでください。

5.1.3 [同期デバッグ方法]グループボックス

[同期デバッグ方法]グループボックスでは、同期動作の方向を設定します。
設定できる項目は以下の通りです。

なし	同期を行いません。 全てのセッションは独立に動作します。
すべてのデバッガで同期	[同期するデバッグ機能] グループボックスでチェックされた機能に関して、全てのセッション間で双方向に同期を行います。
マスタデバッガ	[同期するデバッグ機能] グループボックスでチェックされた機能に関して、指定したセッションをマスタとし、他の全てのセッションへ単方向の同期を行います。ドロップダウンリストボックスより、マスタとするセッションの [Debbuger ID] を選択してください。

5.1.4 [同期デバッグ機能]グループボックス

[同期デバッグ機能]グループボックスでは、同期させる動作を設定します。
設定できる項目は以下の通りです。

リセット	[CPU のリセット] 機能、[リセット後実行] 機能を同期します。 [リセット後実行] 機能は、[実行] チェックボックスもチェックされている必要があります。
実行	[実行] 機能、[リセット後実行] 機能を同期します。 [リセット後実行] 機能は、[リセット] チェックボックスもチェックされている必要があります。
ブレーク/プログラムの停止	すべての要因によるデバイスのブレークおよび、[プログラムの停止] 機能を同期します。ブレーク種別毎に同期を設定することはできません。
ステップ	各種ステップ機能を同期します。 他コアがユーザプログラム実行中に同期ステップを行った場合、ステップ実行終了時の他コア側動作は、[ブレーク/プログラムの停止] 設定に依存します。
接続	すべてのセッションで E10A-USB エミュレータの [接続] を同期します。
ダウンロード	すべてのセッションで [ダウンロード] 機能を同期します。ダウンロードモジュールのファイル名が同一の場合のみ同期します。
初期化	すべてのセッションで [初期化] 機能を同期します。

【注】　設定可能な機能はデバッグ対象のデバイスにより異なりますので、別冊の「SHxxxx　ご使用時の補足説明　2.2.1 同期デバッグ機能」を参照してください。

5.1.5 [メモリ更新]オプション

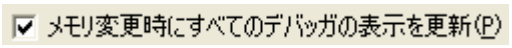


図 5.2　[メモリ更新]オプション

このオプションがチェックされている場合、全セッションのメモリデータ（メモリウィンドウ、ウォッチウィンドウ等）を表示するすべての High-performance Embedded Workshop のウィンドウは、同期デバッグ中のセッションのいずれかのメモリが変更された時、必ず更新されます。【注】
オプションが選択されていない場合、メモリが変更された時にはローカルセッションのメモリウィンドウのみが更新されます。セッション間でメモリを共有している場合、他のセッションのメモリ関連ウィンドウで正しいメモリ内容を表示するには、それらのウィンドウでマニュアルリフレッシュを実行する必要があります。

【注】ユーザプログラム実行中のセッションに対しても、メモリリードを行うため、ショートブレークが発生します。

5.1.6 [同期デバッグモード]ドロップダウンリストボックス

[同期デバッグモード]ドロップダウンリストボックスでは、同期デバッグ時に使用する High-performance Embedded Workshop の状態を設定します。

Internal	1 つの High-performance Embedded Workshop で指定したすべてのセッションを開きます。このモードはすべてのセッションが同じワークスペースの中にある場合のみ選択可能です。
Parallel	指定したセッション毎に異なる High-performance Embedded Workshop で開きます。

5.2 エミュレーション環境を設定する

この節では、エミュレーションを行うための環境を設定する方法を説明します。

5.2.1 [Configuration]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、[Emulator System]ツールバーボタン をクリックすると、[Configuration]ダイアログボックスが開きます。

5.2.2 [General]ページ

[General]ページでは、E10A-USB エミュレータの基本設定（固有部）を行います。
本設定は CPU 毎に設定が可能です。

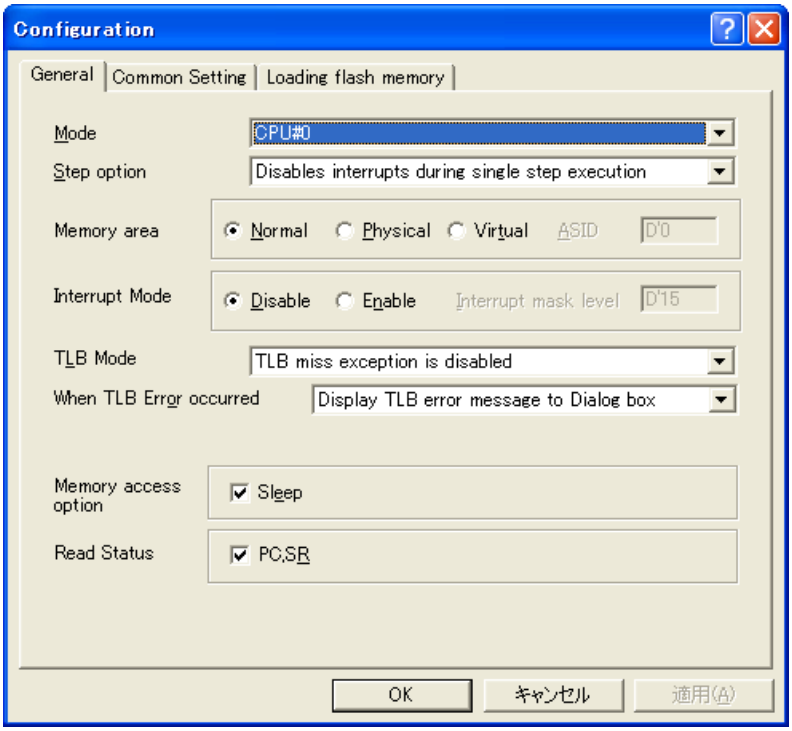


図 5.3 [Configuration]ダイアログボックス（[General]ページ）

設定できる項目は以下の通りです。

[Mode]	CPU 番号を表示します。
[Step option]	ステップ中の割り込みの開放/マスクを設定します。 Disable interrupts during single step execution：ステップ開始時に割り込み【注 1】を受け付けません。 Enable interrupts during single step execution：ステップ開始時に割り込み【注 1】を受け付けます。
[Reset assert(Auto Connect)]	E10A-USB エミュレータ接続時または、[CPU のリセット] 機能、[リセット後実行] 機能使用時、E10A-USB エミュレータからリセット信号を発行します。

注意

別冊の SHxxxx ご使用時の補足説明「1.5　H-UDI ポートコネクタとチップ間の推奨接続例」の記載通りに結線されていない場合は、絶対に[Reset assert(Auto Connect)]オプションを使用しないでください。ユーザシステムの故障につながります。

【注】1. ブレーク中に発生した割り込みも含みます。

【注】　本ダイアログボックスで設定可能な項目はご使用のエミュレータにより異なります。詳細につきましては、オンラインヘルプをご参照ください。

ご使用のデバイスによっては下記の[General]ページでは、E10A-USB エミュレータの基本設定（固有部）を行います。

5.2.3 [Common Setting]ページ

[Common Setting]ページでは、E10A-USB エミュレータの基本設定（共有部）を行います。本設定は各 CPU で共有されます。

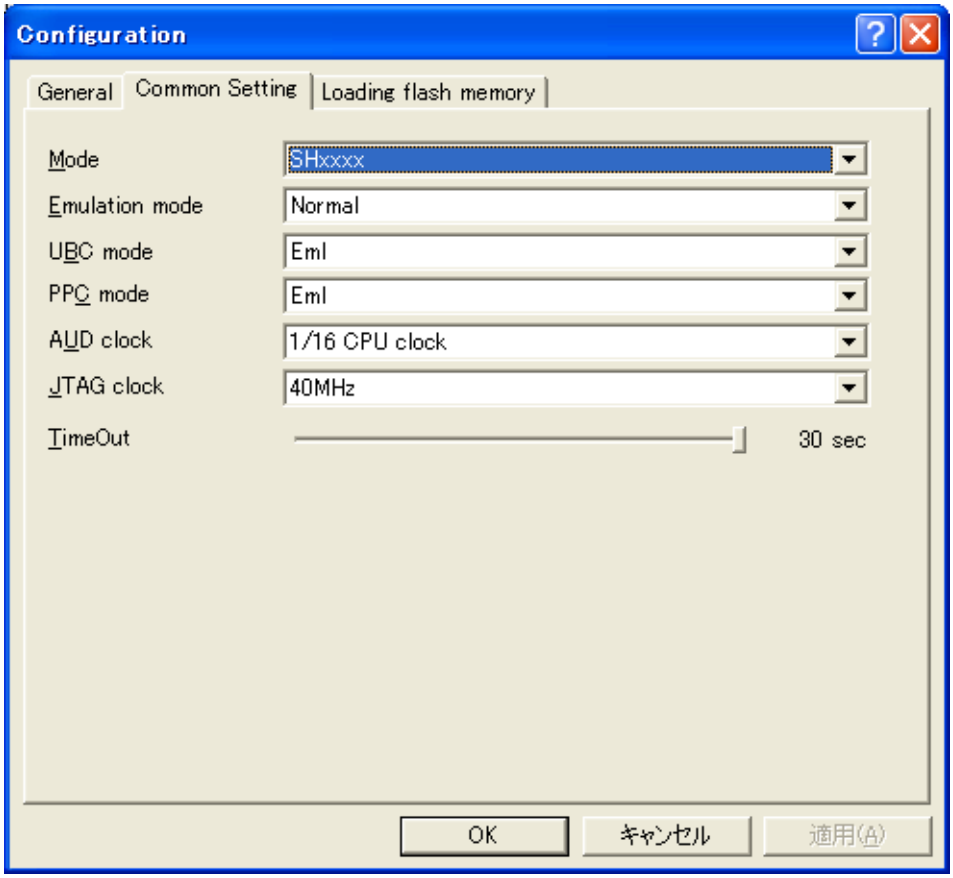


図 5.4 [Configuration]ダイアログボックス（[General]ページ）

設定できる項目は以下の通りです。

[Mode] マイコン名を表示します。

[Emulation mode] ユーザプログラム実行時のエミュレーションモードを選択します。
Normal：通常の実行を行います。
No break：PC ブレークポイント、ハードウェアブレークポイントを一時的に無効にしてユーザプログラムを実行します。

[UBC mode] UBC モードの設定を行います。
Eml：E10A-USB エミュレータにより、Event Condition として使用します。
User：UBC をユーザに開放します。

[PPC mode] PPC モードの設定を行います。
Eml：E10A-USB エミュレータにより、[パフォーマンス解析機能] として使用します。
User：PPC をユーザに開放します。【注】

[AUD clock] AUD トレース取得時のクロックです。
周波数が低いと、リアルタイムトレース使用時にデータ抜けの発生頻度が高くなります。
周波数は、サポートデバイスの AUD clock 上限を超えないように設定してください。
各デバイスの AUD clock 上限は、別冊の SHxxxx ご使用時の補足説明「2.2.4 JTAG（H-UDI）クロック（TCK）AUD クロック（AUDCK）使用時の注意事項」を参照してください。

[JTAG clock] AUD トレース以外の通信クロックです。
周波数が低いと、ダウンロードが遅くなります。
周波数は、サポートデバイスの TCK 上限を超えないように設定してください。
各デバイスの TCK 上限は、別冊の SHxxxx ご使用時の補足説明「2.2.4 JTAG (H-UDI) クロック (TCK) AUD クロック (AUDCK) 使用時の注意事項」を参照してください。

[Timeout] タイムアウトエラー発生までの待ち時間を設定します。
3 秒～30 秒までの範囲で 3 秒刻みで設定が可能です。

【注】 ”User”を選択した場合、使用できない機能があります。
使用できない機能は製品によって異なりますので、オンラインヘルプを参照してください。

【注】 本ダイアログボックスの有無および設定可能な項目はご使用のデバイスによって異なります。
詳細につきましては、オンラインヘルプをご参照ください。

5.2.4 フラッシュメモリへダウンロードする

[Loading flash memory]ページでは、外部フラッシュメモリへのダウンロードを行う場合の設定を行います。

ご使用のデバイスによっては、本機能は使用できません。

詳細につきましては、「6.23、7.25 フラッシュメモリへのダウンロード機能」を参照してください。

本設定は CPU 毎に設定が可能です。

【注】 各 CPU で別々のデータを外部フラッシュメモリへダウンロードする場合は、同期ダウンロード機能は使用しないでください。

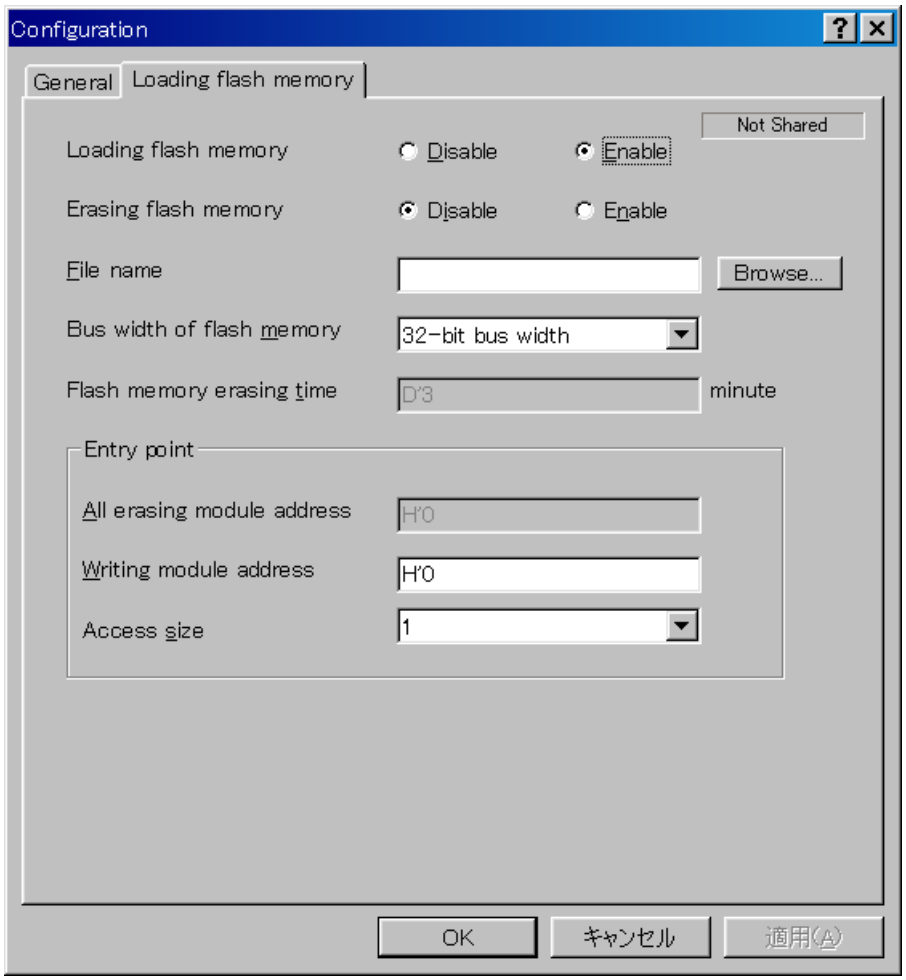


図 5.5 [Configuration]ダイアログボックス（[Loading flash memory]ページ）

設定できる項目は以下の通りです。

[Loading flash memory]
フラッシュメモリへのダウンロードを行う場合、Enable にします。
Enable 時は、High-performance Embedded Workshop 上でダウンロードを行う場合、常にライトモジュールを呼び出します。
Disable　：フラッシュメモリへのダウンロードを行いません。
Enable　：フラッシュメモリへのダウンロードを行います。

[Erasing flash memory]
フラッシュメモリへの書き込みを行う前に消去を行う場合、Enable にします。
Disable　：フラッシュメモリの消去を行いません。
Enable　：フラッシュメモリの消去を行います。

[File name]
ライト/消去モジュール名を設定します。設定したファイルは、フラッシュメモリへロードする前に RAM 領域へロードします。

[Bus width of flash memory]
フラッシュメモリのバス幅の設定を行います。

[Flash memory erasing time]
フラッシュメモリ消去時の TIMEOUT 値を設定します。デフォルトは 3 分となっていますが、消去に時間がかかる場合は値を大きくしてください。
設定できる値は、最小：D'0、最大：D'65535 です。正の整数値のみ入力可能です。

[Entry point]
ライト/消去モジュールの呼び出し先アドレス/アクセスサイズを設定します。
（RAM アドレスである必要があります。）
[All erasing module address]　：消去モジュールの呼び出し先アドレスを入力します。
[Writing module address]　：ライトモジュールの呼び出し先アドレスを入力します。
[Access size]　：ライト/消去モジュールをロードする RAM 領域のアクセスサイズを選択します。

5.3 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コードとして見る方法を説明します。

【注】 ブレークを検出すると、High-performance Embedded Workshop はプログラムカウンタ(PC)の場所を表示します。多くの場合、例えば、Elf/Dwarf2 をベースにしたプロジェクトがもとのパスから移動した場合、ソースファイルを自動的に見つけることができない場合があります。この場合、High-performance Embedded Workshop はソースファイルブラウザダイアログボックスを開くので、ユーザは手動でファイルを探すことができます。

5.3.1 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、[Workspace]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】 1. プログラムをダウンロードする場合、ロードモジュールとして High-performance Embedded Workshop に登録する必要があります。
登録方法については、「4.3 E10A-USB エミュレータ起動時の設定」を参照してください。
2. 外部 RAM にプログラムをダウンロードする場合は、必ずダウンロード対象領域のバスコントローラの設定を行ってから、ダウンロードを実行してください。
特に SDRAM の初期化、バス幅の設定がターゲットシステムに合っているかを十分にご確認ください。
3. 各 CPU でそれぞれソースレベルデバッグを行うためには、同期ダウンロード機能を使用するか、それぞれ別にデバッグ情報ファイルをロードしてください。

5.3.2 ソースコードを表示する

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

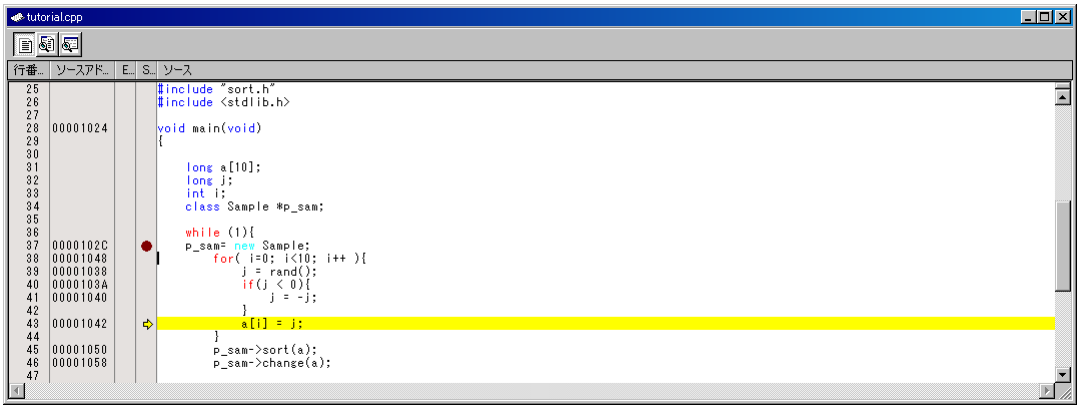


図 5.6　[エディタ]ウィンドウ

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	(Source address カラム)	アドレス情報
2 列目	(Event カラム)	イベント情報(イベントコンディション)
3 列目	(S/W ブレークポイントカラム)	PC、ブックマーク、ブレークポイント情報

Source address カラム

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

Event カラム

Event カラムには下記を表示します。

- イベントコンディションのアドレス条件を設定します。
  アドレス条件が設定できるイベントコンディションチャネルの本数分設定可能です。
  本数は製品によって異なります。

Event カラムをダブルクリックすることによって、上記のビットマップが現れます。

この設定は、ポップアップメニューからも可能です。

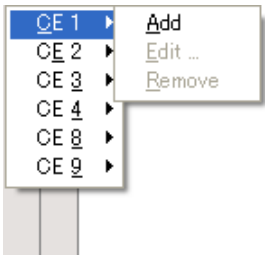


図 5.7　ポップアップメニュー

【留意事項】

[Edit]メニューや[Eventpoint]ウィンドウによって、各チャネルの条件にアドレス条件以外を追加した場合、Event カラムの表記は消えます。

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

 　PC Break を設定している

 　PC 位置

➲すべてのソースファイルでカラムをオフにするには

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

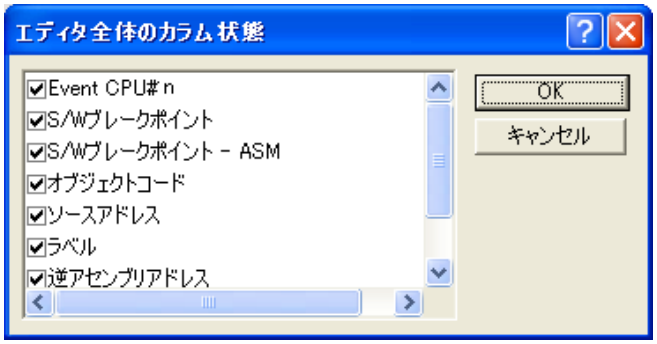


図 5.8　[エディタ全体のカラム状態]ダイアログボックス

➲1 つのソースファイルでカラムをオフにするには

1. 削除したいカラムのあるソースファイルを開き、[編集]メニューをクリックしてください。
2. [カラム]メニュー項目をクリックしてください。
   カスケードしたメニュー項目が現れます。各カラムを、このポップアップメニューに表示します。カラムが有効である場合、名前の横にチェックマークがあります。エントリをクリックすると、カラムの表示、非表示を切り替えます。

5.3.3 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ときは、ウィンドウ上部の[逆アセンブリ]ツールバーボタンをクリックしてください。

開いているソースファイルに対応するアセンブリ言語コードを表示します。

ソースファイルが存在しなくてもアセンブリ言語レベルでコードを表示したい場合は、[表示]->[逆アセンブリ...]を選択するか、[逆アセンブリ]ツールバーボタン  をクリックします。

[逆アセンブリ]ウィンドウは現在の PC の場所で開きます。

また、ディスアセンブルニモニック(可能なときはラベルも一緒に)を表示する[Address]、[Code](オプション)を表示します。

また、[混合表示] ツールバーボタン  を選択すると、ソースとコードの両方を表示することができます。以下は[混合表示]を選択した場合の表示例です。

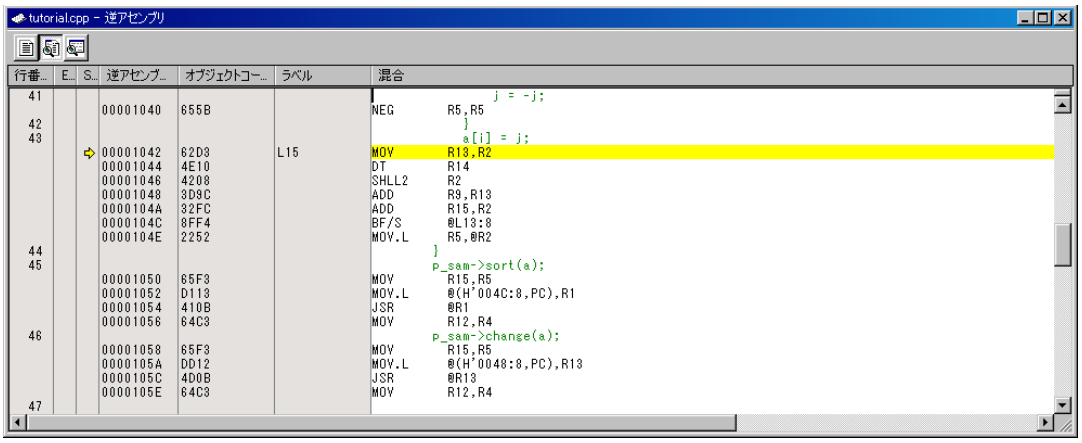


図 5.9 [逆アセンブリ]ウィンドウ

5.3.4 アセンブリ言語コードを修正する

修正したい命令をダブルクリックすることによって、アセンブリ言語コードを修正することができます。[アセンブル]ダイアログボックスが開きます。

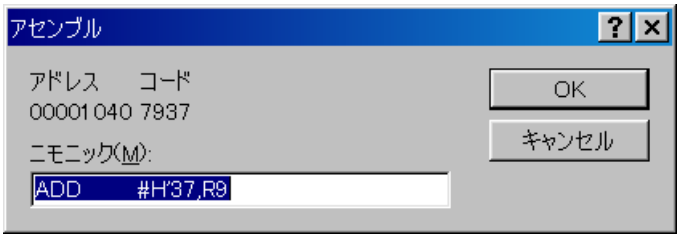


図 5.10　[アセンブル]ダイアログボックス

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

【注】　アセンブリ言語表示は、実際のメモリ上の機械語コードからディスアセンブルします。メモリの内容を修正すると、ダイアログボックス（および[逆アセンブリ]ウィンドウ）には新しいアセンブリ言語コードを表示します。しかし、[エディタ]ウィンドウの表示内容は変更しません。これはソースファイルにアセンブラを含む場合も同じです。

5.3.5 特定のアドレスを見る

[逆アセンブリ]ウィンドウを使って作成したプログラムを見ているとき、プログラム内のほかのところも見たいときがあります。そのような場合、プログラム内のコードをスクロールせずに特定のアドレスに直接行くことができます。[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]を選択します。

図 5.11　[アドレス指定]ダイアログボックス

エディットボックスにアドレスまたはラベル名を入力して、[OK]ボタンをクリックするか”Enter”キーを押します。[逆アセンブリ]ウィンドウを更新して新しいアドレスコードを表示します。

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

5.3.6 現在のプログラムカウンタアドレスを見る

High-performance Embedded Workshop でアドレスまたは値を入力できるところでは、式も入力することができます。先頭にハッシュ文字を付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。したがって、[表示アドレス設定]ダイアログボックスを開いて"#pc"という式を入力すると、[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 のオフセットも表示することができます。

5.4 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。本設定は CPU 毎に設定が可能です。

【注】 デバッグ対象デバイスによっては、本機能はサポートしていません。
各製品の仕様についてはオンラインヘルプを参照してください。

5.4.1 [モニタ]ウィンドウを開く

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

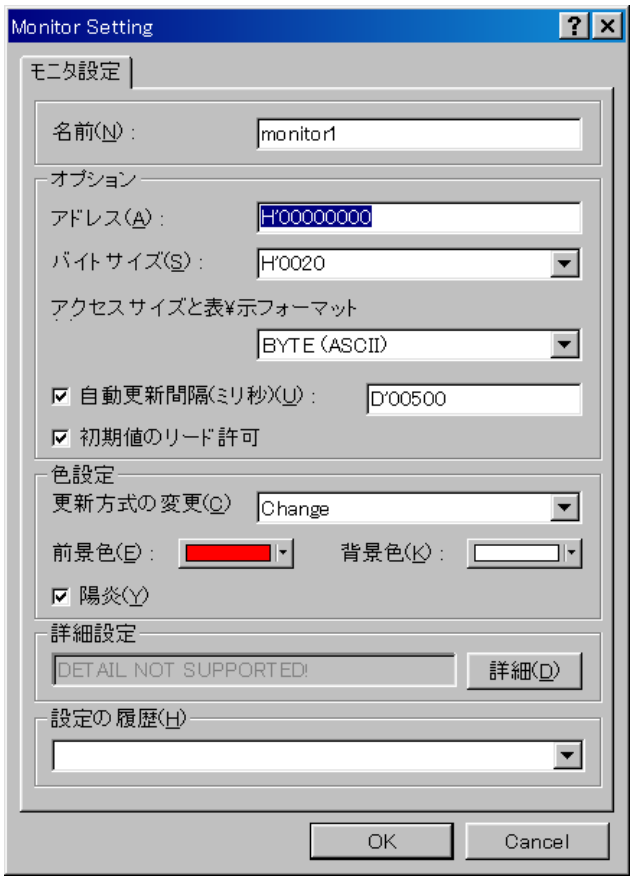


図 5.12 [Monittor Setting]ダイアログボックス

[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。
- [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 change　：色の変更は行いません。
  - Change　：色を変更します。
    色は前景色オプション、背景色オプションで設定します。
  - Gray　：値の変更のないデータを灰色表示します。
  - Appear　：値の変更があると表示します。変更がなければ表示しません。
- [前景色]　表示文字色を設定します。
  ([Change]選択時有効)
- [背景色]　背景色を設定します。
  ([Change]選択時有効)
- [陽炎]　チェックボックスにチェックがある場合、一定間隔更新のないデータの色を背景色オプションで設定した色に戻します。一定間隔とは、モニタ取得間隔の一回分です。
  ([Change], [Gray], [Appear] 選択時有効)

[詳細設定]　E10A-USB エミュレータではサポートしていません。

[設定の履歴]　前回の設定内容を呼び出します。

【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。

設定完了後、[OK]ボタンをクリックすると[モニタ]ウィンドウが開きます。

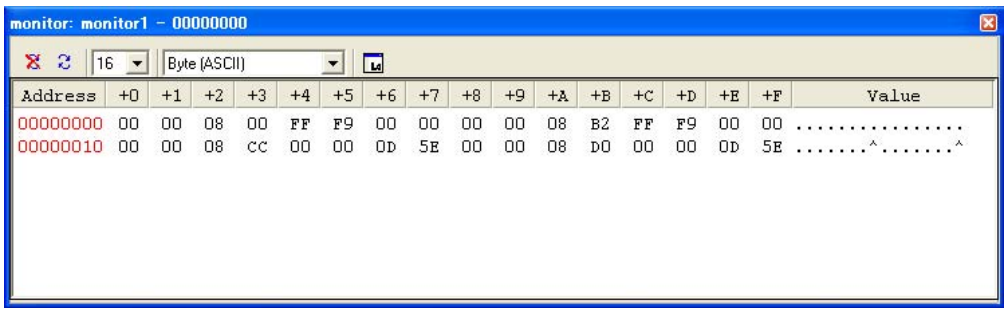


図 5.13　[モニタ]ウィンドウ

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。

5.4.2 モニタの設定内容を変更する

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

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

5.4.3 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。

アドレスの表示文字が黒色となり、表示更新を停止します。

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

5.4.4 モニタ設定を削除する

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

5.4.5 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。

5.4.6 [モニタ]ウィンドウを非表示にする

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

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

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

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

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

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

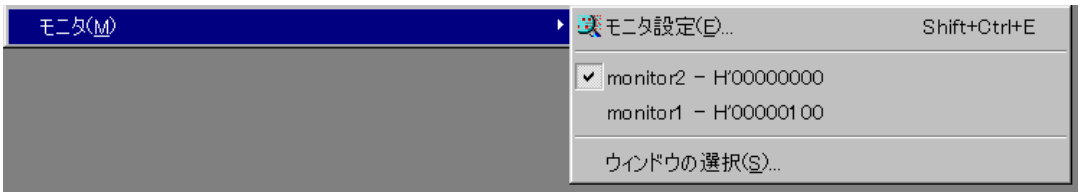


図 5.14　モニタ設定リスト

5.4.7 [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の条件の確認、新規モニタ条件の追加、編集、削除などの操作を連続的に行うことができます。

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

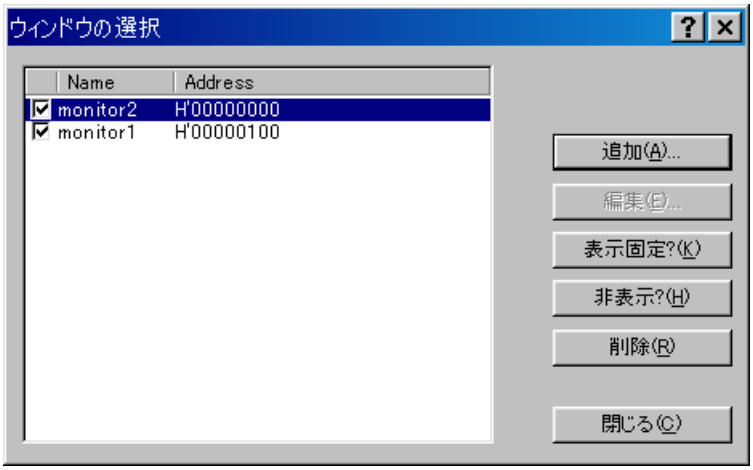


図 5.15　[ウィンドウの選択]ダイアログボックス

5.5 現在の状態を表示する

デバッグプラットフォームの現在の状態を知るには[ステイタス]ウィンドウを表示します。
[ステイタス]ウィンドウを開くには、[表示->CPU->ステイタス]を選択するか、
[ステイタスの表示]ツールバーボタン  をクリックします。

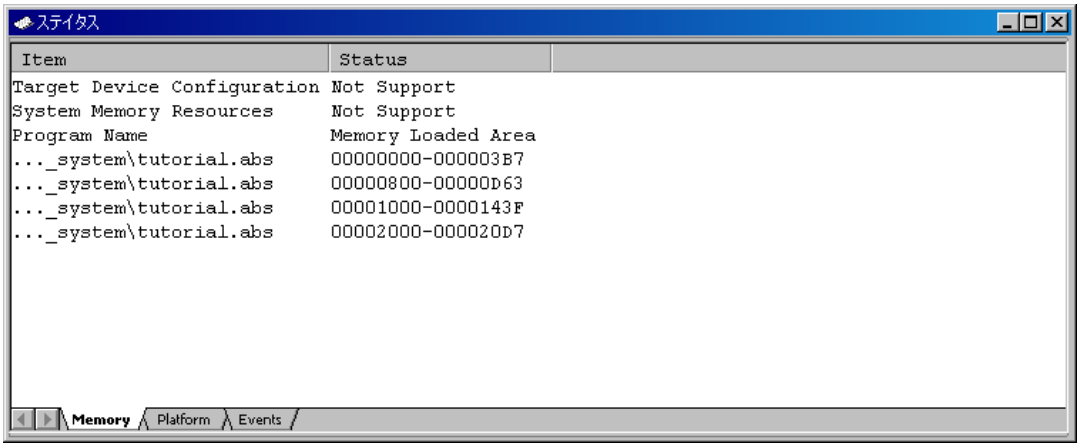


図 5.16　[ステイタス]ウィンドウ

[ステイタス]ウィンドウには、3 枚のシートがあります。

- [Memory]シート
  メモリマッピングおよび現在ロードしたオブジェクト・ファイルが使用するメモリエリアなど、現在のメモリステータスに関する情報を含んでいます。
- [Platform]シート
  CPU 種別および動作モードなど、エミュレータのステイタス情報、実行状態および実行統計情報を含んでいます。
- [Events]シート
  リソース情報およびブレークポイント等のイベント情報に関する情報を含んでいます。

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

5.6 イベントポイントを使用する

E10A-USB エミュレータは High-performance Embedded Workshop 標準の PC ブレークポイントとは別に、より複雑な条件指定によるブレーク、トレース、実行時間測定を行うイベントポイント機能を持っています。

5.6.1 PC ブレークポイントとは

PC ブレークポイントは指定アドレスの命令フェッチが行われた場合にユーザプログラムの実行を停止します。

最大 255 ポイントまで設定できます。

同期実行、同期ステップ、同期ブレークすべてが有効時にのみ設定できます。

本設定は CPU 毎に設定が可能です。

5.6.2 Event condition とは

Event condition は単一アドレス指定以外に、データ条件など、より複雑な条件指定が可能なポイントです。

条件成立時の動作としてユーザプログラムの停止以外にパフォーマンス測定の開始/終了条件として利用可能です。パフォーマンス測定の開始/終了条件とする場合、[パフォーマンス解析]ウィンドウから設定してください。

複数の Event condition を組み合わせることにより、より複雑な条件設定が可能です。

ご使用のデバイスによって、CPU 毎に設定が可能か、共有されるかが異なります。

【注】 1. パフォーマンス測定の開始/終了条件とした場合、ステップ実行はできません。
また、ハードウェアブレーク命令フェッチアドレス条件および PC ブレークポイントで停止後、そのアドレスから実行を再開する場合、シングルステップ機能を使用するため動作できません。ハードウェアブレーク命令フェッチアドレス条件および PC ブレークポイントを解除した上で再開してください。

2. １つのチャネルでブレーク条件とパフォーマンス測定の開始/終了条件を同時に使用することはできません。パフォーマンス測定の開始/終了条件とした場合、ブレーク条件設定は無効となります。

3. 設定できる内容は、製品ごとに異なります。各製品の設定内容については、オンラインヘルプを参照してください。

5.6.3 [イベントポイント]ウィンドウを開く

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

[イベントポイント]ウィンドウには、2 枚のシートがあります。

- [Breakpoint]シート
  PC ブレークポイントの設定内容を表示します。また、PC ブレークポイントの設定、変更および解除を行うことができます。
- [Event condition]シート
  イベントコンディションチャネルの設定内容を表示、設定します。

5.6.4 PC ブレークポイントを設定する

[Breakpoint]シートでは PC ブレークポイントの設定内容の表示、変更および追加ができます。

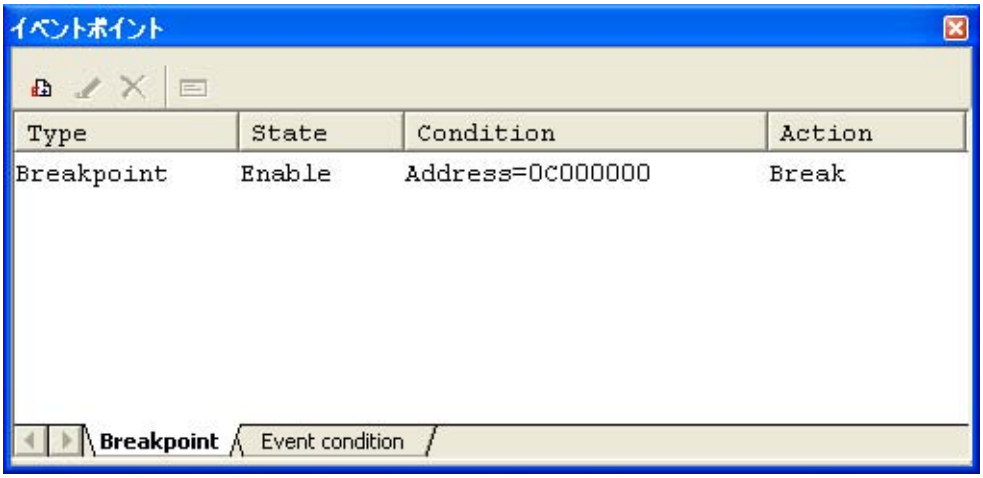


図 5.17　[イベントポイント]ウィンドウ（[Breakpoint]シート）

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

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

【注】 1. PC ブレークポイントは同期実行、同期ステップ、同期ブレークすべてが有効時にのみ設定できます。
2. 設定できる内容は、製品ごとに異なります。各製品の設定内容については、オンラインヘルプを参照してください。

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

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

5.6.5 追加

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

5.6.6 編集

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

5.6.7 有効

選択しているブレークポイントを有効にします。

5.6.8 無効

選択しているブレークポイントを無効にします。無効にした場合は、ブレークポイントはリストには残りますが、指定した条件が一致してもブレークは成立しません。

5.6.9 削除

選択しているブレークポイントを削除します。ブレークポイントを削除しないで、詳細情報は保持したまま、条件が一致してもブレークを成立させないようにするには、Disable オプションを使用します(「5.6.8 無効」参照)。

5.6.10 すべてを削除

全てのブレークポイントを削除します。

5.6.11 ソースを表示

ブレークポイントを 1 つ選択している場合のみ有効です。ブレークポイントのある[エディタ]ウィンドウをオープンします。

5.6.12 [Breakpoint]ダイアログボックス

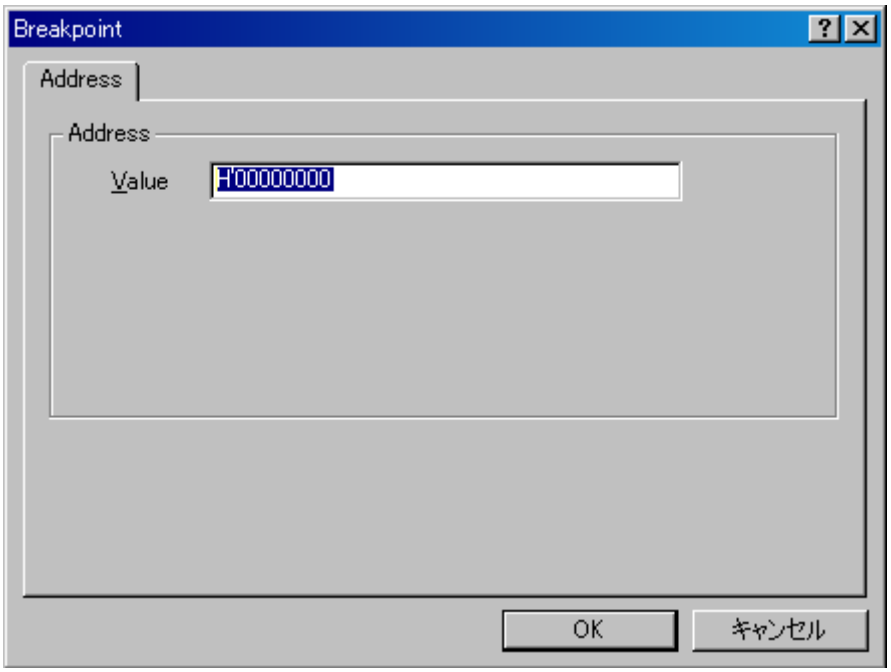


図 5.18 [Breakpoint]ダイアログボックス

本ダイアログボックスでは、ブレーク条件を設定します。
設定するブレークポイントアドレスを [Value]エディットボックスで指定します。
また、#PC のように PC レジスタを指定することも可能です。ブレークポイントは 255 個まで設定できます。
設定できる内容は製品によって異なります。詳しくは、各製品のオンラインヘルプを参照してください。

[Value]の設定時に、アドレスに多重定義関数あるいはメンバ関数を含むクラス名を入力した場合、[Select Function]ダイアログボックスが開くので設定する関数を選択します。
指定したブレーク条件は、[OK]ボタンをクリックすることにより設定します。
[キャンセル]ボタンをクリックすると、設定しないでダイアログボックスを閉じます。

5.6.13 イベントコンディションを設定する

[Event condition]シートではイベントコンディションの設定内容の表示、変更および追加ができます。

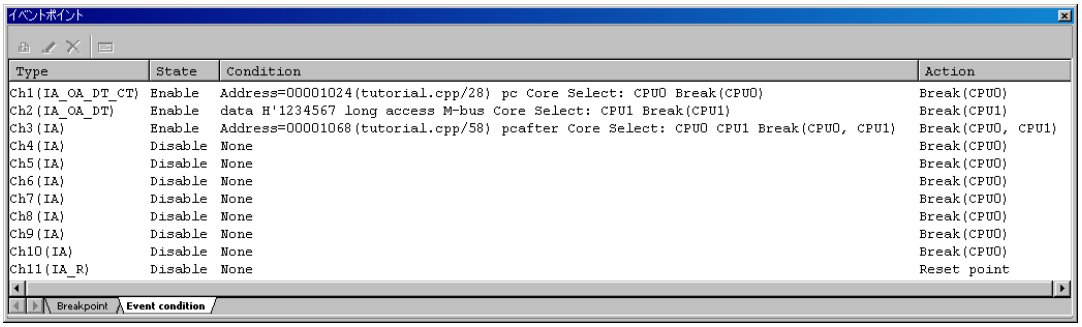


図 5.19 [イベントポイント]ウィンドウ（[Event condition]シート）

イベント条件を表示、設定します。

条件検出チャネル本数や設定できる内容は製品によって違いますので、各製品のオンラインヘルプを参照してください。

シート内に表示する項目は以下の通りです。

[Type] チャネル番号を表示します。

[State] 該当チャネルの有効/無効を示します。

- Enable : 有効
- Disable : 無効

[Condition] 設定されている条件を表示します。表示内容はチャネルにより異なります。

[Action] 条件成立時の動作を表示します。

- Break : 実行停止
- Trace : トレース取得
- Sequential : チャネルの条件がシーケンシャルに成立した場合イベント成立
- PAn_Start_Point : パフォーマンス解析チャネル n の測定を開始
- PAn_End_Point : パフォーマンス解析チャネル n の測定を停止

本ウィンドウでチャネルをダブルクリックすると、[Event condition]ダイアログボックスが開き、条件を変更することができます。

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

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

5.6.14 編集...

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

5.6.15 有効

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

5.6.16 無効

選択しているチャネルを無効にします。無効にした場合は、指定した条件が一致しても［Action］は成立しません。

5.6.17 削除

選択しているチャネルの条件を初期化します。チャネルを初期化しないで、詳細情報は保持したまま、条件が一致しても［Action］を成立させないようにするには、Disable オプションを使用します(「5.6.16 無効」参照)。

5.6.18 すべてを削除

全てのチャネルの条件を初期化します。

5.6.19 ソースを表示

チャネルを 1 つ選択している場合のみ有効です。チャネルのある[エディタ]ウィンドウをオープンします。
チャネルにアドレス値が設定されていない場合は使用できません。

5.6.20 シーケンシャル設定

チャネルのシーケンシャル条件を設定します。

5.6.21 イベントコンディションの編集

PC ブレークポイント、イベントコンディションに対する設定以外の操作方法はすべて共通となっています。
以下イベントコンディションを例に設定以外の操作方法について説明します。

5.6.22 イベントコンディションの設定内容を変更する

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

5.6.23 イベントコンディションを有効にする

イベントコンディションを選択後ポップアップメニューから[有効]を選択すると、選択しているイベントコンディションを有効にします。

5.6.24 イベントコンディションを無効にする

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

5.6.25 イベントコンディションを削除する

イベントコンディションを選択後ポップアップメニューから[削除]を選択すると、選択しているイベントコンディションを削除します。イベントコンディションを削除しないで、詳細情報は保持したまま、条件が成立してもイベントを発生させないようにするには、[無効]オプションを使用します(「5.6.24 イベントコンディションを無効にする」参照)。

5.6.26 イベントコンディションをすべて削除する

ポップアップメニューから[すべて削除]を選択すると、すべてのイベントコンディションを削除します。

5.6.27 イベントコンディションのソース行を表示する

イベントコンディションを選択後ポップアップメニューから[ソースを表示]を選択すると、ブレークポイントのある[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。[ソースを表示]メニューは対応するソースファイルを持つイベントコンディションを 1 個選択しているときのみ有効となります。

5.7 トレース情報を見る

Trace 機能の説明は、「2.2 トレース機能」を参照してください。

5.7.1 [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。

5.7.2 トレース情報を取得する

[トレース]ウィンドウのポップアップメニューに[設定...]メニューがあります。[設定...]メニューを選択すると[Acquisition]ダイアログボックスが表示されます。このダイアログボックス内の[Trace Type]で[I-Trace]を選択すると内蔵トレース機能を使用してトレース情報を取得します。

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。

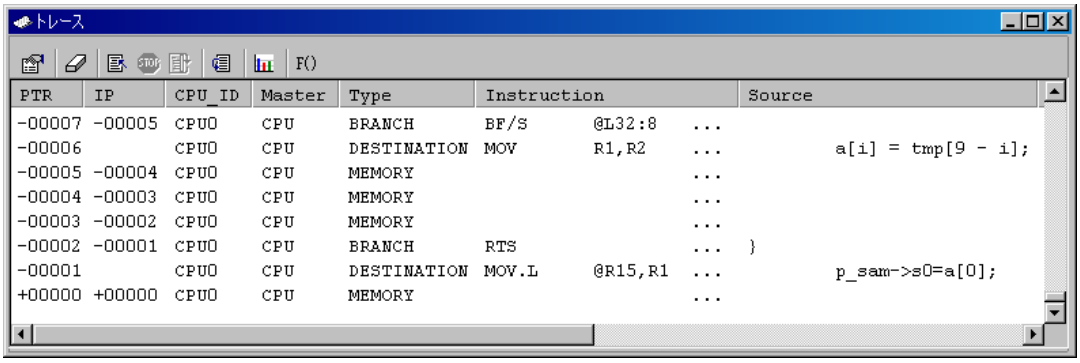
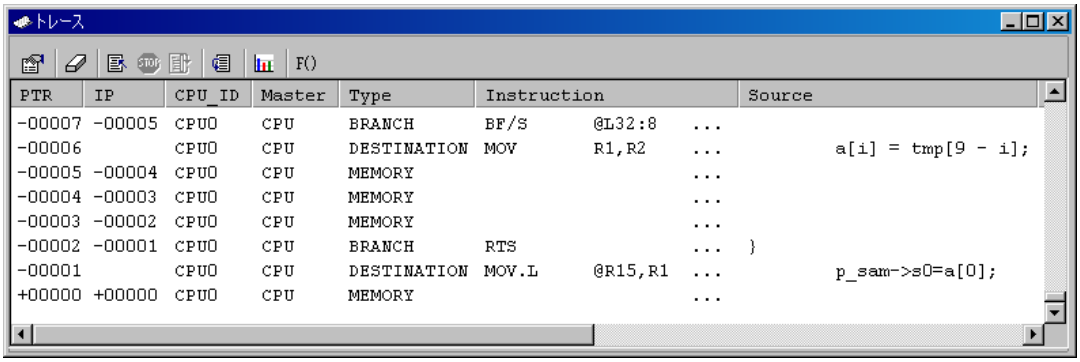


図 5.20　[トレース]ウィンドウ（I-Trace の場合）

表示する項目は以下の通りです。

[PTR]	トレースバッファ内ポインタ (最後に実行した命令が + 0 となります)
[IP]	取得したトレース情報数
[CPU_ID]	CPU コア種別
	CPU0：CPU0 によるトレース
	CPU1：CPU1 によるトレース
[Master]	トレース事象を発生させたマスタを表示します。
	CPU：CPU がマスタ
	DMA：DMAC がマスタ
[Type]	トレース情報の種別を表示します。
	BRANCH：分岐元
	DESTINATION：分岐先
	MEMORY：メモリアクセス
	PC-RELATIVE：PC 相対アクセス
	INSTRUCTION：外部空間からのフェッチ
	S_TRADE：Trace(x)関数を実行したことを示します
	OPERAND PRE-FETCH：PREF 命令の実行
[BranchType]	分岐種別を表示します。
	GENERAL：一般分岐
	SUBROUTINE：サブルーチン分岐
	EXCEPTION：例外分岐
[Bus]	サイクルのアクセス種類を表示します。
	F-Bus：F バス
	M-Bus：M バス
	I-Bus：I バス
	DMA：DMA アクセス
[R/W]	データアクセスがリードアクセスかライトアクセスかを表示します。
	READ：リードアクセス
	WRITE：ライトアクセス
[Address]	命令アドレス
[Data]	データ値を表示します。
[Size]	アクセスサイズを表示します。
	BYTE：バイト
	WORD：ワード
	LONG：ロング
[Instruction]	命令ニモニック
[Time stamp]	タイムスタンプを表示します。値は、Bφになります。
[Source]	C/C++またはアセンブラソース
[Label]	ラベル情報

[トレース]ウィンドウのポップアップメニューに[設定...]メニューがあります。
[設定...]メニューを選択すると[Acquisition]ダイアログボックスが表示されます。
このダイアログボックス内の[Trace Type]で[AUD function]を選択すると AUD トレース機能を使用してトレース情報を取得します。



PTR	IP	CPU_ID	Master	Type	Instruction		Source
-00007	-00005	CPU0	CPU	BRANCH	BF/S	@L32:8 ...	
-00006		CPU0	CPU	DESTINATION	MOV	R1,R2 ...	a[i] = tmp[9 - i];
-00005	-00004	CPU0	CPU	MEMORY		...	
-00004	-00003	CPU0	CPU	MEMORY		...	
-00003	-00002	CPU0	CPU	MEMORY		...	
-00002	-00001	CPU0	CPU	BRANCH	RTS	...	}
-00001		CPU0	CPU	DESTINATION	MOV.L	@R15,R1 ...	p_sam->s0=a[0];
+00000	+00000	CPU0	CPU	MEMORY		...	

図 5.21　[トレース]ウィンドウ（AUD trace の場合）

表示する項目は以下の通りです。
なお、製品によっては表示されない情報もあります。

[PTR]	トレースバッファ内ポインタ (最後に実行した命令が + 0 となります)
[IP]	取得したトレース情報数
[CPU_ID]	CPU コア種別 CPU0：CPU0 によるトレース CPU1：CPU1 によるトレース
[Master]	トレース事象を発生させたマスタを表示します。 CPU：CPU がマスタ
[Type]	トレース情報の種別を表示します。 BRANCH：分岐元 DESTINATION：分岐先 MEMORY：メモリアクセス S_TRADE：Trace(x)関数を実行したことを示します LOST：トレース情報が失われたことを示します。(リアルタイム) CPU-Wait：ストールが発生したことを示します。(ノンリアルタイム)
[BranchType]	分岐種別を表示します。 GENERAL：一般分岐 SUBROUTINE：サブルーチン分岐 EXCEPTION：例外分岐
[Bus]	サイクルのアクセス種類を表示します。 M-Bus：M バス I-Bus：I バス
[R/W]	データアクセスがリードアクセスかライトアクセスかを表示します。 READ：リードアクセス WRITE：ライトアクセス
[Address]	命令アドレス(AUD トレース:ベースアドレスがトレースバッファ内にない場合は、差分のみ表示)
[Data]	データ値を表示します。
[Size]	アクセスサイズを表示します。 BYTE：バイト WORD：ワード LONG：ロング
[Instruction]	命令ニモニック
[Time stamp]	タイムスタンプがないため、値は 0 固定になります。
[Source]	C/C++またはアセンブラソース
[Label]	ラベル情報

【注】　表示する内容は製品によって違いますので、各製品のオンラインヘルプを参照してください。
サポートするチップによっては、AUD トレース機能がない場合があります。

デバッグ対象デバイスによっては、以下の項目を表示するものがあります。
各製品の仕様については、別冊の「SHxxxx ご使用時の補足説明」、またはオンラインヘルプを参照してください。

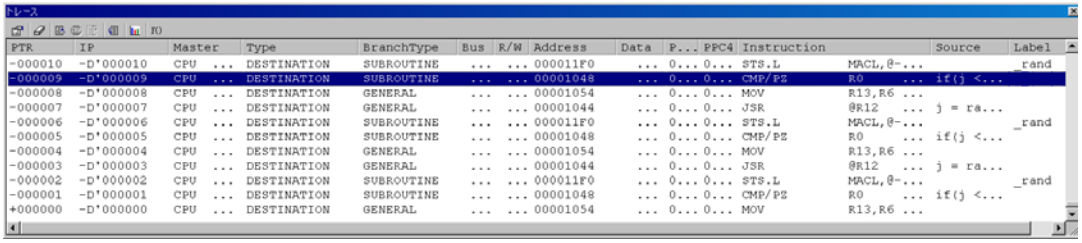


図 5.22　[トレース]ウィンドウ（Type2）

[PTR]	トレースバッファ内ポインタ (最後に実行した命令が + 0 となります)
[IP]	取得したトレース情報数
[Master](Bus Master)	アクセスを行った CPU 番号または、バスマスタの種別
[Type]	トレース情報種別
	BRANCH：分岐元
	DESTINATION：分岐先
	MEMORY：メモリアクセス
	S_TRACE：Trace(x)関数実行したことを示す
	LOST：トレース情報が失われたことを示す　(リアルタイムモード時のみ)
	CPU-WAIT：トレース情報出力のために CPU が待たされたことを示す (ノンリアルタイムモード時のみ)
[Branch Type]	分岐種別(分岐トレース取得時のみ)
	GENRAL:一般分岐
	SUBROUTINE:サブルーチン分岐
	EXCEPTION：例外分岐
[Bus]	どこのバスに対するアクセスであるかを表示
[R/W]	発生したデータアクセスが、リードアクセスかライトアクセスかを表示
[Address]	アドレス
[Data]	発生したデータアクセスのデータを表示
	[Type]が S_TRACE の場合は、関数 Trace(x)の変数 x 値を表示
[PPC]	パフォーマンスカウンタ出力
[Instruction]	命令ニモニック
[Source]	C/C++またはアセンブラソース
[Label]	ラベル情報

[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。
また、マウスでカラムをドラッグすることにより表示順序を変更することができます。

5.7.3 トレース情報取得条件を設定する

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

トレース情報の取得条件は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[Acquisition]ダイアログボックスで設定します。

ご使用のデバイスによって、表示されるダイアログボックスおよび設定内容が CPU 毎に設定が可能か、共有されるかが異なります。
以下のダイアログボックスを表示するデバイスでは、設定内容は、各 CPU で共有されます。
各製品の仕様については、別冊の「SHxxxx ご使用時の補足説明」、またはオンラインヘルプを参照してください。

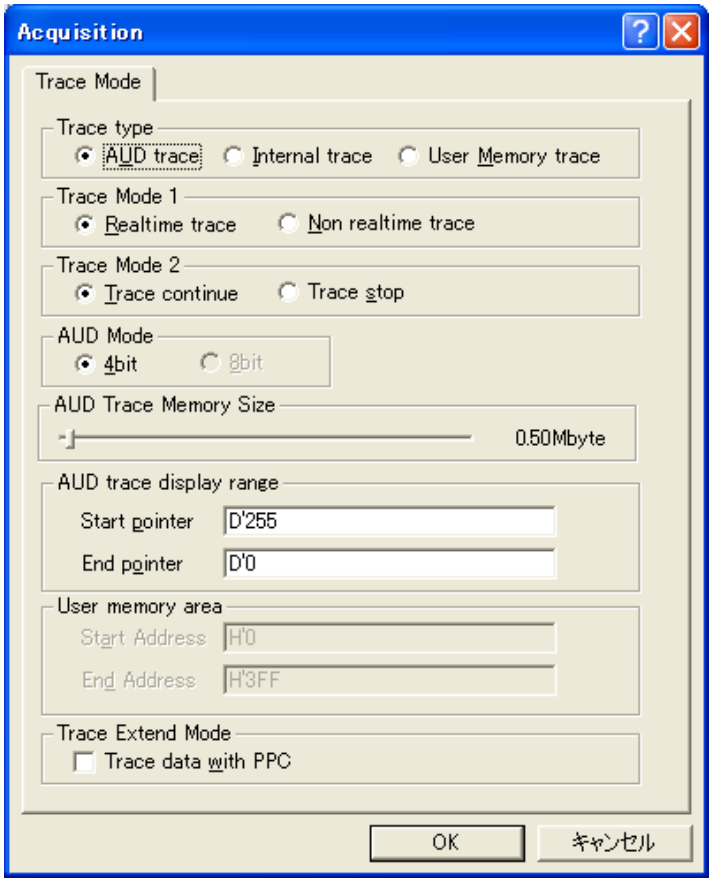


図 5.23 [Acquisition]ダイアログボックス([Trace mode]ページ)

[Trace type]：トレース機能の種類を選択します

[AUD trace]	AUD トレース機能を使用します
[Internal trace]	内蔵トレース機能を使用します
[User Memory trace]	トレースデータメモリ出力機能を使用します

[Trace Mode 1]：トレース情報が連続して発生した場合の動作を決めるオプション
[AUD trace]、[User Memory trace]選択時のみ使用可能です。

[Realtime trace]	一部のトレース情報が出力されないモード
[Non realtime trace]	トレースが出力されるまで CPU が待たされるモード

[Trace Mode 2]： E10A-USB エミュレータのトレースバッファが Full になったときの動作を決めるオプション
[AUD trace]、[User Memory trace]選択時のみ使用可能です。

[Trace continue]	古いトレース情報を上書きして、常に最新の情報を取得します
[Trace stop]	以前のトレース情報は取得しないモード

[AUD Mode]： デバッグ対象のデバイスによっては、AUD 端子 8 ビットモードを選択できます。
各製品の仕様については、別冊の「SHxxxx ご使用時の補足説明」、またはオンラインヘルプを参照してください。
[AUD trace]選択時のみ使用可能です。

[AUD trace Memory Size]：E10A-USB エミュレータのトレースバッファメモリサイズを設定するオプション
[AUD trace]選択時のみ使用可能です。

[AUD trace display range]：トレースウィンドウの表示範囲を設定するオプション
[AUD trace]選択時のみ使用可能です。

[Start pointer]	設定された値からトレース表示
[End pointer]	設定された値までトレース表示

[User Memory area]：トレースウィンドウの表示範囲を設定するオプション
[User Memory trace]選択時のみ使用可能です。

[Start]	トレース結果を書き込むメモリ範囲の先頭アドレスを指定します。
[End address]	トレース結果を書き込むメモリ範囲の終了アドレスを指定します。

[Trace Extend Mode]：

[Trace data with PPC]	トレースウィンドウにパフォーマンスカウンタを出力します。 （この機能を有効にした場合は、分岐トレースの分岐元が表示されなくなります。）

指定した内容は、[OK]ボタンをクリックすることにより設定します。[Cancel]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。

また、以下のダイアログボックスが表示されるデバイスでは、[Display Type] グループボックス以外の設定は CPU0 用/CPU1 用 High-performance Embedded Workshop に共有されます。[Display Type] グループボックスの設定は CPU0 用/CPU1 用 High-performance Embedded Workshop に共有されません。各々の High-performance Embedded Workshop で設定できます。

各製品の仕様については、別冊の「SHxxxx ご使用時の補足説明」、またはオンラインヘルプを参照してください。

(1) [Trace mode]ページ
トレース情報の取得条件を設定します。

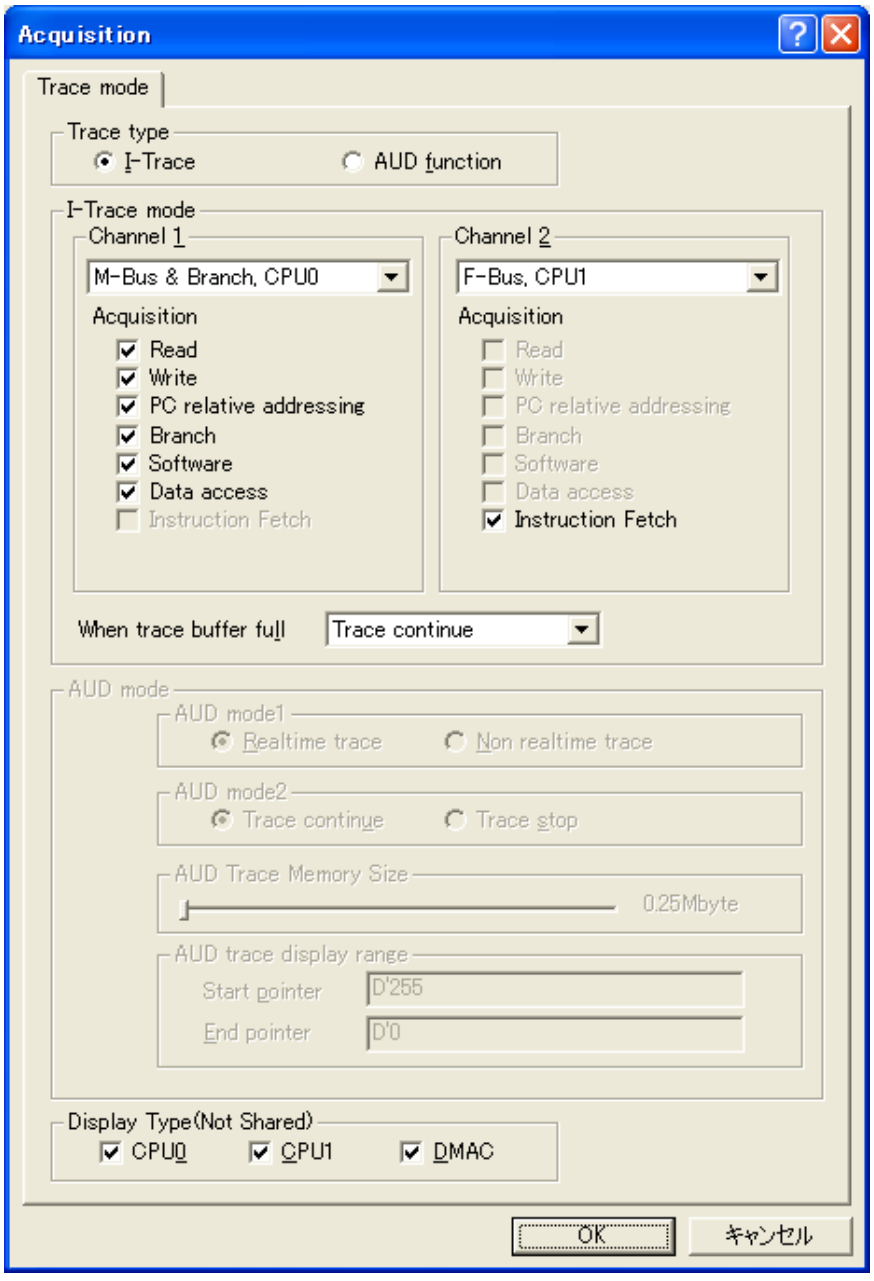


図 5.24 [Acquisition]ダイアログボックス([Trace mode]ページ)

本ダイアログボックスでは、トレース情報の取得方法、取得条件を設定します。
設定できる項目は以下の通りです。

[Trace mode] ：Trace mode 条件を設定します。
[Trace Type] ：トレース機能の種類を選択します
[I-Trace] ：内蔵トレース機能を使用します
[AUD function] ：AUD トレース機能を使用します。

- 内蔵トレース [Trace mode]ページ ([I-Trace]選択時)

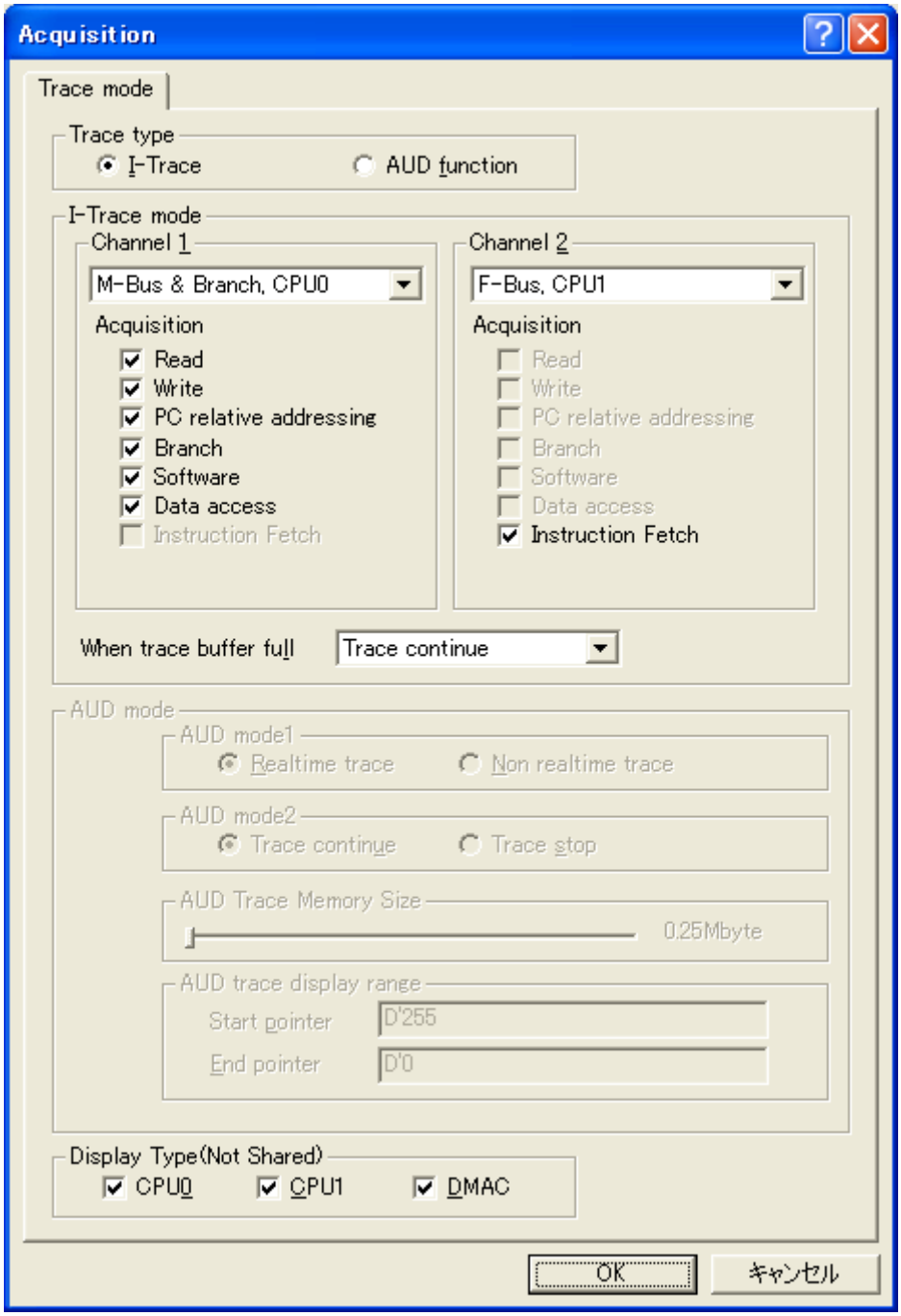


図 5.25 [Acquisition]ダイアログボックス(I-Trace 選択時)

本ダイアログボックスでは、トレース情報の取得条件を設定します。

・[I-Trace]選択時

[I-Trace mode]　：内蔵トレースを取得するバスと条件を設定します。

[Channel 1]　：トレース取得条件を Channel 1 に設定します。
Channel 1 と Channel 2 に同じ条件を設定できません。

[Type]
- [M-Bus & Branch,CPU0]：CPU0 の M-Bus と分岐を条件に取得します。
- [I-Bus,CPU0]：CPU0 の I-Bus を条件に取得します。
- [F-Bus,CPU0]：CPU0 の F-Bus を条件に取得します。
- [M-Bus & Branch,CPU1]：CPU1 の M-Bus と分岐を条件に取得します。
- [I-Bus,CPU1]：CPU1 の I-Bus を条件に取得します。
- [F-Bus,CPU1]：CPU1 の F-Bus を条件に取得します。
- [DMAC]：DMAC を条件に取得します。

[Channel 2]　：トレース取得条件を Channel 2 に設定します。
Channel 1 と Channel 2 に同じ条件を設定できません。

[Type]
- [M-Bus & Branch,CPU0]：CPU0 の M-Bus と分岐を条件に取得します。
- [I-Bus,CPU0]：CPU0 の I-Bus を条件に取得します。
- [F-Bus,CPU0]：CPU0 の F-Bus を条件に取得します。
- [M-Bus & Branch,CPU1]：CPU1 の M-Bus と分岐を条件に取得します。
- [I-Bus,CPU1]：CPU1 の I-Bus を条件に取得します。
- [F-Bus,CPU1]：CPU1 の F-Bus を条件に取得します。
- [None]：条件を設定しません。

[Acquisition]　：内蔵トレースの取得条件を設定します。
- [Read]：リードを条件にします。
- [Write]：ライトを条件にします。
- [PC relative addressing]：実行アドレスを条件にします。
- [Branch]：分岐を条件にします。
- [Software]：ソフトウェアトレースを条件にします。
- [Data access]：データアクセスを条件にします。
- [Instruction Fetch]：フェッチサイクルを条件にします。

[When trace buffer full]　：トレースバッファがすべて埋まった時点での動作を指定します。
- [Trace continue]：トレース取得を続けます。古いトレースバッファの内容は消去されます。
- [Trace stop]：トレース取得を停止します。
- [Break(CPU0)]：CPU0 をブレークします。
- [Break(CPU1)]：CPU1 をブレークします。
- [Break(CPU0,CPU1)]：CPU0 と CPU1 をブレークします。

- AUD トレース [Trace mode]ページ ([AUD function]選択時)

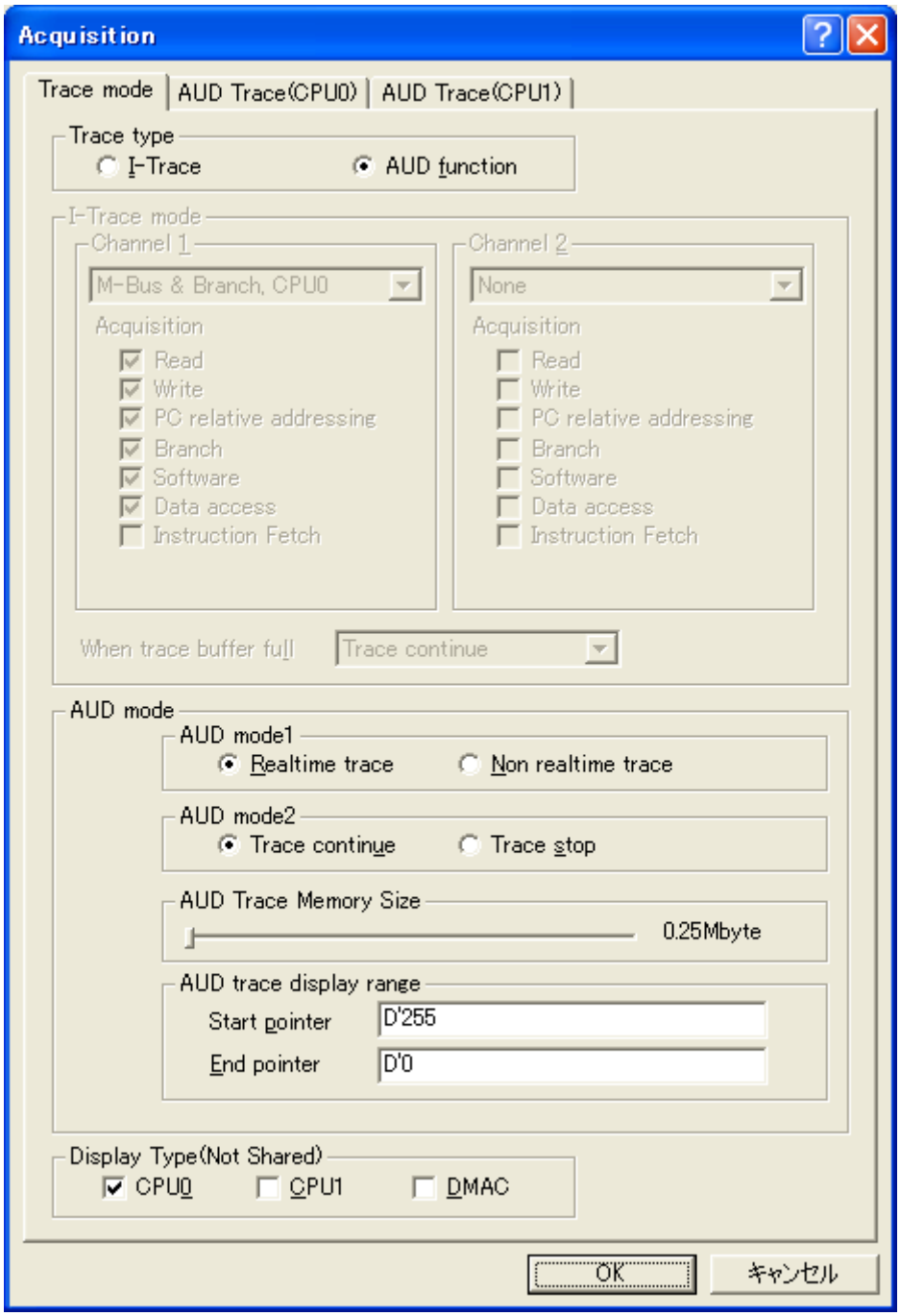


図 5.26 [Acquisition]ダイアログボックス(AUD function 選択時)

・ [AUD function]選択時

[AUD mode1] ：トレース取得時のリアルタイムの設定

[Realtime trace] ：トレース情報の発生が集中し、AUD 端子からの出力が間に合わなくなると CPU はトレース情報の出力を一時的に停止します。

このため､ユーザプログラムはリアルタイムに動作しますが､トレース情報が一部取得できないことがあります。

[Non realtime trace] ：トレース情報の発生が集中し、AUD 端子からの出力が間に合わなくなると CPU の動作を

一時的に停止し､トレース情報の出力を優先します。このため､ユーザプログラムのリアルタイム性がなくなります。

[AUD mode2] ：トレースバッファがすべて埋まった時点での動作を指定します。

[Trace continue] ：トレース取得を続けます。古いトレースバッファの内容は消去されます。

[Trace stop] ：トレースバッファがフルになった場合、トレース取得しません。

[AUD trace Memory Size] ：E10A-USB エミュレータのトレースバッファメモリサイズを指定します。

[AUD trace display range] ：AUD トレース情報の表示範囲を設定します。

[Start pointer] ：AUD トレースの先頭ポインタを設定します。

[End pointer] ：AUD トレースの終了ポインタを設定します。

・ [I-Trace]/ [AUD function]共通

[Display Type] ：トレースウインドウに表示する情報を指定します。

[CPU0] ：CPU コア種別(CPU_ID)が CPU0 のトレースを表示します。

[CPU1] ：CPU コア種別(CPU_ID)が CPU1 のトレースを表示します。

[DMAC] ：CPU コア種別(CPU_ID)が DMA のトレースを表示します。

指定した内容は、[OK]ボタンをクリックすることにより設定します。[Cancel]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。

（2） [AUD Trace(CPU0)]および[AUD Trace(CPU1)]ページ

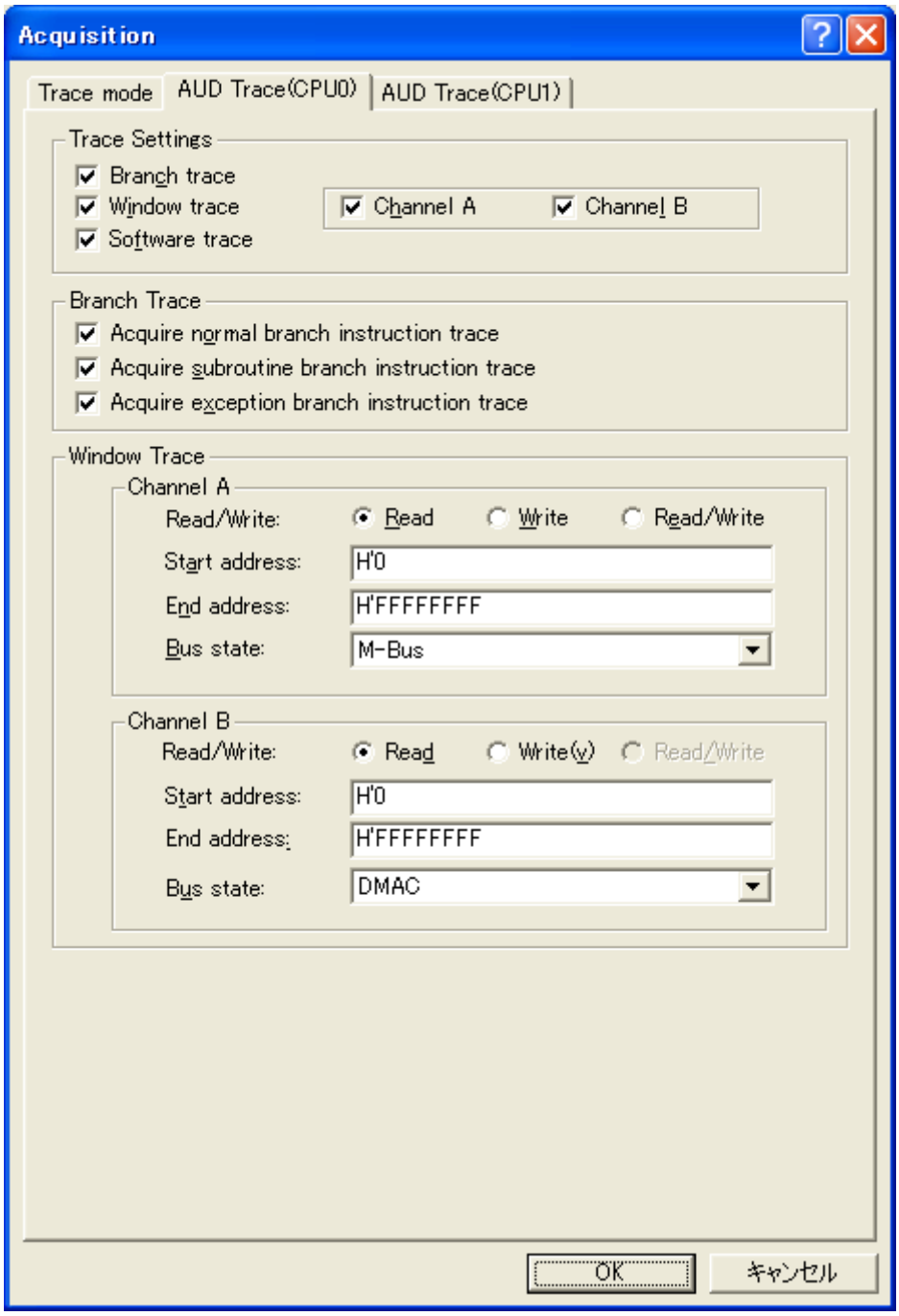


図 5.27 [Acquisition]ダイアログボックス([AUD Trace(CPU0)]ページ)

AUD Trace(CPU0)および AUD Trace(CPU1)の設定

AUD function 選択時[Trace Settings]から AUD トレースの条件を選択します。

[Branch trace] : 分岐元および分岐先のアドレスを条件として設定します。

[Window trace] :ウィンドウトレース機能。指定範囲内メモリアクセス情報を取得します。

[Channel A] : ウィンドウトレース情報をチャネル A から取得するかどうかを設定します。

[Channel B] : ウィンドウトレース情報をチャネル B から取得するかどうかを設定します。

[Software trace] : ソフトウェアトレース機能。ソフトウェアトレース命令のトレースを取得します。

[Trace Settings]の[Branch Trace]にチェックをいれた場合、取得分岐条件を選択します。

[Acquire normal branch instruction trace] :通常分岐を取得する分岐条件に指定します。

[Acquire subroutine branch instruction trace] : サブルーチン分岐を取得する分岐条件に指定します。

[Acquire exception branch instruction trace] :例外分岐を取得する分岐条件に指定します。

[Trace Settings]の[Window trace]にチェックをいれた場合、[Window Trace]グループボックスで[Channel A]、[Channel B]の条件を設定します。

[Channel A] : AUD トレースを取得する条件を設定します。

[Read/Write] : ウィンドウトレースの Read/Write を指定します。

[Read] : Read を条件に指定します。

[Write] : Write を条件に指定します。

[Read/Write] : Read/Write を条件に指定します。

[Start address] : ウィンドウトレースの先頭アドレスを指定します。

[End address] : ウィンドウトレースの終了アドレスを指定します。

[Bus state] : ウィンドウトレースを取得するバスを指定します。

[M-Bus] : M-Bus を指定します。

[DMAC] : DMA を指定します。

[Channel B] : AUD トレースを取得する条件を設定します。

[Read/Write] : ウィンドウトレースの Read/Write を指定します。

[Read] : Read を条件に指定します。

[Write] : Write を条件に指定します。

[Read/Write] : Read/Write を条件に指定します。

[Start address] : ウィンドウトレースの先頭アドレスを指定します。

[End address] : ウィンドウトレースの終了アドレスを指定します。

[Bus state] : ウィンドウトレースを取得するバスを指定します。

[M-Bus] : M-Bus を指定します。

[DMAC] : DMA を指定します。

5.7.4 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。

本設定は CPU 毎に設定が可能です。
[Trace Find]ダイアログボックスは下記ページより構成されています。

表 5.1 [Trace Find]ダイアログボックスのページ構成

ページ	設定項目
[General]	検索範囲を指定します。
[Address]	アドレス条件を指定します。
[Data]	データ条件を指定します。
[Type]	トレース情報のタイプを指定します。
[R/W]	アクセスサイクルの種類を指定します。
[Size]	アクセスサイズを指定します。

【注】 [General], [Address]以外の項目はご使用のエミュレータにより異なります。
詳細につきましては、オンラインヘルプをご参照ください。

各ページで条件を設定し、[OK]ボタンをクリックすることにより、サーチ条件を設定し、検索を開始します。[キャンセル]ボタンをクリックすると、設定しないでダイアログボックスを閉じます。
検索の結果一致するトレースレコードが見つかった場合は当該レコード行を強調表示します。
一致するトレースレコードが見つからなかった場合は、メッセージダイアログボックスを表示します。
トレース情報の検索は各ページで設定した条件がすべて一致するトレース情報のみを検索します。

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

(1) [General]ページ

検索範囲を指定します。

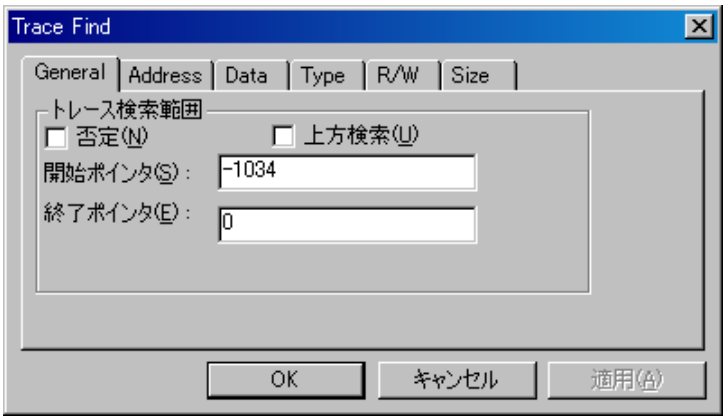


図 5.28 [Trace Find]ダイアログボックス([General]ページ)

[トレース検索範囲]	検索範囲を指定します。
[否定]	チェックすると他のページで設定した項目の否定条件で検索します。
[上方検索]	チェックすると上方検索を行います。
[開始ポインタ]	検索を開始する PTR の値を入力します。
[終了ポインタ]	検索を終了する PTR の値を入力します。

【注】 検索範囲入力時、[開始ポインタ]オプションに検索を終了する PTR の値、
[終了ポインタ]オプションに検索を開始する PTR の値を指定することも可能です。

(2) [Address]ページ

アドレス条件を指定します。

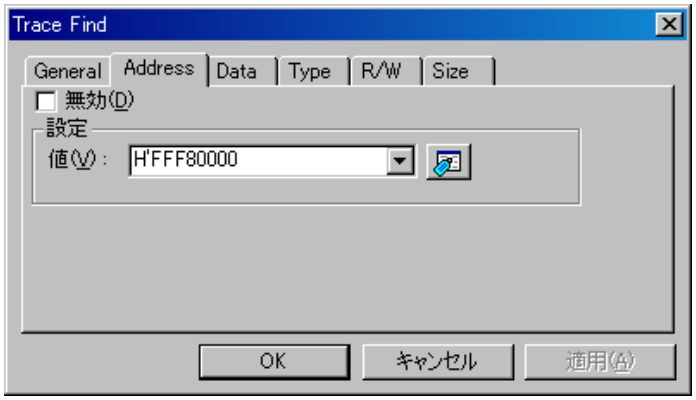


図 5.29 [Trace Find]ダイアログボックス([Address]ページ)

[無効] チェックすると、アドレスを検出しません。

[設定] 指定したアドレスを検出します。

[値] アドレス値を入力します。([無効]選択時無効)

(3) [Data]ページ

データ条件を指定します。

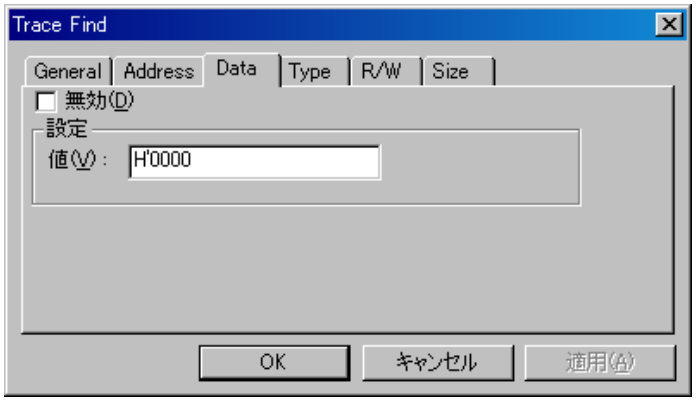


図 5.30 [Trace Find]ダイアログボックス([Data]ページ)

[無効] チェックすると、データを検出しません。

[設定] 指定したデータを検出します。

[値] データ値を入力します。
([無効]選択時無効)

（4）[R/W]ページ

アクセスサイクルの種類を指定します。

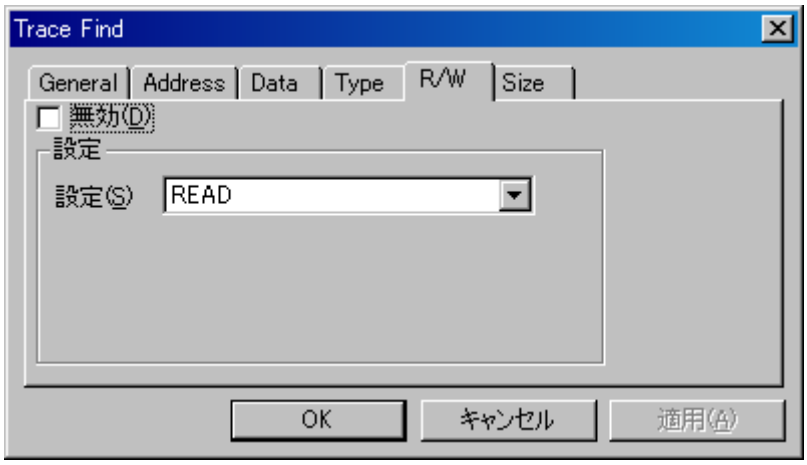


図 5.31 [Trace Find]ダイアログボックス([R/W]ページ)

[無効] チェックすると、リード/ライト条件を検出しません。

[設定] 指定したリード/ライト条件を検出します。

[設定] リード/ライト条件を選択します。
([無効]選択時無効)

READ : リードサイクル

WRITE : ライトサイクル

（5）[Type]ページ

トレース情報のタイプを指定します。

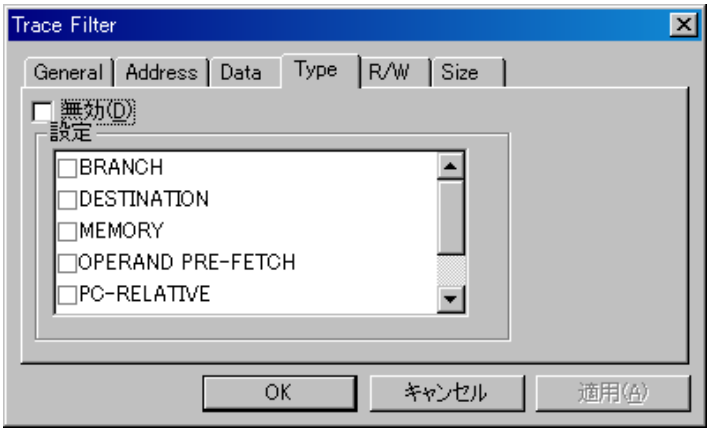


図 5.32 [Trace Find]ダイアログボックス([Type]ページ)

[無効] チェックすると、タイプ条件を検出しません。

[設定] 指定したタイプ条件を検出します。

[設定] タイプ条件を選択します。
([無効]選択時無効)

(6) [Size]ページ

アクセスサイズを指定します。

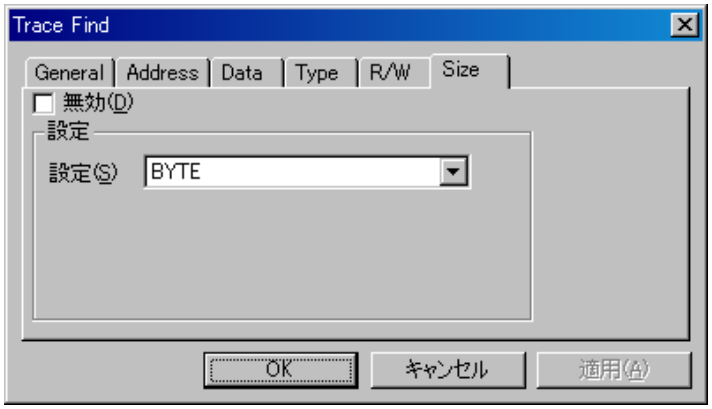


図 5.33 [Trace Find]ダイアログボックス([Size]ページ)

[無効] チェックすると、サイズ条件を検出しません。

[設定] 指定したサイズ条件を検出します。

[設定] サイズ条件を選択します。
([無効]選択時無効)

5.7.5 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。
その際にトレース情報を保持しているトレースバッファは空になります。
複数の[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。

5.7.6 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、「5.7.10 取得したトレース情報から必要なレコードを抽出する」
を参照してください。

5.7.7 [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[逆アセンブリ]ウィンドウを開いてソース表示し、選択した行をカーソルで示します。

5.7.8 ソース表示を整形する

ポップアップメニューで[ソーストリム]を選択すると、ソースプログラムの左側の空白を取り除きます。
取り除いた状態だと[ソーストリム]メニューの左にチェックが付きます。
チェックありの状態で[ソーストリム]メニューを選択すると取り除いた空白を元に戻します。

5.7.9 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。

5.7.10 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[Acquisition]設定と異なり、取得したトレース情報に対し何度もフィルタ設定を変更することで必要な情報が簡単に抽出でき、データの分析に役立ちます。

フィルタ機能を使用してもトレースバッファの内容は変更されません。

トレースバッファは有限ですので、[Acquisition]設定により有用なトレース情報をより多く取得することで、より効果的にデータの分析が可能となります。

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

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

本設定は CPU0 用/CPU1 用 High-performance Embedded Workshop に共有されません。
各々の High-performance Embedded Workshop で設定できます。

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

表 5.2 [Trace Filter]ダイアログボックスのページ構成

ページ	設定項目
[General]	フィルタ範囲を指定します。
[Address]	アドレス条件を指定します。
[Data]	データ条件を指定します。
[Type]	トレース情報のタイプを指定します。
[R/W]	アクセスサイクルの種類を指定します。
[Size]	アクセスサイズを指定します。

【注】 [General], [Address]以外の項目はご使用のエミュレータにより異なります。
詳細につきましては、オンラインヘルプをご参照ください。

各ページでフィルタ条件を設定し、[OK]ボタンをクリックすることにより、フィルタ条件にしたがいフィルタリングを行います。
[キャンセル]ボタンをクリックすると、[Trace Filter]ダイアログボックスを開いた時点の設定のままダイアログボックスを閉じます。
フィルタリングは各ページで設定したフィルタ条件が 1 つ以上一致するトレース情報のみを[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。

(1) [General]ページ
フィルタ範囲を指定します。

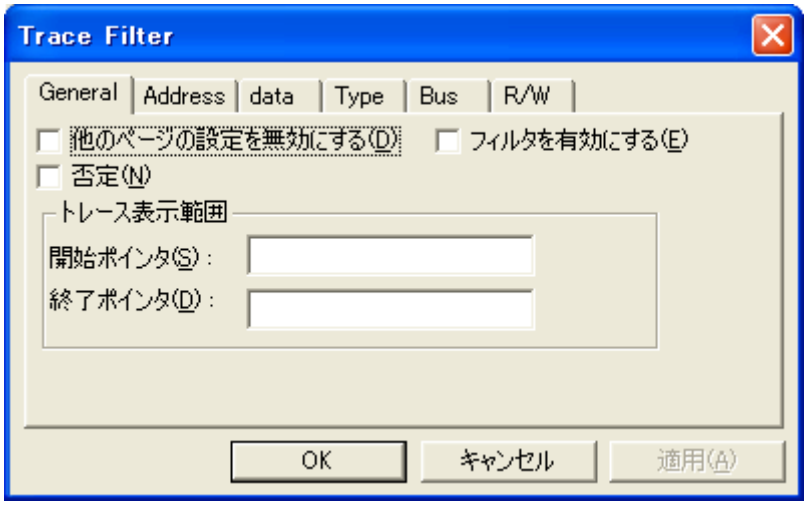


図 5.34　[Trace Filter]ダイアログボックス([General]ページ)

[他のページ設定を無効にする]	チェックすると、サイクル番号のみ指定できます。他のオプションはすべて無効になります。
[フィルタを有効にする]	チェックすると、フィルタを有効にします。
[否定]	チェックすると他のページで設定した項目の否定条件でフィルタリングします。
[トレース表示範囲]	フィルタ範囲を指定します。
[開始ポインタ]	フィルタを開始する PTR の値を入力します。
[終了ポインタ]	フィルタを終了する PTR の値を入力します。

【注】　フィルタ範囲入力時、[開始ポインタ]オプションにフィルタを終了する PTR の値、[終了ポインタ]オプションにフィルタを開始する PTR の値を指定することも可能です。

（2）[Address]ページ
アドレス条件を指定します。

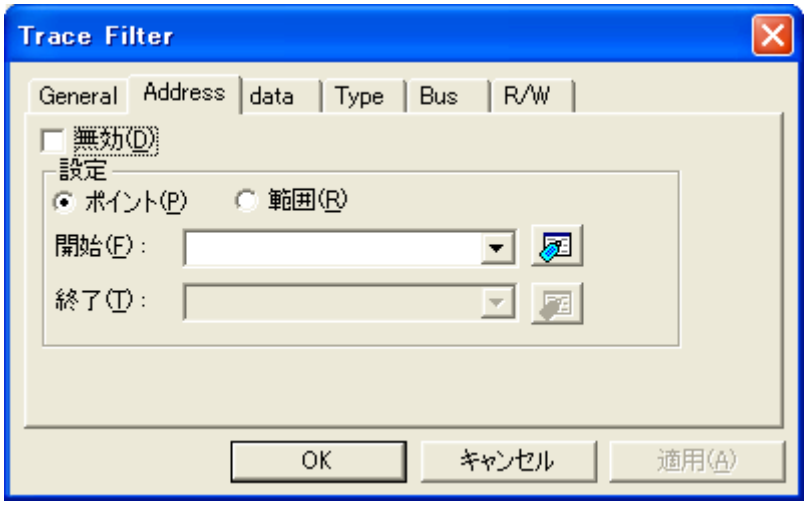


図 5.35 [Trace Filter]ダイアログボックス([Address]ページ)

[無効]	チェックすると、アドレスを検出しません。
[設定]	指定したアドレスを検出します。
[ポイント]	単一アドレスを指定します。 ([無効]選択時無効)
[範囲]	アドレス範囲を指定します。 ([無効]選択時無効)
[開始]	単一アドレスまたはアドレス範囲の開始アドレスを入力します。 ([無効]選択時無効)
[終了]	アドレス範囲の終了アドレスを入力します。 ([範囲]選択時有効)

【注】 アドレス範囲入力時、[開始]オプションにアドレス範囲の終了アドレス、[終了]オプションにアドレス範囲の開始アドレスを指定することも可能です。

(3) [Data]ページ

データ条件を指定します。

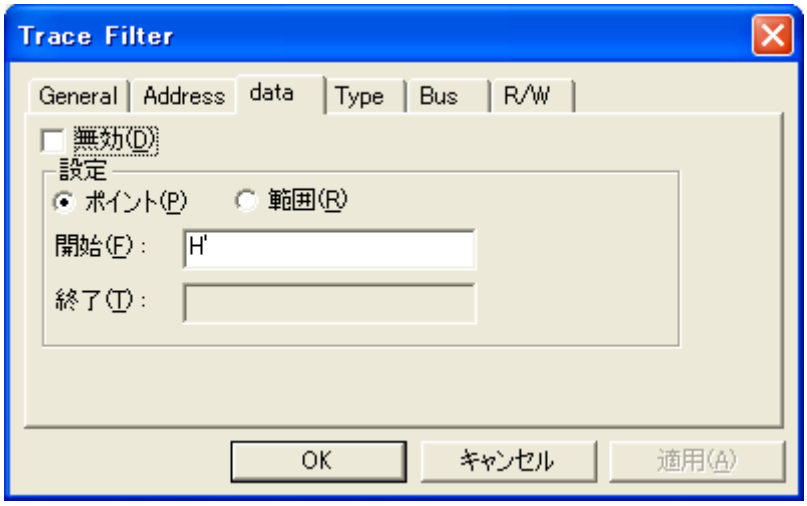


図 5.36　[Trace Filter]ダイアログボックス([Data]ページ)

[無効]	チェックすると、データを検出しません。
[設定]	指定したデータを検出します。
[ポイント]	単一データを指定します。 ([無効]選択時無効)
[範囲]	データ範囲を指定します。 ([無効]選択時無効)
[開始]	単一データまたはデータ範囲の最小値を入力します。 ([無効]選択時無効)
[終了]	データ範囲の最大値を入力します。 ([範囲]選択時有効)

【注】　データ範囲入力時、[開始]オプションにデータの最大値、[終了]オプションにデータの最小値を指定することも可能です。

（4）[R/W]ページ

アクセスサイクルの種類を指定します。

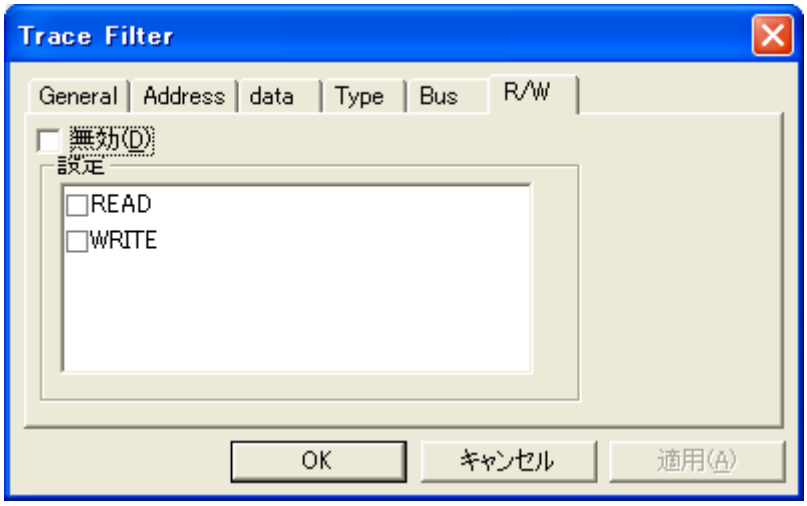


図 5.37　[Trace Filter]ダイアログボックス([R/W]ページ)

[無効]　チェックすると、リード/ライト条件を検出しません。

[設定]　指定したリード/ライト条件を検出します。

[READ]　チェックすると、リードサイクルを検出します。
([無効]選択時無効)

[WRITE]　チェックすると、ライトサイクルを検出します。
([無効]選択時無効)

(5) [Type]ページ

トレース情報のタイプを指定します。
タイムスタンプ取得時は無効です。

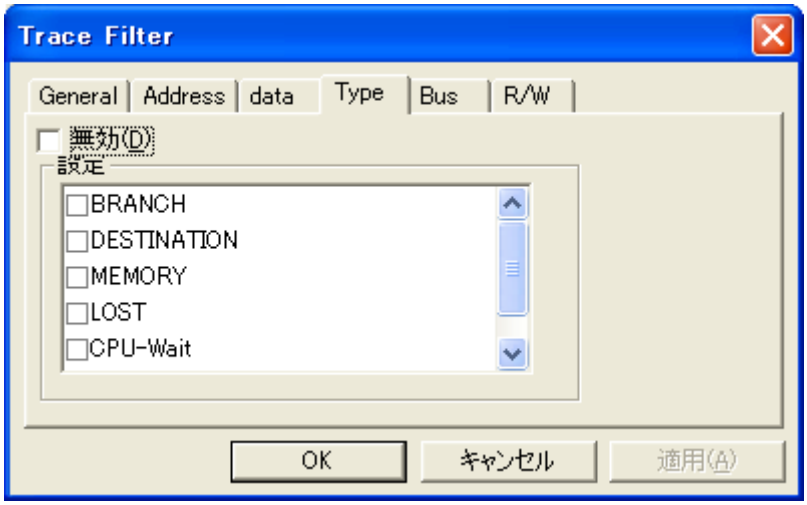


図 5.38 [Trace Filter]ダイアログボックス([Type]ページ)

[無効] チェックすると、タイプ条件を検出しません。

[設定] 指定したタイプ条件を検出します。
([無効]選択時無効)

(6) [Size]ページ

アクセスサイズを指定します。

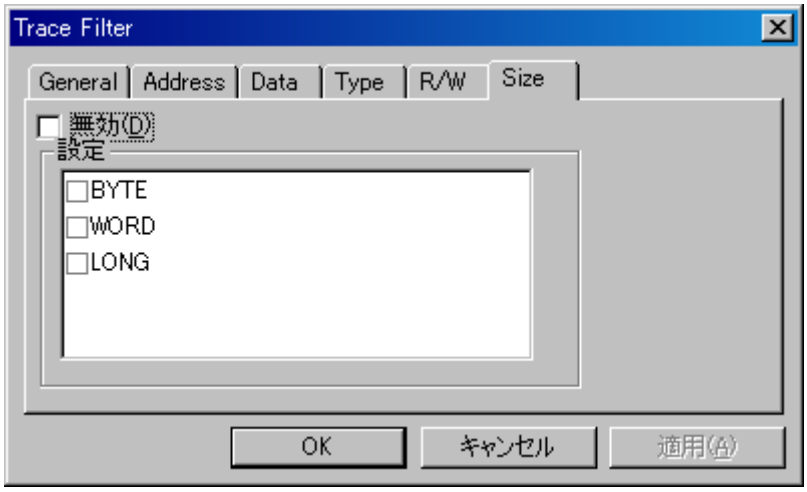


図 5.39 [Trace Filter]ダイアログボックス([Size]ページ)

[無効] チェックすると、サイズ条件を検出しません。

[設定] 指定したサイズ条件を検出します。
([無効]選択時無効)

5.7.11 統計情報を解析する

指定された条件で統計情報の解析を実行するには、ポップアップメニューから[統計...]を選択します。

[統計]ダイアログボックスが開きます。

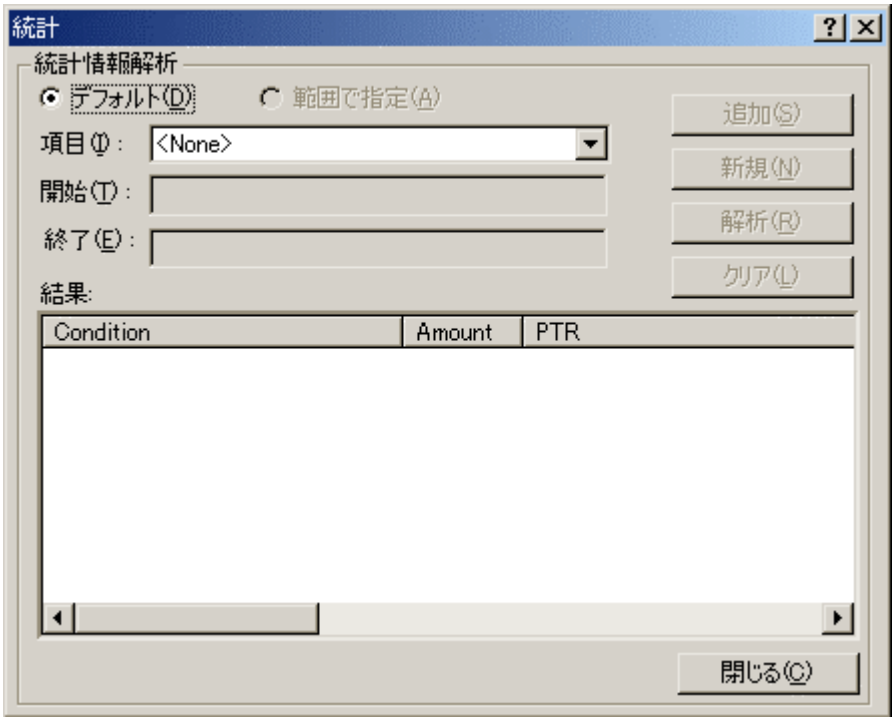


図 5.40 [統計]ダイアログボックス

[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。
[クリア]	設定の初期化を行います。
[結果]	すべての条件と統計情報解析結果を表示します。
[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、完全一致したものだけをカウントします。ただし、大文字小文字は区別しません。
また、空白の数も考慮しません。

5.7.12 取得したトレース情報から関数呼び出し箇所を抽出する

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

図 5.41 [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。
関数の呼び出しを含んだトレース情報に対して本機能を使用することにより、関数の呼び出し順序を調べることができます。

5.8 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 Performance Analysis 機能を使用します。
Performance Analysis 機能はマイコン内蔵の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。

本設定は CPU0 用/CPU1 用 High-performance Embedded Workshop に共有されません。
各々の High-performance Embedded Workshop で設定できます。

【注】　測定条件、チャネル本数は製品によって異なります。

5.8.1 [パフォーマンス解析]ウィンドウを開く

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

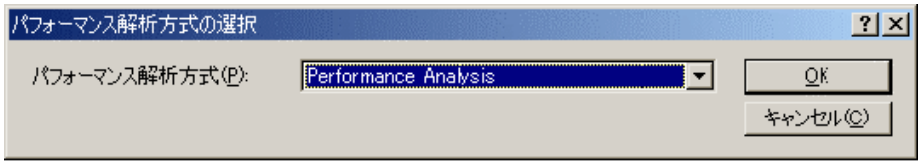


図 5.42　[パフォーマンス解析方式の選択]ダイアログボックス

[OK]ボタンをクリックすると[パフォーマンス解析]ウィンドウが開きます。

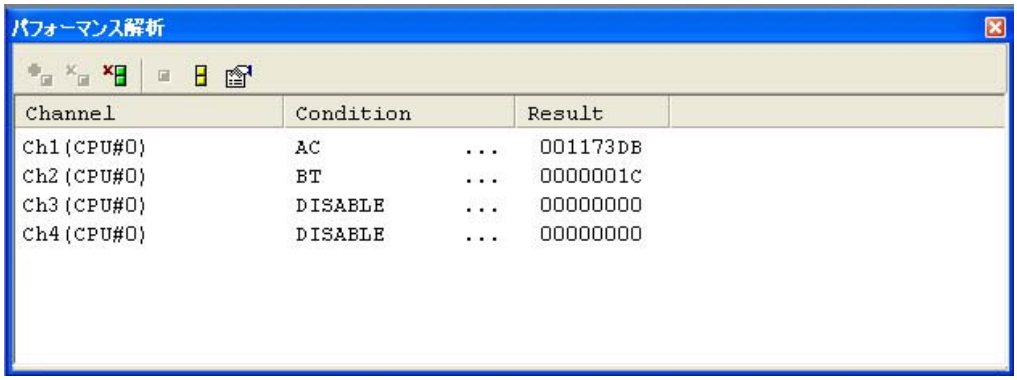


図 5.43　[パフォーマンス解析]ウィンドウ

[パフォーマンス解析]ウィンドウ内の不要なカラムは非表示にすることができます。
カラムを非表示にする場合はヘッダカラム上で右クリックすると表示されるポップアップメニューより非表示にしたいカラムを選択してください。
カラムを再表示する場合は再度ポップアップメニューより該当のカラムを選択してください。

5.8.2 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[Performance Analysis Properties]ダイアログボックスを表示します。

5.8.3 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。

5.8.4 測定条件を削除する

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

5.8.5 すべての測定条件を削除する

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

6. チュートリアル[SH-2A 編]

6.1 はじめに

E10A-USB 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[Parallel]モードの動作を説明します。

特に説明のない場合は CPU1 側の High-performance Embedded Workshop の操作として説明します。

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、CPU0 側と CPU1 側の High-performance Embedded Workshop はそれぞれ 10 個のランダムデータを昇順/降順にソートします。

チュートリアルプログラムでは、以下の処理を行います。

main 関数でソートするランダムデータを生成します。

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

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

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

【留意事項】

1. Tutorial.abs は、ビッグエンディアンで動作します。リトルエンディアンで動作させる場合、再コンパイルを行ってください。
   再コンパイルを行った場合、本章で説明しているアドレスと異なることがあります。
2. 本章は、一般的な E10A-USB エミュレータの使用例です。各製品の仕様については、別冊の「SHxxxx ご使用時の補足説明」、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を、実際にロードされたアドレスの上位 16 ビットで置き換えて操作してください。
   [例] Tutorial.abs のロードされたアドレスが H'0C00xxxx である場合、マニュアルでは PC アドレスが H'0000006c となっていますが、H'0C00006c（上位ビット H'0000 -> H'0C00 に変更）として、入力してください。
4. ご使用のデバイスによっては、表示されるアドレスやデータ値が本章の説明と異なることがあります。

6.2 High-performance Embedded Workshop の起動

[スタート]メニューの[プログラム]から[Renesas]–>[High-performance Embedded Workshop]–>[High-performance Embedded Workshop]を選択してください。

6.3 同期デバッグの設定

(1) [ようこそ！]ダイアログボックスが表示されます。

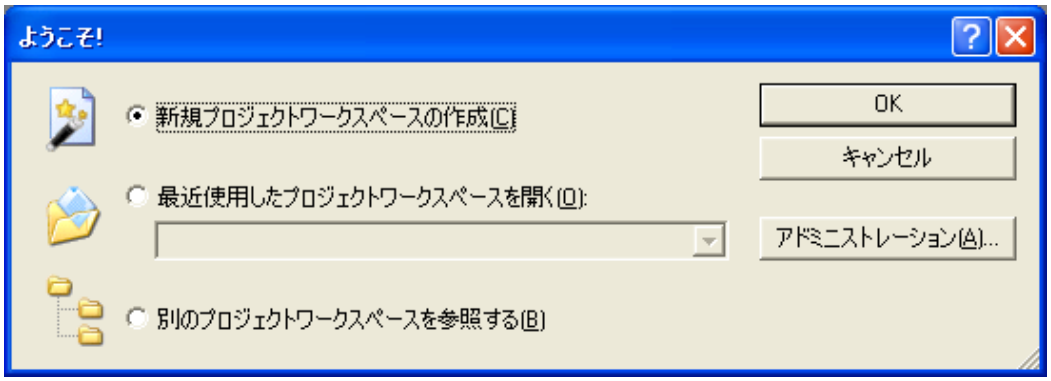


図 6.1 [ようこそ！]ダイアログボックス

ここでは、[キャンセル]ボタンを押してください。

(2) [デバッグ]メニューの[同期デバッグ]を選択し、[同期デバッグ]ダイアログボックスを開いてください。

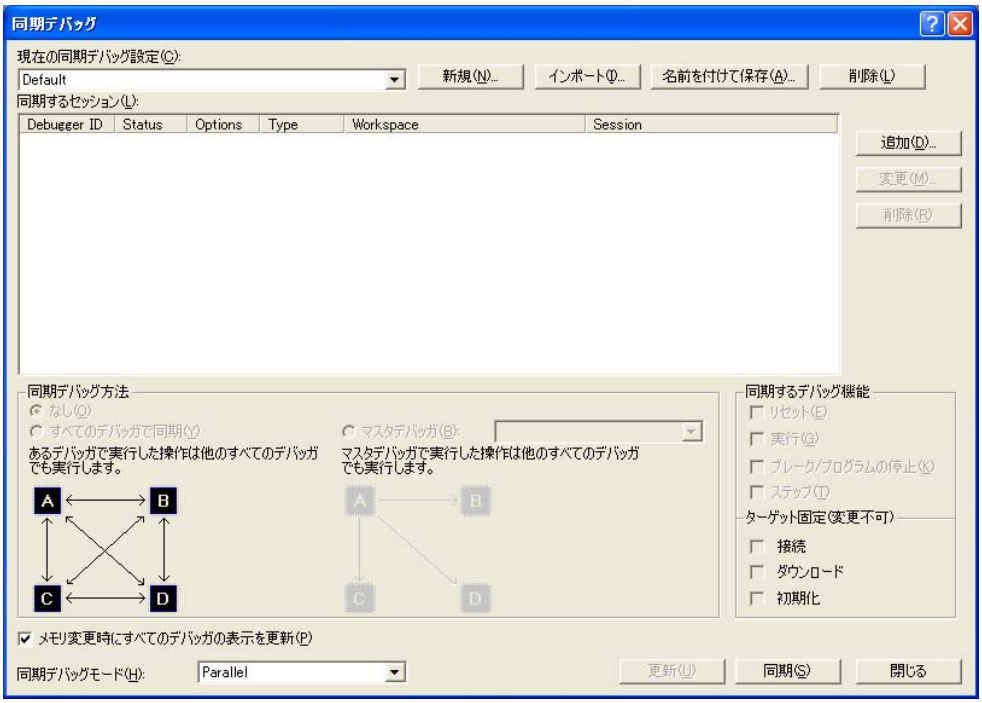


図 6.2 [同期デバッグ]ダイアログボックス

(3) [新規]ボタンをクリックし、[同期デバッグ設定名]を入力してください。ここでは、SH2A-DUAL_Tutorialと入力します。
(4) [追加]ボタンをクリックし、[セッションの追加]ダイアログボックスを開いてください。

[参照]ボタンより、
〈OS がインストールされているドライブ〉
¥WorkSpace¥Tutorial¥E10A-USBM¥SH2A-DUAL¥SH2A-DUAL¥Tutorial¥xxxx¥xxxx.hws
を読み込んでください。xxxx は対象の製品グループを示します。
[プロジェクト]ドロップダウンリストボックスから[CPU0]を選択し、[OK]ボタンをクリックし、ダイアログボックスを閉じてください。

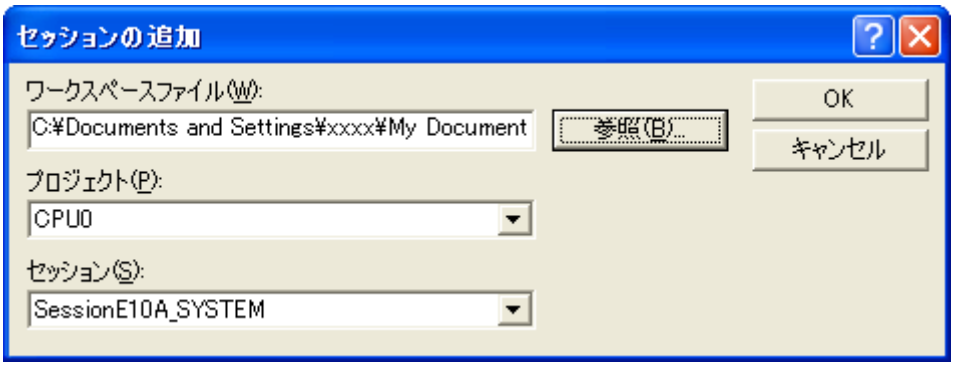


図 6.3 [セッションの追加]ダイアログボックス

【注】ワークスペースの読み込みに失敗する場合は、「3.9 システムチェック」の手順にしたがってワークスペースを開いてから読み込みを行ってください。

(5) 再度[追加]ボタンをクリックし、[セッションの追加]ダイアログボックスを開いてください。
[ワークスペースファイル]部に先ほど読み込んだワークスペースが表示されていることを確認してください。異なる場合は、(4)の手順でワークスペースを読み込んでください。
[プロジェクト]ドロップダウンリストボックスから[CPU1]を選択し、[OK]ボタンをクリックし、ダイアログボックスを閉じてください。

(6) 同期デバッグ状態の設定を行います。

ここでは、[同期デバッグ方法]を[すべてのデバッガで同期]ラジオボタンを選択、[同期するデバッグ機能]を[実行]、[ブレーク/プログラムの停止]、[ステップ]の全てのチェックボックスをチェック、[同期デバッグモード]ドロップダウンリストボックスから[Parallel]を選択してください。

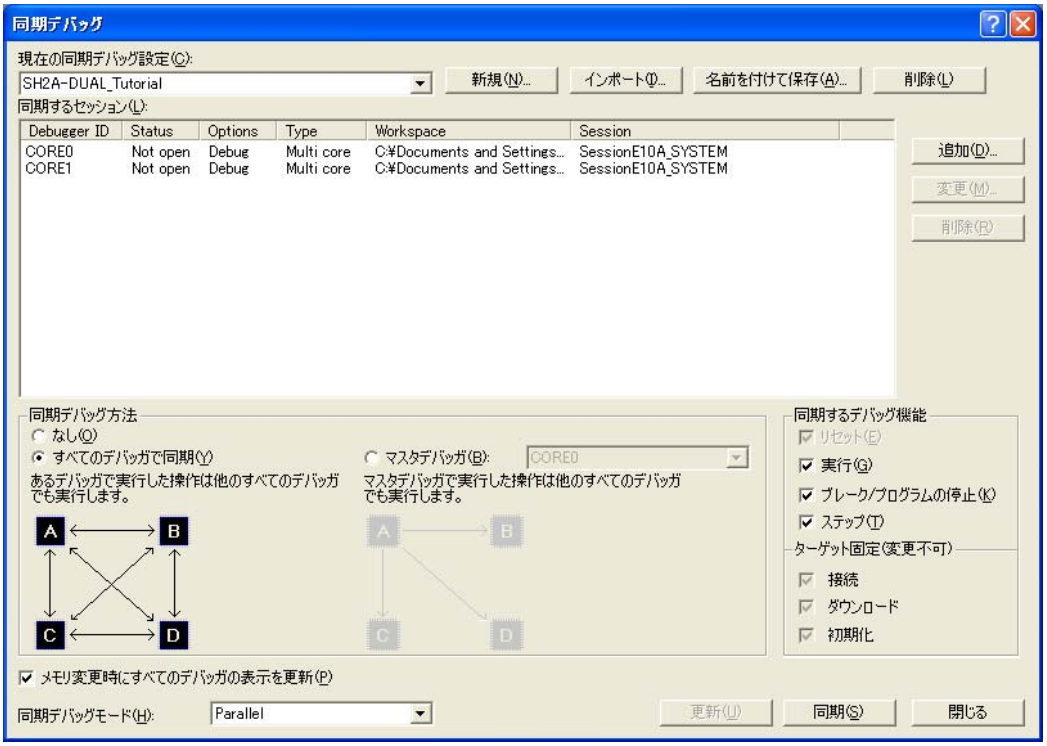


図 6.4 [同期デバッグ] ダイアログボックス

[同期]ボタンをクリックし、「3.9 システムチェック」の手順にしたがって High-performance Embedded Workshop を起動してください。

6.4 E10A-USB エミュレータのセットアップ

プログラムをダウンロードする前に、E10A-USB エミュレータの通信クロックをセットアップする必要があります。

- AUD clock
  AUD トレース取得時のクロックです。
  周波数が低いと、リアルタイムトレース機能使用時にデータ抜けの発生頻度が高くなります。
  周波数は、サポートデバイスの AUD clock 上限を超えないように設定してください。
  AUD トレース機能を使用するときのみ必要です。

- JTAG (H-UDI)clock（TCK）
  AUD トレース以外の通信クロックです。
  周波数が低いと、ダウンロードが遅くなります。
  周波数は、サポートデバイスの TCK 保証範囲の上限を超えないように設定してください。
  各製品の AUD clock 上限は、別冊の SHxxxx ご使用時の補足説明の「2.2.3 トレース機能 (4) AUD トレースの注意事項」を参照してください。
  また、各製品の TCK 上限は、別冊の SHxxxx ご使用時の補足説明の「2.2.4 JTAG（H-UDI）クロック（TCK）使用時の注意事項」を参照してください。

以下に、通信クロックを設定する方法について説明します。

6.5 [Configuration]ダイアログボックスの設定

通信クロックを設定するために、CPU1 用の High-performance Embedded Workshop にて[基本設定]メニューから[エミュレータ]を選択し、さらに[システム…]を選択してください。[Configuration]ダイアログボックスが表示されます。

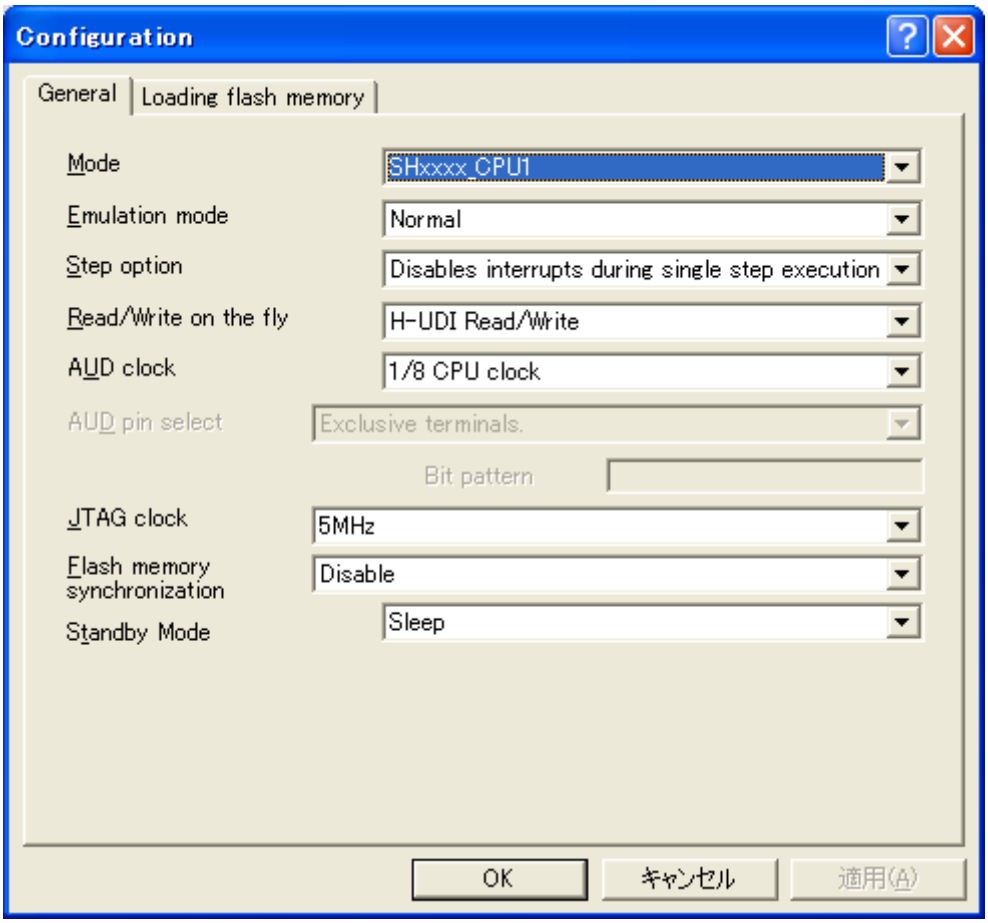


図 6.5 [Configuration]ダイアログボックス

[AUD clock]コンボボックスと、[JTAG clock]コンボボックスに適切な値を設定してください。デフォルトでも動作します。

【留意事項】

本ウィンドウで設定できる内容は、製品ごとに異なります。各製品の設定内容については、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、コンフィグレーションを設定してください。

6.6 ダウンロード先メモリの動作チェック

ダウンロードを行うメモリが正常に動作することをチェックします。

ダウンロード先のメモリが SDRAM/DRAM 等の場合、ダウンロードする前に CPU のバスコントローラの設定をする必要があります。使用するメモリに従った設定を前もって適切に行ってください。なお、バスコントローラは、[IO]ウィンドウから設定することができます。

バスコントローラの設定などのメモリ設定が完了したら、CPU1 用の High-performance Embedded Workshop にて[メモリ]ウィンドウでメモリ内容を表示、編集し、メモリが正常に動作することを確認します。

【留意事項】

メモリ動作チェックは上記だけでは不完全な場合があります。メモリチェック用プログラムを作成し、チェックすることをお勧めします。

[表示]メニューの[CPU]サブメニューから[メモリ...]を選択し、[表示開始アドレス]エディットボックスに“H’00000000”を入力し、[スクロール開始アドレス]エディットボックスに“H’00000000”を、[スクロール終了アドレス]エディットボックスに“H’FFFFFFFF”を入力してください。

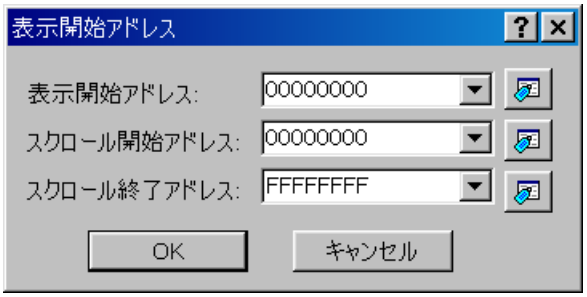


図 6.6 [表示開始アドレス]ダイアログボックス

[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。

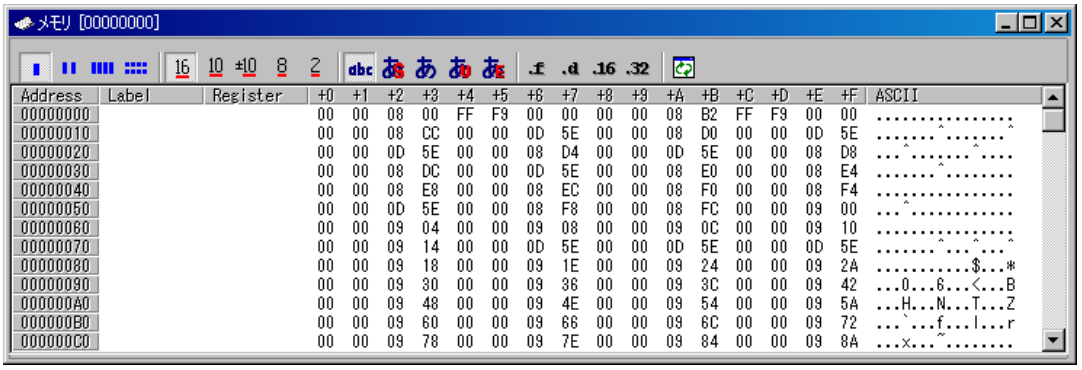


図 6.7 [メモリ]ウィンドウ

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

6.7 チュートリアルプログラムのダウンロード

6.7.1 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。
CPU0 用 High-performance Embedded Workshop および CPU1 用 High-performance Embedded Workshop でそれぞれソースレベルデバッグを行うために、CPU0/ CPU1 でそれぞれ別にデバッグ情報ファイルをロードしてください。

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

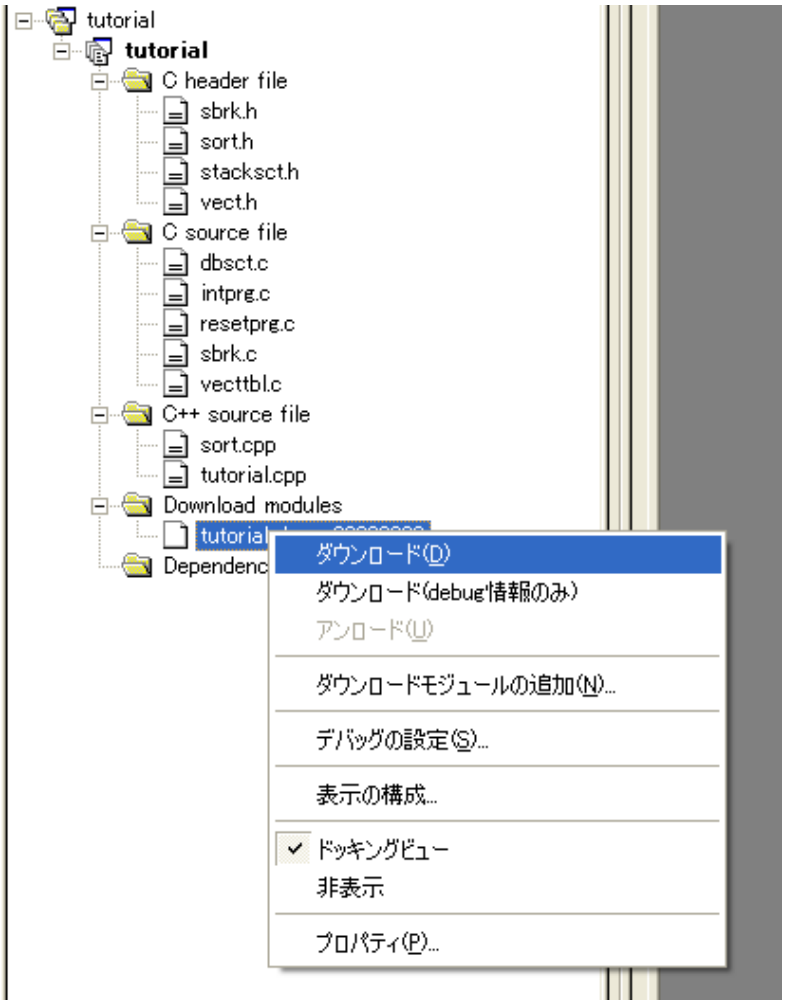


図 6.8 チュートリアルプログラムのダウンロード

6.7.2 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。

CPU1 用の High-performance Embedded Workshop にて[C++ source file]の[tutorial.cpp]をダブルクリックします。

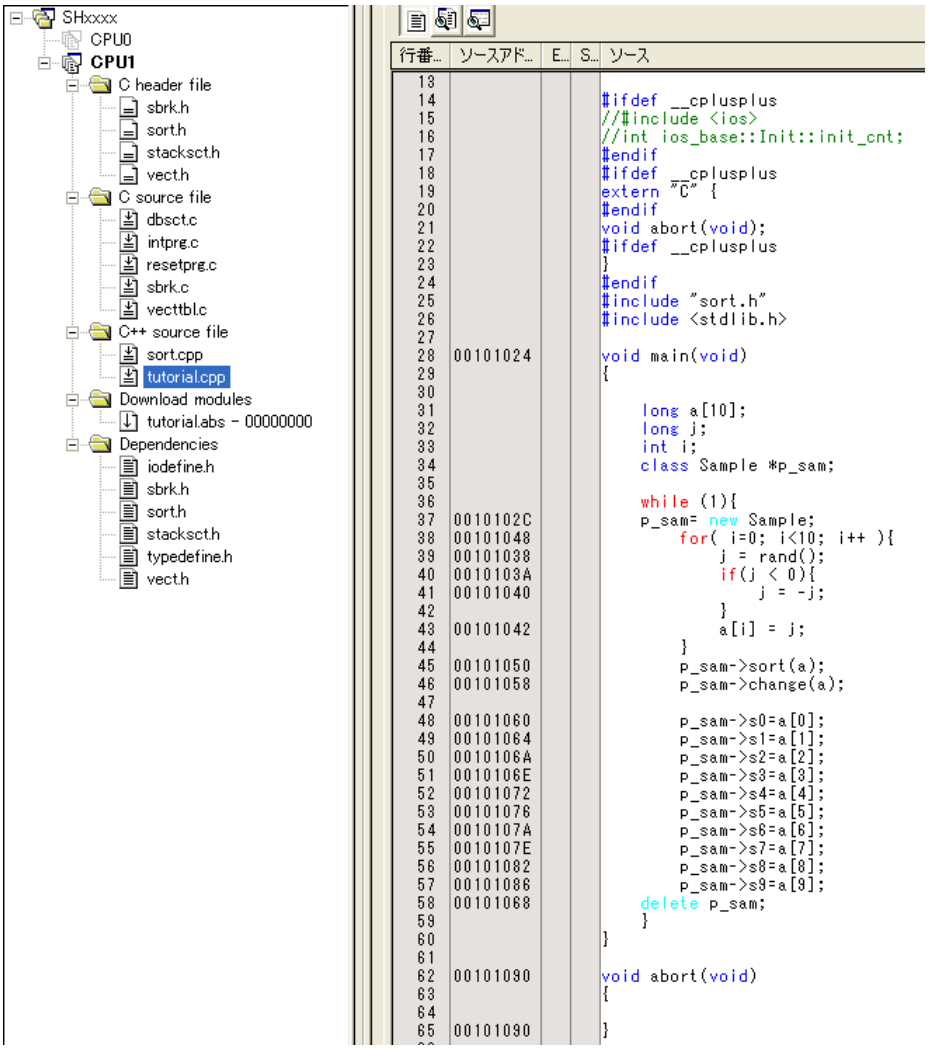


図 6.9 [エディタ]ウィンドウ（ソースプログラムの表示）

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

[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。

6.8 PC ブレークポイントの設定

簡単なデバッグ機能の 1 つに PC ブレークポイントがあります。
[エディタ]ウィンドウにおいて、PC ブレークポイントを簡単に設定できます。
例えば、sort 関数のコール箇所に PC ブレークポイントを設定します。

CPU1 用の High-performance Embedded Workshop にて sort 関数コールを含む行の[S/W ブレークポイント]カラムをダブルクリックしてください。

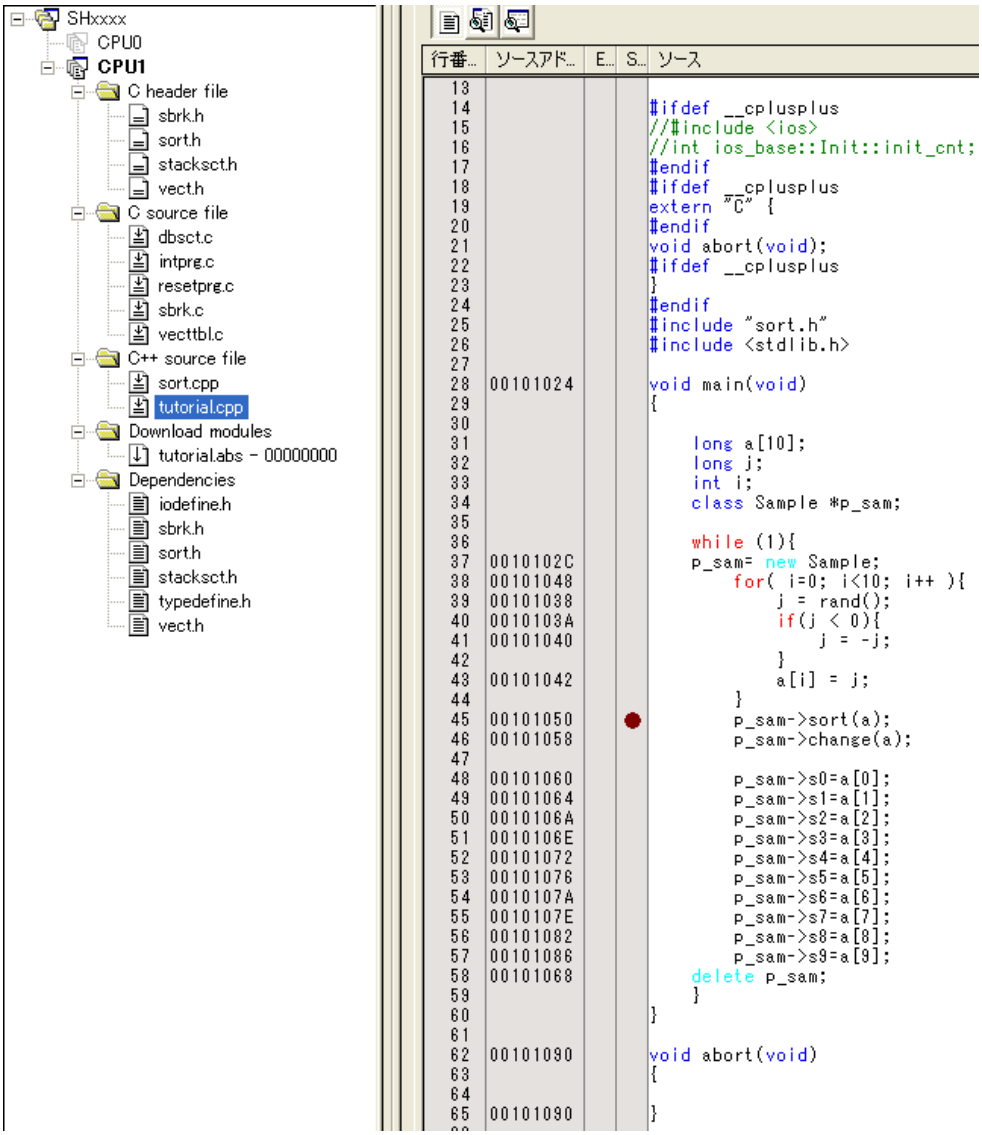


図 6.10 [エディタ]ウィンドウ（PC ブレークポイントの設定）

sort 関数を含む行に”●”と表示されます。この表示により PC ブレークポイントが設定されたことを示しています。

【留意事項】

PC ブレークポイントは、ROM 領域には設定できません。

6.9 レジスタ内容の変更

プログラムを実行する前に、プログラムカウンタおよびスタックポインタの値を設定してください。

CPU0 用および CPU1 用の High-performance Embedded Workshop にて[表示]メニューの[CPU]サブメニューから[レジスタ]を選択してください。[レジスタ]ウィンドウが表示されます。

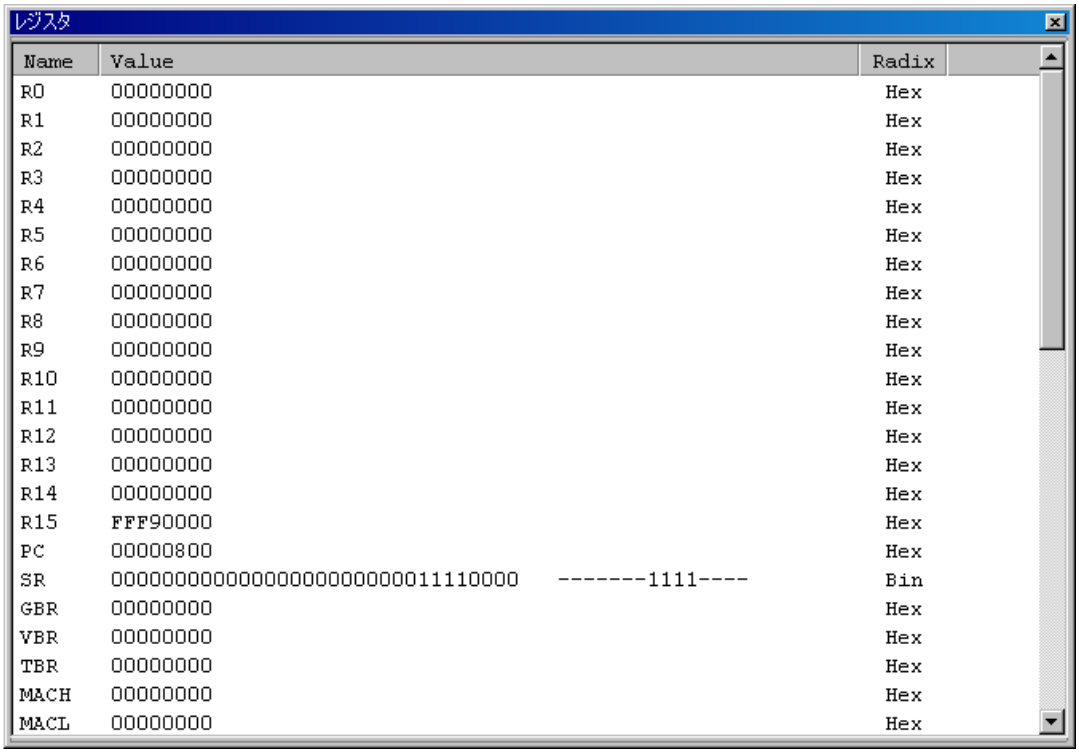


図 6.11 [レジスタ]ウィンドウ

プログラムカウンタ（PC）を変更する場合には、[レジスタ]ウィンドウで[PC]の数値エリアをマウスでダブルクリックすると、以下のダイアログボックスが表示され、値の変更が可能です。本チュートリアルプログラムでは、H´00000800 を設定し、[OK]ボタンをクリックしてください。

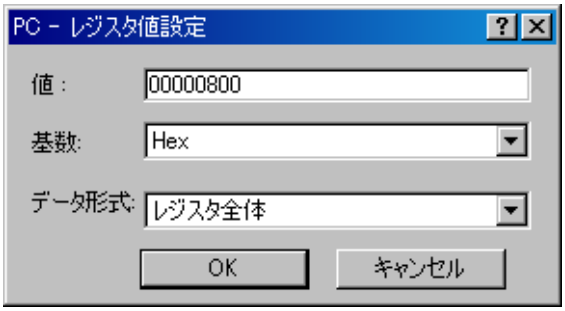


図 6.12 [レジスタ]ダイアログボックス（PC）

同じようにして、スタックポインタ（SP）を変更します。本チュートリアルプログラムでは、H’FFF90000 を設定してください。

フラッシュメモリ内蔵デバイスをご使用の場合、スタックポインタ（SP）は内蔵 RAM の最終アドレスを指定してください。

デバイスごとに内蔵 RAM の領域は異なります。ご使用のデバイスのハードウェアマニュアルを参照してください。

図 6.13 [レジスタ]ダイアログボックス（R15）

6.10 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。

[同期デバッグ]ダイアログボックスで同期実行([すべてのデバッガで同期]、[同期するデバッグ機能]の[実行]チェックボックス)が有効になっているので、プログラムを実行する場合は、いずれかの CPU の High-performance Embedded Workshop にて[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。

ここでは、CPU1 用のセッションで実行操作を行います。

図 6.14 [実行]ボタン

実行を開始すると、ステータスバーに”** RUNNING”と表示します。プログラムは CPU1 用の High-performance Embedded Workshop はブレークポイントを設定したところまで実行されます。CPU0 用の High-performance Embedded Workshop は同期ブレーク([すべてのデバッガで同期]、[同期するデバッグ機能]の[ブレーク/プログラムの停止]チェックボックス設定)が有効なため、CPU1 のブレークにしたがい CPU0 もブレークします。

それぞれの High-performance Embedded Workshop ではプログラムが停止した位置を示すために[S/W ブレークポイント]カラム中に矢印が表示されます。また、[BREAK POINT]メッセージがステータスバーに表示されます。

【留意事項】

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

< OS がインストールされているドライブ >
¥WorkSpace¥Tutorial¥E10A-USBM¥SH2A-DUAL¥SH2A-DUAL¥Tutorial¥CPU1¥source

2. 正常に実行できない場合、[デバッグ]メニューから[CPU のリセット]を選択し、一度リセットを発行してから、図 6.12 よりやり直してください。

```
28  00101024        void main(void)
29                  {
30
31                      long a[10];
32                      long j;
33                      int i;
34                      class Sample *p_sam;
35
36                      while (1){
37  0010102C            p_sam= new Sample;
38  00101048                for( i=0; i<10; i++ ){
39  00101038                    j = rand();
40  0010103A                    if(j < 0){
41  00101040                        j = -j;
42                              }
43  00101042                    a[i] = j;
44                          }
45  00101050                p_sam->sort(a);
46  00101058                p_sam->change(a);
47
48  00101060                p_sam->s0=a[0];
49  00101064                p_sam->s1=a[1];
50  0010106A                p_sam->s2=a[2];
51  0010106E                p_sam->s3=a[3];
52  00101072                p_sam->s4=a[4];
53  00101076                p_sam->s5=a[5];
54  0010107A                p_sam->s6=a[6];
55  0010107E                p_sam->s7=a[7];
56  00101082                p_sam->s8=a[8];
57  00101086                p_sam->s9=a[9];
58  00101068            delete p_sam;
59                      }
60                  }
61
62  00101090        void abort(void)
```

図 6.15 [エディタ]ウィンドウ（ブレーク状態）

[ステイタス]ウィンドウで最後に発生したブレークの要因が確認できます。

[表示]メニューの[CPU]サブメニューから[ステイタス]を選択してください。
[ステイタス]ウィンドウが表示されますので、[Platform]シートを開いて Cause of last break の Status を確認してください。

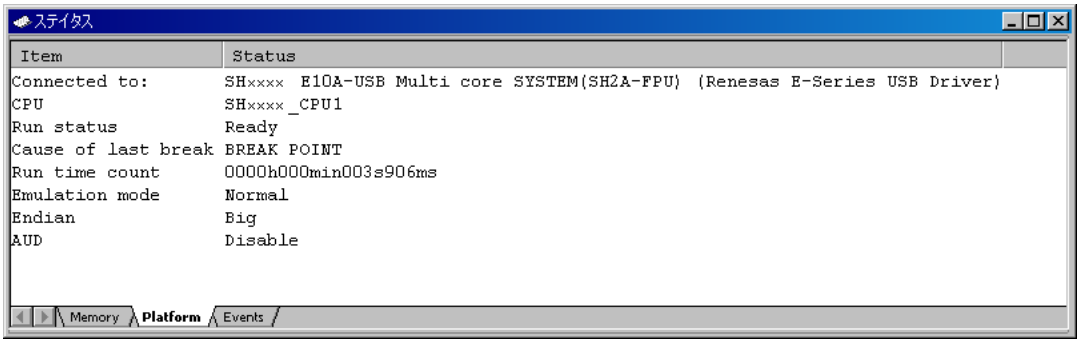


図 6.16 [ステイタス]ウィンドウ

【留意事項】

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

6.11 ブレークポイントの確認

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

CPU1 用の High-performance Embedded Workshop にて[表示]メニューの[コード]サブメニューから[イベントポイント]を選択してください。[イベントポイント]ウィンドウが表示されます。[Breakpoint]シートを開きます。

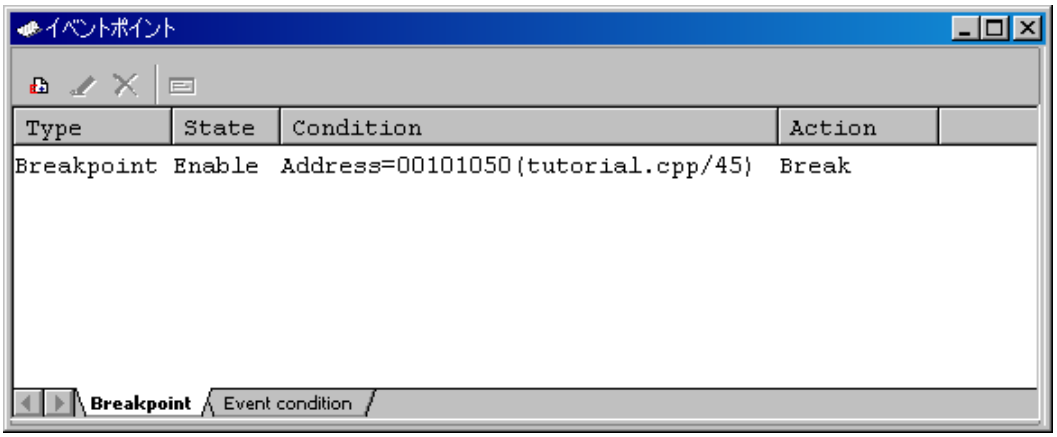


図 6.17 [イベントポイント]ウィンドウ

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

6.12 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。

CPU1 用の High-performance Embedded Workshop にて[表示]メニューの[シンボル]サブメニューから[ラベル]を選択してください。[ラベル]ウィンドウが表示され、モジュール内のシンボル情報が参照できます。

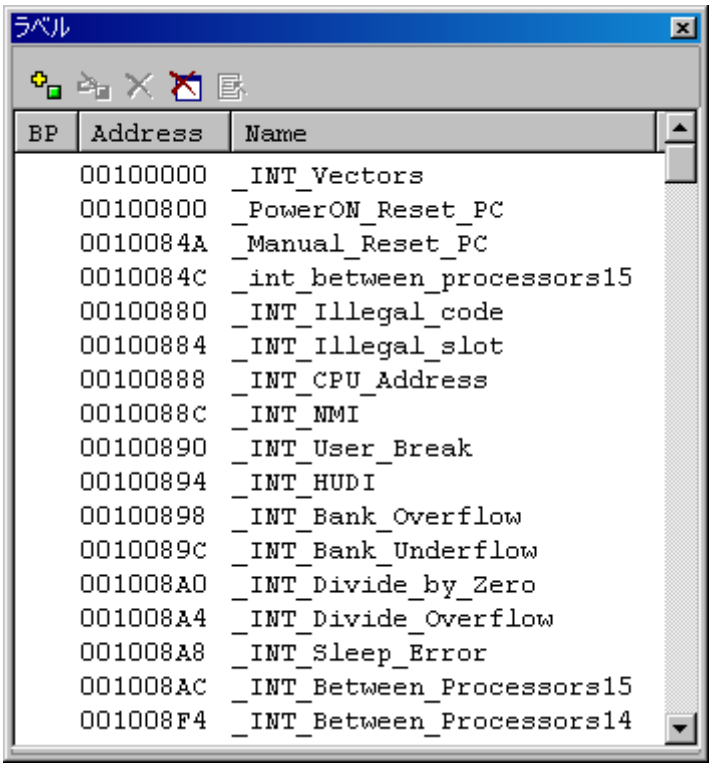


図 6.18 [ラベル]ウィンドウ

6.13 メモリ内容の確認

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

CPU1 用の High-performance Embedded Workshop にて[表示]メニューの[CPU]サブメニューから[メモリ]を選択し、[表示開始アドレス]エディットボックスに” _main”を入力し、[スクロール開始アドレス]エディットボックスに”H’00000000”を、[スクロール終了アドレス]エディットボックスに”H’FFFFFFFF”を入力してください。

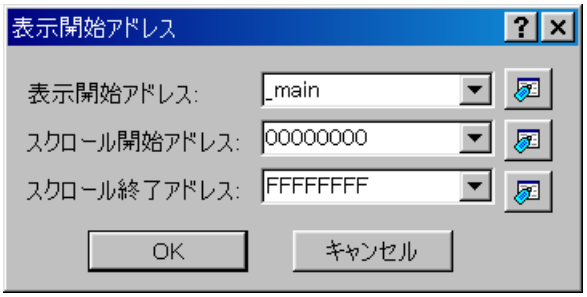


図 6.19 [表示開始アドレス]ダイアログボックス

[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。

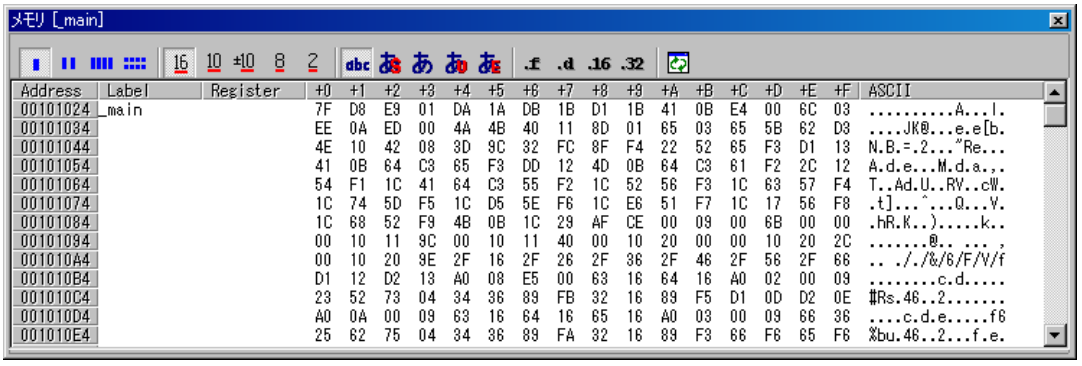


図 6.20 [メモリ]ウィンドウ

6.14 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。例えば、以下の手順で、プログラムのはじめに宣言した long 型の配列 a を見ることができます。

CPU1 用の High-performance Embedded Workshop の[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。

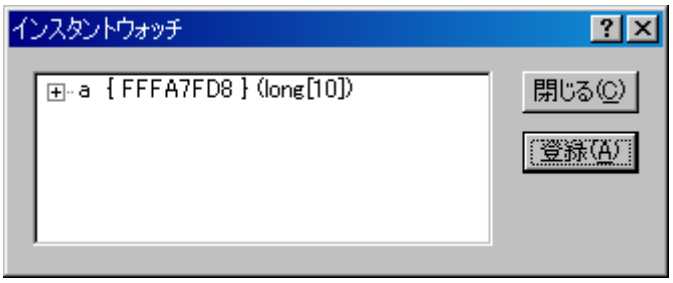


図 6.21 [インスタントウォッチ]ダイアログボックス

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

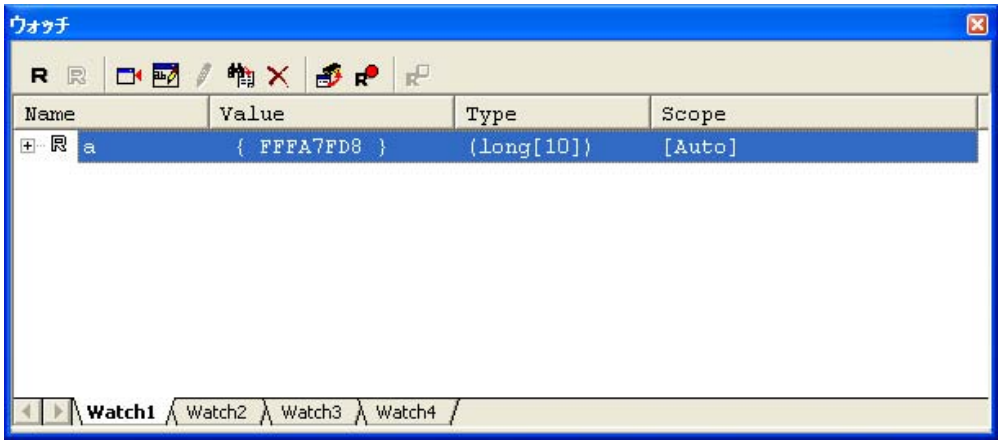


図 6.22 [ウォッチ]ウィンドウ（配列の表示）

また、変数名などを指定して、[ウォッチ]ウィンドウに変数など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ので、p_sam を入力してください。

図 6.23 [シンボル登録]ダイアログボックス

[OK]ボタンをクリックします。

[ウォッチ]ウィンドウに、インスタンス p_sam が表示されます。

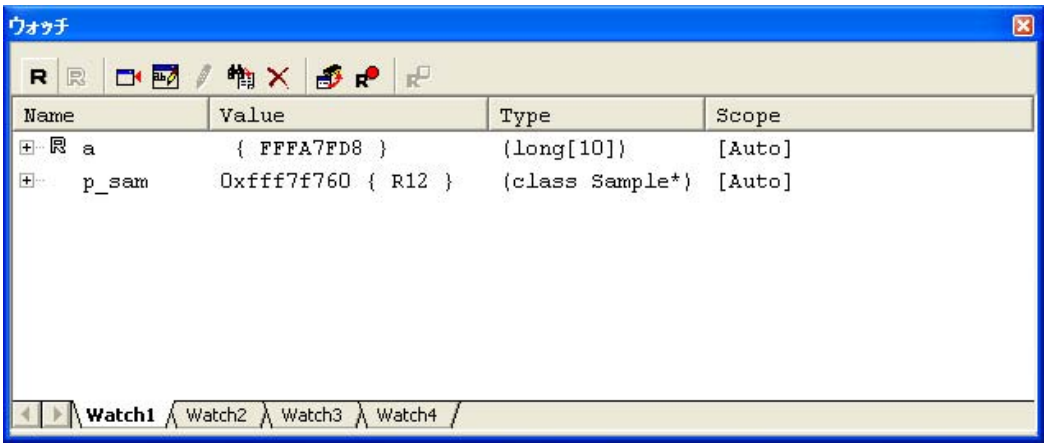


図 6.24 [ウォッチ]ウィンドウ（変数などの表示）

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

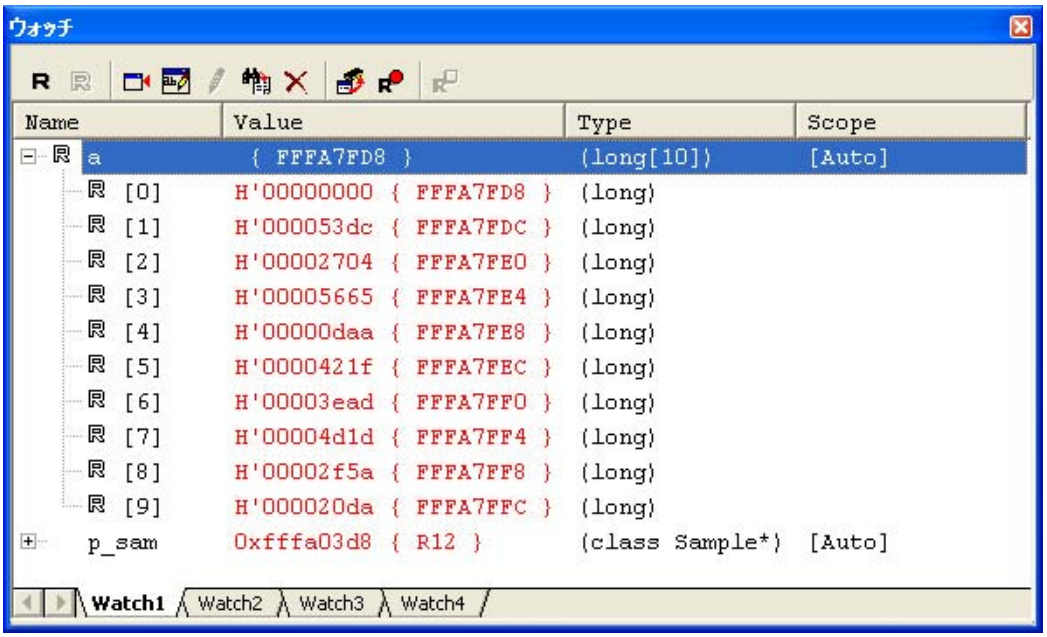


図 6.25 [ウォッチ]ウィンドウ（配列要素の表示）

6.15 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、main 関数のローカル変数を調べます。
この関数は、4 つのローカル変数 a, j, i, p_sam を宣言します。

CPU1 用の High-performance Embedded Workshop の[表示]メニューの[シンボル]サブメニューから[ローカル]を選択してください。[ローカル]ウィンドウが表示されます。

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

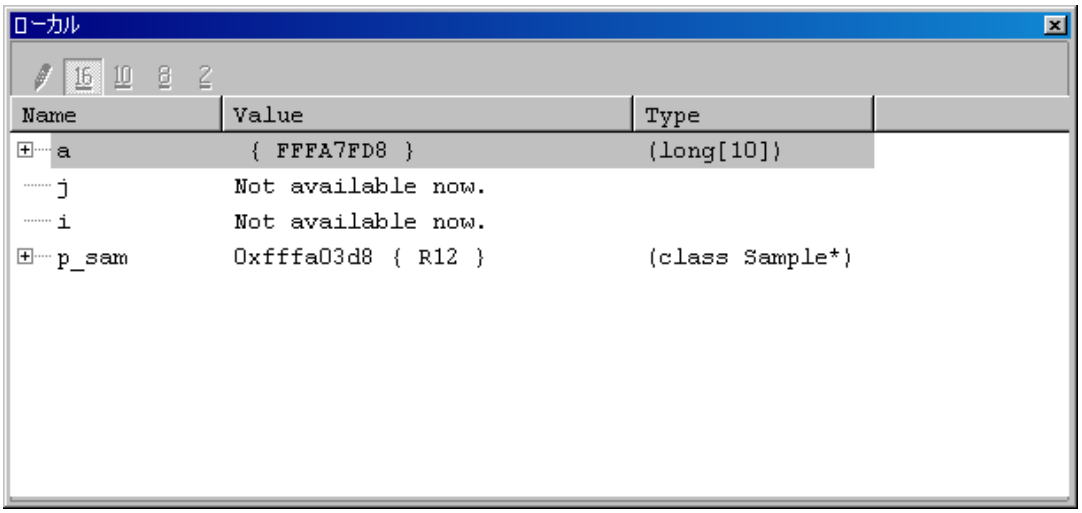


図 6.26 [ローカル]ウィンドウ

[ローカル]ウィンドウの配列 a の左側にある”+”マークをクリックし、配列 a の構成要素を表示させてください。
sort 関数実行前と実行後の配列 a の要素を参照すると、ランダムデータが降順にソートされていることがわかります。

6.16 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 6.1 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。

6.16.1 ステップインの実行

ステップイン機能はコール関数の中に入り、コール関数の先頭のステートメントで停止します。

sort 関数の中に入るためには、CPU1 用の High-performance Embedded Workshop の[デバッグ]メニューから[ステップイン]を選択するか、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。

[同期デバッグ]ダイアログボックスで同期ステップ（[すべてのデバッガで同期]、[同期するデバッガ機能]の[ステップ]チェックボックス）が有効なため、この操作により同期ステップインが行われます。

図 6.27 [ステップイン]ボタン

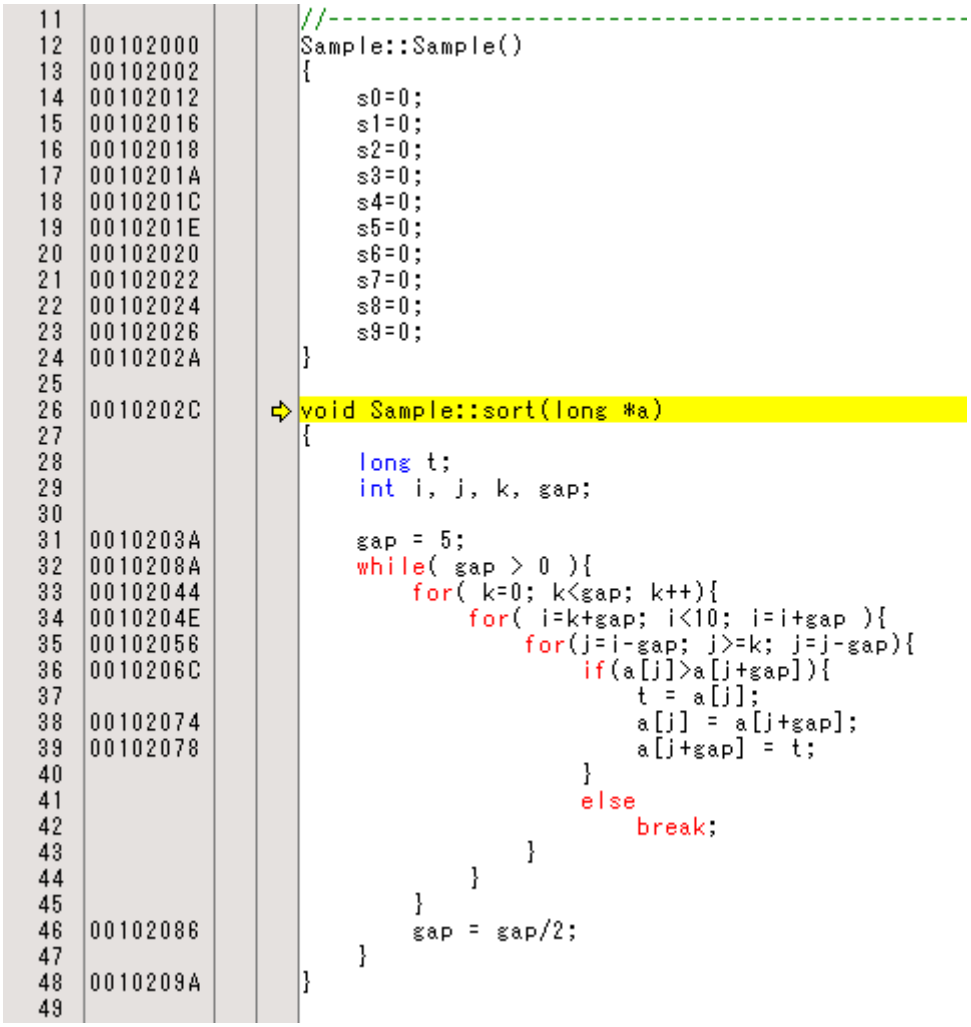


図 6.28 [エディタ]ウィンドウ（ステップイン）

CPU1 用の High-performance Embedded Workshop の[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。

6.16.2 ステップアウトの実行

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

sort 関数の中から抜け出すために、CPU1 用の High-performance Embedded Workshop の[デバッグ]メニューから[ステップアウト]を選択するか、またはツールバーの[ステップアウト]ボタンをクリックしてください。

【留意事項】

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

図 6.29 [ステップアウト]ボタン

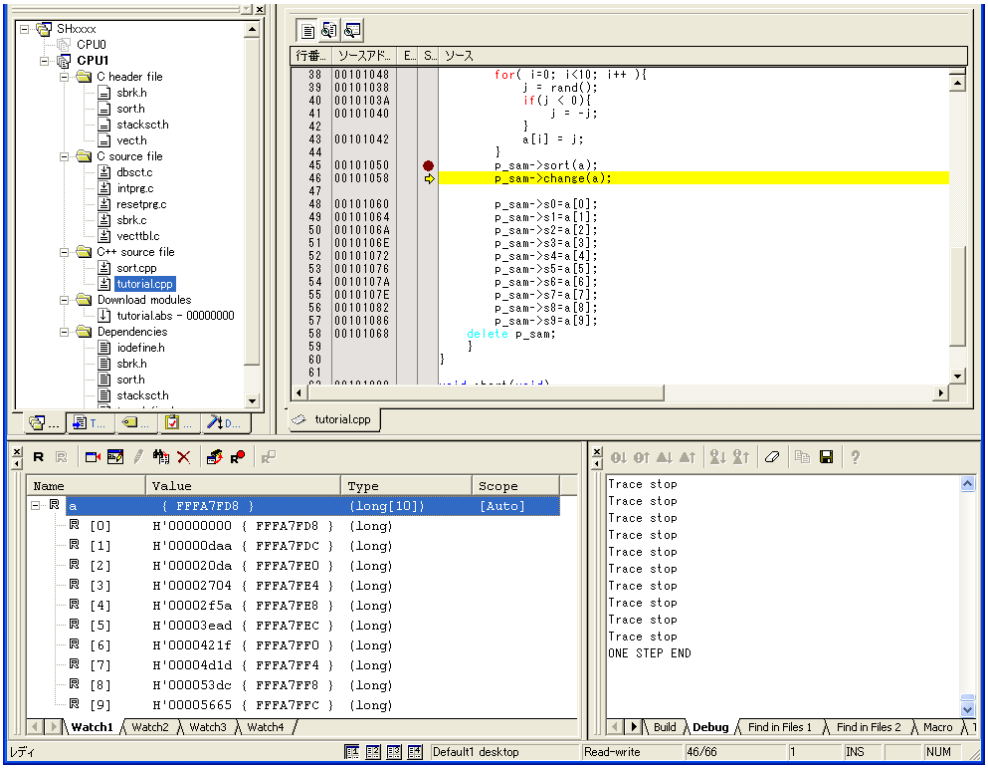


図 6.30 [High-performance Embedded Workshop]ウィンドウ（ステップアウト）

[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。

CPU0 用の High-performance Embedded Workshop の同期開始時のソース位置によっては、CPU0 側のステップアウトが終了しない場合があります。この場合は、ツールバー上の[STOP]ボタンを選択して終了させてください。

6.16.3 ステップオーバの実行

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

「6.16.2 ステップアウトの実行」の手順を実行し change 関数に移動してください。
次に、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、CPU1 用の High-performance Embedded Workshop の[デバッグ]メニューから[ステップオーバ]を選択するか、またはツールバーの[ステップオーバ]ボタンをクリックしてください。

図 6.31 [ステップオーバ]ボタン

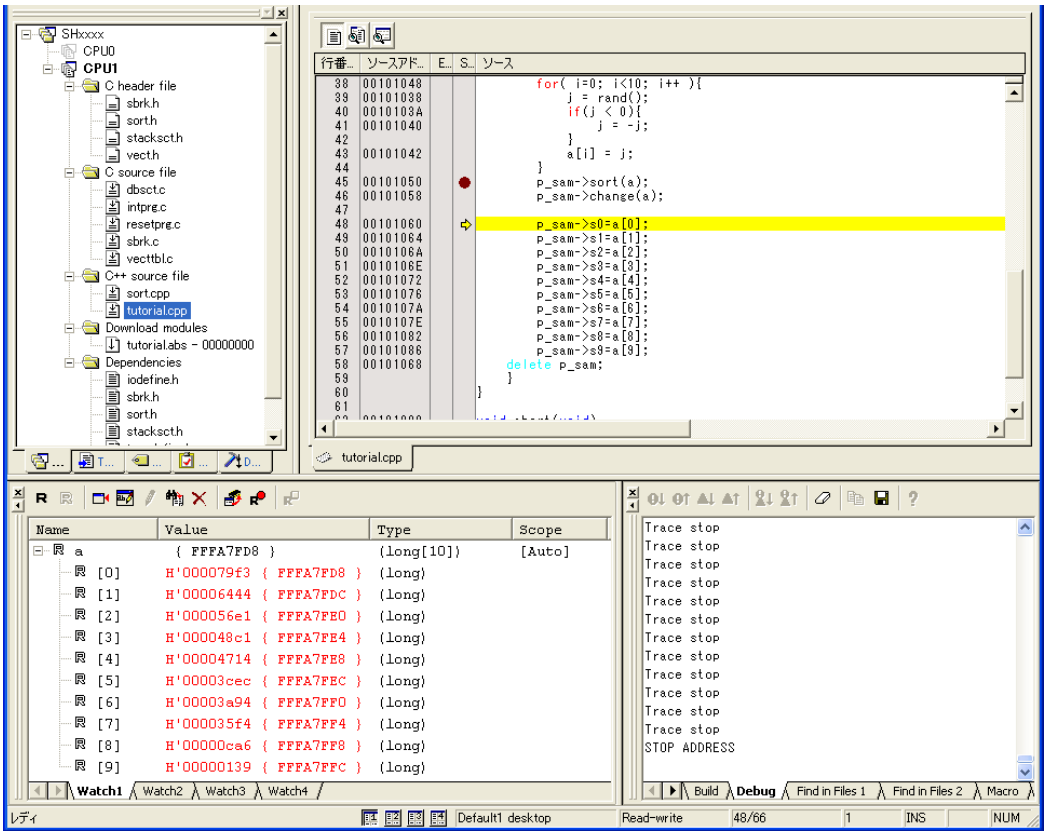


図 6.32 [High-performance Embedded Workshop]ウィンドウ（ステップオーバ）

6.17 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。

ブレークを全て解除してください。

main 関数の残り部分を実行するために、CPU1 用の High-performance Embedded Workshop の[デバッグ]メニューから[実行]を選択するか、ツールバー上の[実行]ボタンを選択してください。

図 6.33 [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、CPU1 用の High-performance Embedded Workshop の[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[STOP]ボタンを選択してください。

CPU0 用の High-performance Embedded Workshop は同期ブレーク([すべてのデバッガで同期]、[同期するデバッグ機能]の[ブレーク/プログラムの停止]チェックボックス)が有効なため、CPU1 のブレークにしたがい CPU0 もブレークします。

図 6.34 [STOP]ボタン

6.18 ブレーク機能

E10A-USB エミュレータは、PC ブレーク機能とハードウェアブレーク機能を持っています。
High-performance Embedded Workshop では、PC ブレークポイントの設定を[イベントポイント]ウィンドウの[Breakpoint]シートで、また、ハードウェアブレーク条件の設定を[Event condition]シートでそれぞれ行うことができます。

以下にブレーク機能の概要と設定方法について説明します。

6.18.1 PC ブレーク機能

E10A-USB エミュレータは、255 ポイントまで PC ブレークを設定することができます。
本章では、「6.8 PC ブレークポイントの設定」でご紹介した以外の設定方法を説明します。

CPU1 用の High-performance Embedded Workshop の[表示]メニューの[コード]サブメニューから[イベントポイント]を選択してください。[イベントポイント]ウィンドウが表示されます。
[Breakpoint]シートを開きます。

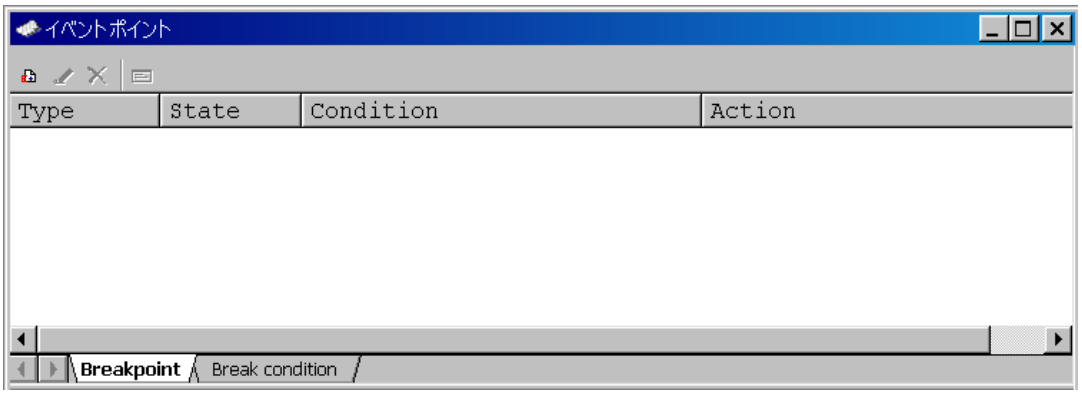


図 6.35 [イベントポイント]ウィンドウ（PC ブレーク設定前）

マウスの右ボタンで[イベントポイント]ウィンドウをクリックし、ポップアップメニューから[追加…]を選択してください。

[Address]エディットボックスにアドレス H’00101060 を入力してください。

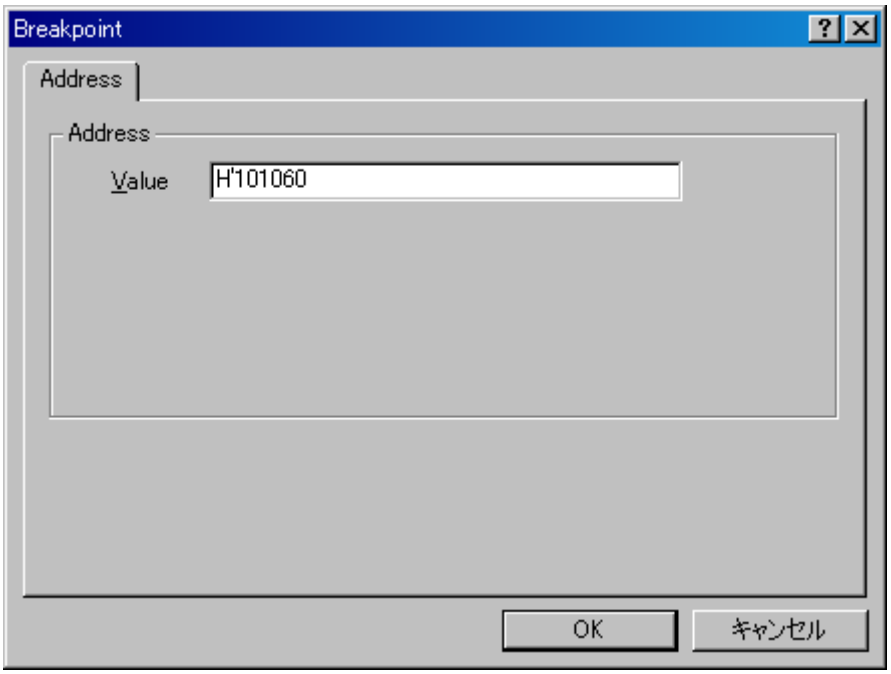


図 6.36 [Breakpoint]ダイアログボックス

【留意事項】

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

[OK]ボタンをクリックしてください。

[イベントポイント]ウィンドウには、設定された PC ブレークポイントが表示されます。

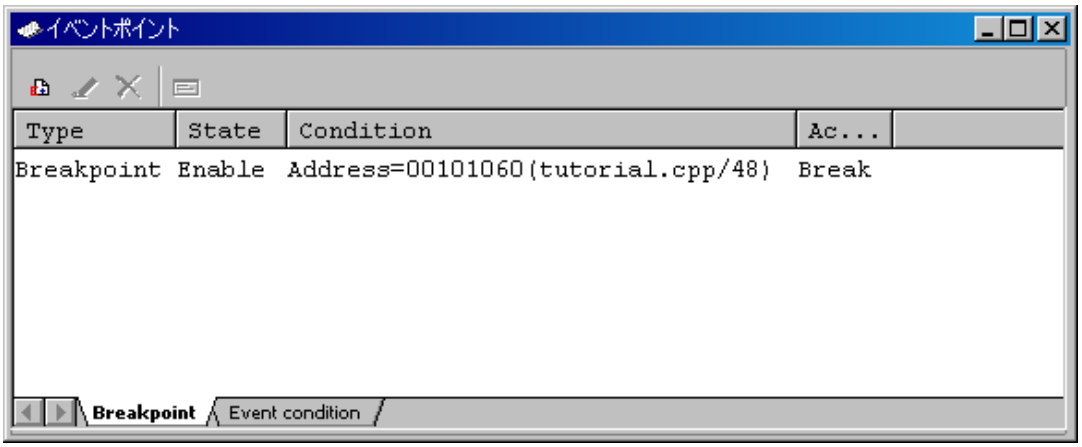


図 6.37 [イベントポイント]ウィンドウ（PC ブレーク設定時）

【留意事項】

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

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

「6.9 レジスタ内容の変更」で設定したプログラムカウンタ、スタックポインタ（PC=H’00000800、R15=H’FFF90000）を CPU0 用および CPU1 用の High-performance Embedded Workshop の[レジスタ]ウィンドウに設定して、CPU0 用または CPU1 用いずれかの High-performance Embedded Workshop の[実行]ボタンをクリックしてください。

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

設定した PC ブレークポイントまで、プログラムを実行して停止します。

```
 28  00101024        void main(void)
 29                  {
 30
 31                      long a[10];
 32                      long j;
 33                      int i;
 34                      class Sample *p_sam;
 35
 36                      while (1){
 37  0010102C            p_sam= new Sample;
 38  00101048                for( i=0; i<10; i++ ){
 39  00101038                    j = rand();
 40  0010103A                    if(j < 0){
 41  00101040                        j = -j;
 42                              }
 43  00101042                    a[i] = j;
 44                          }
 45  00101050                p_sam->sort(a);
 46  00101058                p_sam->change(a);
 47
 48  00101060                p_sam->s0=a[0];
 49  00101064                p_sam->s1=a[1];
 50  0010106A                p_sam->s2=a[2];
 51  0010106E                p_sam->s3=a[3];
 52  00101072                p_sam->s4=a[4];
 53  00101076                p_sam->s5=a[5];
 54  0010107A                p_sam->s6=a[6];
 55  0010107E                p_sam->s7=a[7];
 56  00101082                p_sam->s8=a[8];
 57  00101086                p_sam->s9=a[9];
 58  00101068            delete p_sam;
 59                      }
 60  }
```

図 6.38 実行停止時の[エディタ]ウィンドウ（PC ブレーク）

[ステイタス]ウィンドウの表示内容は、以下のようになります。

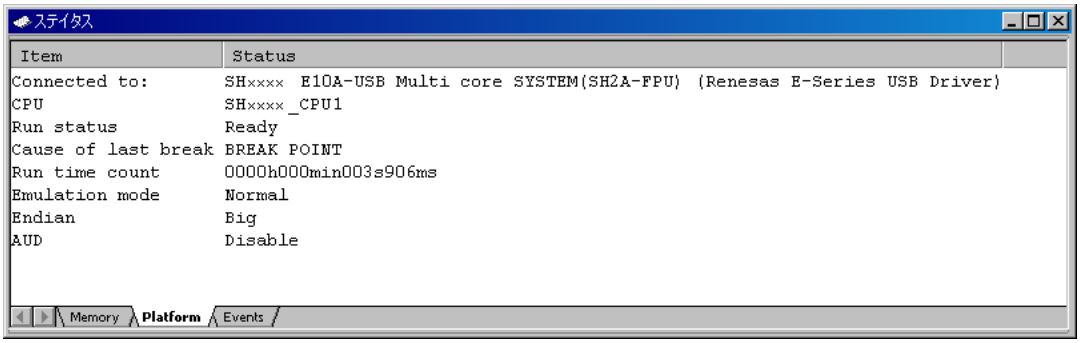


図 6.39 [ステイタス]ウィンドウの表示内容（PC ブレーク）

【留意事項】

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

6.19 ハードウェアブレーク機能

ハードウェアブレーク条件 Ch1(IA_OA_DT_CT)にアドレスバス条件を設定する方法を説明します。

CPU1 用の High-performance Embedded Workshop にて[表示]メニューの[コード]サブメニューから[イベントポイント]を選択してください。[イベントポイント]ウィンドウが表示されます。

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

次は Ch1(IA_OA_DT_CT)を設定します。

[Event condition]タブをクリックしてください。

ハードウェアブレーク条件 Event condition は、CPU 毎に 11 ポイントまで独立に条件を設定することができます。

ここでは、ハードウェアブレーク条件 Ch1(IA_OA_DT_CT)を設定します。

【留意事項】

ハードウェアブレーク条件の本数は、製品ごとに異なります。各製品の仕様については、オンラインヘルプを参照してください。

[イベントポイント]ウィンドウ内の Ch1(IA_OA_DT_CT)行を選択してください。
Ch1(IA_OA_DT_CT)行が強調表示されますので、ダブルクリックしてください。

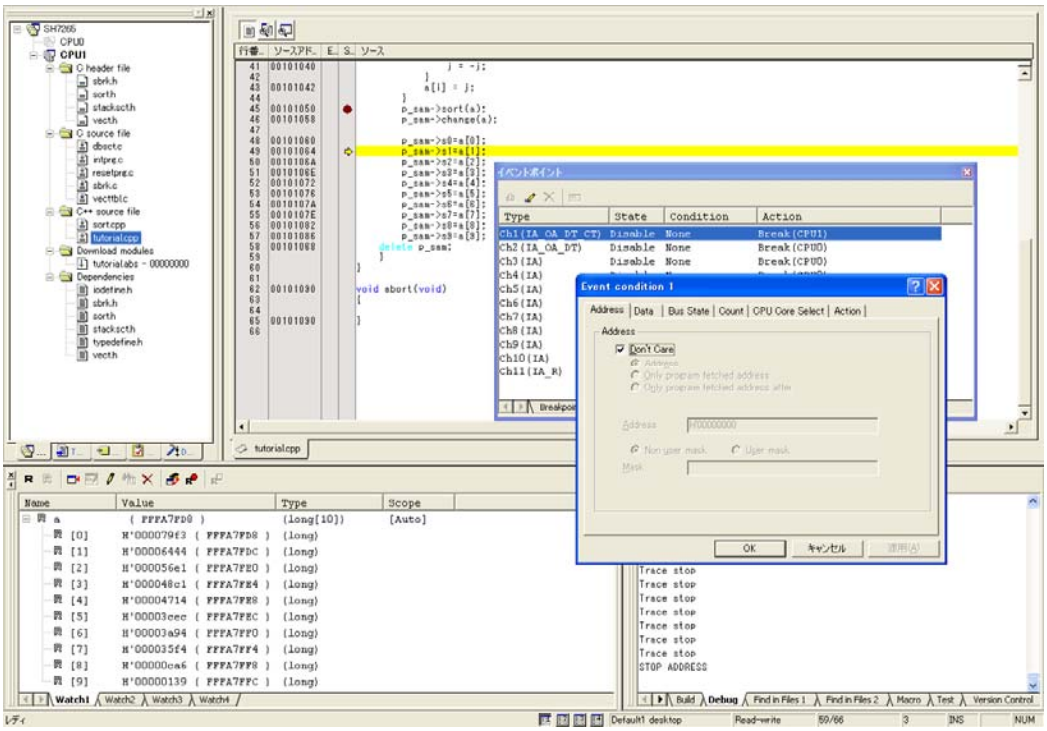


図 6.40 [High-performance Embedded Workshop]ウィンドウ（[Ch1(IA_OA_DT_CT)]）

[Event condition 1]ダイアログボックスが表示されます。
[Address]ページの[Don’t care]チェックボックスを無効にします。
[Only program fetched address]ラジオボタンを選択して、値として[Address]エディットボックスにアドレス H'00101050 を入力してください。

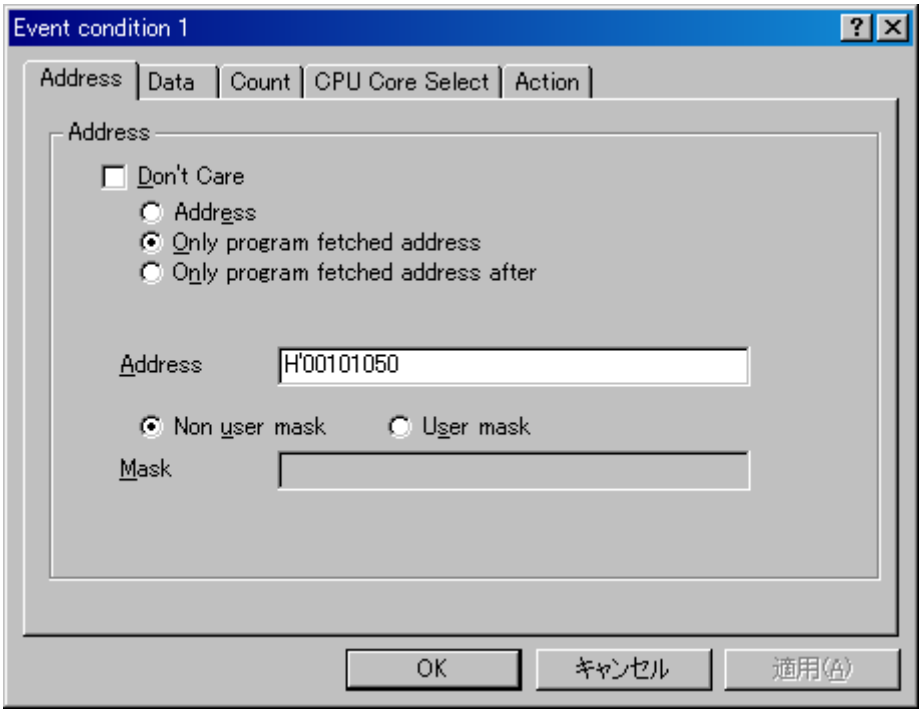


図 6.41 [Address]ページ（[Event condition1]ダイアログボックス）

【留意事項】

本ダイアログボックスで設定できる内容は、製品ごとに異なります。各製品の設定内容については、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。
State 行の 1 ポイント目の表示が”Disable”から”Enable”に変わります。
Condition 行の 1 ポイント目の表示が”None”から” Address=H'00101050(tutorial.cpp/45) pc Core Select: CPU1 Break(CPU1)”に変わります。
Action 行の 1 ポイント目の表示が”Break(CPU1)”と表示されます。

「6.9 レジスタ内容の変更」で設定したプログラムカウンタ、スタックポインタ（PC=H’00000800、R15=H’FFF90000）を CPU0 用および CPU1 用の High-performance Embedded Workshop の[レジスタ]ウィンドウに設定して、CPU0 用または CPU1 用のいずれかの High-performance Embedded Workshop の[実行]ボタンをクリックしてください。
デバイスごとに内蔵 RAM の領域は異なります。ご使用のデバイスのハードウェアマニュアルを参照してください。
正常に実行できない場合は、一旦リセットを発行してから上記手順を実行してください。

Break Condition 1 の条件まで、プログラムを実行して停止します。

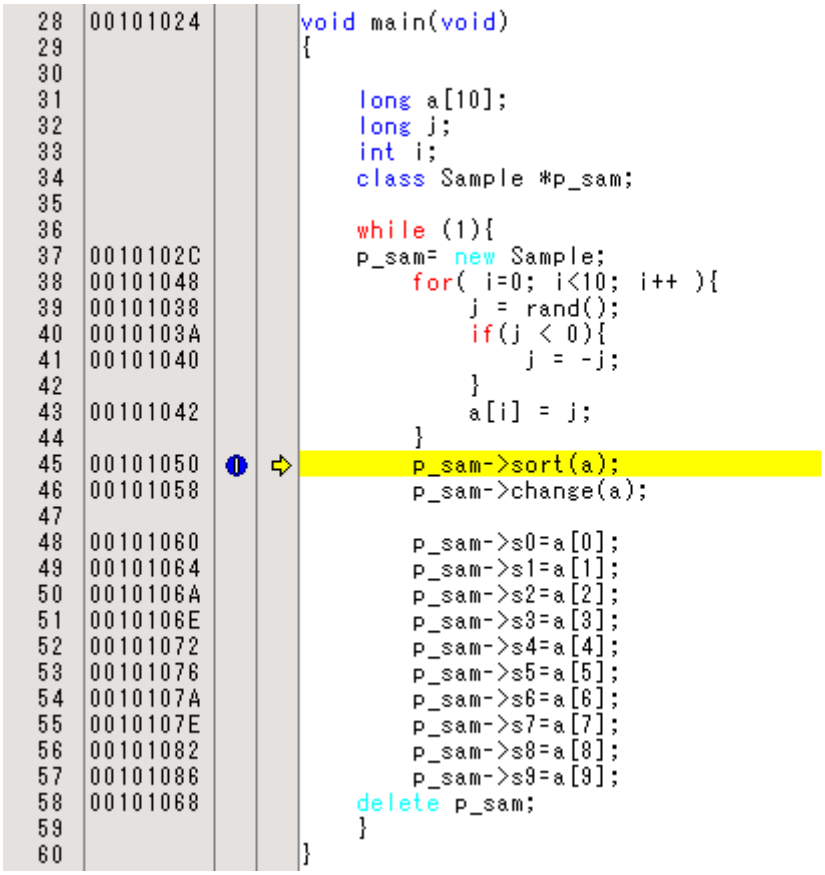


図 6.42 実行停止時の[エディタ]ウィンドウ（[Ch1(IA_OA_DT_CT)]）

[ステイタス]ウィンドウの表示内容は、以下のようになります。

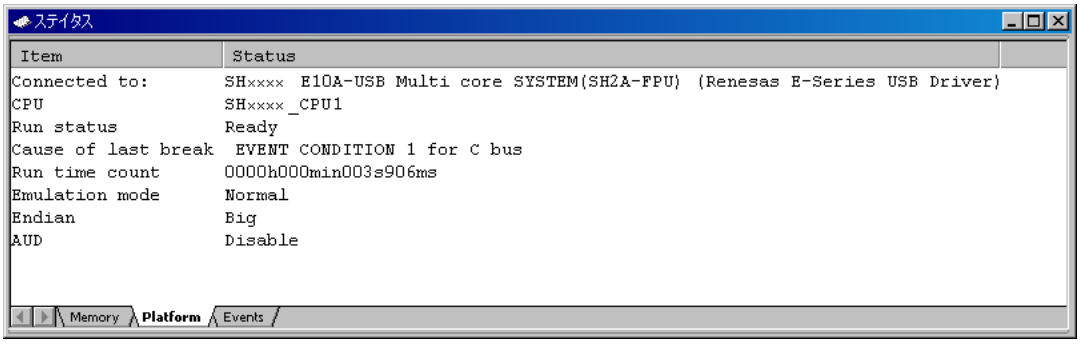


図 6.43 [ステイタス]ウィンドウの表示内容（[Ch1(IA_OA_DT_CT)]）

【留意事項】

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

6.19.1 シーケンシャルブレーク条件の設定

E10A-USB エミュレータは、シーケンシャルブレーク機能を持っています。

ハードウェアブレーク条件を次のように設定します。

1. Ch1(IA_OA_DT_CT)
   アドレス H'00101050 を実行した直後にブレーク条件が成立します。
2. Ch2(IA_OA_DT)
   アドレス H'00101042 を実行した直後にブレーク条件が成立します。

前の章でご紹介した設定方法にしたがって設定してください。

次に、これらのブレークポイントをシーケンシャルとする設定を行います。
[イベントポイント]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[Combination action(Sequential or PtoP)]を選択してください。
[Combination action(Sequential or PtoP)]ダイアログボックスが開きます。

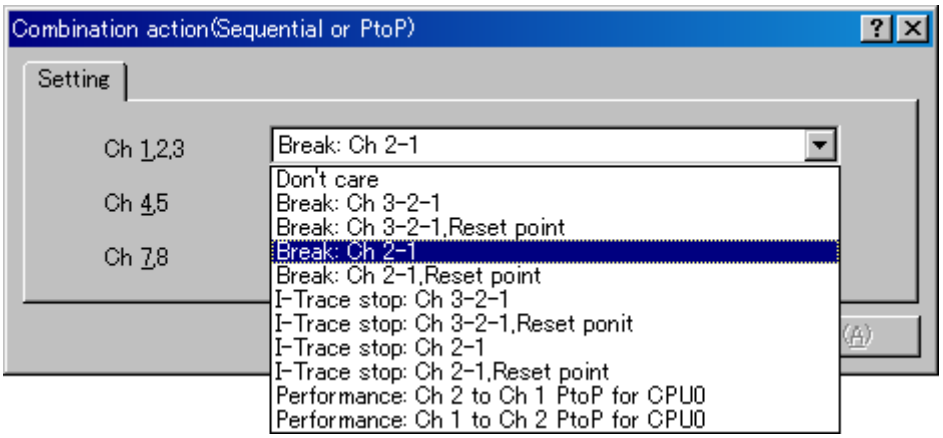


図 6.44 [Combination action(Sequential or PtoP)]ダイアログボックス

【留意事項】

本ダイアログボックスで表示される内容は、製品ごとに異なります。各製品の設定内容については、オンラインヘルプを参照してください。

[Break:Ch 2-1]を選択し、OK ボタンをクリックしてください。

設定完了後、[イベントポイント]ウィンドウの状態は以下のようになっています。

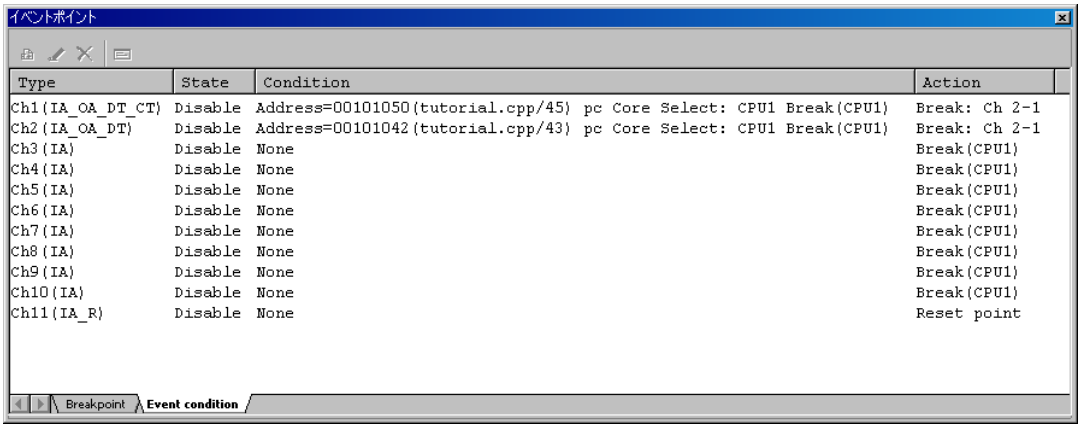


図 6.45 [Event condition]ページ

[Combination action(Sequential or PtoP)]設定直後は Ch1(IA_OA_DT_CT)および Ch2(IA_OA_DT)が無効になります。Ch1(IA_OA_DT_CT)および Ch2(IA_OA_DT)をそれぞれ選択して[イベントポイント]ウィンドウを右クリックし、[有効]を選択してください。

【留意事項】

本ダイアログボックスで表示される内容は、製品ごとに異なります。各製品の設定内容については、オンラインヘルプを参照してください。

次に、「6.9 レジスタ内容の変更」で設定したプログラムカウンタ、スタックポインタ（PC=H’00000800、R15=H’FFF90000）を CPU0 用および CPU1 用の High-performance Embedded Workshop の[レジスタ]ウィンドウに設定して、CPU0 用および CPU1 用の High-performance Embedded Workshop の[実行]ボタンをクリックしてください。

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

Event Condition 1 の条件まで、プログラムを実行して停止します。

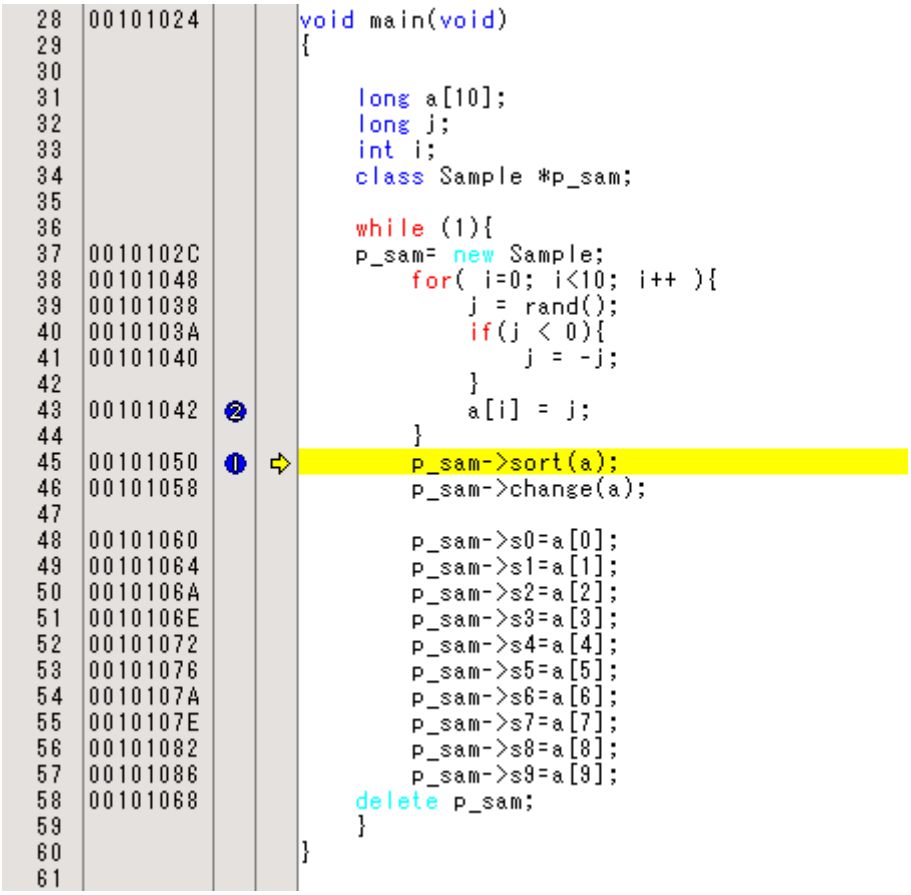


図 6.46 実行停止時の[エディタ]ウィンドウ（シーケンシャルブレーク）

[ステイタス]ウィンドウの表示内容は、以下のようになります。

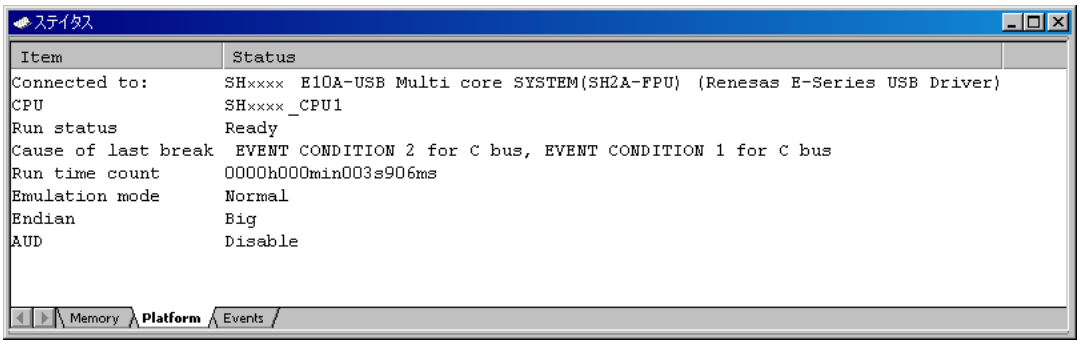


図 6.47 [ステイタス]ウィンドウの表示内容（シーケンシャルブレーク）

【留意事項】

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

設定したシーケンシャルブレーク条件を削除します。マウスの右ボタンで[イベントポイント]ウィンドウをクリックすることによって開くポップアップメニューから[すべてを削除]を選択し、設定されているハードウェアブレーク条件をすべて解除してください。

次に、[イベントポイント]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[シーケンシャル設定]を選択してください。

[Combination action(Sequential or PtoP)]ダイアログボックス（図 6.44）が開きます。

[Ch 1,2,3]ドロップダウンリストボックスより、[Don’t care]を選択し、OK ボタンをクリックしてください。

6.20 トレース機能

E10A-USB エミュレータには以下の 2 種類の分岐命令トレース機能があります。

- 内蔵トレース機能
  設定内容および表示内容については「5.7 トレース情報を見る」を参考にしてください。

- AUD トレース機能
  デバイスの AUD 端子を E10A-USB エミュレータに接続している場合に有効な、大容量のトレース機能です。トレース取得できるイベントの数は、分岐元／分岐先の組を 1 個とすると最大 262,144 個です。

設定内容および表示内容については「5.7 トレース情報を見る」を参考にしてください。

6.20.1 トレースウィンドウの表示方法

CPU1 用の High-performance Embedded Workshop にて[表示]メニューの[コード]サブメニューから[トレース]を選択してください。トレースの取得結果が表示されます。

6.20.2 内蔵トレース機能

分岐元／分岐先情報を最新の数分岐分トレースして表示します。
内蔵トレースでは、分岐命令のタイプを指定して取得することができます。
分岐命令のタイプの指定方法は、以下です。

[表示]メニューから[トレース]を選択してください。
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。[Acquisition]ダイアログボックスが表示されます。

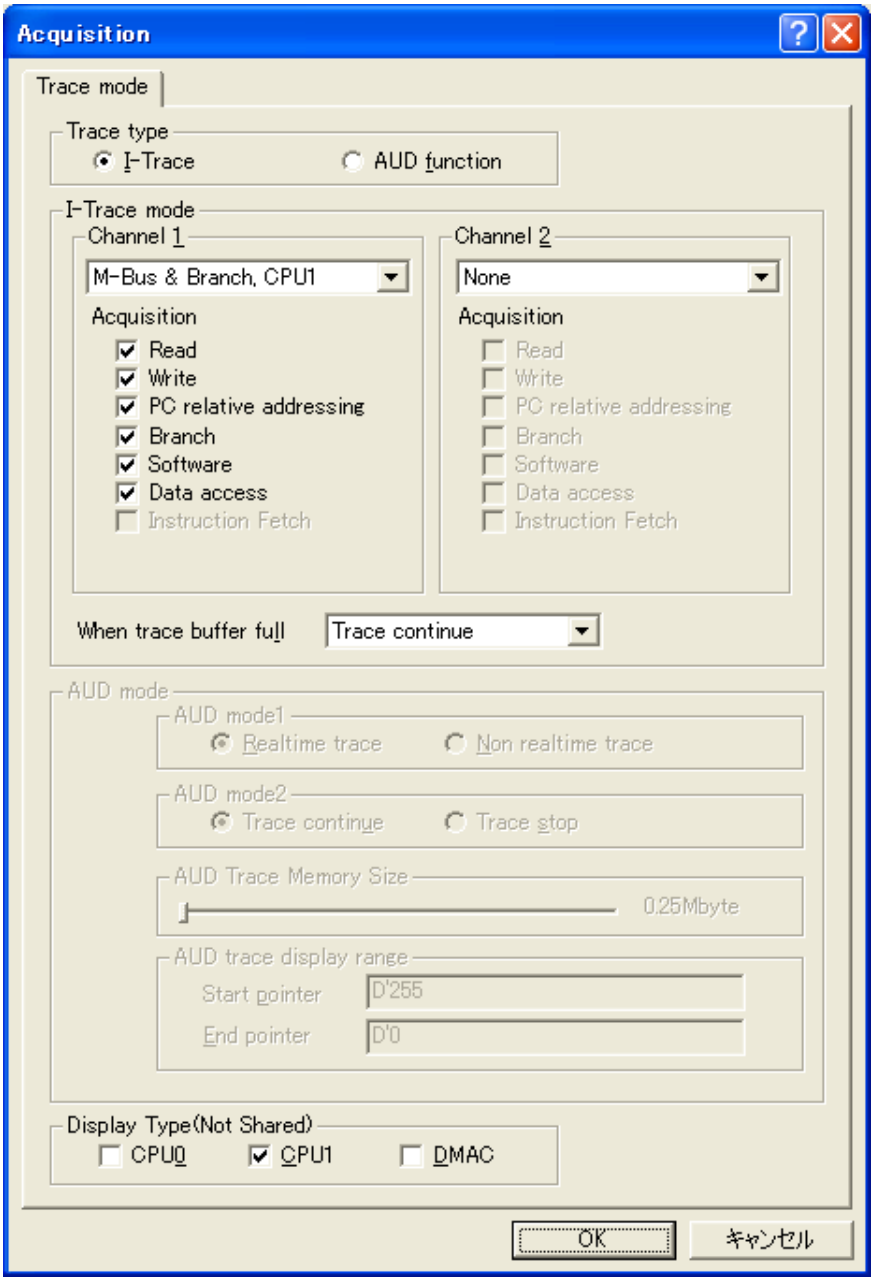


図 6.48 [Acquisition]ダイアログボックス

【留意事項】

本ウィンドウで設定できる内容は、製品ごとに異なります。各製品の設定内容については、オンラインヘルプを参照してください。

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

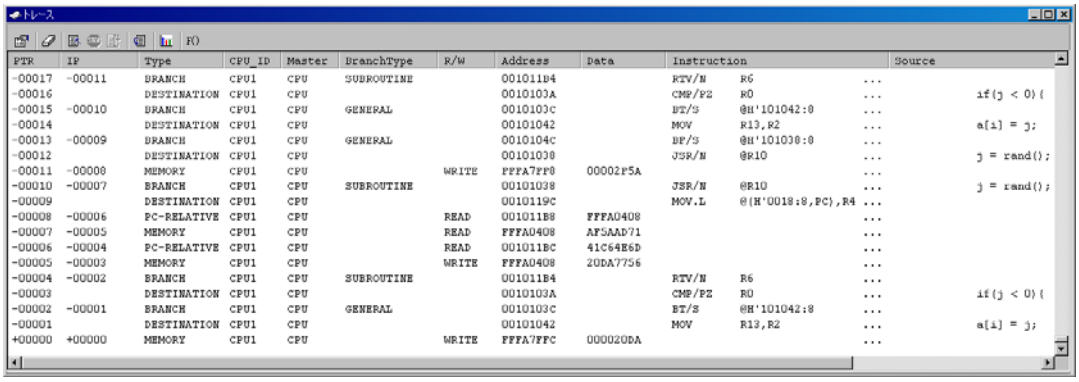


図 6.49 [トレース]ウィンドウ

必要ならば、タイトルバーの下のヘッダバーをドラッグして、カラムの幅を調節してください。

【留意事項】 取得できる情報の種類、数、トレース表示内容は、製品によって異なります。
各製品の仕様については、オンラインヘルプを参照してください。

6.20.3 AUD トレース機能

デバイスの AUD 端子を E10A-USB エミュレータに接続している場合に有効なトレース機能です。

AUD トレースの設定方法を以下に説明します。

(1) トレース取得方法の設定

[トレース]ウィンドウを表示してください。

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。[Acquisition]ダイアログボックスが表示されます。

[Trace type]で[AUD function]を選択してください。

[Trace mode]ページで、トレースの取得条件を設定します。

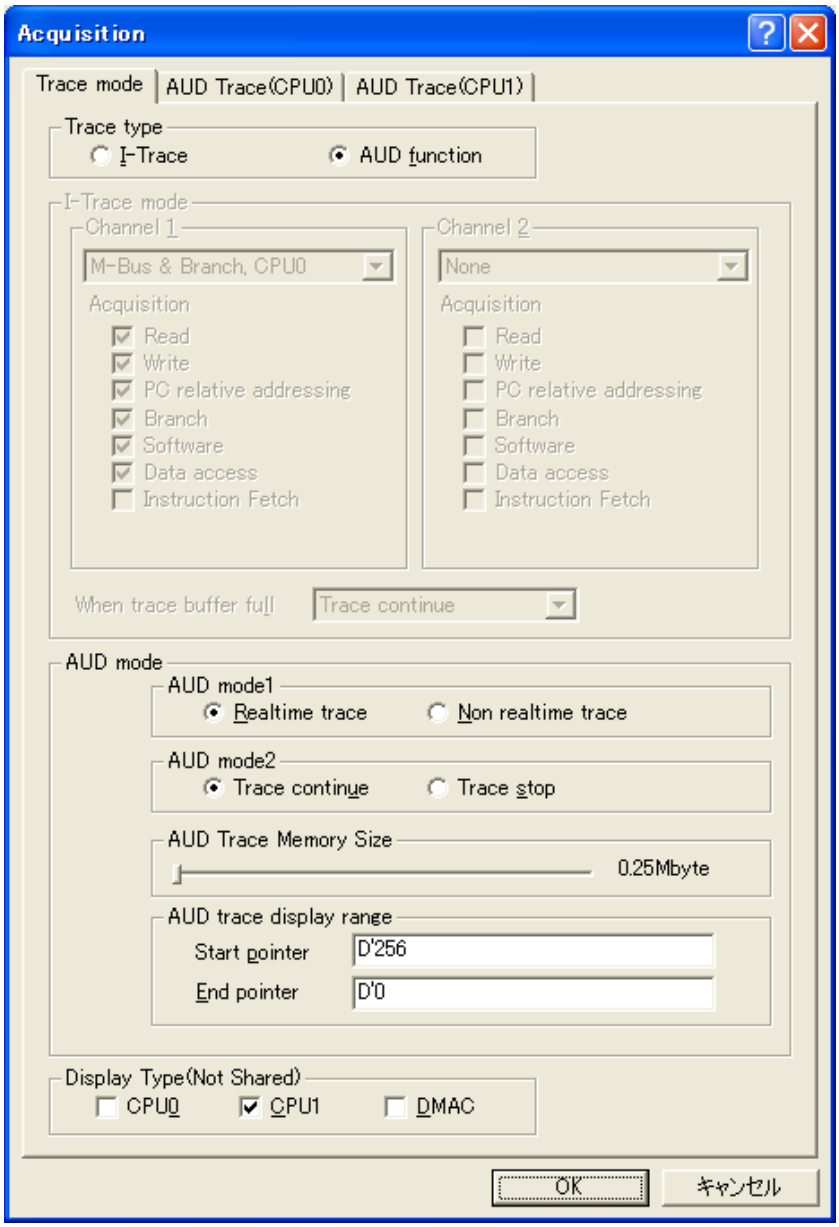


図 6.50 [Acquisition]ダイアログボックス

【留意事項】

AUD トレース機能をサポートしない製品では、本ページは使用できません。
本ウィンドウで設定できる内容は、製品ごとに異なります。各製品の設定内容については、オンラインヘルプを参照してください。

以下の表で、各オプションを説明します。

[AUD トレース取得モード]

種別	モード	説明
トレース出力が連続して発生した場合の取得モード	Realtime trace モード	トレース情報の発生が集中し、AUD 端子からの出力が間に合わなくなると CPU はトレース情報の出力を一時的に停止します。 このため、ユーザプログラムはリアルタイムに動作しますが、 トレース情報が一部取得できないことがあります。
	Non realtime trace モード	トレース情報の発生が集中し、AUD 端子からの出力が間に合わなくなると CPU の動作を一時的に停止し、トレース情報の出力を優先します。このため、ユーザプログラムのリアルタイム性がなくなります。
E10A-USB エミュレータのトレースバッファがフルになった場合の取得モード	Trace continue モード	古い情報を上書きして、常に最新の情報を取得します。
	Trace stop モード	その後のトレースを取得しません。 ユーザプログラムは継続して実行されます。

[AUD trace display range]

種別	説明
Start pointer	先頭ポインタを設定します。デフォルトは、D'255 が設定されています。
End pointer	最終ポインタを設定します。デフォルトは、D'0 が設定されています。

【留意事項】

本ウィンドウで設定できる内容は、製品ごとに異なります。各製品の設定内容については、オンラインヘルプを参照してください。

(2) トレース結果の表示

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

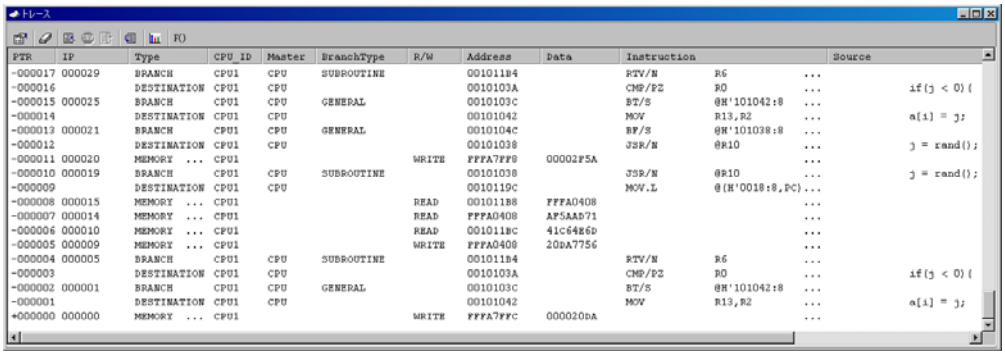


図 6.51 [トレース]ウィンドウ（表示例）

6.21 スタックトレース機能

E10A-USB エミュレータでは、スタック情報を用いて、現在の PC がある関数がどの関数からコールされているかを表示します。

【留意事項】

本機能は、Elf/Dwarf2 形式のデバッグ情報を持ったロードモジュールをロードした場合のみ使用できます。

Elf/Dwarf2 形式のデバッグ情報を持ったロードモジュールは、SHC/C++コンパイラ(OEM、バンドル販売品を含む)V6.0 以降でサポートしています。

CPU1 用の High-performance Embedded Workshop にて sort 関数内の行の[Event]カラムをダブルクリックして、Event ポイントを設定してください。

```
26 0010202C      void Sample::sort(long *a)
27               {
28                   long t;
29                   int i, j, k, gap;
30
31 0010203A          gap = 5;
32 0010208A          while( gap > 0 ){
33 00102044              for( k=0; k<gap; k++){
34 0010204E                  for( i=k+gap; i<10; i=i+gap ){
35 00102056                      for(j=i-gap; j>=k; j=j-gap){
36 0010206C  ●                       if(a[j]>a[j+gap]){
37                                       t = a[j];
38 00102074                              a[j] = a[j+gap];
39 00102078                              a[j+gap] = t;
40                                   }
41                                   else
42                                       break;
43                               }
44                           }
45                       }
46 00102086              gap = gap/2;
47               }
48 0010209A      }
49
50 0010209E      void Sample::change(long *a)
51               {
```

図 6.52 [エディタ]ウィンドウ（ハードウェアブレークの設定）

「6.9　レジスタ内容の変更」で設定したプログラムカウンタ、スタックポインタ（PC=H’00000800、R15=H’FFF90000）を CPU0 用および CPU1 用の High-performance Embedded Workshop の[レジスタ]ウィンドウに設定して、CPU0 用および CPU1 用の High-performance Embedded Workshop の[実行]ボタンをクリックしてください。
デバイスごとに内蔵 RAM の領域は異なります。ご使用のデバイスのハードウェアマニュアルを参照してください。
正常に実行できない場合は、一旦リセットを発行してから上記手順を実行してください。

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

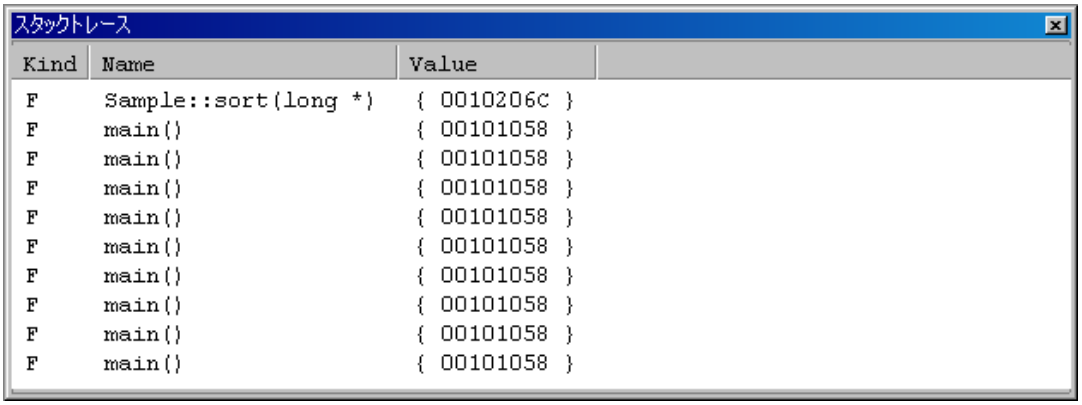


図 6.53　[スタックトレース]ウィンドウ

現在 PC が sort()関数内にあり、sort()関数は main()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[Event]カラムを再度ダブルクリックして、ハードウェアブレークを解除します。

【留意事項】

本機能の詳細はオンラインヘルプを参照してください。

6.22 パフォーマンス測定機能

E10A-USB エミュレータには、マイコンのパフォーマンスを測定する機能があります。

- パフォーマンス測定機能
  マイコン内蔵のカウンタを使用して、各種イベント発生回数、サイクル数を測定する機能です。
  開始条件、終了条件を設定できます。
  測定できる項目はサポートデバイスによって異なります。

6.22.1 パフォーマンス測定機能

マイコン内蔵のカウンタを使用して各種イベント発生回数、サイクル数を測定する例を説明します。

(1) 設定方法

CPU1 用の High-performance Embedded Workshop にて[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]を選択してください。
[パフォーマンス解析方法の選択]ダイアログボックスが開きますので、[OK]ボタンを押してください。

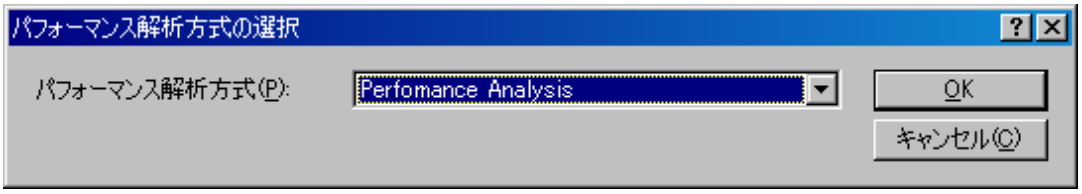


図 6.54 [パフォーマンス解析方法の選択]ダイアログボックス

[パフォーマンス解析]ウィンドウが表示されます。

本ウィンドウ内のチャネル行をダブルクリックしてください。
[Performance Analysis]ダイアログボックスが開きます。
本ダイアログボックスで測定するイベントや測定条件を設定することができます。

【留意事項】

本ダイアログボックスで表示される内容は、製品ごとに異なります。
各製品の設定内容については、オンラインヘルプを参照してください。

条件設定後、[OK]ボタンを押し、ユーザプログラムを実行すると、実行終了後に[パフォーマンス解析]ウィンドウに結果が表示されます。

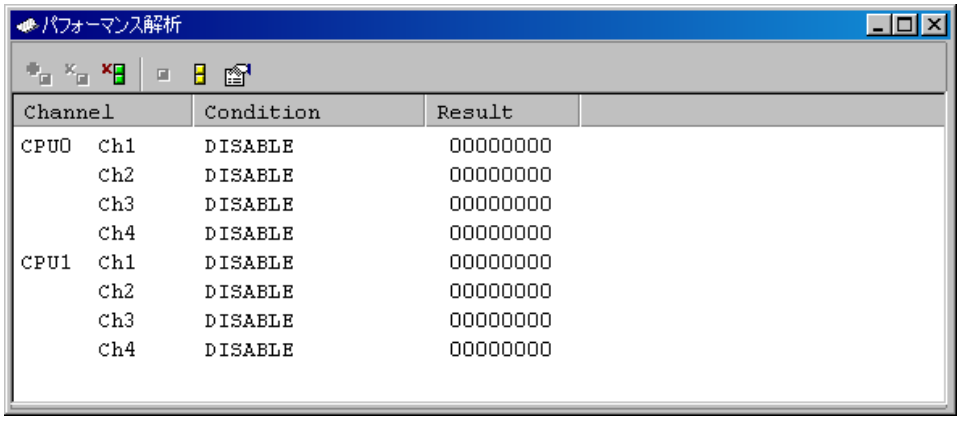


図 6.55 [パフォーマンス解析]ウィンドウ

【留意事項】

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

6.23 フラッシュメモリへのダウンロード機能

E10A-USB エミュレータは、外部フラッシュメモリ領域へダウンロードすることができます。
本機能を使用するためには、ご使用のフラッシュメモリにライトするプログラム（以後、ライトモジュールと呼びます）とフラッシュメモリを消去するプログラム（以後、消去モジュールと呼びます）、またそれらのモジュールをダウンロードし実行するRAM 領域が必要です。

【留意事項】

1. **ライトモジュールと消去モジュールは、お客様の方で用意して頂く必要があります。**
2. **ご使用のデバイスによっては、本機能は使用できません。**
   本機能をご使用になれないデバイスでは、図 6.56 [Loading flash memory]ページは表示されません。

(a)ライト/消去モジュールと E10A-USB エミュレータファームウェアとのインタフェース

ライト/消去モジュールは、E10A-USB エミュレータファームウェアから分岐します。
E10A-USB エミュレータファームウェアからライト/消去モジュールへ正常に分岐、またはライト/消去モジュールからE10A-USB エミュレータファームウェアに正常に戻ってくるようにするため、以下の条件を必ず守ってください。

- ライト/消去モジュールは、すべてアセンブル言語で記述してください。
- ライト/消去モジュール呼び出し前、呼び出し後で全ての汎用/制御レジスタ値を退避、復帰してください。
- ライト/消去モジュールは、処理終了後、必ずコール元に戻る構造としてください。
- ライト/消去モジュールは、モトローラ形式のファイルにしてください。

また、フラッシュメモリアクセスに必要な情報を正確に渡すため、以下のインタフェースで作成してください。

表 6.2 モジュールインタフェース

項番	モジュール名	引数	リターン値
1	ライトモジュール	R4(L):ライトアドレス R5(L):アクセスサイズ 0x4220=バイト 0x5720=ワード 0x4C20=ロング R6(L):ライトデータ	R0(L):終了コード 正常終了=0 異常終了=0 以外
2	消去モジュール	R4(L):アクセスサイズ 0x4220=バイト 0x5720=ワード 0x4C20=ロング	なし

【注】(L)はロングサイズであることを示します。

【留意事項】

・ ライトモジュール
ライトデータは、R6 レジスタにアクセスサイズ分設定されます。
R6 レジスタは、アクセスサイズがワードまたはバイトの場合、上位ビットには 0 が設定されます。

(b) フラッシュメモリダウンロード方法

フラッシュメモリへダウンロードするには、[基本設定]メニュー→[エミュレータ]→[システム…]から開く[Configuration]ダイアログボックスの[Loading flash memory]ページで必要な設定を行う必要があります。

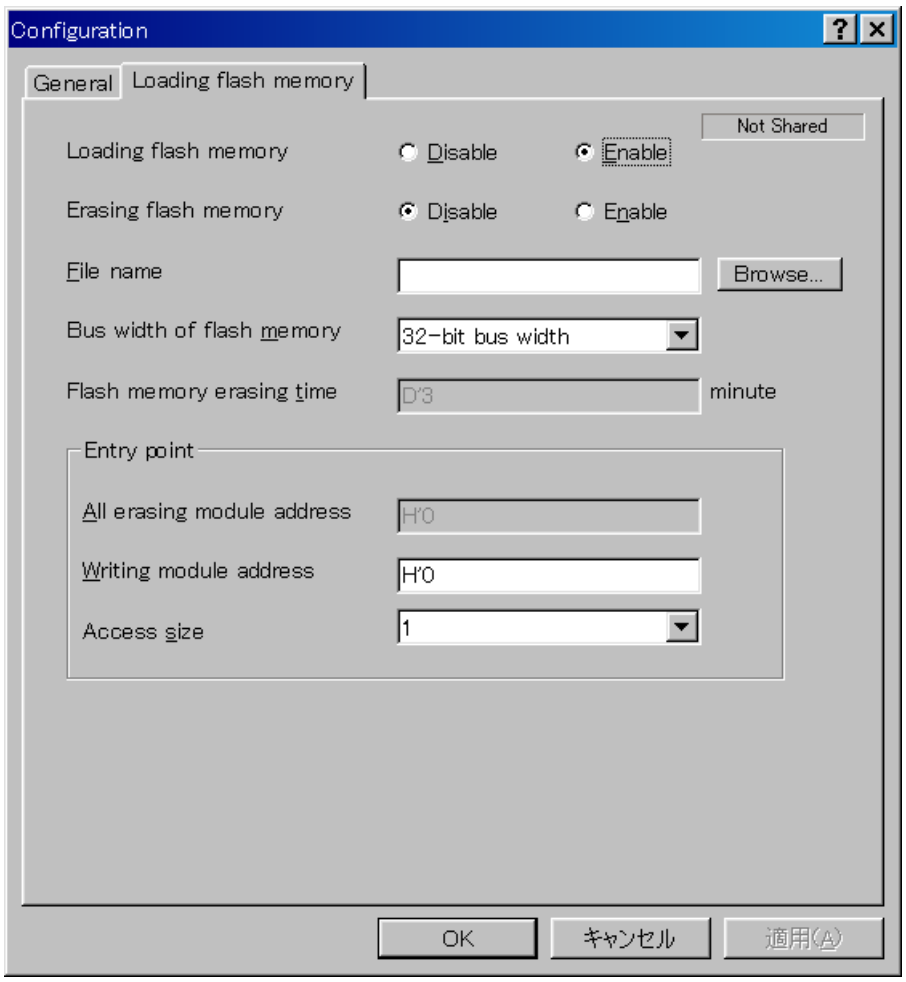


図 6.56 [Loading flash memory]ページ

[Loading flash memory]ページのオプションを以下の表で説明します。

表 6.3 [Loading flash memory]ページのオプション

オプション	説明
[Loading flash memory] ラジオボタン	フラッシュメモリへのダウンロードを行う場合、Enable にします。 Enable 時は、[File]メニューから[File load]を選択してダウンロードを行う場合、常にライトモジュールを呼び出します。 Enable：フラッシュメモリへのダウンロードを行う Disable：フラッシュメモリへのダウンロードを行わない
[Erasing flash memory] ラジオボタン	フラッシュメモリ書き込みの前に消去を行う場合、Enable にします。 Enable 時は、ライトモジュールを呼び出す前に消去モジュールを呼び出します。 Enable：フラッシュメモリの消去を行う Disable：フラッシュメモリの消去を行わない
[File name]エディットボックス	ライト/消去モジュールを含む S タイプロードモジュールのファイル名を設定します。設定したファイルは、フラッシュメモリへロードする前に RAM 領域へロードします。 ファイル名の入力の文字数は、最大 128 文字です。
[Bus width of flash memory] リストボックス	フラッシュメモリのバス幅の設定を行います。
[Flash memory erasing time]エディットボックス 【注】	フラッシュメモリ消去時の TIMEOUT 値を設定します。デフォルトは 3 分となっていますが、消去に時間がかかる場合は値を大きくしてください。 入力値の基数は 10 進数です。「H'」を付けると 16 進数になります。
[Entry point]グループボックス	ライト/消去モジュールの呼び出し先アドレス/アクセスサイズを設定します。 [All erasing module address] エディットボックス：消去モジュールの呼び出し先アドレスを入力します。 [Writing module address] エディットボックス：ライトモジュールの呼び出し先アドレスを入力します。 [Access size] ドロップダウンリストボックス：ライト/消去モジュールをロードする RAM 領域のアクセスサイズを選択します。

【注】設定できる値は、D'1～D'65535 ですが、設定値によって、TIMEOUT 時間が長くなります。したがって、使用しているフラッシュメモリの消去時間を考慮して、できるだけ最小の値を入力することをお勧めします。

(c) フラッシュメモリダウンロード機能使用時の注意事項

フラッシュメモリダウンロード時には、以下の注意事項があります。

- フラッシュメモリダウンロードをイネーブルにしている場合、フラッシュメモリ領域以外へのダウンロードはできません。
- フラッシュメモリ領域へはダウンロードのみ可能です。メモリライト、PC ブレーク等の操作は RAM 領域のみに行ってください。
- フラッシュメモリの消去をイネーブルにしている場合、消去を行っている間は[Stop]ボタンで停止できません。
- ライトモジュール、消去モジュールの各領域は、必ず MMU 無効空間としてください。

(d) フラッシュメモリダウンロード例

(株)Intel 製フラッシュメモリ（型名 : G28F640J5-150）にダウンロードする例をご紹介します。

なお、各エミュレータのインストール先フォルダ中の¥Fmtool フォルダにサンプルを提供しています。このサンプルを参考にして、お客様の仕様に合ったプログラムを作成してください。

表 6.4 ボード仕様

項目		内容
SDRAM アドレス		H'0C000000～H'0FFFFFFF
フラッシュメモリアドレス		H'00000000～H'01FFFFFF
フラッシュメモリバス幅		32 ビット
動作環境	CPU 内部周波数	167 MHz
	バス周波数	55.7 MHz
	CPU 内部モジュール周波数	27.84 MHz
	エンディアン	ビッグエンディアン

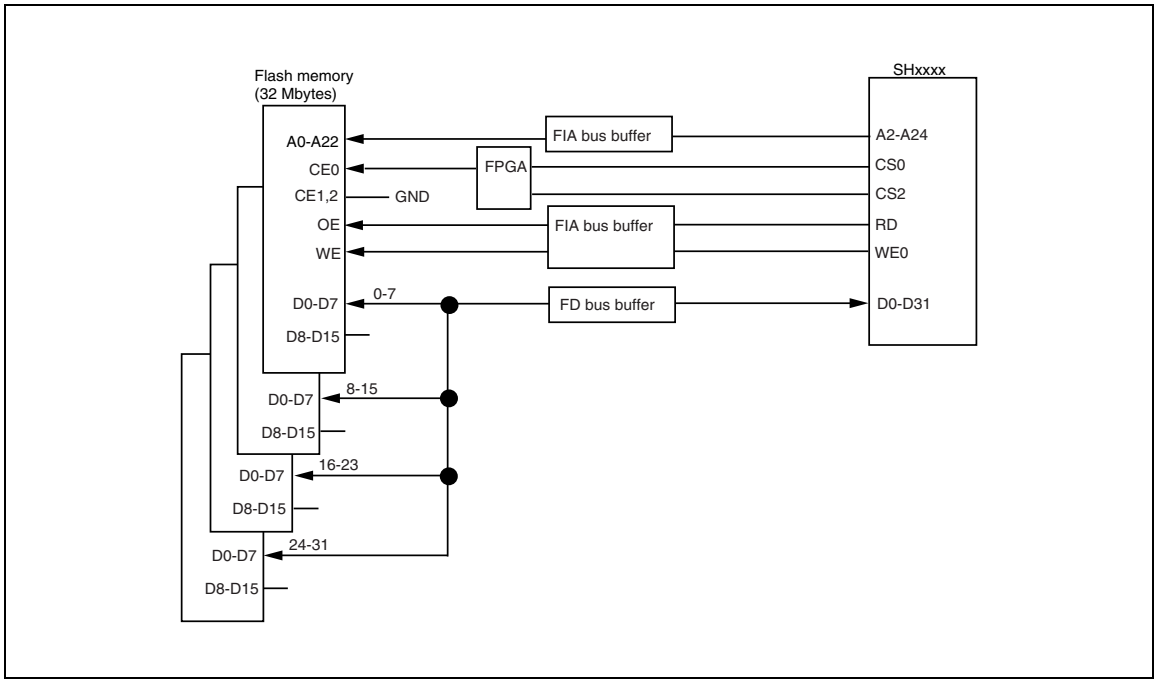


図 6.57 フラッシュメモリ結線図

表 6.5 サンプルプログラム仕様

項目	内容
使用する RAM エリア	H'0C001000～H'0C0015BF
ライトモジュール開始アドレス	H'0C001100
消去モジュール開始アドレス	H'0C001000

(i) SDRAMを使用するため、バスコントローラを設定します。

(ii) [Configuration]ウィンドウの[Loading flash memory]ページの各オプションを以下のように設定します。

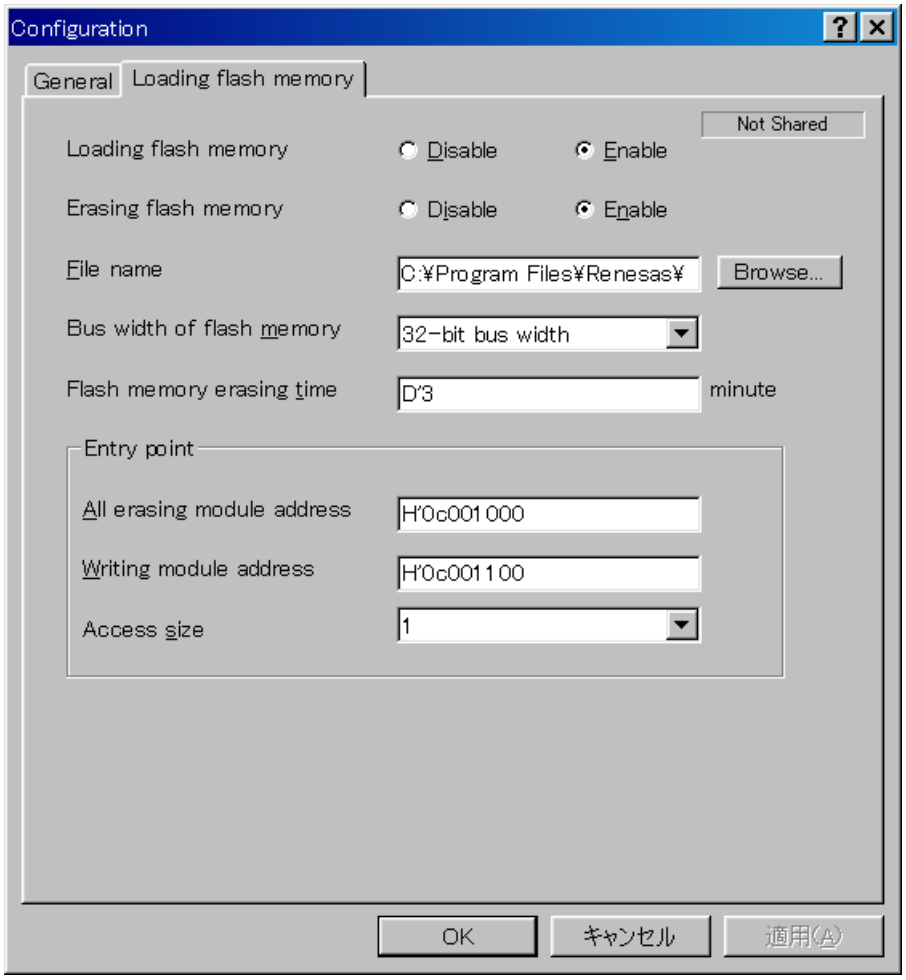


図 6.58 [Loading flash memory]ページ

【注】

1. フラッシュメモリにデータが既に書かれている場合、必ず[Erasing flash memory]を[Enable]にしてください。[Disable]の場合、ベリファイエラーが発生します。
2. [Erasing flash memory]を選択した場合、消去には約 1 分間かかります（サンプル例の場合）。

(iii) ダウンロードするオブジェクトを選択し、フラッシュメモリ領域にダウンロードを行ってください。

7.　チュートリアル[SH-4A 編]

7.1　はじめに

E10A-USB 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[Parallel]モードの動作を説明します。

特に説明のない場合は CPU0 側の High-performance Embedded Workshop の操作として説明します。

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、CPU0 側と CPU1 側の High-performance Embedded Workshop はそれぞれ 10 個のランダムデータを昇順/降順にソートします。

チュートリアルプログラムでは、以下の処理を行います。

main 関数でソートするランダムデータを生成します。

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

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

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

【留意事項】

1. Tutorial.abs は、ビッグエンディアンで動作します。リトルエンディアンで動作させる場合、再コンパイルを行ってください。
   再コンパイルを行った場合、本章で説明しているアドレスと異なることがあります。
2. 本章は、一般的な E10A-USB エミュレータの使用例です。各製品の仕様については、別冊の「SHxxxx ご使用時の補足説明」、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を、実際にロードされたアドレスの上位 16 ビットで置き換えて操作してください。
   [例] Tutorial.abs のロードされたアドレスが H’0C00xxxx である場合、マニュアルでは PC アドレスが H’0000006c となっていますが、H’0C00006c（上位ビット H’0000 -> H’0C00 に変更）として、入力してください。
4. ご使用のデバイスによっては、表示されるアドレスやデータ値が本章の説明と異なることがあります。

7.2　High-performance Embedded Workshop の起動

[スタート]メニューの[プログラム]から[Renesas]→[High-performance Embedded Workshop]→[High-performance Embedded Workshop]を選択してください。

7.3　同期デバッグの設定

(1) [ようこそ!]ダイアログボックスが表示されます。

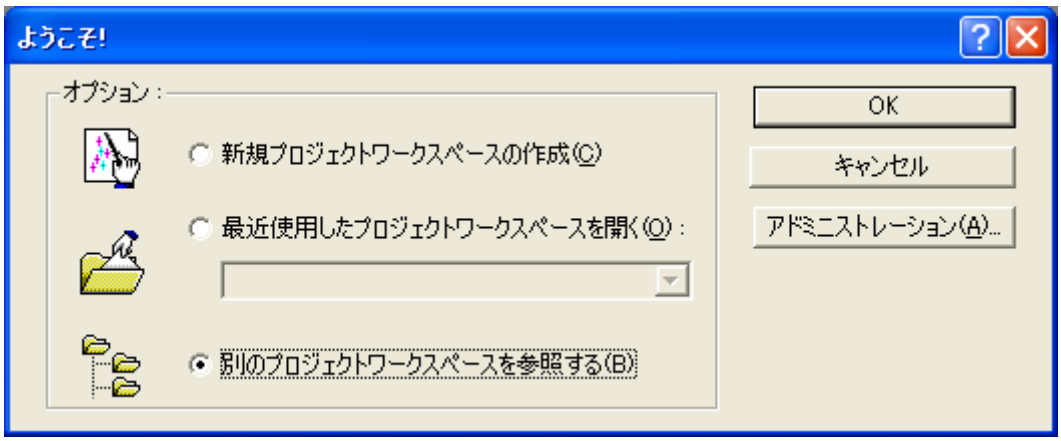


図 7.1　[ようこそ!]ダイアログボックス

ここでは、 [キャンセル]ボタンを押してください。

(2) [デバッグ]メニューの[同期デバッグ]を選択し、[同期デバッグ]ダイアログを開いてください。

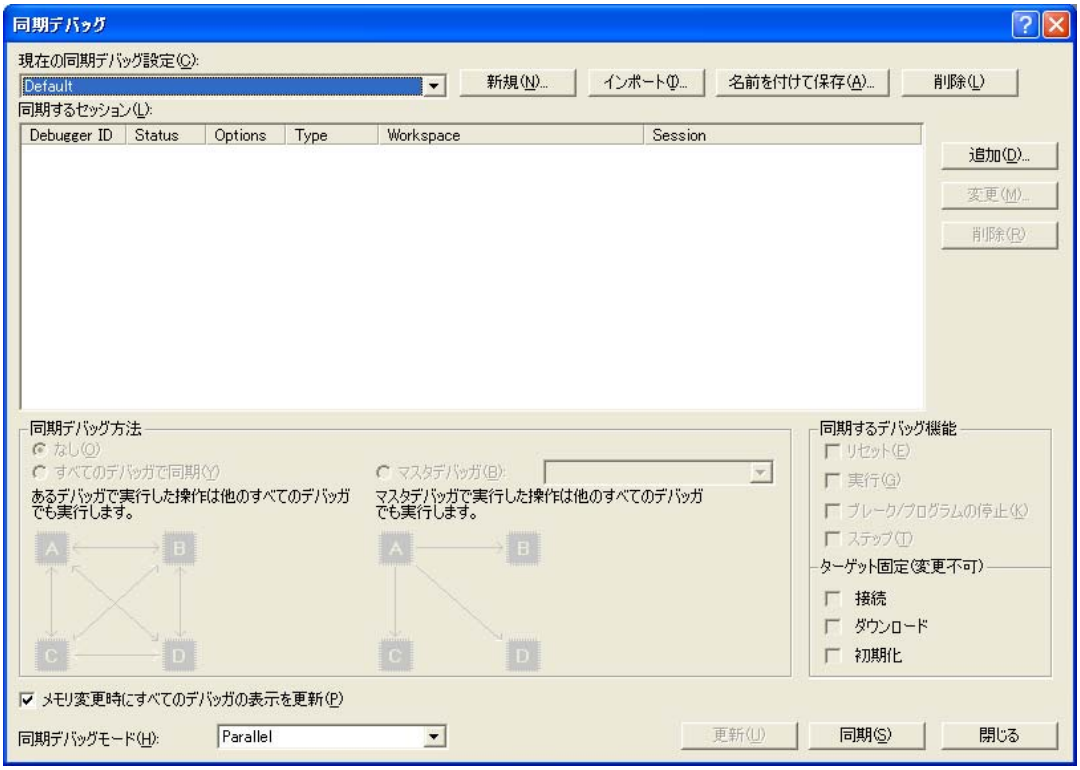


図 7.2　[同期デバッグ]　ダイアログボックス

(3) [新規]ボタンをクリックし、[同期デバッグ設定名]を入力してください。
ここでは、SH4AM_Tutorialと入力します。

(4) [追加]ボタンをクリックし、[セッションの追加]ダイアログを開いてください。
[参照]ボタンより、
〈OSがインストールされているドライブ〉
¥WorkSpace¥Tutorial¥E10A-USBM¥SH4AM¥Tutorial¥SH4AM_Internal¥SH4AM_Internal.hwsを読み込んでください。
[プロジェクト]ドロップダウンリストから[tutorial_CPU0]を選択し、[OK]ボタンをクリックし、ダイアログボックスを閉じてください。

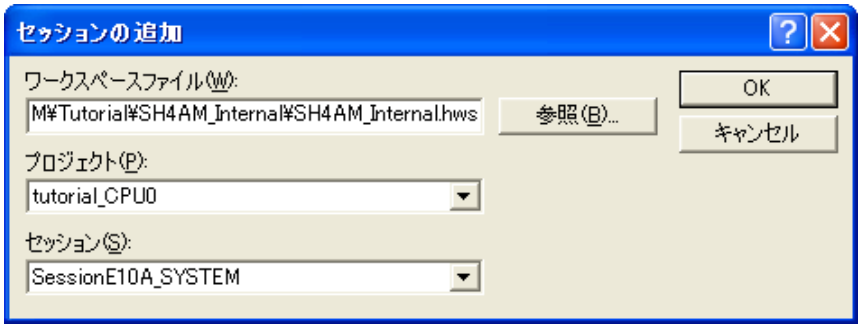


図 7.3　[セッションの追加] ダイアログボックス

【注】ワークスペースの読み込みに失敗する場合は、「3.9 システムチェック」の手順にしたがってワークスペースを開いてから読み込みを行ってください。

(5)　再度[追加]ボタンをクリックし、[セッションの追加] ダイアログを開いてください。
[ワークスペースファイル]部に先ほど読み込んだワークスペースが表示されていることを確認してください。
異なる場合は、(4)の手順でワークスペースを読み込んでください。
[プロジェクト]ドロップダウンリストから[tutorial_CPU1]を選択し、[OK]ボタンをクリックし、ダイアログボックスを閉じてください。

(6)　同期デバッグ状態の設定を行います。
ここでは、[同期デバッグ方法]を[すべてのデバッガで同期]ラジオボタンを選択、[同期するデバッグ機能]を[実行]、[ブレーク/プログラムの停止]、[ステップ]の全てのチェックボックスをチェック、[同期デバッグモード]ドロップダウンリストボックスから[Parallel]を選択してください。

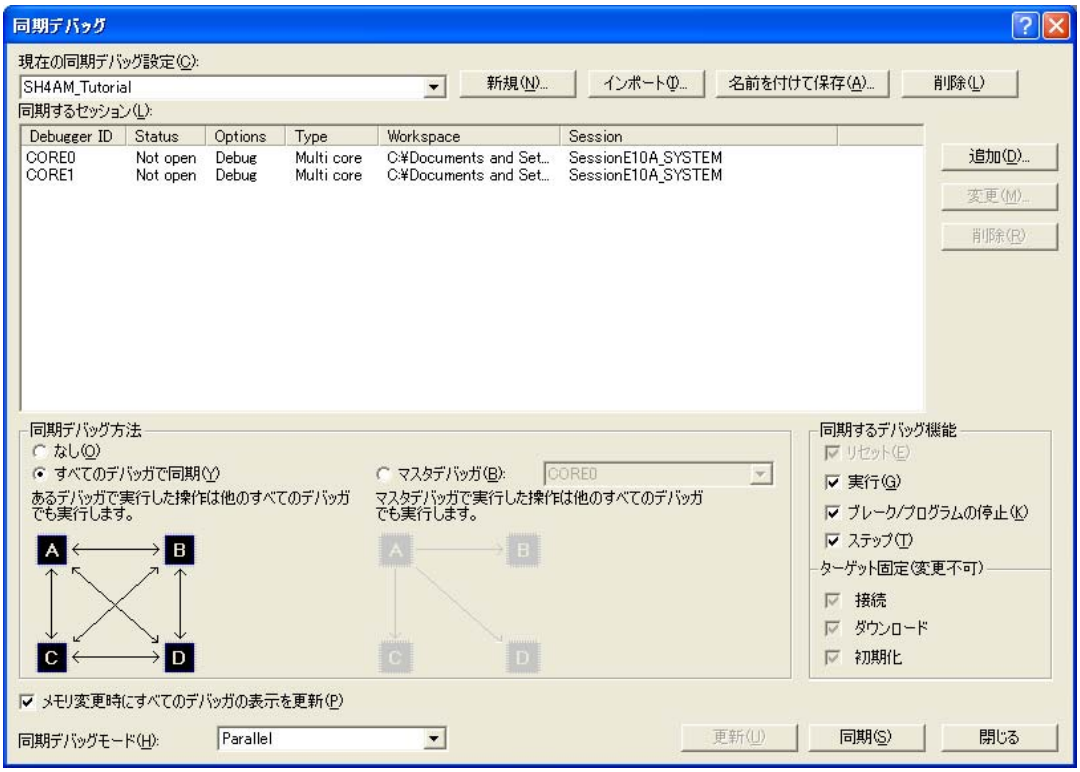


図 7.4　[同期デバッグ]　ダイアログボックス

[同期]ボタンをクリックし、「3.9 システムチェック」にしたがって High-performance Embedded Workshop を起動してください。

7.4　E10A-USB エミュレータのセットアップ

プログラムをダウンロードする前に、E10A-USB エミュレータの通信クロックをセットアップする必要があります。

- AUD clock
  AUD トレース取得時のクロックです。
  周波数が低いと、リアルタイムトレース機能使用時にデータ抜けの発生頻度が高くなります。
  周波数は、サポートデバイスの AUD clock 上限を超えないように設定してください。
  AUD トレース機能を使用するときのみ必要です。

- JTAG (H-UDI)clock（TCK）
  AUD トレース以外の通信クロックです。
  周波数が低いと、ダウンロードが遅くなります。
  周波数は、サポートデバイスの TCK 保証範囲の上限を超えないように設定してください。
  両クロックとも、制限事項は、別冊の SHxxxx ご使用時の補足説明の「2.2.4　JTAG(H-UDI) クロック（TCK）、AUD クロック（AUDCK）使用時の注意事項」をご参照ください。

以下に、通信クロックを設定する方法について説明します。

7.5　[Configuration]ダイアログボックスの設定

通信クロックを設定するために、CPU0 用の High-performance Embedded Workshop にて[基本設定]メニューから[エミュレータ]を選択し、さらに[システム…]を選択してください。[Configuration]ダイアログボックスが表示されます。

[Common Setting]ページを開いてください。

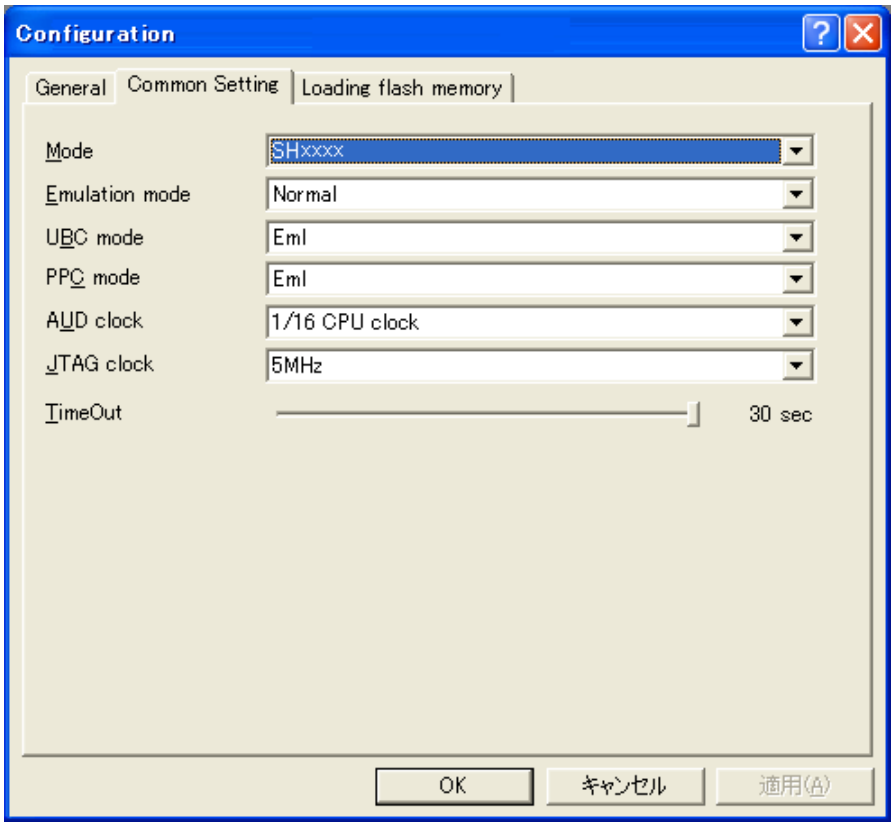


図 7.5　[Configuration]ダイアログボックス

[AUD clock]コンボボックスと、[JTAG clock]コンボボックスに適切な値を設定してください。デフォルトでも動作します。

【留意事項】

本ウィンドウで設定できる内容は、製品ごとに異なります。各製品の設定内容については、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、コンフィグレーションを設定してください。

7.6　ダウンロード先メモリの動作チェック

ダウンロードを行うメモリが正常に動作することをチェックします。

ダウンロード先のメモリが SDRAM/DRAM 等の場合、ダウンロードする前に CPU のバスコントローラの設定をする必要があります。使用するメモリに従った設定を前もって適切に行ってください。なお、バスコントローラは、[IO]ウィンドウから設定することができます。

バスコントローラの設定などのメモリ設定が完了したら、CPU0 用の High-performance Embedded Workshop にて[メモリ]ウィンドウでメモリ内容を表示、編集し、メモリが正常に動作することを確認します。

【留意事項】

メモリ動作チェックは上記だけでは不完全な場合があります。メモリチェック用プログラムを作成し、チェックすることをお勧めします。

[表示]メニューの[CPU]サブメニューから[メモリ…]を選択し、[表示開始アドレス]エディットボックスに“H’00000000”を入力し、[スクロール開始アドレス]エディットボックスに“H’00000000”を、[スクロール終了アドレス]エディットボックスに“H’FFFFFFFF”を入力してください。

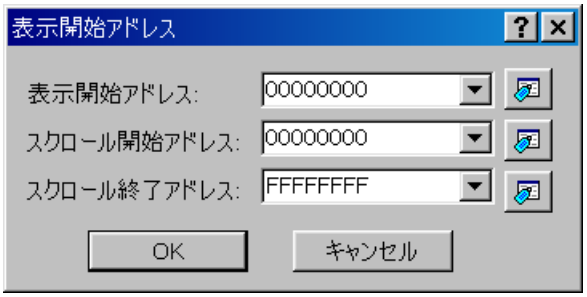


図 7.6　[表示開始アドレス]ダイアログボックス

[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。

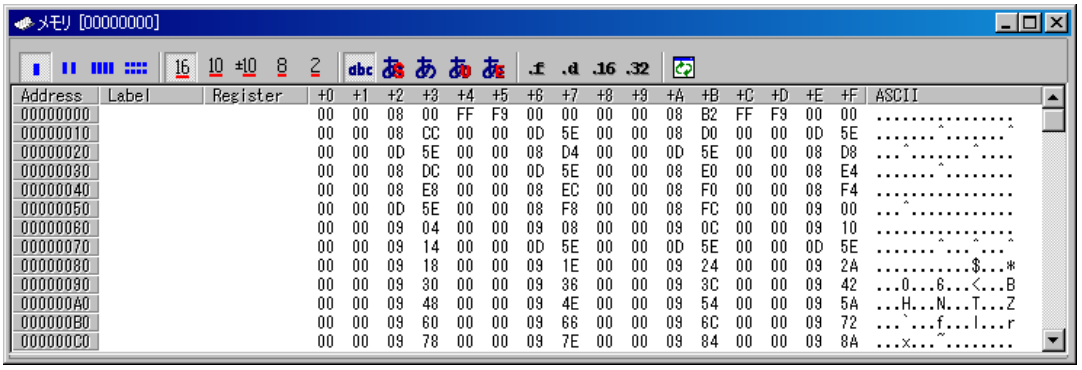


図 7.7　[メモリ]ウィンドウ

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

7.7　チュートリアルプログラムのダウンロード

7.7.1　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。
CPU0 用 High-performance Embedded Workshop および CPU1 用 High-performance Embedded Workshop でそれぞれソースレベルデバッグを行うために、CPU0/ CPU1 でそれぞれ別にデバッグ情報ファイルをロードしてください。

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

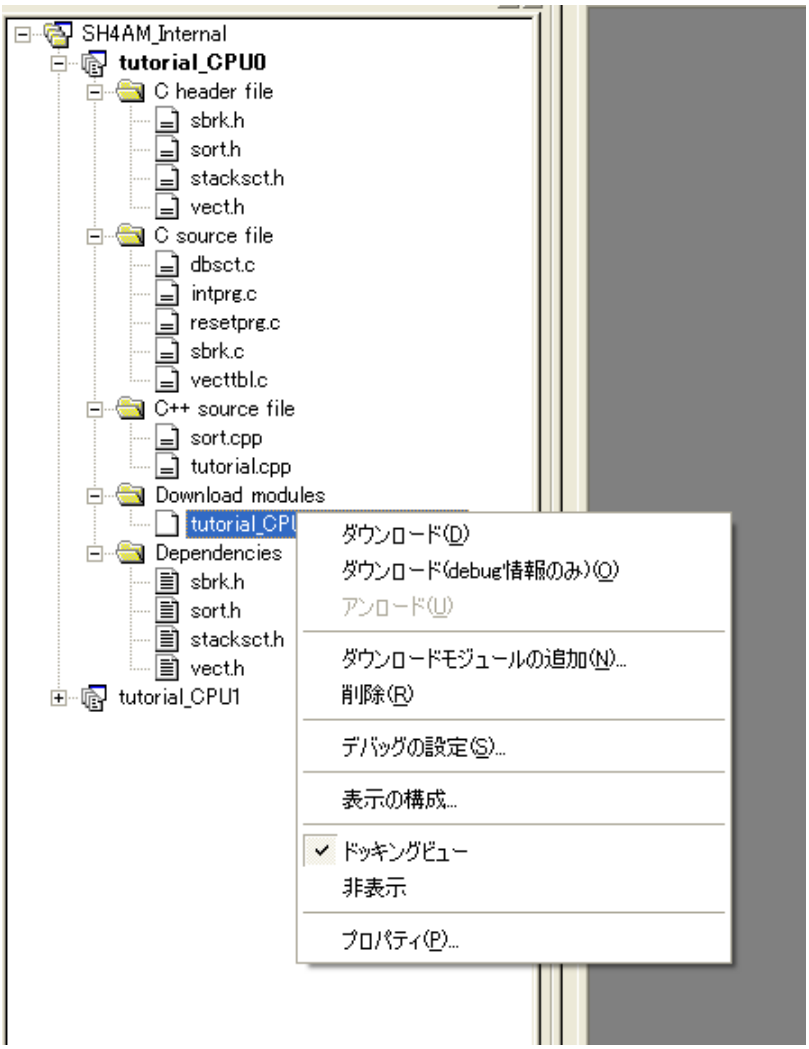


図 7.8　チュートリアルプログラムのダウンロード

7.7.2　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。

CPU0 用の High-performance Embedded Workshop にて[C++ source file]の[tutorial.cpp]をダブルクリックします。

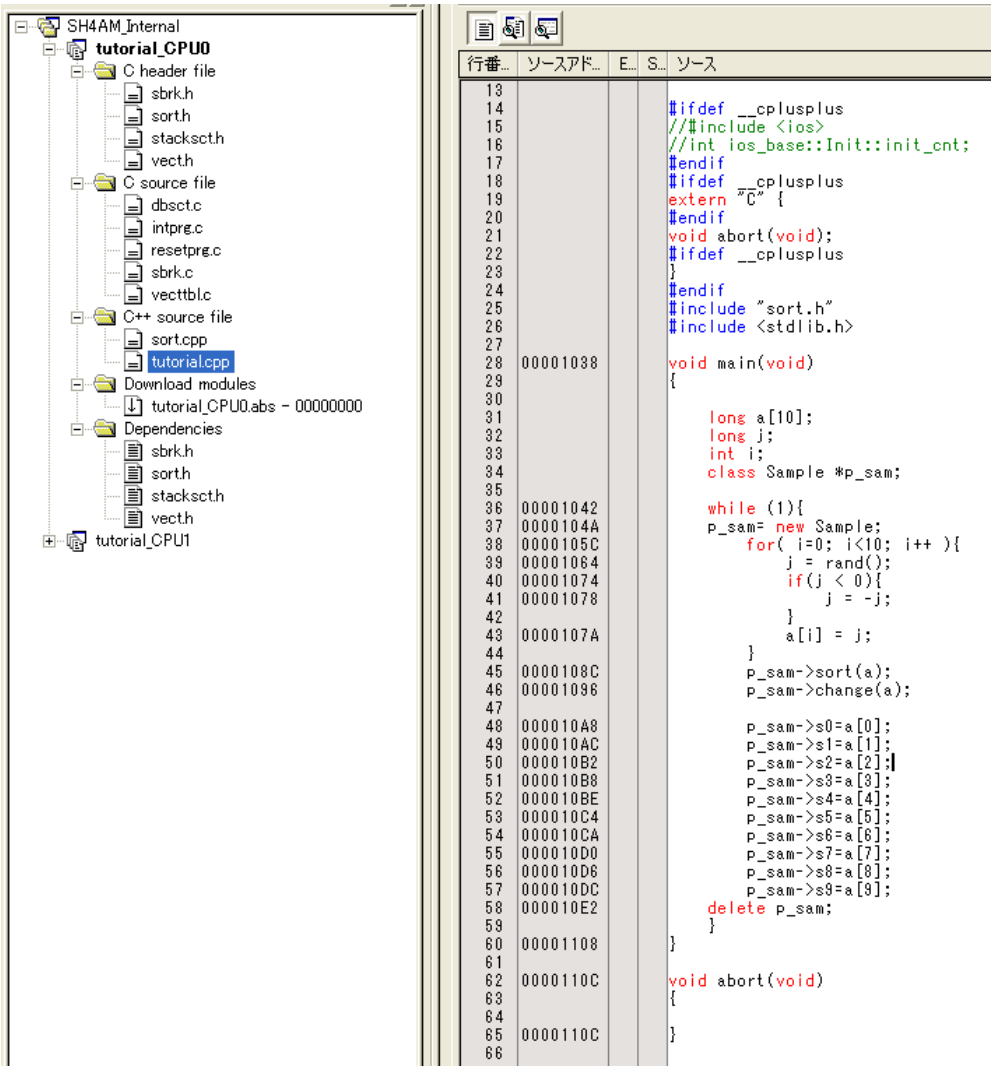


図 7.9　[エディタ]ウィンドウ（ソースプログラムの表示）

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

[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。

7.8　PC ブレークポイントの設定

簡単なデバッグ機能の 1 つに PC ブレークポイントがあります。
[エディタ]ウィンドウにおいて、PC ブレークポイントを簡単に設定できます。
例えば、sort 関数のコール箇所に PC ブレークポイントを設定します。

CPU0 用の High-performance Embedded Workshop にて sort 関数コールを含む行の[S/W ブレークポイント]カラムをダブルクリックしてください。

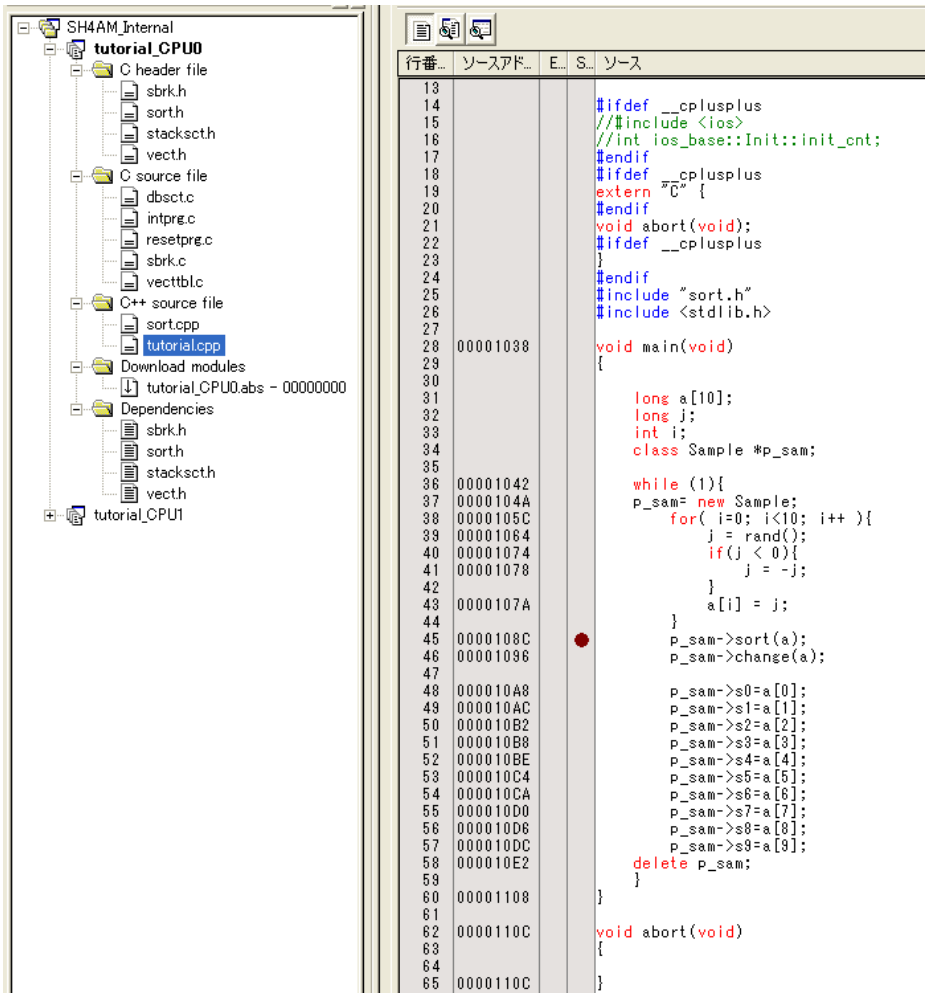


図 7.10　[エディタ]ウィンドウ（PC ブレークポイントの設定）

sort 関数を含む行に”● ”と表示されます。この表示により PC ブレークポイントが設定されたことを示しています。

【留意事項】　PC ブレークポイントは、ROM 領域には設定できません。

7.9　レジスタ内容の変更

プログラムを実行する前に、プログラムカウンタおよびスタックポインタの値を設定してください。

CPU0 用および CPU1 用の High-performance Embedded Workshop にて[表示]メニューの[CPU]サブメニューから[レジスタ]を選択してください。
[レジスタ]ウィンドウが表示されます。

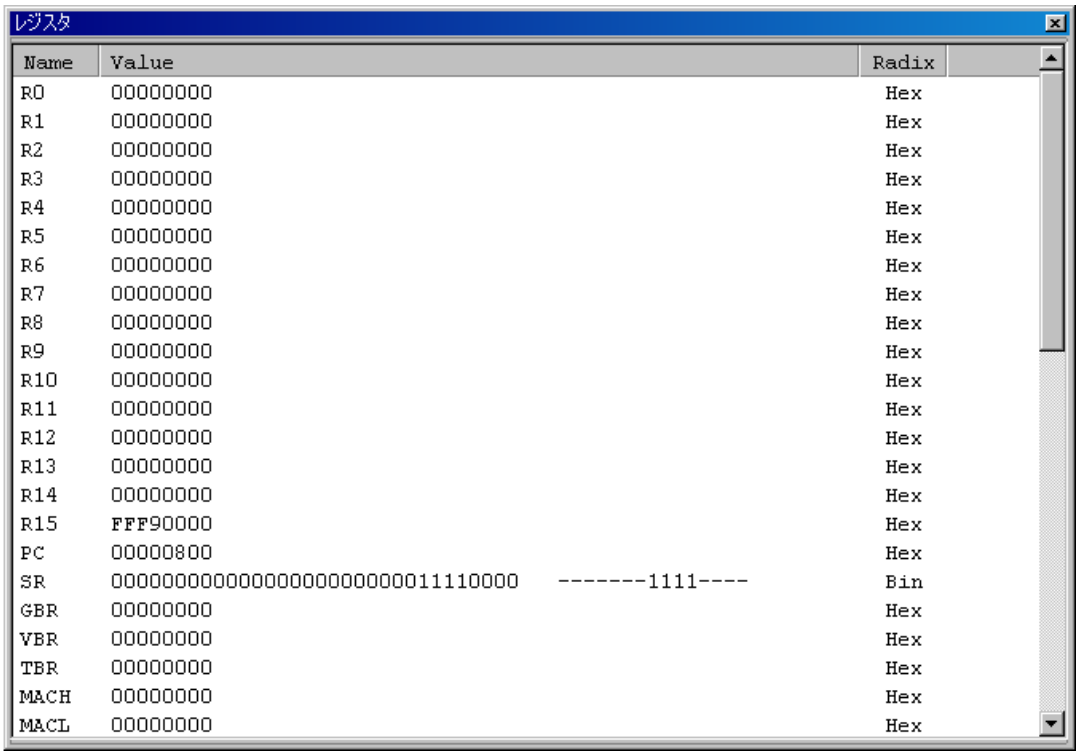


図 7.11　[レジスタ]ウィンドウ

プログラムカウンタ（PC）を変更する場合には、[レジスタ]ウィンドウで[PC]の数値エリアをマウスでダブルクリックすると、以下のダイアログボックスが表示され、値の変更が可能です。本チュートリアルプログラムでは、CPU0 側を H´00000800、CPU1 側を H’00100800 と設定し、[OK]ボタンをクリックしてください。

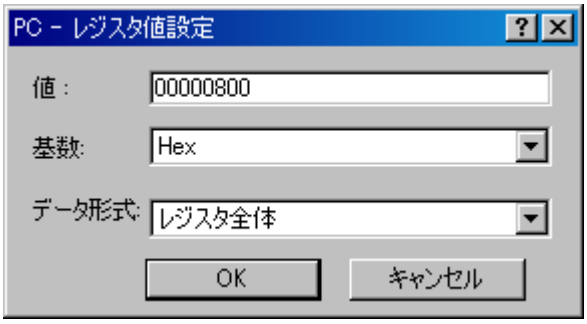


図 7.12　[レジスタ]ダイアログボックス（PC）

同じようにして、スタックポインタ（SP）を変更します。本チュートリアルプログラムでは、CPU0 側を H’00010000、CPU1 側を H’00110000 と設定してください。

フラッシュメモリ内蔵デバイスをご使用の場合、スタックポインタ（SP）は内蔵 RAM の最終アドレスを指定してください。

デバイスごとに内蔵 RAM の領域は異なります。ご使用のデバイスのハードウェアマニュアルを参照してください。

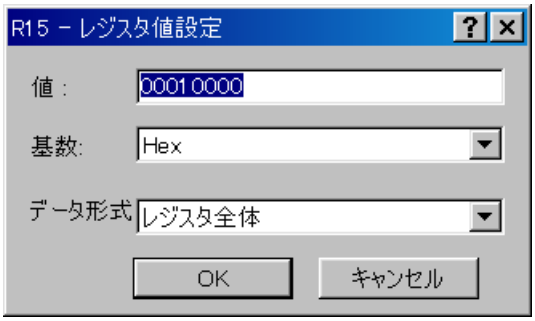


図 7.13　[レジスタ]ダイアログボックス（R15）

7.10　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。
[同期デバッグ]ダイアログボックスで同期実行が有効([すべてのデバッガで同期]、[同期するデバッグ機能]の[実行]チェックボックス)が有効になっているので、プログラムを実行する場合は、いずれかの CPU 用の High-performance Embedded Workshop にて[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。ここでは、CPU0 用の High-performance Embedded Workshop で実行操作を行います。

図 7.14　[実行]ボタン

実行を開始すると、ステータスバーに”** RUNNING”と表示し、その後、CPU ステイタス取得機能をサポートしている製品では、実行 PC アドレスと SR レジスタ値の表示となります。

プログラムは CPU0 用の High-performance Embedded Workshop はブレークポイントを設定したところまで実行されます。CPU1 用の High-performance Embedded Workshop は同期ブレーク([すべてのデバッガで同期]、[同期するデバッグ機能]の[ブレーク/プログラムの停止]チェックボックス)のため CPU0 のブレークにしたがい CPU1 もブレークします。

それぞれの High-performance Embedded Workshop ではプログラムが停止した位置を示すために[S/W ブレークポイント]カラム中に矢印が表示されます。

また、CPU0 用 High-performance Embedded Workshop では[BREAK POINT] 、CPU1 用 High-performance Embedded Workshop では[SYNCRONIZATION BREAK CPU#0]メッセージがステータスバーに表示されます。

【留意事項】

1. ブレーク後にソースファイルを表示する際に、ソースファイルパスを問い合わせる場合があります。ソースファイルの場所は以下です。
   〈OS がインストールされているドライブ〉
   ¥WorkSpace¥ Tutorial¥E10A-USBM¥SH4AM¥SH4AM¥Tutorial¥CPU0¥source
2. 正常に実行できない場合、[デバッグ]メニューから[CPU のリセット]を選択し、一度リセットを発行してから、図 7.8 よりやり直してください。

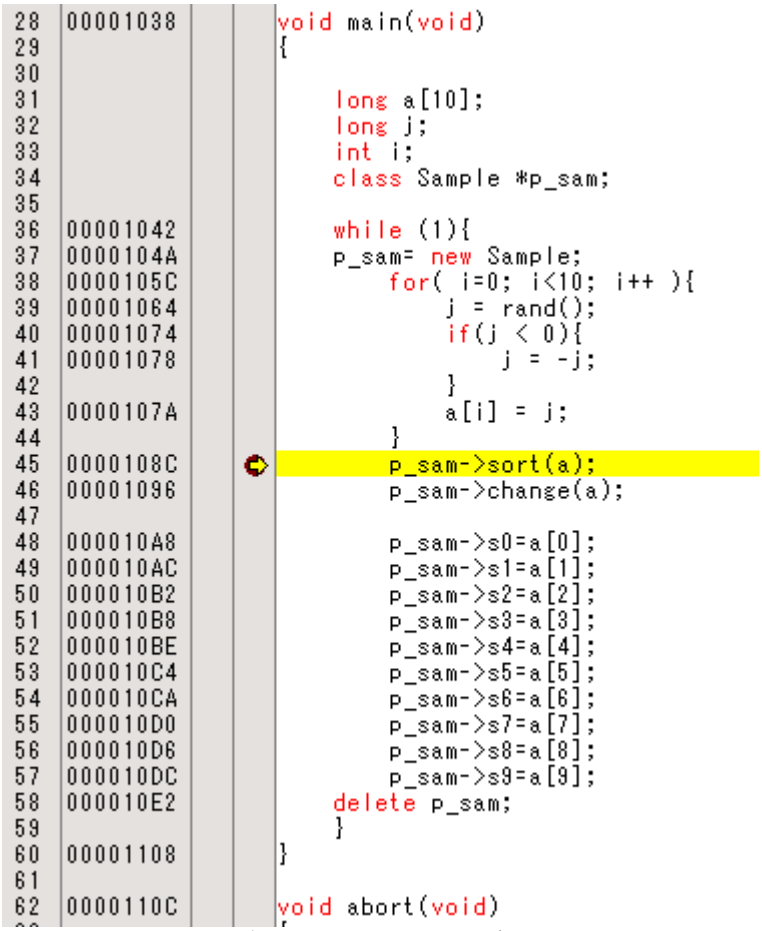


図 7.15　[エディタ]ウィンドウ（ブレーク状態）

[ステイタス]ウィンドウで最後に発生したブレークの要因が確認できます。

[表示]メニューの[CPU]サブメニューから[ステイタス]を選択してください。
[ステイタス]ウィンドウが表示されますので、[Platform]シートを開いて Cause of last break の Status を確認してください。

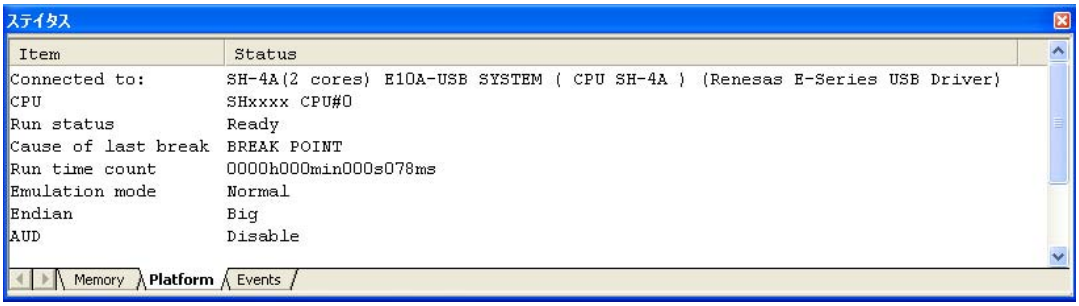


図 7.16　[ステイタス]ウィンドウ

【留意事項】

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

7.11　ブレークポイントの確認

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

CPU0 用の High-performance Embedded Workshop にて[表示]メニューの[コード]サブメニューから[イベントポイント]を選択してください。[イベントポイント]ウィンドウが表示されます。[Breakpoint]シートを開きます。

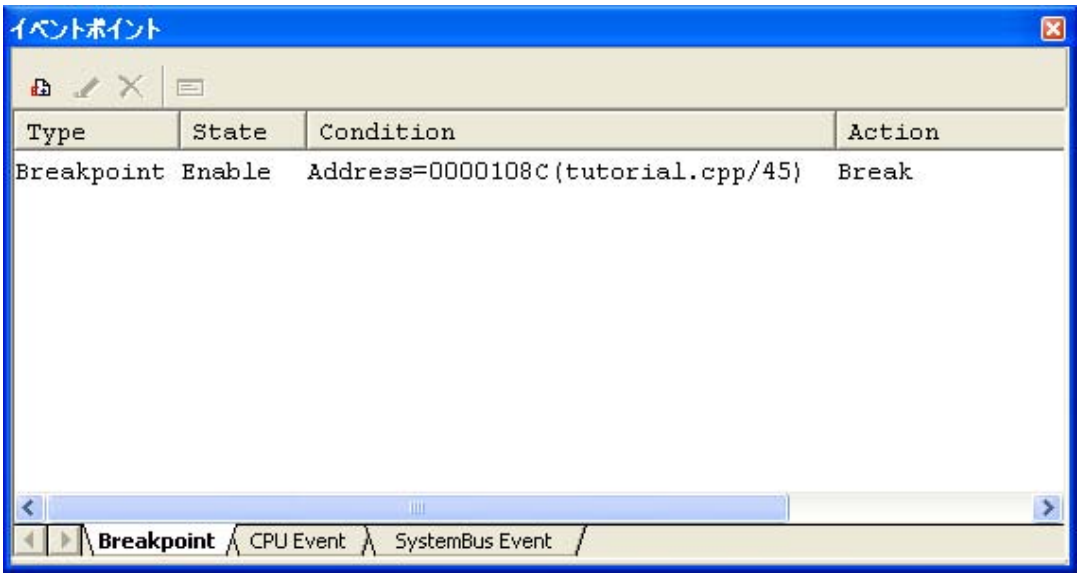


図 7.17　[イベントポイント]ウィンドウ

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。

7.12　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。

CPU1 用の High-performance Embedded Workshop にて[表示]メニューの[シンボル]サブメニューから[ラベル]を選択してください。[ラベル]ウィンドウが表示され、モジュール内のシンボル情報が参照できます。

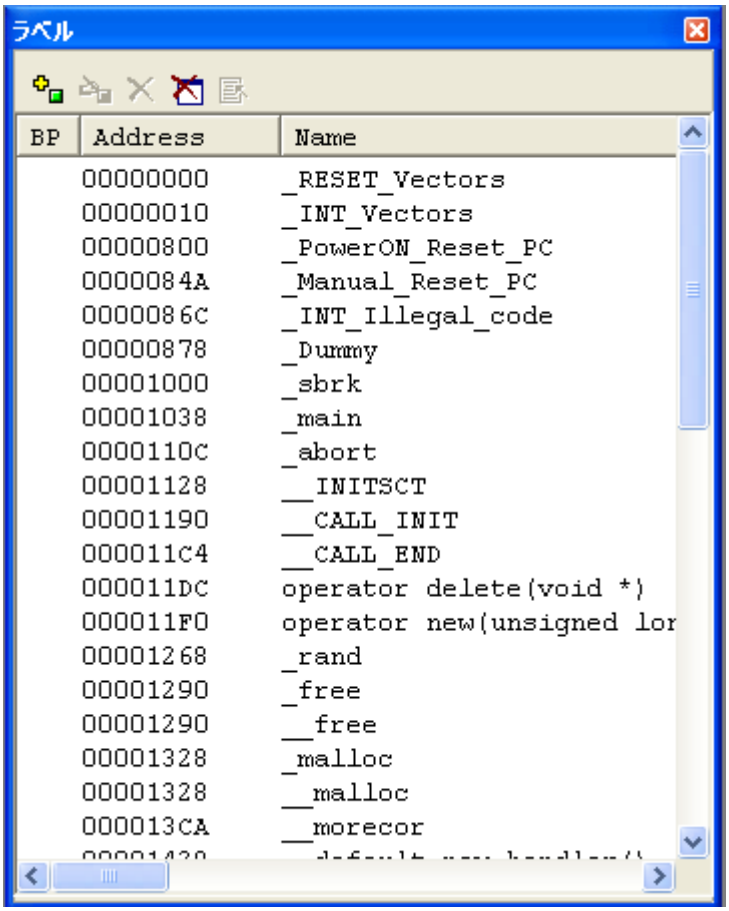


図 7.18　[ラベル]ウィンドウ

7.13　メモリ内容の確認

Label 名を指定することによって、Label が登録されているメモリの内容を[メモリ]ウィンドウで確認することができます。

例えば、以下のように、ワードサイズで_main に対応するメモリ内容を確認します。

CPU0 用の High-performance Embedded Workshop にて[表示]メニューの[CPU]サブメニューから[メモリ]を選択し、[表示開始アドレス]エディットボックスに” _main”を入力し、
[スクロール開始アドレス]エディットボックスに”H’00000000”を、
[スクロール終了アドレス]エディットボックスに”H’FFFFFFFF”を入力してください。

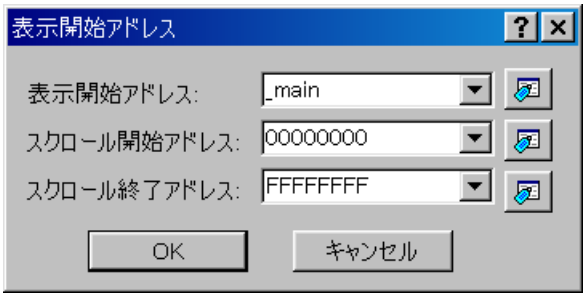


図 7.19　[表示開始アドレス]ダイアログボックス

[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。

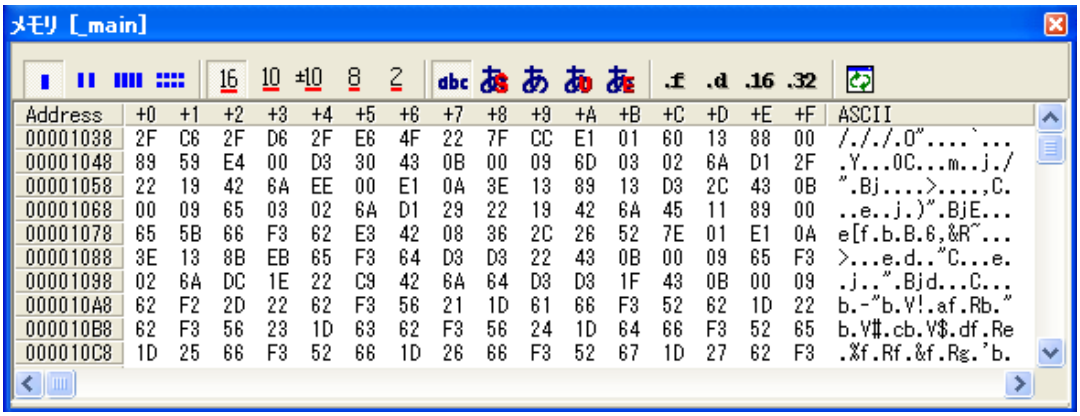


図 7.20　[メモリ]ウィンドウ

7.14　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。例えば、以下の手順で、プログラムのはじめに宣言した long 型の配列 a を見ることができます。

CPU0 用の High-performance Embedded Workshop の[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。

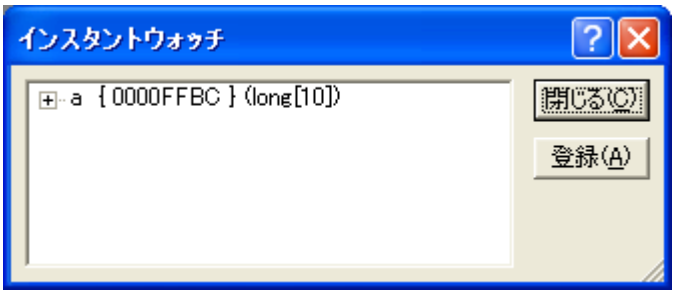


図 7.21　[インスタントウォッチ]ダイアログボックス

[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。

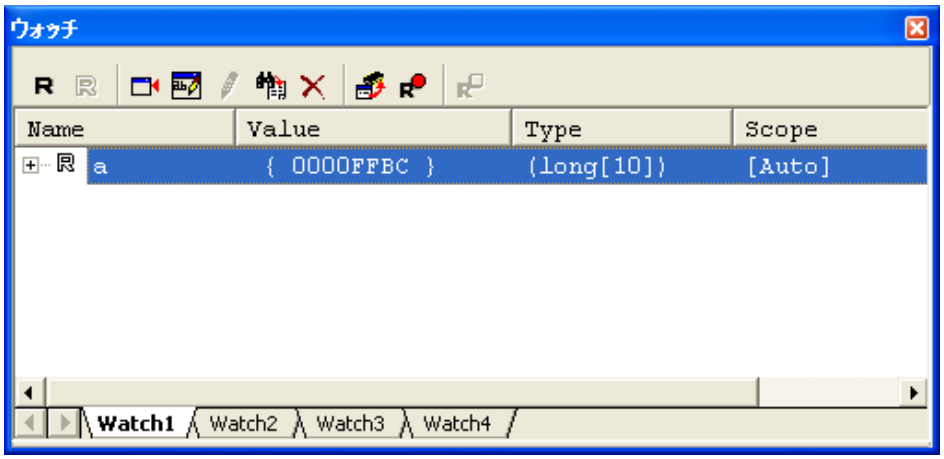


図 7.22　[ウォッチ]ウィンドウ（配列の表示）

また、変数名などを指定して、[ウォッチ]ウィンドウに変数など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ので、p_sam を入力してください。

図 7.23　[シンボル登録]ダイアログボックス

[OK]ボタンをクリックします。

[ウォッチ]ウィンドウに、インスタンス p_sam が表示されます。

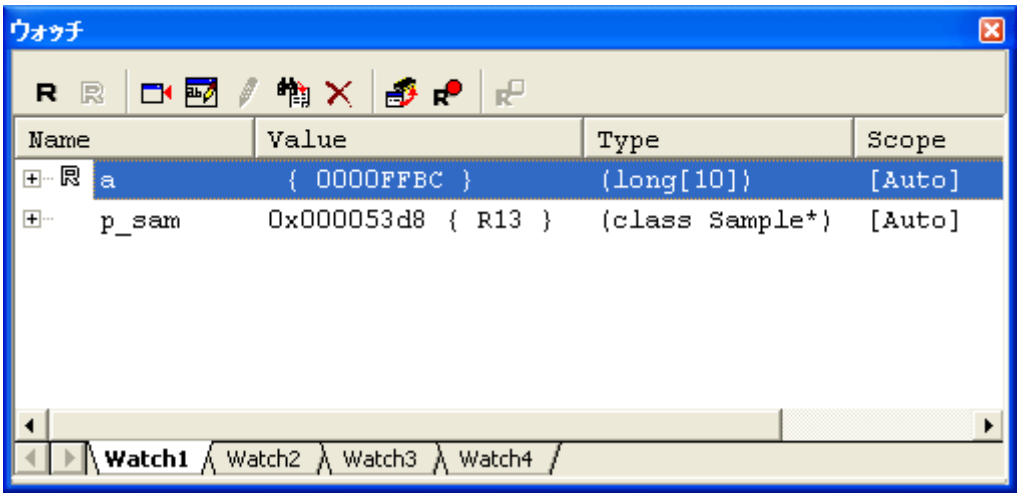


図 7.24　[ウォッチ]ウィンドウ（変数などの表示）

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

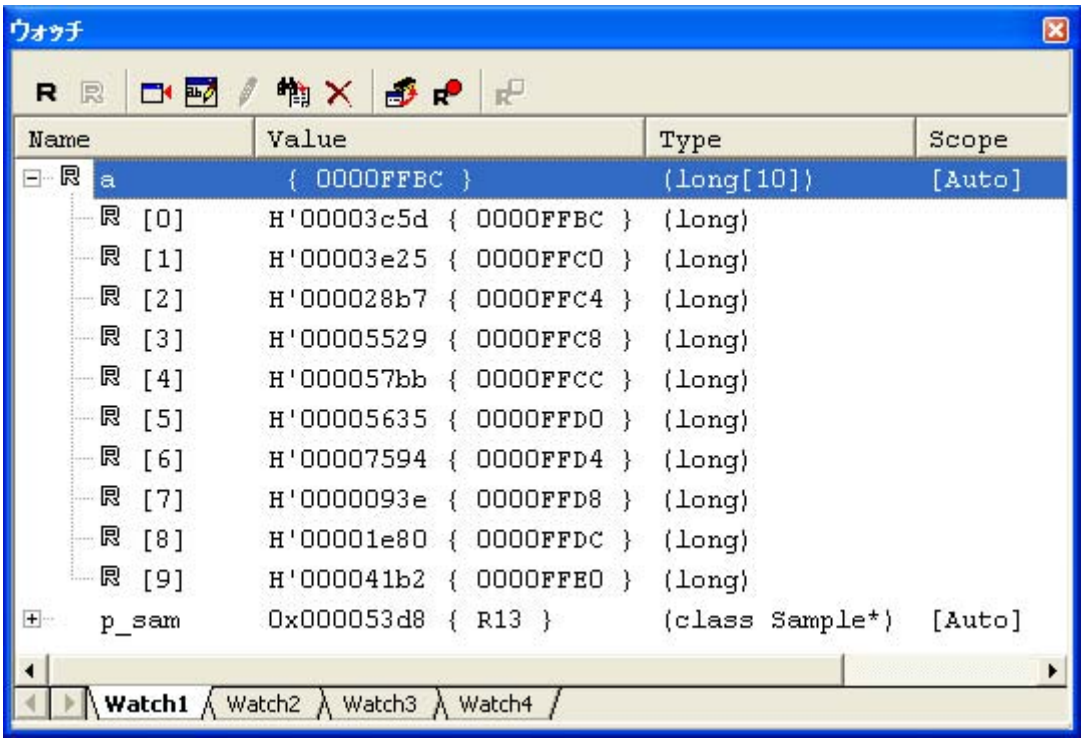


図 7.25　[ウォッチ]ウィンドウ（配列要素の表示）

7.15　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、main 関数のローカル変数を調べます。
この関数は、4 つのローカル変数 a, j, i, p_sam を宣言します。

CPU0 用の High-performance Embedded Workshop の[表示]メニューの[シンボル]サブメニューから[ローカル]を選択してください。[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。

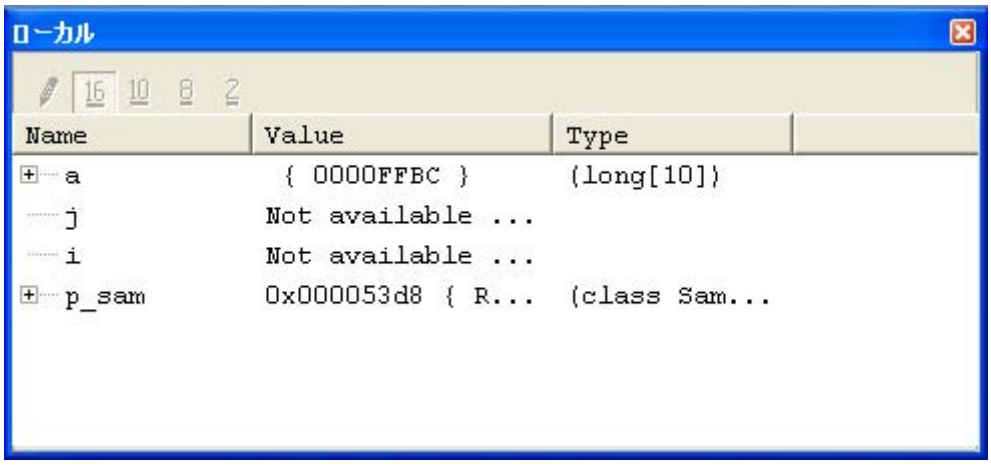


図 7.26　[ローカル]ウィンドウ

[ローカル]ウィンドウの配列 a の左側にある”+”マークをクリックし、配列 a の構成要素を表示させてください。
sort 関数実行前と実行後の配列 a の要素を参照すると、ランダムデータが降順にソートされていることがわかります。

7.16　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 7.1　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。

7.16.1　ステップインの実行

ステップイン機能はコール関数の中に入り、コール関数の先頭のステートメントで停止します。

sort 関数の中に入るためには、CPU0 用の High-performance Embedded Workshop の[デバッグ]メニューから[ステップイン]を選択するか、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。[同期デバッグ]ダイアログボックスで同期ステップが有効（[すべてのデバッガで同期]、[同期するデバッガ機能]の[ステップ]チェックボックス）が有効になっているので、この操作により同期ステップインが行われます。

図 7.27　[ステップイン]ボタン

図 7.28 [エディタ]ウィンドウ（ステップイン）

CPU0 用の High-performance Embedded Workshop の[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。

7.16.2　ステップアウトの実行

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

sort 関数の中から抜け出すために、CPU0 用の High-performance Embedded Workshop の[デバッグ]メニューから[ステップアウト]を選択するか、またはツールバーの[ステップアウト]ボタンをクリックしてください。

【留意事項】

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

図 7.29　[ステップアウト]ボタン

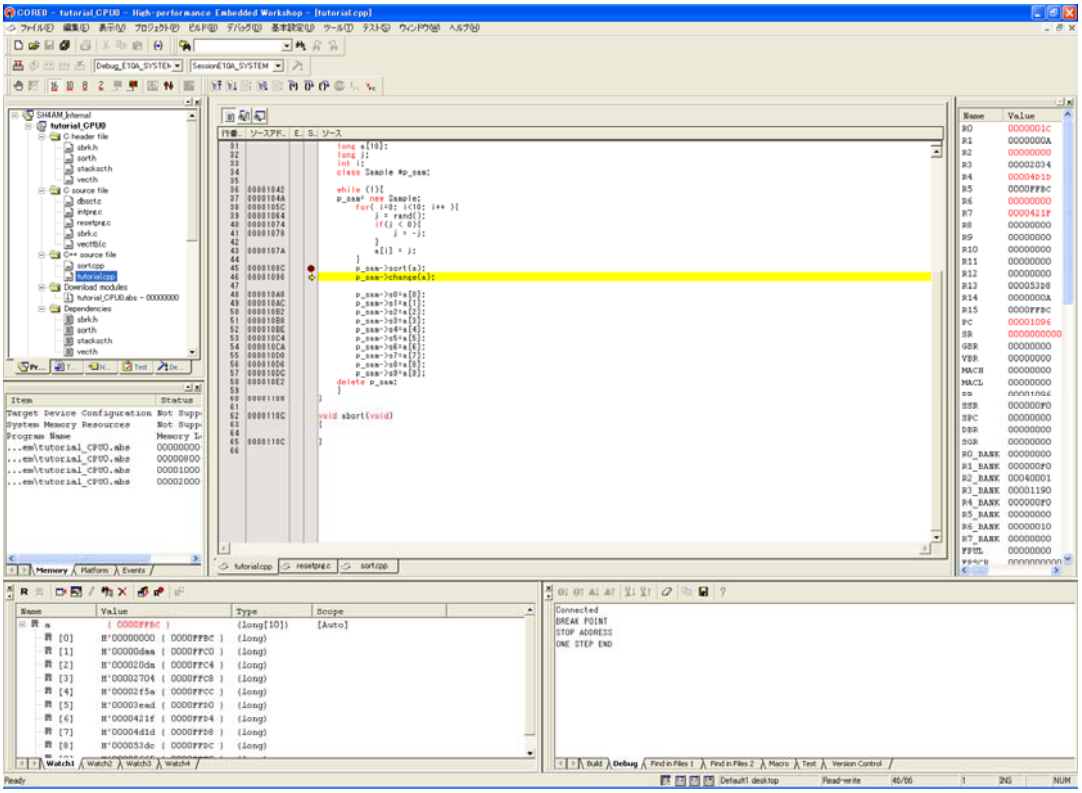


図 7.30　[High-performance Embedded Workshop]ウィンドウ（ステップアウト）

[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。

CPU1 用の High-performance Embedded Workshop の同期開始時のソース位置によっては、CPU1 側のステップアウトが終了しない場合があります。

この場合は、ツールバー上の[STOP]ボタンを選択して終了させてください。

7.16.3　　ステップオーバの実行

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

「7.16.1　ステップインの実行」の手順を実行し change 関数に移動してください。
次に、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、CPU0 用の High-performance Embedded Workshop の[デバッグ]メニューから[ステップオーバ]を選択するか、またはツールバーの[ステップオーバ]ボタンをクリックしてください。

図 7.31　[ステップオーバ]ボタン

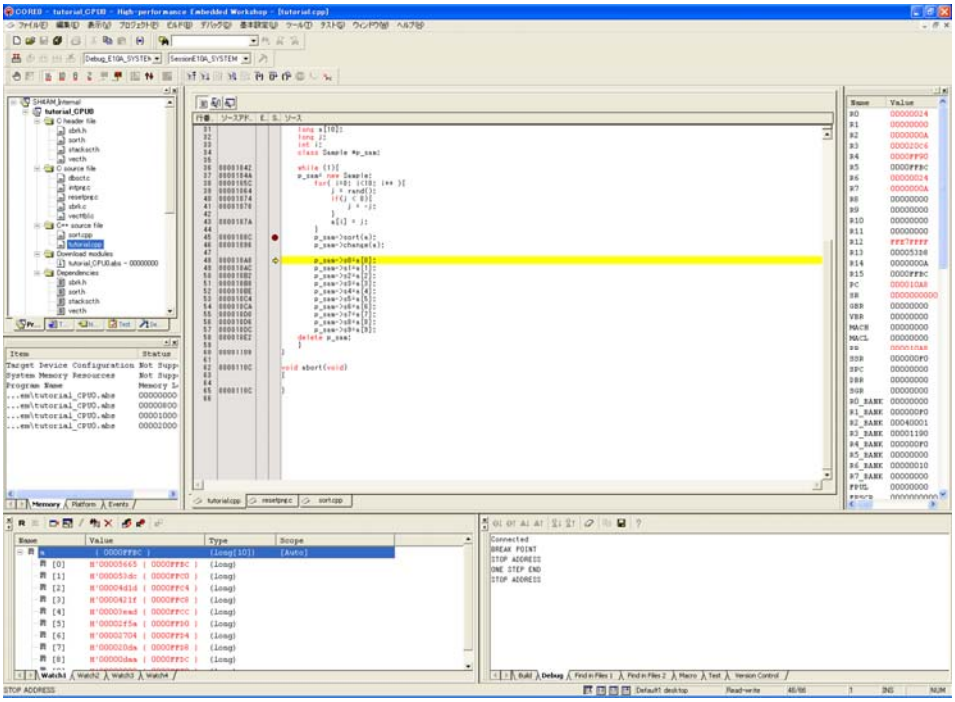


図 7.32　[High-performance Embedded Workshop]ウィンドウ（ステップオーバ）

7.17　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。

ブレークを全て解除してください。

main 関数の残り部分を実行するために、CPU0 用の High-performance Embedded Workshop の[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 7.33　[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、CPU0 用の High-performance Embedded Workshop の[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[STOP]ボタンを選択してください。

CPU0 用の High-performance Embedded Workshop は同期ブレーク ([すべてのデバッガで同期]、[同期するデバッグ機能]の[ブレーク/プログラムの停止]チェックボックス)が有効なため CPU0 のブレークにしたがい CPU1 もブレークします。

図 7.34　[STOP]ボタン

7.18　ブレーク機能

E10A-USB エミュレータは、PC ブレーク機能とハードウェアブレーク機能を持っています。
High-performance Embedded Workshop では、PC ブレークポイントの設定を[イベントポイント]ウィンドウの[Breakpoint]シートで、また、ハードウェアブレーク条件の設定を[Event condition]シートでそれぞれ行うことができます。

以下にブレーク機能の概要と設定方法について説明します。

7.18.1　PC ブレーク機能

E10A-USB エミュレータは、255 ポイントまで PC ブレークを設定することができます。
本章では、「7.8 PC ブレークポイントの設定」でご紹介した以外の設定方法を説明します。

CPU0 用の High-performance Embedded Workshop の[表示]メニューの[コード]サブメニューから[イベントポイント]を選択してください。[イベントポイント]ウィンドウが表示されます。
[Breakpoint]シートを開きます。

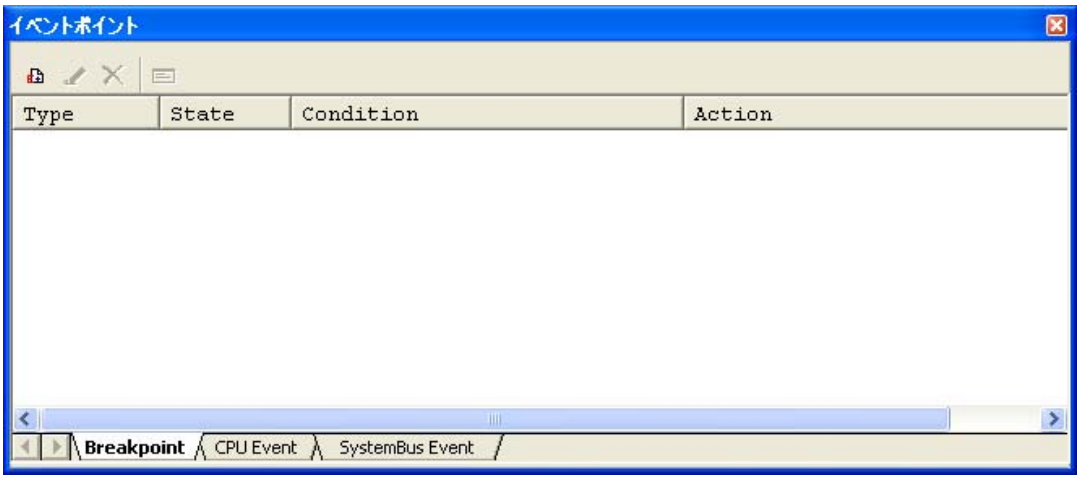


図 7.35　[イベントポイント]ウィンドウ（PC ブレーク設定前）

マウスの右ボタンで[イベントポイント]ウィンドウをクリックし、ポップアップメニューから[追加…]を選択してください。

[Address]エディットボックスにアドレス H’000010A8 を入力してください。

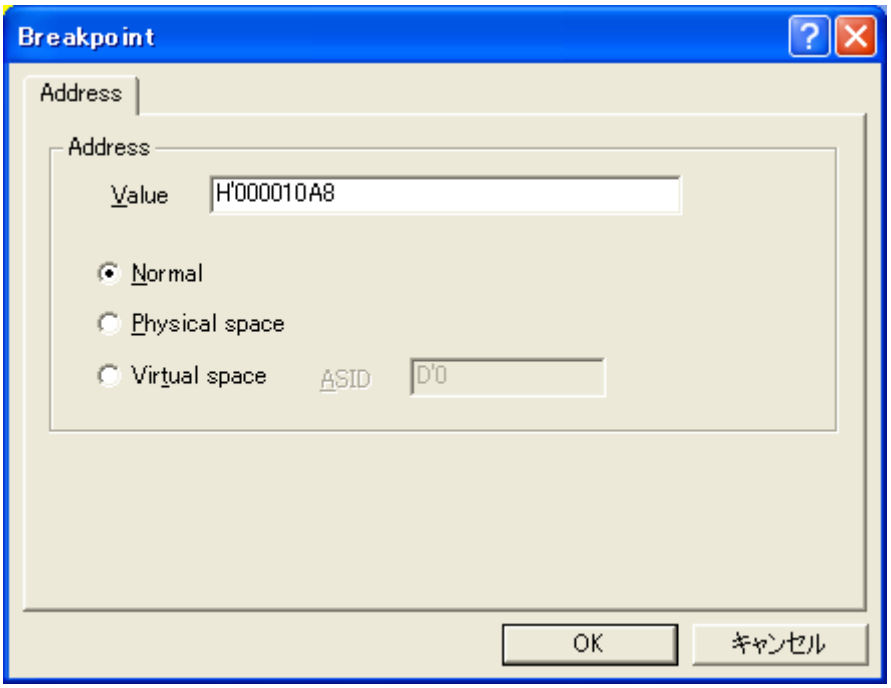


図 7.36　[Breakpoint]ダイアログボックス

【留意事項】

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

[OK]ボタンをクリックしてください。

[イベントポイント]ウィンドウには、設定された PC ブレークポイントが表示されます。

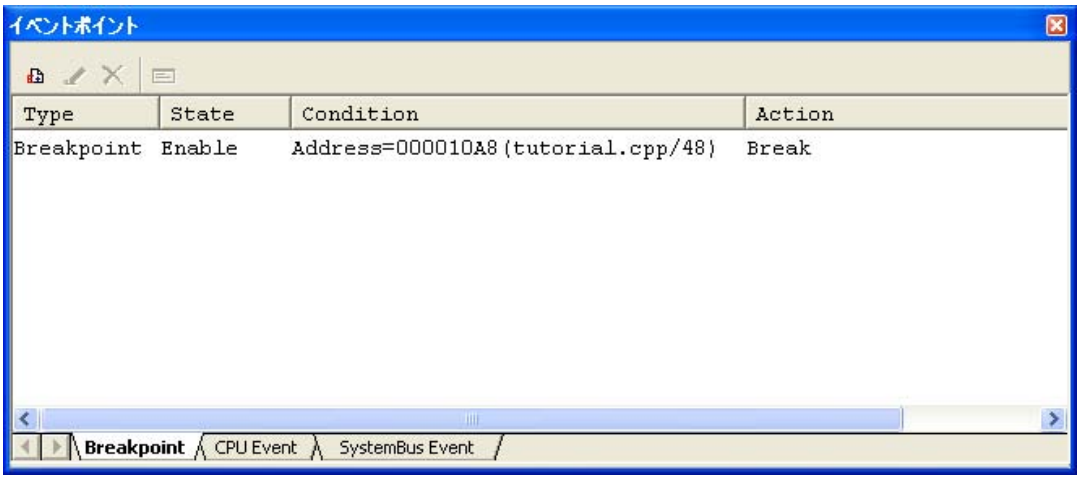


図 7.37　[イベントポイント]ウィンドウ（PC ブレーク設定時）

【留意事項】

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

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

「7.9　レジスタ内容の変更」で設定したプログラムカウンタ、スタックポインタ（CPU0:PC=H’00000800、R15=H’00010000、CPU1:PC=H’00100800、R15=H’00110000）を CPU0 用および CPU1 用の High-performance Embedded Workshop の[レジスタ]ウィンドウに設定して、CPU0 用または CPU1 用いずれかの High-performance Embedded Workshop の[実行]ボタンをクリックしてください。

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

設定した PC ブレークポイントまで、プログラムを実行して停止します。

```
27
28  00001038        void main(void)
29                  {
30
31                      long a[10];
32                      long j;
33                      int i;
34                      class Sample *p_sam;
35
36  00001042            while (1){
37  0000104A            p_sam= new Sample;
38  0000105C                for( i=0; i<10; i++ ){
39  00001064                    j = rand();
40  00001074                    if(j < 0){
41  00001078                        j = -j;
42                              }
43  0000107A                    a[i] = j;
44                          }
45  0000108C                p_sam->sort(a);
46  00001096                p_sam->change(a);
47
48  000010A8                p_sam->s0=a[0];
49  000010AC                p_sam->s1=a[1];
50  000010B2                p_sam->s2=a[2];
51  000010B8                p_sam->s3=a[3];
52  000010BE                p_sam->s4=a[4];
53  000010C4                p_sam->s5=a[5];
54  000010CA                p_sam->s6=a[6];
55  000010D0                p_sam->s7=a[7];
56  000010D6                p_sam->s8=a[8];
57  000010DC                p_sam->s9=a[9];
58  000010E2            delete p_sam;
59                      }
60  00001108        }
```

図 7.38 実行停止時の[エディタ]ウィンドウ（PC ブレーク）

[ステイタス]ウィンドウの表示内容は、以下のようになります。

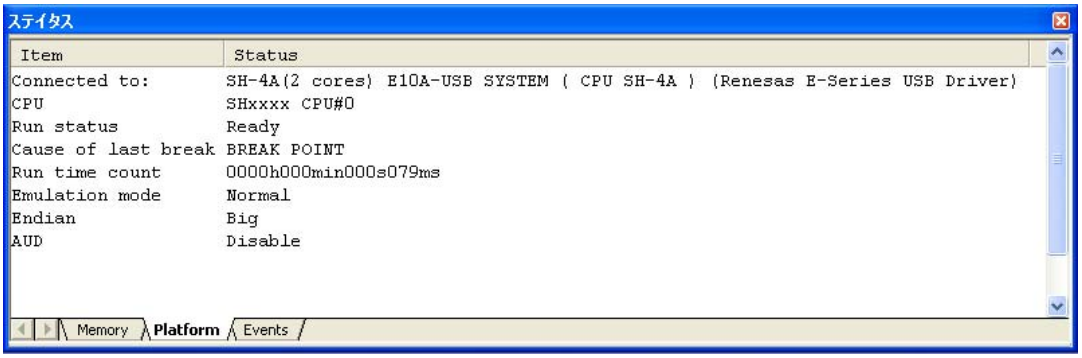


図 7.39 [ステイタス]ウィンドウの表示内容（PC ブレーク）

【留意事項】

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

7.19　ハードウェアブレーク機能

ハードウェアブレーク条件 Ch1(IA_OA)にアドレスバス条件を設定する方法を説明します。

CPU0 用の High-performance Embedded Workshop にて[表示]メニューの[コード]サブメニューから[イベントポイント]を選択してください。[イベントポイント]ウィンドウが表示されます。
先ほど設定した PC ブレークポイントを削除します。マウスの右ボタンで[イベントポイント]ウィンドウをクリックすることによって開くポップアップメニューから[すべてを削除]を選択し、設定されている PC ブレークポイントをすべて解除してください。
次は Ch1(IA_OA)を設定します。
[CPU Event]タブをクリックしてください。

ハードウェアブレーク条件 Event condition は、CPU 毎に 10 ポイントまで独立に条件を設定することができます。ここでは、ハードウェアブレーク条件 Ch1(IA_OA)を設定します。

【留意事項】

ハードウェアブレーク条件の本数は、製品ごとに異なります。各製品の仕様については、オンラインヘルプを参照してください。

[イベントポイント]ウィンドウ内の Ch1(IA_OA)行を選択してください。Ch1(IA_OA)行が強調表示されますので、ダブルクリックしてください。

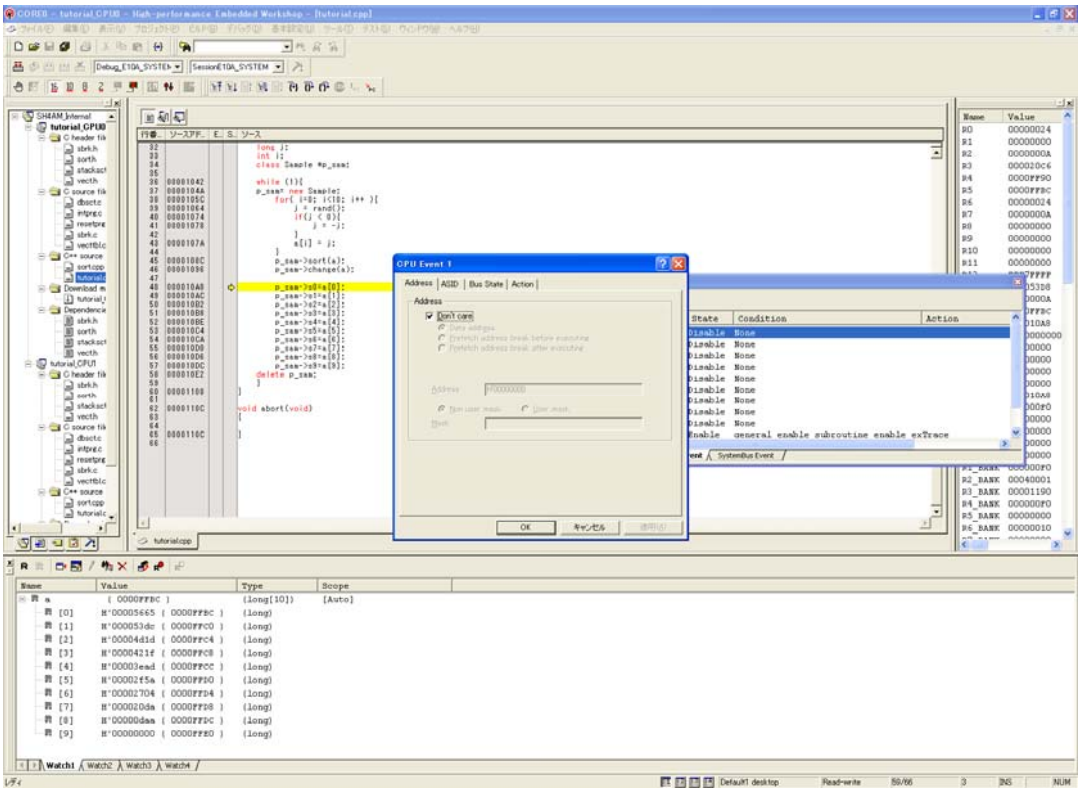


図 7.40　[High-performance Embedded Workshop]ウィンドウ（[Ch1(IA_OA)]）

[Ch1(IA_OA)]ダイアログボックスが表示されます。

[Address]ページの[Don’t care]チェックボックスを無効にします。

[Prefetch address break before executing]ラジオボタンを選択して、値として[Address]エディットボックスにアドレス H'0000108C を入力してください。

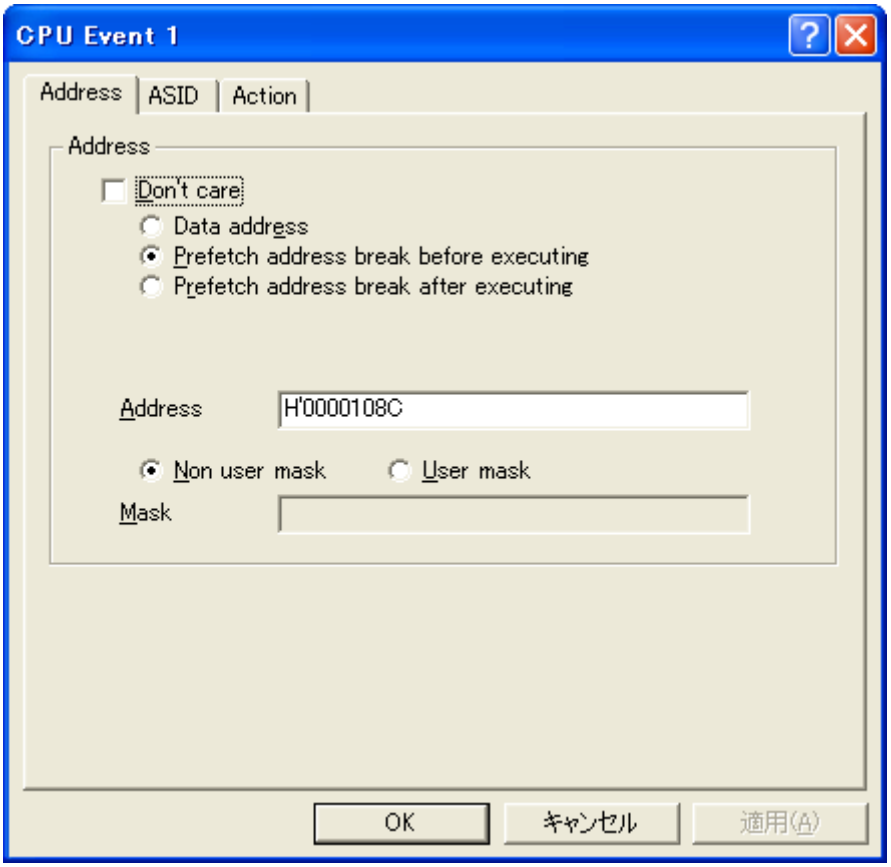


図 7.41　[Address]ページ（[CPU Event 1]ダイアログボックス）

【留意事項】

本ダイアログボックスで設定できる内容は、製品ごとに異なります。
各製品の設定内容については、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。

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

Condition 行の 1 ポイント目の表示が”None”から” Address=H'0000108C(tutorial.cpp/45) pc”に変わります。

Action 行の 1 ポイント目の表示が”Break”と表示されます。

「7.9　レジスタ内容の変更」で設定したプログラムカウンタ、スタックポインタ（CPU0:PC=H’00000800、R15=H’00010000、CPU1:PC=H’00100800、R15=H’00110000）を CPU0 用および CPU1 用の High-performance Embedded Workshop の[レジスタ]ウィンドウに設定して、CPU0 用または CPU1 用いずれかの High-performance Embedded Workshop の[実行]ボタンをクリックしてください。

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

Break Condition 1 の条件まで、プログラムを実行して停止します。

```
28  00001038          void main(void)
29                    {
30
31                        long a[10];
32                        long j;
33                        int i;
34                        class Sample *p_sam;
35
36  00001042              while (1){
37  0000104A              p_sam= new Sample;
38  0000105C                  for( i=0; i<10; i++ ){
39  00001064                      j = rand();
40  00001074                      if(j < 0){
41  00001078                          j = -j;
42                                }
43  0000107A                      a[i] = j;
44                            }
45  0000108C                  p_sam->sort(a);
46  00001096                  p_sam->change(a);
47
48  000010A8                  p_sam->s0=a[0];
49  000010AC                  p_sam->s1=a[1];
50  000010B2                  p_sam->s2=a[2];
51  000010B8                  p_sam->s3=a[3];
52  000010BE                  p_sam->s4=a[4];
53  000010C4                  p_sam->s5=a[5];
54  000010CA                  p_sam->s6=a[6];
55  000010D0                  p_sam->s7=a[7];
56  000010D6                  p_sam->s8=a[8];
57  000010DC                  p_sam->s9=a[9];
58  000010E2              delete p_sam;
59                        }
60  00001108          }
```

図 7.42　実行停止時の[エディタ]ウィンドウ（[Ch1(IA_OA)]）

[ステイタス]ウィンドウの表示内容は、以下のようになります。

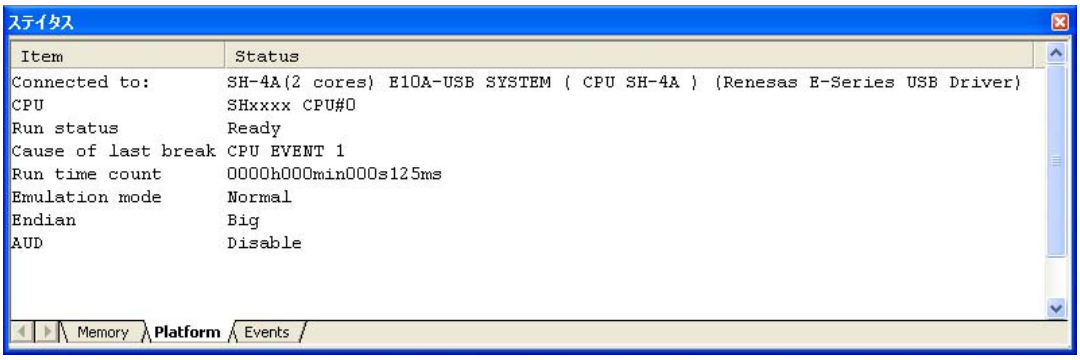


図 7.43　[ステイタス]ウィンドウの表示内容（[Ch1(IA_OA)]）

【留意事項】

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

7.20　トレース機能

E10A-USB エミュレータには 3 種類のトレース機能があります。
トレース機能では、以下に示すイベントを取得することができます。

(a) 分岐発生情報
分岐元／分岐先アドレスを取得します。

(b) 指定範囲内メモリアクセス情報
指定した範囲内のメモリアクセスをトレース取得します。
メモリ範囲は 2 つまで指定できます。Ch5(OA)、Ch6(OA)にそれぞれ範囲を指定することができます。またそれぞれトレース取得するバスサイクルとして、リードサイクル、ライトサイクル、またはリードライトサイクルを選択できます。
本機能は、以降、「ウィンドウトレース機能」と呼びます。

(c) ソフトウェアトレース
特殊な命令を実行した場合に、実行時の PC 値と 1 つの汎用レジスタ内容をトレース取得します。
あらかじめ、C ソース上に Trace(x)関数（x は変数名）を記述し、コンパイル、リンクしてください。詳細は SHC/C++コンパイラマニュアルを参照してください。
ロードモジュールを E10A-USB エミュレータにロードし、ソフトウェアトレース機能を有効にして実行すると、Trace(x)関数を実行した PC 値と、x に対応する変数の値と、ソースが表示されます。

7.20.1　内蔵トレース機能

デバイスに内蔵されているトレースバッファを使用して実現します。

【留意事項】

1. トレース取得できる分岐命令の数、トレース表示内容は、製品によって異なります。
各製品の仕様については、オンラインヘルプを参照してください。
2. 製品によっては、内蔵トレース機能はサポートしておりません。各製品の仕様については、オンラインヘルプを参照してください。
3. 製品によっては、内蔵トレース機能が拡張されています。各製品の仕様については、オンラインヘルプを参照してください。

7.20.2　AUD トレース機能

デバイスの AUD 端子を E10A-USB エミュレータに接続している場合に有効な、大容量のトレース機能です。トレース取得するイベントが発生した場合、AUD 端子からリアルタイムにトレース情報が出力されます。

トレース取得できるイベントの数は、分岐元／分岐先の組を 1 個とすると最大 1,048,544 個です。
トレースウィンドウに表示される数は、最大 65,535 個毎となります。

(1)　トレース取得モード

AUD トレース機能では、トレースを取得する際の取得方法において以下のモードを持っています。
表 7.2 に、AUD トレースのトレース取得モードを示します。

表 7.2　AUD トレース取得モード

種別	モード	説明
トレース出力が連続して発生した場合の取得モード	Realtime trace モード	トレース情報の発生が集中し、AUD 端子からの出力が間に合わなくなると CPU はトレース情報の出力を一時的に停止します。 このため、ユーザプログラムはリアルタイムに動作しますが、 トレース情報が一部取得できないことがあります。
	Non realtime trace モード	トレース情報の発生が集中し、AUD 端子からの出力が間に合わなくなると CPU の動作を一時的に停止し、トレース情報の出力を優先します。このため、ユーザプログラムのリアルタイム性がなくなります。
E10A-USB エミュレータのトレースバッファがフルになった場合の取得モード	Trace continue モード	古い情報に新しい情報を上書きして、常に最新の情報を取得します。
	Trace stop モード	その後のトレースを取得しません。 ユーザプログラムは継続して実行されます。

(2)　　トレース表示内容

プログラムの実行停止後、[トレース]ウィンドウにトレース結果を表示します。
[トレース]ウィンドウへは、以下を表示します。

PTR　　：トレースバッファ内ポインタ（最後に実行した命令が +0 となります）
IP　　：一番最近のトレース情報から遡って、いくつ前の情報であるかを示します。
　　分岐命令は、分岐元と分岐先で 1 つとカウントします。
Master　　：アクセスを行ったバスマスタの種別
Type　　：トレース取得情報の種類を表示します。
Branch Type　：分岐種別（分岐トレースの場合のみ表示されます。）
　　AUD トレースでは、PPC オプションを有効にした場合のみ表示されます。
Bus　　：どこのバスに対するアクセスであるかを表示
R/W　　：発生したデータアクセスが、リードアクセスかライトアクセスかを表示
Address　：トレース取得アドレスを表示します。
Data　　：トレース取得データを表示します。
PPC　　：パフォーマンスカウンタ出力
Instruction、Source、Label：
　　トレース取得アドレスのニモニック、該当するソースと、ラベル情報を表示します。「Source」カラムをダブルクリックすると、[エディタ]ウィンドウの該当個所にカーソルが移動します。

また、Type 列、BUS 列、R/W 列、Address 列、Data 列は、選択されているトレース種別によってそれぞれ以下の意味を持ちます。

表 7.3　[トレース]ウィンドウ表示内容

トレース種別	Type 列	BUS 列	R/W 列	Address 列	Data 列
分岐トレース	BRANCH【注 1】	表示なし	表示なし	分岐元アドレス【注 1】	表示なし
	DESTINATION	表示なし	表示なし	分岐先アドレス	表示なし
範囲内メモリアクセストレース	MEMORY	アクセス対象バス	READ/WRITE	メモリアクセスアドレス	メモリアクセスデータ【注 1】
ソフトウェアトレース	S_TRACE	表示なし	表示なし	Trace(x)関数実行アドレス	変数 x データ
システムバストレース	MEMORY	表示なし	READ/WRITE	メモリアクセスアドレス	メモリアクセスデータ (ライトのみ) 【注 1】
データロスト【注 2】	LOST	表示なし	表示なし	表示なし	表示なし
CPU ウエイト発生【注 2】	CPU-WAIT	表示なし	表示なし	表示なし	表示なし

【注】 1.　PPC オプション使用時は表示されません。
　　2.　デバッグ対象のデバイスによっては、[Lost]、[CPU-WAIT]が出力されません。
　　このため、トレースデータの出力が間に合わなかったことや、トレースデータ出力のために CPU がウエイトを発生したことがわかりませんので、ご了承ください。

7.20.3　メモリ出力トレース機能

トレース結果を指定したメモリ範囲に書き出す機能です。
メモリ出力トレースの設定方法を以下に説明します。

(1) トレース取得方法の設定

[トレース]ウィンドウを表示してください。
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。[Acquisition]ダイアログボックスが表示されます。
[Trace mode]ページで、トレースの取得条件を設定します。

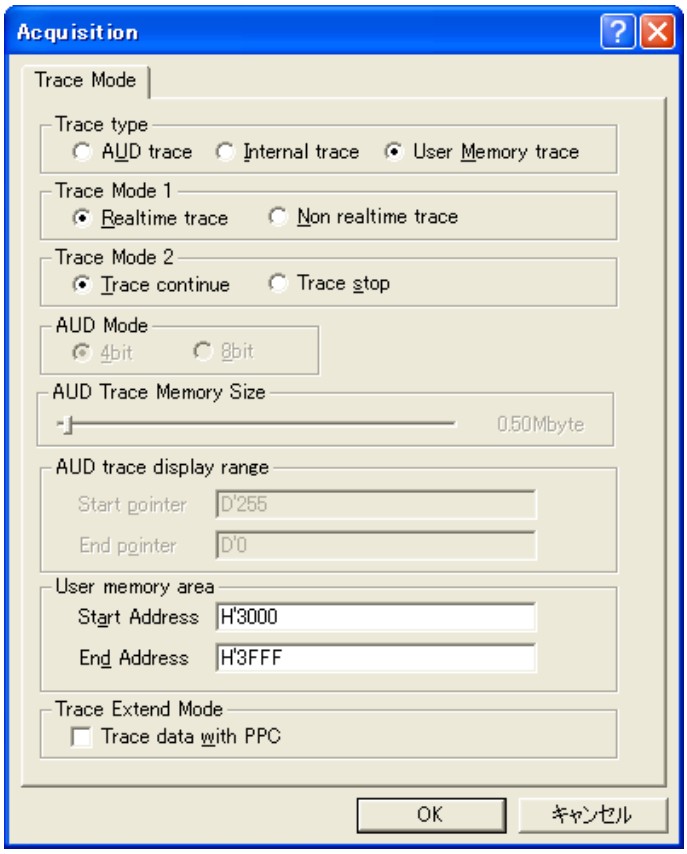


図 7.44　[Trace Mode]ページ

以下の表で、各オプションを説明します。

[トレース取得モード]

種別	モード	説明
トレース出力が連続して発生した場合の取得モード	Realtime trace モード	トレース情報の発生が集中し、メモリへの出力がまにあわなくなると、CPU は、すべてのトレース情報を出力できないことがあります。このため、ユーザプログラムはリアルタイムに動作しますが、トレース情報が一部取得できないことがあります。
	Non realtime trace モード	トレース情報の発生が集中し、メモリへの出力がまにあわなくなると、トレース情報が出力し終わるまで CPU は動作を停止します。このため、ユーザプログラムのリアルタイム性はありません。
E10A-USB エミュレータのトレースバッファがフルになった場合の取得モード	Trace continue モード	古い情報を上書きして、常に最新の情報を取得します。
	Trace stop モード	その後のトレースを取得しません。 ユーザプログラムは継続して実行されます。

【留意事項】

本ウィンドウで設定できる内容は、製品ごとに異なります。各製品の設定内容については、オンラインヘルプを参照してください。

(2) トレース結果の表示

[User Memory area]の[Start]エディットボックスに結果を出力するメモリ範囲の先頭アドレス、[End Address]エディットボックスに結果を出力するメモリ範囲の終了アドレスを指定してください。

ここで指定するアドレスはお使いになる環境に応じて入力してください。

この時トレース出力結果によりメモリ内容を上書きしますので、プログラムのダウンロードされている範囲は指定しないでください。

「7.18.1　PC ブレーク機能」の例でプログラムを実行してください。

実行停止後に[トレース]ウィンドウにトレース結果を表示します。

次に表示例をご紹介します。

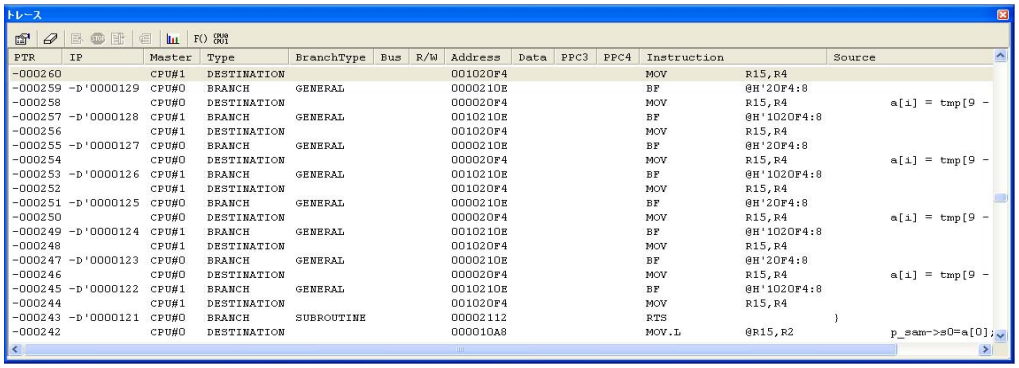


図 7.45　[トレース]ウィンドウ（表示例）

7.20.4　トレースウィンドウの便利な機能

トレースウィンドウでは、以下の便利な機能をサポートしています。

(1) 指定データの検索
(2) 指定データの抽出
(3) 指定データをフィルタリングして再表示
(4) 分岐先アドレスから、次の分岐元アドレスまでの情報補填

これらの機能の使用方法については、「5.7　トレース情報を見る」を参照してください。

(5) ユーザプログラム実行中のトレース設定内容変更
デバッグ対象のデバイスによっては、トレースの設定をユーザプログラム実行中に変更することができます。
各製品の仕様については、オンラインヘルプを参照してください。

7.21　MMU サポート

本機能は、サポートデバイスが MMU を内蔵しているものに限ります。

(1) [TLB]ウィンドウ

E10A-USB エミュレータでは、TLB テーブルの内容を簡単に表示、編集することができます。
[表示]メニューから[CPU]→[TLB]を選択することによって、表示、編集できます。
詳しくは、オンラインヘルプをご覧ください。

(2) VP_MAP 変換機能

MMU を内蔵しているデバイスにおいては、内部のアドレス(論理アドレス)を実メモリのアドレス(物理アドレス)に変換することができます。このアドレス変換はデバイス内蔵のアドレス変換テーブル(TLB)にしたがって行われます。MMU はエミュレーション実行時や E10A-USB エミュレータのメモリアクセスコマンド実行状態でも機能しています。
そのため、MMU のアドレス変換機能が有効な状態でメモリをアクセスするコマンドを使用した場合、MMU によって変換されたアドレスにアクセスします。指定のアドレスが TLB 内にない場合、TLB ミスヒットとなり、ユーザプログラムで TLB を書き換えなければなりません。
そこで、E10A-USB エミュレータでは、MMU を内蔵しているデバイスに対しては、VP_MAP テーブルに従ったアドレス変換機能をサポートしています。
VP_MAP テーブルとはエミュレーションコマンド VPMAP_SET コマンドで作成する E10A-USB エミュレータ用アドレス変換テーブルです。

(例)まず、変換テーブルを作成します。
(論理アドレス H’10000～H’10fff を物理アドレス H’4000000～H’4000fff に、
論理アドレス H’11000～H’11fff を物理アドレス H’0～H’fff に変換するテーブルを作成する)

```
>vs  10000  10fff  4000000 (RET)
>vs  11000  11fff  0 (RET)
>vd  (RET)
<VADDR_TOP>                <VADDR_END>      <PADDR_TOP>
```

00010000　　00010fff　　04000000
00011000　　00011fff　　00000000
DISABLE

次に、VP_MAP テーブルを有効にします(無効時はアドレス変換はしません)。
>ve　enable (RET)
>vd　(RET)
<VADDR_TOP>　　<VADDR_END>　　<PADDR_TOP>
00010000　　00010fff　　04000000
00011000　　00011fff　　00000000
ENABLE
この場合、論理アドレスと物理アドレスの対応は以下のようになります。

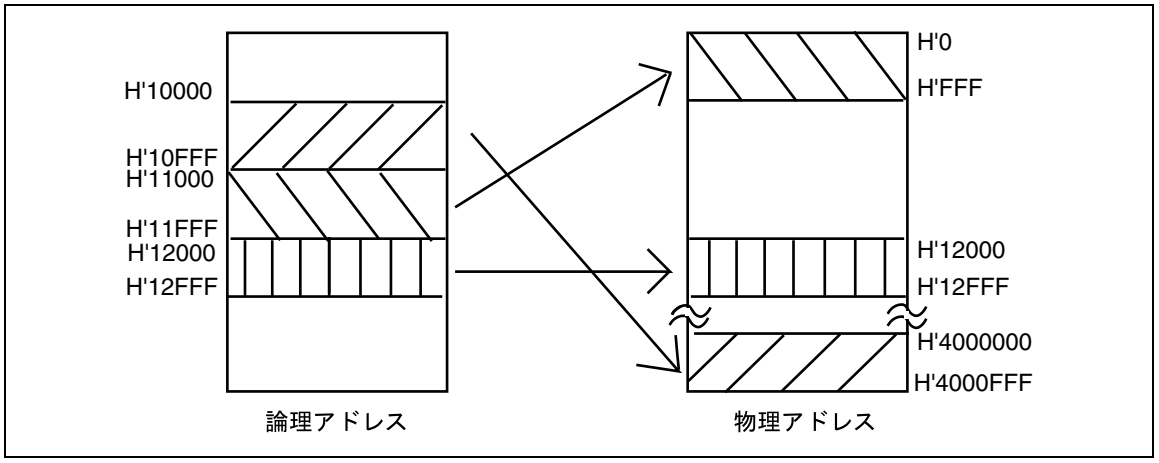


図 7.46　VP_MAP テーブルによるアドレス変換

アドレス変換方法は、[Configuration]ダイアログボックスの[Memory area]グループの各ラジオボタンにより決定されます。
次に、各設定状態でのアドレス変換方法を示します。

(3) Normal ラジオボタンが選択されている場合
VP_MAP の設定が TLB よりも優先されます。VP_MAP が有効で、VP_MAP 範囲内の場合、E10A-USB エミュレータが VP_MAP テーブルを用いてアドレス変換します。
VP_MAP が有効でアドレスが VP_MAP 範囲外のとき、あるいは VP_MAP が無効であるときは、MMU の状態に従います。

(4) Physical ラジオボタンが選択されている場合
アドレス変換は行いません。

(5)　Virtual ラジオボタンが選択されている場合
TLB にしたがってアドレス変換します。
ただし、TLB テーブル範囲外のとき TLB エラーになります。

表 7.4　アドレス変換に使用するテーブル

ラジオボタン設定【注】	VP_MAP		MMU		使用する変換テーブル
	有効/無効	範囲内/外	有効/無効	TLB 範囲内/外	
Normal 指定	有効	範囲内	有効	範囲内	VP_MAP テーブルによって変換
				範囲外	VP_MAP テーブルによって変換
			無効	範囲内/外	VP_MAP テーブルによって変換
		範囲外	有効	範囲内	TLB テーブルによって変換
				範囲外	TLB エラー発生
			無効	範囲内/外	変換しない
	無効	範囲内/外	有効	範囲内	TLB テーブルによって変換
				範囲外	TLB エラー発生
			無効	範囲内/外	変換しない
Virtual 指定	有効/無効	範囲内/外	有効	範囲内	TLB テーブルによって変換
				範囲外	TLB エラー発生
			無効	範囲内	TLB テーブルによって変換
				範囲外	TLB エラー発生
Physical 指定	有効/無効	範囲内/外	有効/無効	範囲内/外	変換しない

【注】[Configuration]ダイアログボックスの[Memory area]グループボックスで指定します。

7.22　スタックトレース機能

E10A-USB エミュレータでは、スタック情報を用いて、現在の PC がある関数がどの関数からコールされているかを表示します。

【留意事項】

本機能は、Elf/Dwarf2 形式のデバッグ情報を持ったロードモジュールをロードした場合のみ使用できます。

Elf/Dwarf2 形式のデバッグ情報を持ったロードモジュールは、SHC/C++コンパイラ(OEM、バンドル販売品を含む)V6.0 以降でサポートしています。

CPU0 用の High-performance Embedded Workshop にて sort 関数内の行の[Event]カラムをダブルクリックして、Event ポイントを設定してください。

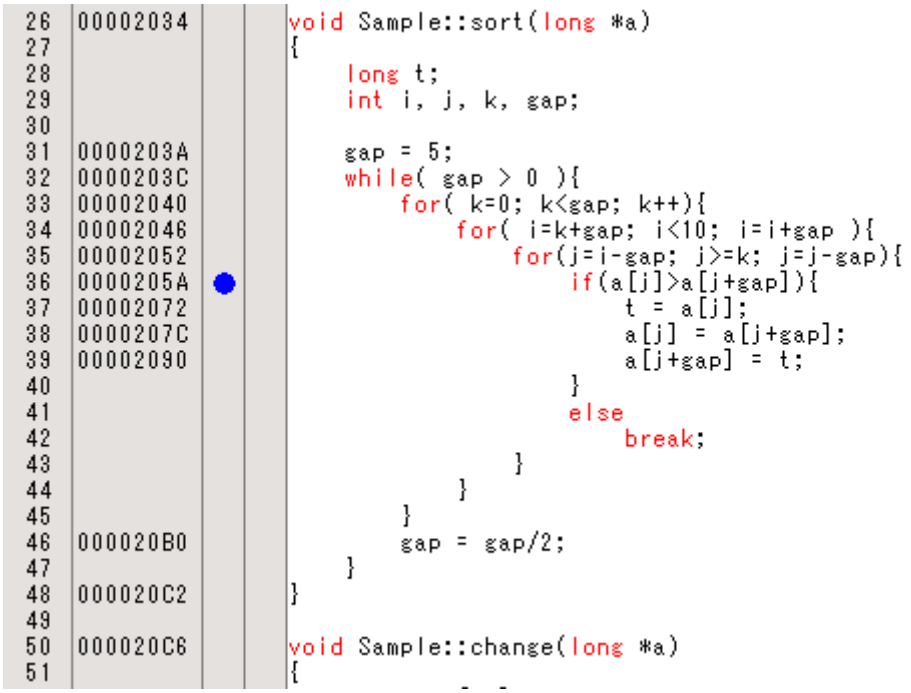


図 7.47　[エディタ]ウィンドウ（ハードウェアブレークの設定）

「7.9　レジスタ内容の変更」で設定したプログラムカウンタ、スタックポインタ（CPU0:PC=H’00000800、R15=H’00010000、CPU1:PC=H’00100800、R15=H’00110000）を CPU0 用および CPU1 用の High-performance Embedded Workshop の[レジスタ]ウィンドウに設定して、CPU0 用または CPU1 用の High-performance Embedded Workshop の[実行]ボタンをクリックしてください。

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

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

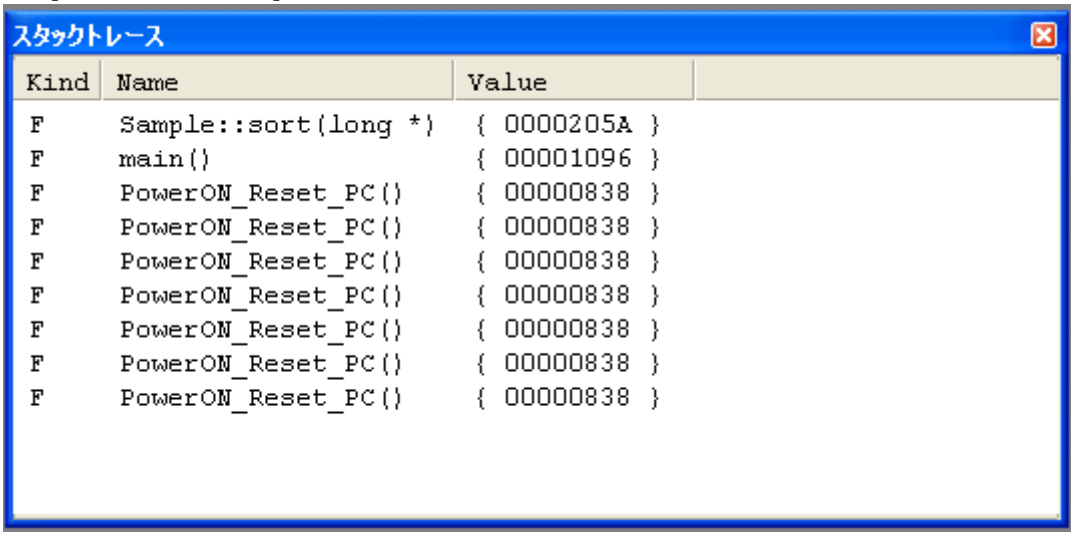


図 7.48　[スタックトレース]ウィンドウ

現在 PC が sort()関数内にあり、sort()関数は main()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[Event]カラムを再度ダブルクリックして、ハードウェアブレークを解除します。

【留意事項】

本機能の詳細はオンラインヘルプを参照してください。

7.23　パフォーマンス測定機能

E10A-USB エミュレータには、マイコンのパフォーマンスを測定する機能があります。

- パフォーマンス測定機能
  マイコン内蔵のカウンタを使用して、各種イベント発生回数、サイクル数を測定する機能です。
  開始条件、終了条件を設定できます。
  測定できる項目はサポートデバイスによって異なります。

7.24　パフォーマンス測定機能

マイコン内蔵のカウンタを使用して各種イベント発生回数、サイクル数を測定する例を説明します。

(1) 設定方法

CPU0 用の High-performance Embedded Workshop にて[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]を選択してください。[パフォーマンス解析方法の選択]ダイアログボックスが開きますので、[OK]ボタンを押してください。

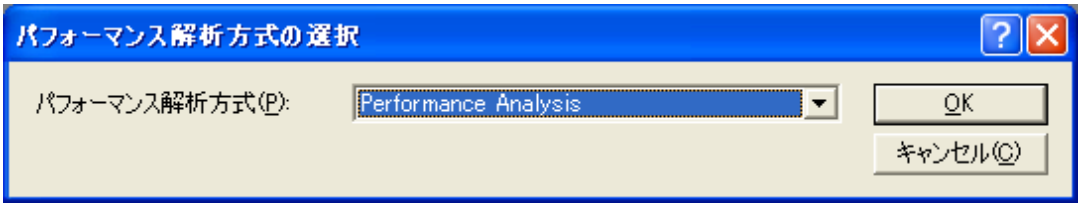


図 7.49　[パフォーマンス解析方法の選択]ダイアログボックス

[パフォーマンス解析]ウィンドウが表示されます。

本ウィンドウをマウスの右ボタンでクリックすることによって開くポップアップメニューより、[設定]を選択してください。[Performance Analysis]ダイアログボックスが開きます。

本ダイアログボックスで測定するイベントや測定条件を設定することができます。

【留意事項】

本ダイアログボックスで表示される内容は、製品ごとに異なります。
各製品の設定内容については、オンラインヘルプを参照してください。

条件設定後、[OK]ボタンを押し、ユーザプログラムを実行すると、実行終了後に[パフォーマンス解析]ウィンドウに結果が表示されます。

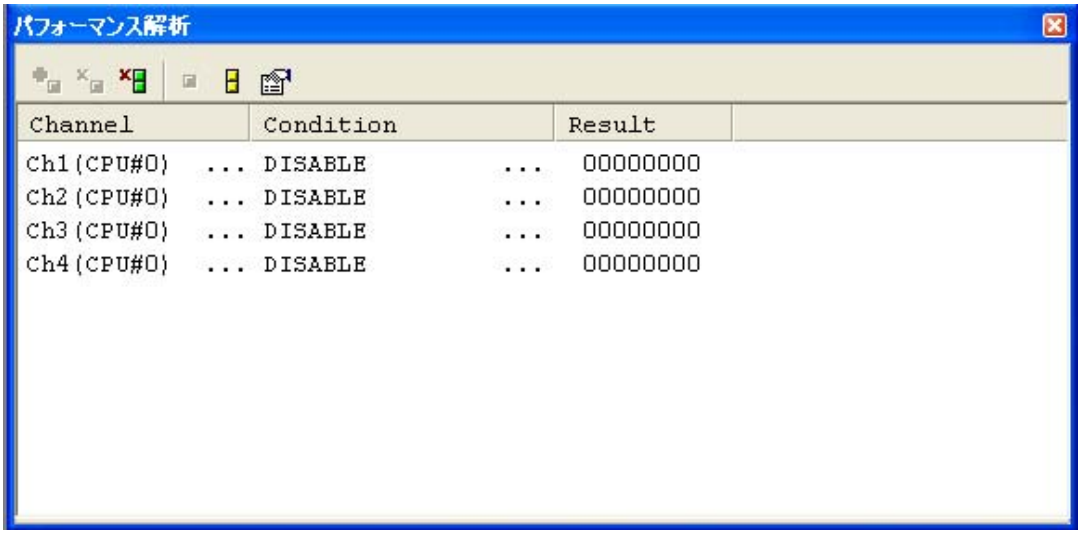


図 7.50　[パフォーマンス解析]ウィンドウ

【留意事項】

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

7.25　フラッシュメモリへのダウンロード機能

E10A-USB エミュレータは、外部フラッシュメモリ領域へダウンロードすることができます。
本機能を使用するためには、ご使用のフラッシュメモリにライトするプログラム（以後、ライトモジュールと呼びます）とフラッシュメモリを消去するプログラム（以後、消去モジュールと呼びます）、またそれらのモジュールをダウンロードし実行するRAM 領域が必要です。

【留意事項】

1. **ライトモジュールと消去モジュールは、お客様の方で用意して頂く必要があります。**
2. **ご使用のデバイスによっては、本機能は使用できません。**
   本機能をご使用になれないデバイスでは、図 6.52　[Loading flash memory]ページは表示されません。

(a)　ライト/消去モジュールと E10A-USB エミュレータファームウェアとのインタフェース

ライト/消去モジュールは、E10A-USB エミュレータファームウェアから分岐します。
E10A-USB エミュレータファームウェアからライト/消去モジュールへ正常に分岐、またはライト/消去モジュールからE10A-USB エミュレータファームウェアに正常に戻ってくるようにするため、以下の条件を必ず守ってください。

- ライト/消去モジュールは、すべてアセンブル言語で記述してください。
- ライト/消去モジュール呼び出し前、呼び出し後で全ての汎用/制御レジスタ値を退避、復帰してください。
- ライト/消去モジュールは、処理終了後、必ずコール元に戻る構造としてください。
- ライト/消去モジュールは、モトローラ形式のファイルにしてください。

また、フラッシュメモリアクセスに必要な情報を正確に渡すため、以下のインタフェースで作成してください。

表 7.5　モジュールインタフェース

項番	モジュール名	引数	リターン値
1	ライトモジュール	R4(L):ライトアドレス R5(L):アクセスサイズ 0x4220=バイト 0x5720=ワード 0x4C20=ロング R6(L):ライトデータ	R0(L):終了コード 正常終了=0 異常終了=0 以外
2	消去モジュール	R4(L):アクセスサイズ 0x4220=バイト 0x5720=ワード 0x4C20=ロング	なし

【注】(L)はロングサイズであることを示します。

【留意事項】

・ ライトモジュール

ライトデータは、R6 レジスタにアクセスサイズ分設定されます。
R6 レジスタは、アクセスサイズがワードまたはバイトの場合、上位ビットには 0 が設定されます。

(b) フラッシュメモリダウンロード方法

フラッシュメモリへダウンロードするには、[基本設定]メニュー→[エミュレータ]→[システム…]から開く[Configuration]ダイアログボックスの[Loading flash memory]ページで必要な設定を行う必要があります。

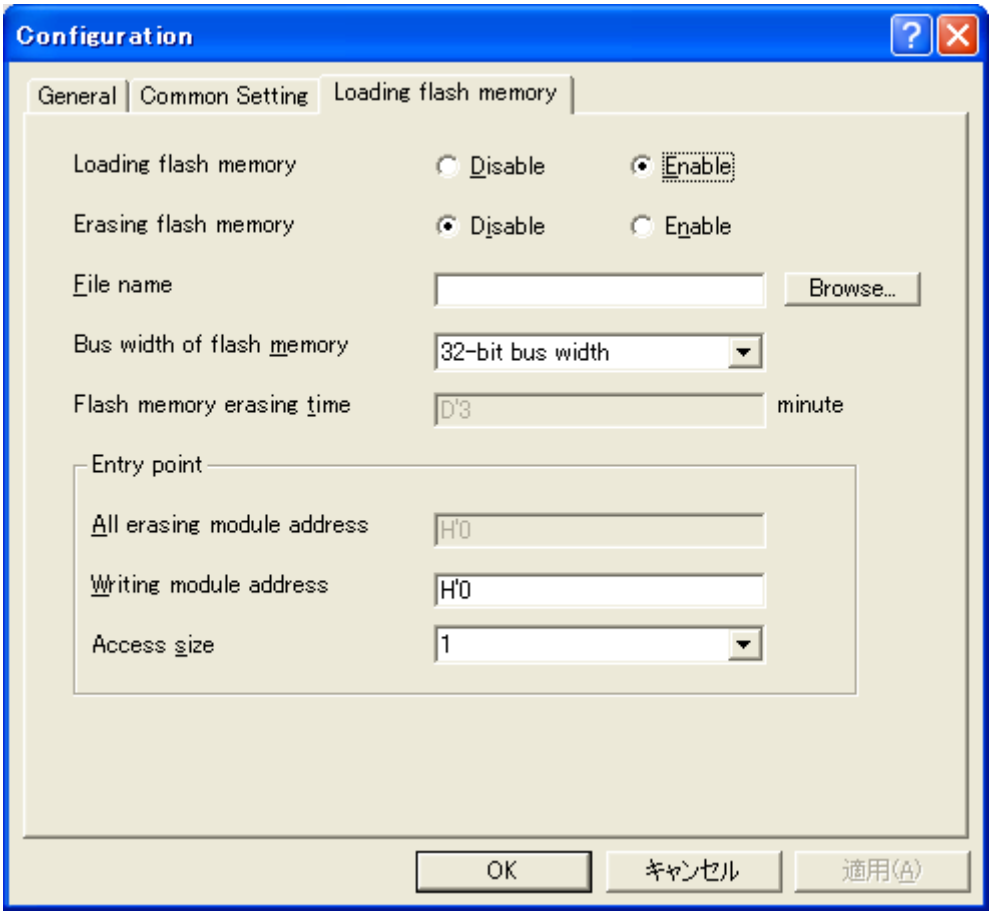


図 7.51 [Loading flash memory]ページ

[Loading flash memory]ページのオプションを以下の表で説明します。

表 7.6　[Loading flash memory]ページのオプション

オプション	説明
[Loading flash memory] ラジオボタン	フラッシュメモリへのダウンロードを行う場合、Enable にします。 Enable 時は、[File]メニューから[File load]を選択してダウンロードを行う場合、常にライトモジュールを呼び出します。 Enable：フラッシュメモリへのダウンロードを行う Disable：フラッシュメモリへのダウンロードを行わない
[Erasing flash memory] ラジオボタン	フラッシュメモリ書き込みの前に消去を行う場合、Enable にします。 Enable 時は、ライトモジュールを呼び出す前に消去モジュールを呼び出します。 Enable：フラッシュメモリの消去を行う Disable：フラッシュメモリの消去を行わない
[File name]エディットボックス	ライト/消去モジュールを含む S タイプロードモジュールのファイル名を設定します。設定したファイルは、フラッシュメモリへロードする前に RAM 領域へロードします。 ファイル名の入力の文字数は、最大 128 文字です。
[Bus width of flash memory] リストボックス	フラッシュメモリのバス幅の設定を行います。
[Flash memory erasing time]エディットボックス 【注】	フラッシュメモリ消去時の TIMEOUT 値を設定します。デフォルトは 3 分となっていますが、消去に時間がかかる場合は値を大きくしてください。 入力値の基数は 10 進数です。「H'」を付けると 16 進数になります。
[Entry point]グループボックス	ライト/消去モジュールの呼び出し先アドレス/アクセスサイズを設定します。 [All erasing module address] エディットボックス：消去モジュールの呼び出し先アドレスを入力します。 [Writing module address] エディットボックス：ライトモジュールの呼び出し先アドレスを入力します。 [Access size] ドロップダウンリストボックス：ライト/消去モジュールをロードする RAM 領域のアクセスサイズを選択します。

【注】設定できる値は、D'1～D'65535 ですが、設定値によって、TIMEOUT 時間が長くなります。したがって、使用しているフラッシュメモリの消去時間を考慮して、できるだけ最小の値を入力することをお勧めします。

(c)　フラッシュメモリダウンロード機能使用時の注意事項

フラッシュメモリダウンロード時には、以下の注意事項があります。

- フラッシュメモリダウンロードをイネーブルにしている場合、フラッシュメモリ領域以外へのダウンロードはできません。
- フラッシュメモリ領域へはダウンロードのみ可能です。メモリライト、PC ブレーク等の操作は RAM 領域のみに行ってください。
- フラッシュメモリの消去をイネーブルにしている場合、消去を行っている間は[Stop]ボタンで停止できません。
- ライトモジュール、消去モジュールの各領域は、必ず MMU 無効空間としてください。

(d)　フラッシュメモリダウンロード例

(株)Intel 製フラッシュメモリ（型名 : G28F640J5-150）にダウンロードする例をご紹介します。なお、各エミュレータのインストール先フォルダの中の¥Fmtool フォルダにサンプルを提供しています。このサンプルを参考にして、お客様の仕様に合ったプログラムを作成してください。

表 7.5　ボード仕様

項目		内容
SDRAM アドレス		H'0C000000 ~ H'0FFFFFFF
フラッシュメモリアドレス		H'00000000 ~ H'01FFFFFF
フラッシュメモリバス幅		32 ビット
動作環境	エンディアン	ビッグエンディアン

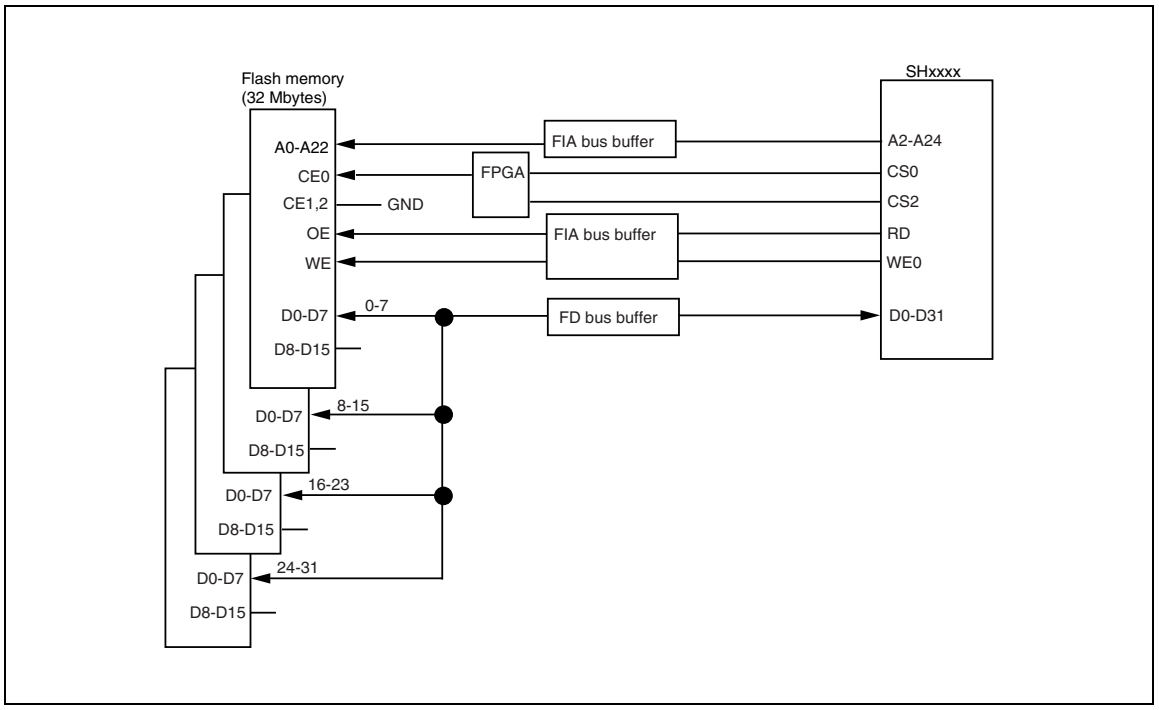


図 7.52　フラッシュメモリ結線図

表 7.7　サンプルプログラム仕様

項目	内容
使用する RAM エリア	H'0C001000～H'0C0015BF
ライトモジュール開始アドレス	H'0C001100
消去モジュール開始アドレス	H'0C001000

(i)　SDRAM を使用するため、バスコントローラを設定します。

(ii) [Configuration]ウィンドウの[Loading flash memory]ページの各オプションを以下のように設定します。

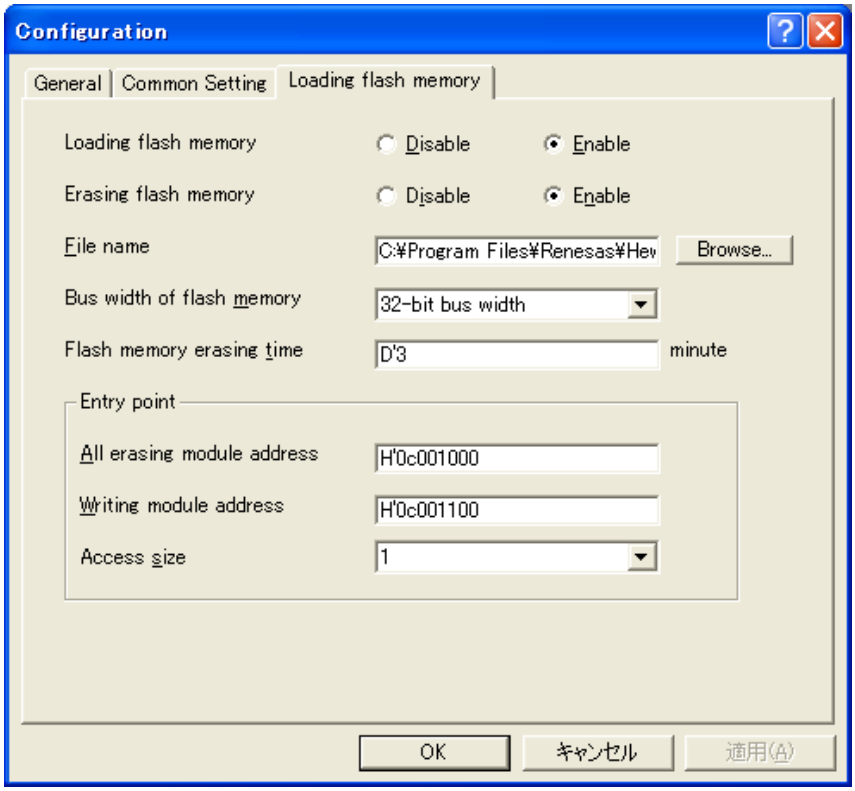


図 7.53　[Loading flash memory]ページ

【注】

1. フラッシュメモリにデータが既に書かれている場合、必ず[Erasing flash memory]を[Enable]にしてください。[Disable]の場合、ベリファイエラーが発生します。
2. [Erasing flash memory]を選択した場合、消去には約 1 分間かかります（サンプル例の場合）。

(iii) ダウンロードするオブジェクトを選択し、フラッシュメモリ領域にダウンロードを行ってください。

8. 保守と保証

第 8 章では、本エミュレータの保守方法と保証内容、修理規定と修理の依頼方法を説明しています。

8.1 ユーザ登録

ご購入頂いた際には WEB でのユーザ登録をお願いします。
ユーザ登録については、本ユーザーズマニュアルの「ユーザ登録について」に従って行ってください。

8.2 保守

(1) 本製品に埃や汚れが付着した場合は、乾いた柔らかい布で拭いてください。
シンナーなどの溶剤を使用した場合は、塗装が剥げたりしますので、使用しないでください。
(2) 長時間使用しない時は、安全のため電源プラグをコンセント等から抜いて保管してください。

8.3 保証内容

本ユーザーズマニュアルの「重要事項」を守った正常な使用状態のもとで、購入後 1 年以内に故障した場合は、無償修理または、無償交換致します。
但し、次の項目による故障の場合は、ご購入から 1 年以内でも有償修理または、有償交換と致します。

- 製品の誤用、濫用または、その他異常な条件下での使用
- 弊社以外のものによる改造、修理、保守または、その他の行為
- ユーザシステムの不備または、誤使用
- 火災、地震または、その他の事故

その際は、ご購入された販売元の担当者へご連絡ください。なお、レンタル中の製品は、レンタル会社または、貸し主とご相談ください。

8.4 修理規定

(1) 有償修理

ご購入後 1 年を超えて修理依頼される場合は、有償修理となります。

(2) 修理をお断りする場合

次の項目に該当する場合には、修理でなく、ユニット交換または、新規購入頂く場合があります。

- 機構部分の故障、破損
- 塗装、メッキ部分の傷、剥がれ、錆
- 樹脂部分の傷、割れなど
- 使用上の誤り、不当な修理、改造による故障、破損
- 電源ショートや過電圧、過電流のため電気回路が大きく破損した場合
- プリント基板の割れ、パターン焼失箇所
- 修理費用より交換の費用が安くなる場合
- 不良箇所が特定できない場合

(3) 修理期間の終了

製品生産中止後、1 年を経過した場合は修理不可能な場合があります。

(4) 修理依頼時の輸送料など

修理依頼時の輸送費などの費用は、お客様でご負担願います。

8.5 修理依頼方法

エミュレータの故障と診断された場合には、以下の手順にて修理を依頼してください。

お客様：故障発生

付録 F「故障症状調査書」に必要事項をご記入の上、故障症状調査書と故障したエミュレータを販売元まで送付してください。

「故障症状調査書」は、迅速な修理を行うためにも詳しく記入してください。

⚠ 注意

エミュレータの輸送方法に関して：

- **修理のために本工ミュレータを輸送される場合、本工ミュレータの梱包箱、クッション材を用いて精密機器扱いで発送してください。工ミュレータの梱包が不十分な場合、輸送中に損傷する恐れがあります。やむをえず他の手段で輸送する場合、精密機器として厳重に梱包してください。また工ミュレータを梱包する場合、必ず工ミュレータ添付の導電性工アキャップもしくは導電性ポリ袋(通常青色の袋)をご使用ください。他の袋を使用した場合、静電気の発生などにより工ミュレータ別の故障を引き起こす恐れがあります。**

付録 A トラブルシューティング

- エディタにテキストファイルが表示されているが、シンタックス色付けが表示されない

  ファイルに名前が付いている（保存した）ことを確認してください。
  また、[基本設定->オプション...]を選んで[オプション]ダイアログボックスを開き、
  [エディタ]タブの[シンタックスカラーリング]チェックボックスがチェックされていることを確認してください。High-performance Embedded Workshop ではファイルの拡張子の属するファイルグループを調べてファイルを色付けするかどうか判断します。
  現在定義されている拡張子とファイルグループを参照するには、 [プロジェクト->ファイルの拡張子...]を選んで[ファイル拡張子]ダイアログボックスを表示してください。
  色付け情報を確認するには、[基本設定->表示形式]を選んで[表示形式]ダイアログボックスの[カラー]タブを参照してください。

- ツールの設定を変えたいが、[ツール->アドミニストレーション…] メニューオプションを選べない

  ワークスペースを開いている間は[ツール->アドミニストレーション...] を選ぶことはできません。[ツールアドミニストレーション]ダイアログボックスを開くには、
  現在のワークスペースを閉じてください。

- 日本語版 Windows®XP, Windows Vista® , Windows®7 上で日本語入力ができない、または日本語の文字が正しく表示されない

  [基本設定->表示形式] を選んで“フォント”フィールドで日本語のフォントを選んでください。

- 自分の PC でワークスペースを開いた。同時に、他の人が他の PC から同じワークスペースを開いた。自分でワークスペースの設定を変えて保存した。
  その後、他の人がワークスペースの設定を変えて保存した。自分が再びワークスペースを開くと、設定が自分の行った設定とは異なっていた。

  最後に保存した設定が有効です。High-performance Embedded Workshop はワークスペースを開くとメモリ内で更新します。ユーザが意識的に設定をファイルに保存しない限り、
  設定はファイルに保存されません。

この他にもルネサスの WEB(www.renesas.com)に掲載されている E10A-USB エミュレータ、High-performance Embedded Workshop に関する FAQ を参照してください。

付録 B メニューー覧

GUI メニューの一覧を表 B.1 に示します。

表 B.1 GUI メニューー覧

メニューー	メニューオプション		ショートカットキー	ツールバーボタン	備考
表示	逆アセンブリ		Ctrl+D		[逆アセンブリ]ウィンドウを表示します
	コマンドライン		Ctrl+L		[コマンドライン]ウィンドウを表示します
	TCL ツールキット		Ctrl+Shift+L		[Console]ウィンドウを表示します
	ワークスペース		Alt+K		[Workspace]ウィンドウを表示します
	アウトプット		Alt+U		[Output]ウィンドウを表示します
	差分				[差分]ウィンドウを表示します
	CPU	レジスタ	Ctrl+R		[レジスタ]ウィンドウを表示します
		メモリ...	Ctrl+M		[メモリ]ウィンドウを表示します
		IO	Ctrl+I		[IO]ウィンドウを表示します
		ステイタス	Ctrl+U		[ステイタス]ウィンドウを表示します
		キャッシュ	Shift+Ctrl+C		[Cache]ウィンドウを表示します
		TLB	Shift+Ctrl+X		[TLB]ウィンドウを表示します
	シンボル	ラベル	Shift+Ctrl+A		[ラベル]ウィンドウを表示します
		ウォッチ	Ctrl+W		[ウォッチ]ウィンドウを表示します
		ローカル	Shift+Ctrl+W		[ローカル]ウィンドウを表示します
	コード	イベントポイント	Ctrl+E		[イベントポイント]ウィンドウを表示します
		トレース	Ctrl+T		[Trace]ウィンドウを表示します
		スタックトレース	Ctrl+K		[スタックトレース]ウィンドウを表示します

表 B.1 GUI メニュー一覧（続き）

メニュー	メニューオプション		ショートカットキー	ツールバーボタン	備考
表示	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		[Performance Analysis]ウィンドウを表示します
基本設定	基数	16 進数			数値の表示 / 入力時の基数のデフォルト設定を 16 進数とします
		10 進数			数値の表示 / 入力時の基数のデフォルト設定を 10 進数とします
		8 進数			数値の表示 / 入力時の基数のデフォルト設定を 8 進数とします
		2 進数			数値の表示 / 入力時の基数のデフォルト設定を 2 進数とします
	エミュレータ	システム...			デバッグプラットフォームの設定を行う[Configuration Properties]ダイアログボックスを表示します
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				ターゲットマイコンをリセットし、PC をリセットベクタアドレスに設定します
	実行		F5		現在の PC からユーザプログラムを実行します
	リセット後実行		Shift+F5		ターゲットマイコンをリセットし、リセットベクタアドレスからユーザプログラムを実行します
	カーソル位置まで実行				現在の PC からテキストカーソルの位置までユーザプログラムを実行します
	カーソル位置を PC 値に設定				テキストカーソルの位置に PC を設定します

表 B.1 GUI メニュー一覧（続き）

メニュー	メニューオプション	ショートカットキー	ツールバーボタン	備考
デバッグ	ラン...			実行時の PC や PC ブレークポイントの設定が可能な[プログラム実行]ダイアログボックスを表示します。
	ステップイン	F11		ユーザプログラムの 1 ブロックを実行して停止します
	ステップオーバ	F10		ユーザプログラムの 1 ブロックを実行して停止しますが、サブルーチンを呼び出す場合は、サブルーチンには入りません
	ステップアウト	Shift+F11		現在の関数の終わりに到達するまでユーザプログラムを実行します
	ステップ...			ステップ動作の設定が可能な[プログラムステップ]ダイアログボックスを表示します
	ステップモード	自動		[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
	接続			デバッグプラットフォームを接続します
	初期化			デバッグプラットフォームを切断し、再接続します
	接続解除			デバッグプラットフォームを切断します 製品によっては使用できません
	ダウンロード			オブジェクトプログラムをロードします
	アンロード			オブジェクトプログラムをアンロードします

付録 C コマンドライン機能

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

コマンドの詳細はオンラインヘルプをご覧ください。

付録 D 注意事項

(1) ロードモジュール作成後のソースファイル位置移動に関する注意事項

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

(2) ソースレベル実行機能

- ソースファイル

ロードモジュールに対応しないソースファイルをプログラムウィンドウに表示しないでください。

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

- Step

標準Cライブラリ等にも移行します。上位関数に戻るには Step Out を使用してください。

また、for および while 文では、1 回のステップでは次の行に進みません。

進める場合はもう一度ステップしてください。

(3) ファイルアクセス中の操作について

ロードモジュールのダウンロード中、[メモリ]ウィンドウでの比較、[保存]、[トレース]ウィンドウでのセーブなどの処理中に他の操作を行わないでください。

ファイルアクセス処理が正しく実行されない場合があります。

(4) ウォッチ機能

- 最適化時の局所変数

最適化オプションでコンパイルされた C ソースの局所変数表示は、生成されたオブジェクトコードによって、正しく表示できないことがあります。

[逆アセンブリ]ウィンドウを表示し、生成されたオブジェクトコードを確認してください。

また、指定した局所変数の割付け領域がない場合があります。

この場合、次のように表示します。

例) 変数名を asc とする。

```
asc = ？  -  target error 2010 (xxxx)
```

- 変数名の指定
  変数名でないシンボル名(関数名)等を指定した場合、内容は表示しません。
  例) 関数名を main とする。
  main =

(5) ラインアセンブル機能

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

(6) コマンドラインインタフェース

- バッチファイル
  バッチファイル実行中に、"Not currently available"が表示される場合は、sleep コマンドを挿入してください。
  sleep させる時間は動作環境によって異なりますので、調整してください。
  例) memory_fill で、"Not currently available"を表示する場合
  sleep d'3000
  memory_fill 0 ffff 0

- コマンドファイルでのファイル指定
  コマンドファイルの指定方法によりカレントディレクトリが移動する場合があります。
  コマンドファイル内のファイル指定は、カレントディレクトリの移動に影響をうけないように絶対パスで記述することをお勧めします。
  例） FILE_LOAD C:¥HEW3¥Tools¥Renesas¥DebugComp¥Platform¥E10A-USBM¥Tutorial¥Tutorial¥Debug_SHxxxx_E10A-USB Multi_SYSTEM¥tutorial.abs

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

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

(9) プログラム実行中の[レジスタ]ウィンドウ動作に関する注意事項

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

(10) ブレーク機能

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

- BREAKPOINT 解除

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

  BREAKPOINT IS DELETED A=xxxxxxxx

  上記メッセージが表示された場合は、[イベントポイント]ウィンドウの[すべてを削除]ボタンまたは[無効]ボタンにより、すべての BREAKPOINT 設定を解除してください。

(11) BREAKPOINTの設定数と[条件を指定して実行]メニューの[テンポラリPCブレークポイント]の設定数

BREAKPOINT の設定数と[条件を指定して実行]メニューの[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 BREAKPOINT を 255 個設定した状態では、[条件を指定して実行]メニューの[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。BREAKPOINT と[条件を指定して実行]メニューの[テンポラリ PC ブレークポイント]は、設定数の合計が 255 個以下で使用してください。

(12) RUN-TIME表示における注意事項

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

(13) Timeout error表示時の注意事項

Timeout error が表示された場合、E10A-USB エミュレータとターゲットマイコンの通信が取れなくなっています。
この場合、一旦ユーザシステムの電源を OFF にし E10A-USB エミュレータの USB コネクタを接続しなおして、High-performance Embedded Workshop から E10A-USB エミュレータを再接続してください。

(14) [Run Program]ダイアログボックスご使用時の注意事項
[デバッグ]メニュー -> [条件を指定して実行]を選択して停止アドレスを指定する際に
以下の注意事項があります。
Disable に設定しているブレークポイントを停止アドレスと設定した場合、
ユーザプログラム停止時にブレークポイントが Enable になりますのでご了承願います。

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

参考値として、以下の環境でのユーザプログラムの停止期間を示します。
環境
ホストコンピュータ：Pentium®4 3 GHz
SH7265：システムクロック周波数 66.6 MHz
JTAG クロック：2.5 MHz
コマンドラインウィンドウから 1 バイトメモリリードを行った場合、停止時間は約 70 ms
になります。

(16) SLEEP命令へのBREAKPOINT設定
SLEEP 命令に Break を設定する場合、BREAKPOINT は使用しないでください。
Break Condition を使用してください。

(17) [Configuration]ダイアログボックスのセッションセーブに関する注意事項
次の設定は、セッションには保存しません。
[General]ページの JTAG クロックの設定
[Loading flash memory]ページの Loading flash memory の設定

(18) ユーザプログラム実行中のウィンドウスクロール
ユーザプログラム実行中に、[メモリ]ウィンドウと[逆アセンブリ]ウィンドウを
スクロールボックスのドラッグにより、スクロールしないでください。
スクロールボックスのドラッグにより、大量のメモリリードが発生し、メモリリード完了
までユーザプログラムの実行が停止します。

(19) memory test機能

[メモリ]メニューから[テスト…]を選択することによって使用する memory test 機能は、本製品ではサポートしておりません。

(20) Flash Memory書き込み中のメモリアクセス

ユーザプログラムの実行などの Flash Memory 書き込み中にメモリウィンドウを開くなどのメモリアクセス動作はできません。
このとき表示される値はダミー値で正しい値ではありません。
Flash Memory 書き込み終了後、再度メモリアクセスを行ってください。

(21) エミュレータ使用時の休止状態について

E10A-USB エミュレータ使用中は、PC をスリープモード、休止状態にしないでください。
スリープモード、休止状態にした場合、E10A-USB エミュレータは使用不可となります。
スリープモード、休止状態からの復帰後はエミュレータを再接続しなおしてください。

(22) Manual Navigator について

Windows Vista®、Windows® 7 でプログラムを実行する場合、下記の操作を行ってください。
回避方法：

1. 管理者権限でログインします。
2. High-performance Embedded Workshop のインストールフォルダ下の Manuals フォルダ内にある man_navi.exe のプロパティを開きます。
3. 「互換性」タブで「管理者としてこのプログラムを実行する」をチェックします。

【注】 64 ビット版の Windows Vista®には対応していません

（23）パラレルモードの注意事項

同期デバッグにてパラレルモードを使用した場合、エミュレータが接続を完了する前の CPU に対してウィンドウの表示や操作を行わないでください。

付録 E ハードウェア診断プログラムについて

E10A-USB エミュレータ用テストプログラムによる故障解析の手順につきましては、“E10A-USB Emulator for Multi-core Microcomputers”の CD 内に含まれる” E10A-USB エミュレータ用テストプログラムマニュアル”(ファイル名: E10A-USBMTMJ.pdf)をご覧ください。

付録 F 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、E10A-USB エミュレータ（HS0005KCU04H、HS0005KCU14H）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に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	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	(1) 本体側のシステム構成 ・E10A-USB エミュレータ(HS0005KCU04H、HS0005KCU14H) シリアル No._____レビジョン_____ （筐体裏面の製品管理シールに表示しています： シリアル No.は数字 5 桁、レビジョンはそれに続くアルファベットです） ・付属 CD-R (HS0005KCU04SR)　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 メーカ名_____ 型式_____ 使用 OS{ Windows®XP, Windows Vista®, Windows®7 }
4	エラー発生時の 設定内容	(1) デバイス：型名_____ (2) 動作周波数：_____MHz
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 Link up しない	エラーメッセージ内容

上記以外のエラーについては、下記に症状を記載いただくようお願いいたします。

SuperH™ファミリ マルチコア マイコン用
E10A-USB エミュレータ ユーザーズマニュアル

発行年月日　2016年3月25日　Rev.5.01
発行　　ルネサス エレクトロニクス株式会社
　　〒135-0061　東京都江東区豊洲3-2-24 (豊洲フォレシア)

ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

http://www.renesas.com

※営業お問合せ窓口の住所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。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■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：http://japan.renesas.com/contact/

Colophon 3.0

SuperH™ファミリ マルチコア マイコン用
E10A-USB エミュレータ
ユーザーズマニュアル

ルネサス エレクトロニクス株式会社

R20UT0363JJ0501