

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

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

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

2010年4月1日  
ルネサスエレクトロニクス株式会社

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

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

## ご注意書き

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

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

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

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

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

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

# SH-4A、SH4AL-DSP用 E200Fエミュレータ

ユーザーズマニュアル

ルネサスマイクロコンピュータ開発環境システム

SH-4A、SH4AL-DSP E200F R0E0200F0EMU00J



## 本資料ご利用に際しての留意事項

1. 本資料は、お客様に用途に応じた適切な弊社製品をご購入いただくための参考資料であり、本資料中に記載の技術情報について弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的、あるいはその他軍用途の目的で使用しないでください。また、輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
4. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりましては、事前に弊社営業窓口で最新の情報をご確認頂きますとともに、弊社ホームページ(<http://www.renesas.com>)などを通じて公開される情報に常にご注意下さい。
5. 本資料に記載した情報は、正確を期すため慎重に制作したのですが、万一本資料の記述の誤りに起因する損害がお客様に生じた場合においても、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断して下さい。弊社は、適用可否に対する責任を負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会下さい。なお、上記用途に使用されたことにより発生した損害等について弊社はその責任を負いかねますのでご了承願います。
8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないで下さい。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
  - 1) 生命維持装置。
  - 2) 人体に埋め込み使用するもの。
  - 3) 治療行為（患部切り出し、薬剤投与等）を行なうもの。
  - 4) その他、直接人命に影響を与えるもの。
9. 本資料に記載された製品のご使用につき、特に最大定格、動作電源電圧範囲、放熱特性、実装条件およびその他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて製品をご使用された場合の故障および事故につきましては、弊社はその責任を負いません。
10. 弊社は製品の品質および信頼性の向上に努めておりますが、特に半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。弊社製品の故障または誤動作が生じた場合も人身事故、火災事故、社会的損害などを生じさせないよう、お客様の責任において冗長設計、延焼対策設計、誤動作防止設計などの安全設計（含むハードウェアおよびソフトウェア）およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特にマイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願い致します。
11. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなきよう、お客様の責任において十分な安全設計をお願い致します。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
12. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断り致します。
13. 本資料に関する詳細についてのお問い合わせ、その他お気付きの点等がございましたら弊社営業窓口までご照会下さい。



---

## 重要事項

---

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

### エミュレータとは：

ここで言うエミュレータとは、株式会社ルネサス テクノロジ（以下、「ルネサス」という）、株式会社ルネサス ソリューションズが製作した次の製品を指します。

- (1) エミュレータ本体、(2) トレースユニット、(3) 拡張プロファイルユニット、
- (4) アナライザユニット

お客様のユーザシステムおよびホストコンピュータは含みません。

### エミュレータの使用目的：

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

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

### エミュレータを使用する人は：

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

特に、本エミュレータを初めて使用する人は、本エミュレータをよく理解し、使い慣れている人から指導を受けることを強く薦めます。

本エミュレータを使用する上で、電気回路、論理回路およびマイクロコンピュータの基本的な知識が必要です。

エミュレータご利用に際して：

- (1) 本エミュレータは、プログラムの開発、評価段階に使用する開発支援装置です。開発の完了したプログラムを量産される場合には、必ず事前に実装評価、試験などにより、お客様の責任において適用可否を判断してください。(2) 本エミュレータを使用したことによるお客様での開発結果については、一切の責任を負いません。
- (3) 弊社は、本製品不具合に対する回避策の提示または、不具合改修などについて、有償もしくは無償の対応に努めます。ただし、いかなる場合でも回避策の提示または不具合改修を保証するものではありません。
- (4) 本エミュレータは、プログラムの開発、評価用に実験室での使用を想定して準備された製品です。国内の使用に際し、電気用品安全法及び電磁波障害対策の適用を受けておりません。
- (5) 本エミュレータは、ULなどの安全規格、IECなどの規格を取得しておりません。したがって、日本国内から海外に持ち出される場合は、この点をご承知おきください。
- (6) ルネサスは、潜在的な危険が存在するおそれのあるすべての起こりうる諸状況や誤使用を予見できません。したがって、このユーザーズマニュアルと本エミュレータに貼付されている警告がすべてではありません。お客様の責任で、本エミュレータを正しく安全に使用してください。

使用制限について：

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

- (1) 運輸、移動体用
- (2) 医療用（人命にかかわる装置用）
- (3) 航空宇宙用
- (4) 原子力制御用
- (5) 海底中継用

このような目的で本エミュレータの採用をお考えのお客様は、ルネサス テクノロジ、ルネサス ソリューションズ、ルネサス販売または特約店へ是非ご連絡頂きますようお願い致します。

製品の変更について：

ルネサスは、本エミュレータのデザイン、性能を絶えず改良する方針をとっています。

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

権利について：

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

---

# 安全事項

---

## 1. シグナル・ワードの定義

ユーザーズマニュアルおよびエミュレータへの表示では、エミュレータを正しくお使い頂き、あなたや他の人々への危害や財産への損害を未然に防止するために、いろいろな絵表示をしています。安全事項では、その絵表示と意味を示し、本エミュレータを安全に正しくご使用されるための注意事項を説明します。

ここに記載している内容をよく理解してからお使いください。



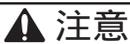
これは、安全警告記号です。潜在的に、人に危害を与える危険に対し注意を喚起するために用います。起こり得る危害又は死を回避するためにこの記号の後に続くすべての安全メッセージに従ってください。



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



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



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



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

注、重要は、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。

上記の4表示に加えて、適宜以下の表示を同時に示します。



表示は、警告・注意を示します。

例： **感電注意**



表示は禁止を示します。

例： **分解禁止**



表示は、強制・指示する内容を示します。

例： **電源プラグをコンセントから抜け**

## ⚠ 警告

### 電源に関して:



- 付属の AC 電源ケーブルがコンセントの形状に合わない場合、AC 電源ケーブルを改造したり、無理に入れるなどの行為は絶対に行わないでください。感電事故または火災の原因となります。
- 日本国外で使用する時は、その国の安全規格に適合している AC 電源ケーブルを使用してください。
- 安全規格に適合している AC 電源ケーブルが調達できない場合は、株式会社ルネサステクノロジ、株式会社ルネサスソリューションズまたは販売元までお問合せください。
- 濡れた手で AC 電源ケーブルのプラグに触れないでください。感電の原因となります。
- 本エミュレータはシグナルグランドとフレームグランドを接続しています。本エミュレータを用いて開発する製品がトランスレス(AC 電源に絶縁トランスを使用していない)製品である場合、感電する危険があります。また、本エミュレータと開発対象製品に修復不可能な損害を与える場合があります。開発中はこれらの危険性を回避するために開発対象製品の AC 電源は絶対トランスを経由して商用電源に接続してください。
- 本エミュレータと同じコンセントに他の装置を接続する場合は、電源電圧および電源電流が過負荷にならないようにしてください。電気定格は AC アダプタの銘板に示してあります。



- AC 電源ケーブルの接地端子は、必ずしっかりした接地接続を行なってください。



- 使用中に異臭・異音がしたり煙が出る場合は、直ちに電源を切り AC 電源ケーブルをコンセントから抜いてください。また、感電事故、または火災の原因になりますので、そのまま使用しないで、株式会社ルネサステクノロジ、株式会社ルネサスソリューションズまたは販売元までご連絡ください。
- 本エミュレータの設置や他の装置との接続時には、AC 電源を切るか AC 電源ケーブルを抜いて怪我や故障を防いでください。

### 本エミュレータの取り扱いに関して:



- 本エミュレータを分解または、改造しないでください。分解または改造された場合、感電などにより傷害を負う可能性があります。
- 通風口から水・金属片・可燃物などの異物を入れないでください。

### 設置に関して:



- 湿度の高いところおよび水等で濡れるところには設置しないでください。水等が内部にこぼれた場合、修理不能な故障の原因となります。

### 使用環境に関して:

- 本エミュレータ使用時の周辺温度の上限(最大定格周辺温度)は35 度です。この最大定格周囲温度を 越えないように注意してください。

## ⚠ 注意

AC アダプタに関して:



- エミュレータに付属のACアダプタ以外は使用しないでください。
- 付属のACアダプタは、本エミュレータ専用です。他の機器に使用しないでください。
- 本エミュレータ付属のACアダプタのDCプラグ極性を以下に示します。



- 付属のACアダプタには電源スイッチがありません。ACアダプタはAC電源ケーブル接続状態では常に動作状態です。電源供給状態はACアダプタのLED点灯にてご確認ください。

本製品の取り扱いに関して:



- 本エミュレータは慎重に扱い、落下・倒れ等による強い衝撃を与えないでください。
- 通信インタフェースケーブルやエミュレーションプローブ部接続用フレキシブルケーブルで本エミュレータを引っ張らないでください。また過度な曲げ方をしないでください。ケーブルが断線する恐れがあります。

設置に関して:



- 本エミュレータは縦置きに設置するように設計しています。横置きの状態で使用しないでください。

---

## はじめに

---

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

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

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

## このマニュアルについて

本マニュアルでは、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グ機能については、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。

Microsoft®, Windows® は米国 Microsoft 社の米国およびその他の国における登録商標です。

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

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

その他、記載されている製品名は各社の商標または登録商標です。

## このマニュアルの記号

このマニュアルで使われている記号の意味を説明します。

表 1: 記号一覧

記号	意味
[Menu->Menu Option]	太字と '->' はメニューオプションを示します (例 [File->Save As...])
FILENAME.C	大文字の名前はファイル名を示します
“ <u>文字列の入力</u> ”	下線は入力する文字列を示します (“” を省く)
Key + Key	キー入力を示します。例えば、CTRL+N キーでは CTRL キーと N キーを同時に押します
☞ (「操作方法」マーク)	このマークが左端にあるとき、その右の文章は何かの操作方法を示します



---

## ユーザ登録について

---

ご購入頂いた際には WEB でのユーザ登録をお願いします。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。

なお、登録なき場合は、フィールドチェンジ、不具合情報の連絡等の保守サービスが受けられなくなりますので、必ずご登録頂きますようお願いいたします。

<http://tool-support.renesas.com/jpn/toolnews/registration/index.html>

上記アドレスにアクセスをお願いいたします。

(ユーザ登録に関するお問合せ先：[regist\\_tool@renesas.com](mailto:regist_tool@renesas.com))



---

# 目次

---

1.	製品概要	1
1.1	梱包内容	5
1.2	ハードウェア構成	5
1.3	エミュレータ機能	17
1.3.1	機能概要	17
1.3.2	イベント検出機能	20
1.3.3	トレース機能	23
1.3.4	ブレーク機能	31
1.3.5	プローブ機能	32
1.3.6	周辺 I/O アナライザ機能	33
1.3.7	パフォーマンス測定機能	33
1.3.8	カバレッジ機能	45
1.3.9	メモリアクセス機能	45
1.3.10	スタックトレース機能	47
1.3.11	ブレーク中のユーザ割込開放機能	47
1.3.12	オンラインヘルプ	47
1.4	使用環境条件	48
2.	セットアップ	49
2.1	エミュレータ使用までのフローチャート	49
2.2	デバッグのインストール	50
2.2.1	CD-R について	50
2.3	エミュレータ本体への接続	50
2.3.1	E200F トレースユニットとユーザシステムとの接続	51
2.3.2	E200F 周辺 I/O アナライザユニットとユーザシステムとの接続	54
2.3.3	E200F 拡張プロファイルユニットの接続	55
2.4	AC アダプタの接続	61
2.5	ホストマシンとの接続	62
2.6	ユーザシステムとの接続	64
2.6.1	E200F H-UDI/AUD プローブとユーザシステムとの接続	65
2.6.2	システムグラウンド系の接続	68
2.7	設定の変更	69
2.7.1	E200F メインユニットのみで実現できる機能の変更方法	70
2.7.2	外部バストレースユニット使用時の機能変更方法	71
2.7.3	拡張プロファイルユニット使用時の機能変更方法	72
2.7.4	モニタ機能の変更	73

3.	ハードウェア仕様	75
3.1	仕様一覧	75
3.2	E200F エミュレータ内インタフェース回路	76
3.3	使用上の注意事項	80
3.3.1	EMI ノイズの最小化	80
3.4	故障解析	81
3.4.1	故障診断プログラムをインストールする	81
3.4.2	診断プログラムを実行する	81
3.4.3	ログファイルを作成する	84
4.	デバッグの準備をする	87
4.1	システムチェック	87
4.2	High-performance Embedded Workshop の起動方法	95
4.2.1	新規にワークスペースを作成する場合(ツールチェイン未使用)	96
4.2.2	新規にワークスペースを作成する場合(ツールチェイン使用)	100
4.2.3	既存のワークスペースを指定する場合	105
4.3	エミュレータ起動時の設定	106
4.4	デバッグセッション	108
4.4.1	セッションを選択する	108
4.4.2	セッションの追加と削除	110
4.4.3	セッション情報を保存する	114
4.5	エミュレータの接続	115
4.6	エミュレータの再接続	117
4.7	エミュレータの終了	117
4.8	アンインストール	118
5.	デバッグ	123
5.1	エミュレーション環境を設定する	123
5.1.1	[Configuration]ダイアログボックスを開く	123
5.1.2	[General]ページ	123
5.1.3	[Main Board]ページ	127
5.1.4	[Bus Board]ページ	129
5.1.5	[Option Board]ページ	132
5.1.6	フラッシュメモリへダウンロードする	133
5.1.7	Memory Mapping ダイアログボックスを開く	135
5.1.8	メモリマップ設定を変更する	136
5.2	プログラムをダウンロードする	137
5.2.1	プログラムをダウンロードする	137
5.2.2	ソースコードを表示する	138
5.2.3	アセンブリ言語コードを表示する	142
5.2.4	アセンブリ言語コードを修正する	143
5.2.5	特定のアドレスを見る	144
5.2.6	現在のプログラムカウンタアドレスを見る	144
5.3	リアルタイムにメモリ内容を表示する	145

5.3.1	[モニタ]ウィンドウを開く.....	145
5.3.2	モニタの設定内容を変更する.....	147
5.3.3	モニタの更新を一時的に停止する.....	147
5.3.4	モニタ設定を削除する.....	147
5.3.5	変数の内容をモニタする.....	148
5.3.6	[モニタ]ウィンドウを非表示にする.....	150
5.3.7	[モニタ]ウィンドウを管理する.....	150
5.4	現在の状態を表示する.....	151
5.5	エミュレータの情報を定期的に読み出し表示する.....	152
5.5.1	[Extended Monitor]ウィンドウを開く.....	152
5.5.2	表示項目を選択する.....	153
5.6	イベントポイントを使用する.....	154
5.6.1	PC ブレークポイントとは.....	154
5.6.2	イベントポイントとは.....	154
5.6.3	[Event]ウィンドウを開く.....	156
5.6.4	PC ブレークポイントを設定する.....	157
5.6.5	Onchip Eventpoint を設定する.....	159
5.6.6	AUD Eventpoint を設定する.....	167
5.6.7	BUS Eventpoint を設定する.....	180
5.6.8	Other Eventpoint を設定する.....	189
5.6.9	ブレークポイント/イベントポイントの編集.....	194
5.6.10	ブレークポイント/イベントポイントを有効にする.....	194
5.6.11	ブレークポイント/イベントポイントを無効にする.....	194
5.6.12	ブレークポイント/イベントポイントを削除する.....	194
5.6.13	ブレークポイント/イベントポイントをすべて削除する.....	194
5.6.14	ブレークポイント/イベントポイントのソース行を表示する.....	194
5.7	トレース情報を見る.....	195
5.7.1	[Internal/AUD/Usermemory trace]ウィンドウを開く.....	195
5.7.2	[BUS/MFI trace]ウィンドウを開く.....	201
5.7.3	Trace カラムを非表示にする.....	205
5.7.4	Trace レコードを検索する.....	205
5.7.5	トレース情報をクリアする.....	210
5.7.6	トレース情報をファイルに保存する.....	210
5.7.7	[エディタ]ウィンドウを表示する.....	210
5.7.8	ソース表示を整形する.....	210
5.7.9	トレース情報の取得を一時的に停止する.....	210
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.....	211
5.7.11	統計情報を解析する.....	217
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.....	218
5.8	パフォーマンスを測定する.....	219
5.8.1	[Onchip Performance Analysis]ウィンドウを開く.....	219
5.8.2	[AUD Performance Analysis]ウィンドウを開く.....	221
5.8.3	[BUS Performance Analysis]ウィンドウを開く.....	224
5.8.4	カラムを非表示にする.....	226
5.8.5	実行効率測定を開始する.....	226
5.8.6	測定条件を削除する.....	226

5.8.7	すべての測定条件を削除する .....	226
5.9	プロファイル情報を見る .....	227
5.9.1	スタック情報ファイル .....	227
5.9.2	プロファイル情報ファイル .....	229
5.9.3	スタック情報ファイルのロード .....	230
5.9.4	プロファイルを有効にする .....	231
5.9.5	測定方法を指定する .....	231
5.9.6	ユーザプログラムを実行し結果を確認する .....	231
5.9.7	[List]シート .....	232
5.9.8	[Tree]シート .....	233
5.9.9	プロファイル - チャートウィンドウ .....	235
5.9.10	表示データの種類および用途 .....	236
5.9.11	プロファイル情報ファイルを作成する .....	237
5.9.12	注意事項 .....	238
5.10	リアルタイムプロファイル情報を見る .....	240
5.10.1	[リアルタイムプロファイル]ウィンドウを開く .....	246
5.10.2	測定範囲を指定する .....	247
5.10.3	測定を開始する .....	248
5.10.4	測定結果をクリアする .....	248
5.10.5	測定範囲を削除する .....	248
5.10.6	測定時間の最小単位を設定する .....	248
5.11	コードカバレッジを測定する .....	250
5.11.1	コードカバレッジウィンドウを開く .....	251
5.11.2	ソースファイルを表示する .....	255
5.11.3	表示アドレスを変更する .....	255
5.11.4	カバレッジ表示範囲を変更する .....	256
5.11.5	カバレッジ情報をクリアする .....	257
5.11.6	カバレッジ情報をファイルに保存する .....	257
5.11.7	カバレッジ情報をファイルからロードする .....	258
5.11.8	最新の情報に更新する .....	258
5.11.9	情報の更新を抑止する .....	258
5.11.10	保存確認ダイアログボックス .....	259
5.11.11	エディタウィンドウへのコードカバレッジ結果表示 .....	260
5.12	複数デバッグングプラットフォームを同期動作させる .....	261
5.12.1	2台のエミュレータを区別する方法 .....	262
6.	チュートリアル .....	265
6.1	はじめに .....	265
6.2	High-performance Embedded Workshop の起動 .....	266
6.3	E200F エミュレータのセットアップ .....	266
6.4	[Configuration]ダイアログボックスの設定 .....	267
6.5	ダウンロード先メモリの動作チェック .....	268
6.6	チュートリアルプログラムのダウンロード .....	270
6.6.1	チュートリアルプログラムをダウンロードする .....	270
6.6.2	ソースプログラムを表示する .....	271

6.7	PC ブレークポイントの設定.....	272
6.8	レジスタ内容の変更.....	273
6.9	プログラムの実行.....	275
6.10	ブレークポイントの確認.....	278
6.11	シンボルの参照.....	279
6.12	メモリ内容の確認.....	280
6.13	変数の参照.....	281
6.14	ローカル変数の表示.....	284
6.15	プログラムのステップ実行.....	285
6.15.1	ステップインの実行.....	285
6.15.2	ステップアウトの実行.....	287
6.15.3	ステップオーバーの実行.....	288
6.16	プログラムの強制ブレーク.....	289
6.17	ブレーク機能.....	290
6.17.1	PC ブレーク機能.....	290
6.18	イベントポイントによるブレーク機能.....	294
6.18.1	Onchip Eventpoint によるブレーク.....	294
6.18.2	Sequential Onchip Eventpoint によるブレーク.....	298
6.19	トレース機能.....	302
6.19.1	[Internal/AUD/Usermemory trace]ウィンドウの表示方法.....	304
6.19.2	[BUS/MFI trace]ウィンドウの表示方法.....	317
6.20	MMU サポート.....	321
6.21	スタックトレース機能.....	324
6.22	フラッシュメモリへのダウンロード機能.....	326
6.23	さてつぎは？.....	331
7.	トラブルシューティング.....	333
8.	保守と保証.....	335
8.1	ユーザ登録.....	335
8.2	保守.....	335
8.3	保証内容.....	335
8.4	修理規定.....	336
8.5	修理依頼方法.....	336

付録 A	メニュー一覧	339
付録 B	コマンドライン機能	345
付録 C	High-performance Embedded Workshop の注意事項	347
付録 D	故障症状調査書	351

# 1. 製品概要

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、測定、および変更に関して、High-performance Embedded Workshop は高性能でしかも直観的な手段を提供することを目的としています。

本システムは、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。

E200F 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。図 1.1 に E200F エミュレータ (型名: R0E0200F0EMU00) を使用したシステム構成外観図、図 1.2 に E200F エミュレータ (型名: R0E0200F2EMU00) を使用したシステム構成外観図を示します。

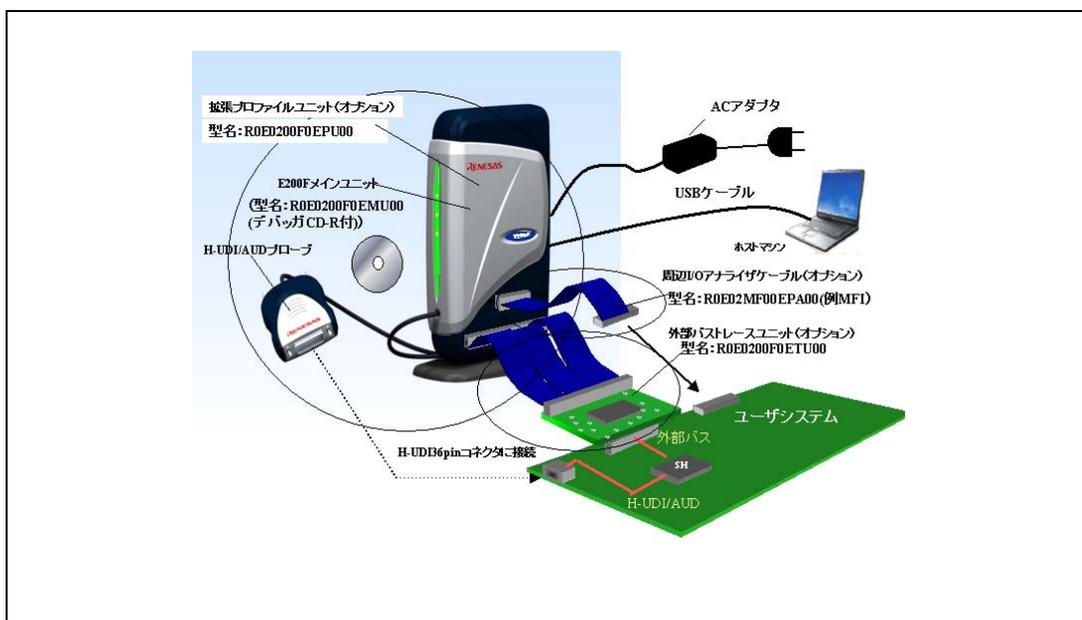


図 1.1 E200F エミュレータ(R0E0200F0EMU00)を使用したシステム構成外観

【注】 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。

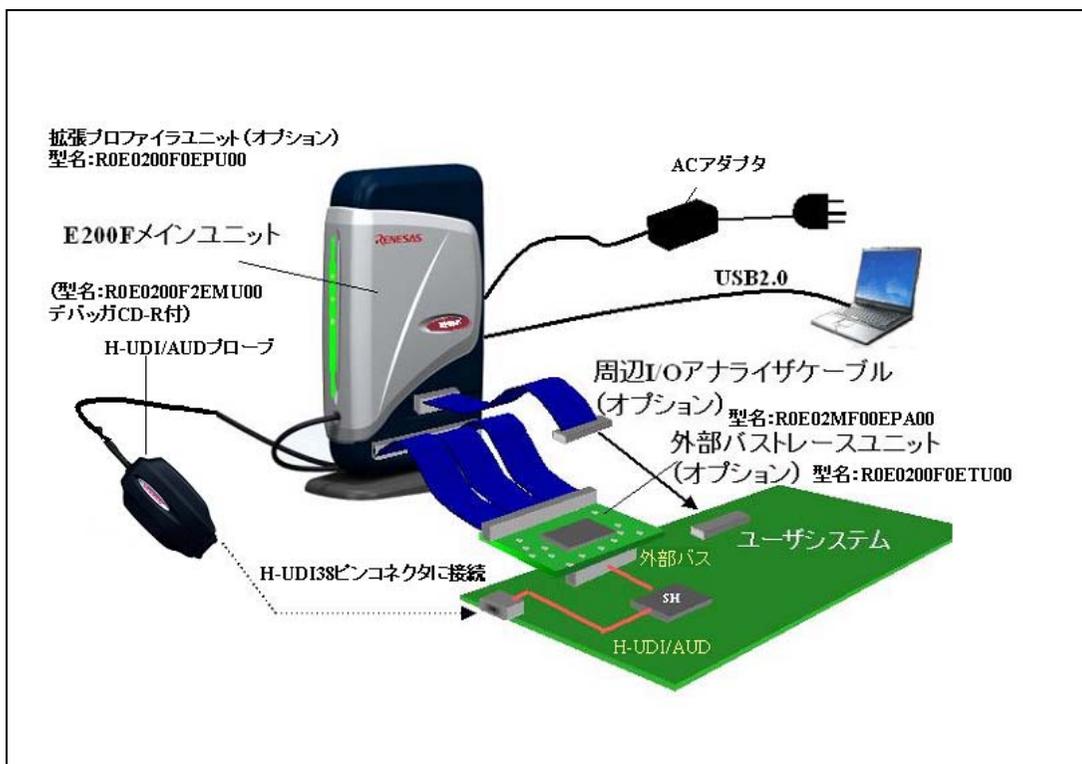


図 1.2 E200F エミュレータ(R0E0200F2EMU00)を使用したシステム構成外観

E200F エミュレータの特長は、以下の通りです。

(1) 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、トレース情報をウィンドウに表示することができます。また、パフォーマンス、プロファイラ機能など、豊富なエミュレーション機能を備えています。

USB2.0 サポートにより高速ダウンロードを実現しました。

周辺 I/O モジュールのアナライザ機能を備えています。サポートしている周辺 I/O は、デバイスによって異なります。

デバッグフェーズごとにエミュレータの機能を入れ替えることが可能です。

豊富なコマンドライン機能を備えています。

(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

Microsoft® Windows® 2000、Microsoft® Windows® XP 環境下で動作する 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) 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。

(5) コンパクトなデバッグ環境

ノート型パソコンをホストマシンとして使用でき、場所を選ばずデバッグ環境を作成することができます。

**⚠ 注意**

E200F 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、E200F 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。「1.4 使用環境条件」を参照してください。
- (4) ホストマシン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、「2 セットアップ」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。

## 1.1 梱包内容

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。E200F 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、ルネサス販売または特約店、ルネサス テクノロジ コンタクトセンタ (csc@renesas.com) までご連絡ください。

## 1.2 ハードウェア構成

E200F エミュレータ (型名: R0E0200F0EMU00) は、図 1.2 に示すように E200F メインユニット、トレースユニット (オプション)、拡張プロファイルユニット (オプション)、周辺 I/O アナライザユニット (オプション)、USB ケーブル、AC アダプタ、外部プローブで構成され、ホストマシンとは USB 2.0 で接続できます。図 1.4 に E200F エミュレータ (型名: R0E0200F2EMU00) のハードウェア構成を示します。



図 1.3 E200F エミュレータ (型名: R0E0200F0EMU00) のハードウェア構成

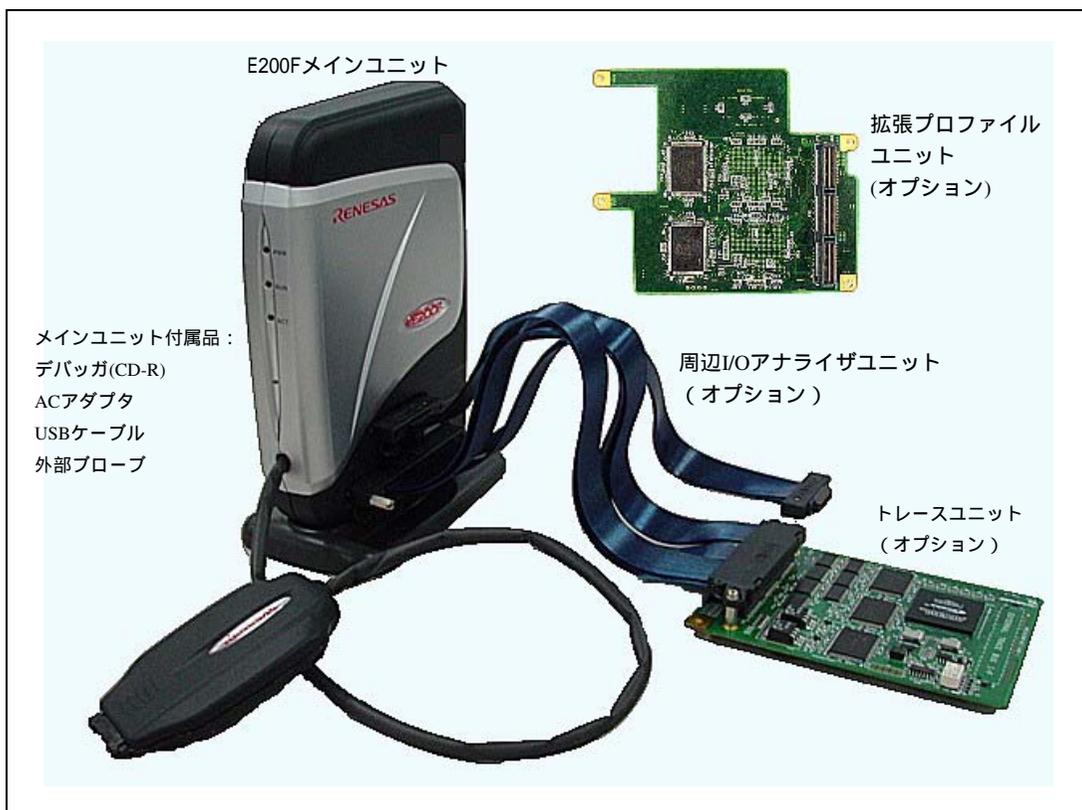


図 1.4 E200F エミュレータ (型名 : R0E0200F2EMU00) のハードウェア構成

E200F エミュレータにおける各部の名称を下記に示します。

(1) 正面部の構成



図 1.5 E200F エミュレータの正面部

- |                   |   |  |
|-------------------|---|--|
| (a) POWER LED 表示  | : | "PWR"と表示してある LED です。点灯時は電源が入っていることを示しています。                   |
| (b) RUN LED 表示    | : | "RUN"と表示してある LED です。点灯時はユーザプログラムが動作していることを示しています。            |
| (c) ACTION LED 表示 | : | "ACT"と表示してある LED です。点灯時は E200F エミュレータが何らかの処理を行っていることを示しています。 |

(2) 背面の構成



図 1.6 E200F エミュレータのホスト側 側面

- (a) 電源スイッチ : “POWER”と表示しています。電源スイッチの “I” で電源が ON、“O” で電源が OFF します。
- (b) DC プラグ : “DC IN”と表示しています。AC アダプタの DC (+12V) 入力用コネクタです。必ず付属品の AC アダプタを接続してください。
- (c) 外部プローブコネクタ : “EXT”と表示しています。外部プローブ接続用コネクタです。必ず付属品の外部プローブを接続してください。
- (d) ホスト側接続用コネクタ (USB コネクタ) : と表示しています。ホストマシン接続用コネクタ(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) 右側面の構成



図 1.7 E200F エミュレータの右側面

- (a) 識別プレート : 他の E200F メインユニットと簡単に識別できるように、ネイビーブルー (R0E0200F0EMU00)、シルバー & レッド (R0E0200F2EMU00) の E200F 専用プレートが張付けられています。
- (b) 周辺 I/O アナライザユニット接続コネクタ : “ANALYZER I/F”と表示しています。周辺 I/O アナライザユニット接続用コネクタです。必ずオプション製品の周辺 I/O アナライザユニットに付属している専用のアナライザケーブルを接続してください。
- (c) トレースユニット接続コネクタ : “TRACE I/F”と表示しています。トレースユニット接続用コネクタです。必ずオプション製品のトレースユニットに付属している専用のトレースケーブルを接続してください。

## 1 製品概要

---

### (4) 左側面の構成



図 1.8 E200F エミュレータの左側面

- (a) 製品管理シール : E200F 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。

## (5) 先端プローブ上面の構成 (型名: R0E0200F0EMU00)



図 1.9 E200F 先端プローブの上面

(a) 先端プローブ固定用ネジ : ユーザシステムと先端プローブを固定するためのネジです。

## (6) 先端プローブ正面の構成 (型名: R0E0200F0EMU00)

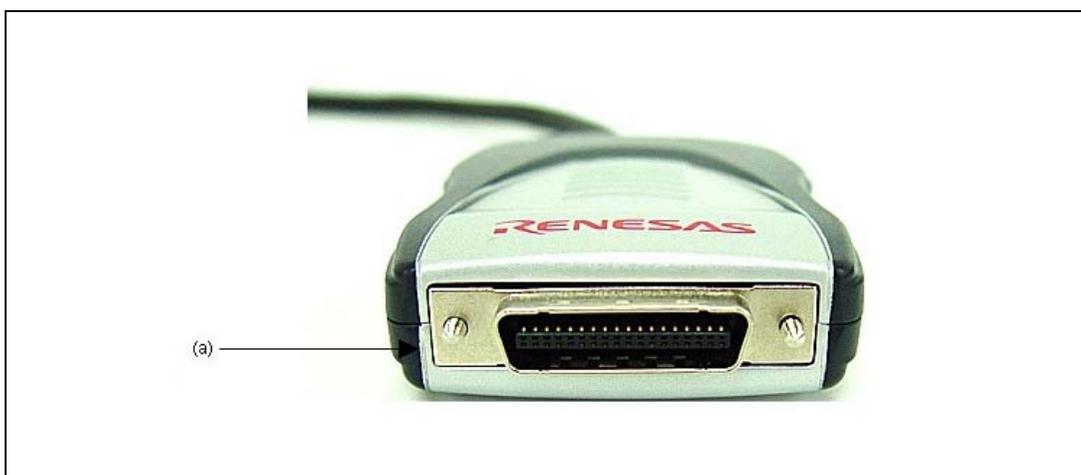


図 1.10 E200F 先端プローブの正面

(a) ユーザシステム接続用 H-UDI ポートコネクタ : オンチップデバッガとして使用する場合のユーザシステム接続用コネクタです。  
(36ピン E10A-USB 同一)

## 1 製品概要

---

### (7) 38ピン先端プローブ上面の構成 (型名: R0E0200F2EMU00)



図 1.11 38ピン先端プローブの上面 (型名: R0E0200F2EMU00)

### (8) 38ピン先端プローブ正面の構成 (型名: R0E0200F2EMU00)



図 1.12 38ピン先端プローブの正面 (型名: R0E0200F2EMU00)

(a) ユーザシステム接続用  
H-UDI ポートコネクタ (38ピン)

: オンチップデバッグとして使用する場合のユーザシステム接続用コネクタで  
す。(38ピン E10A-USB 同一)

(9) 先端プローブの収納

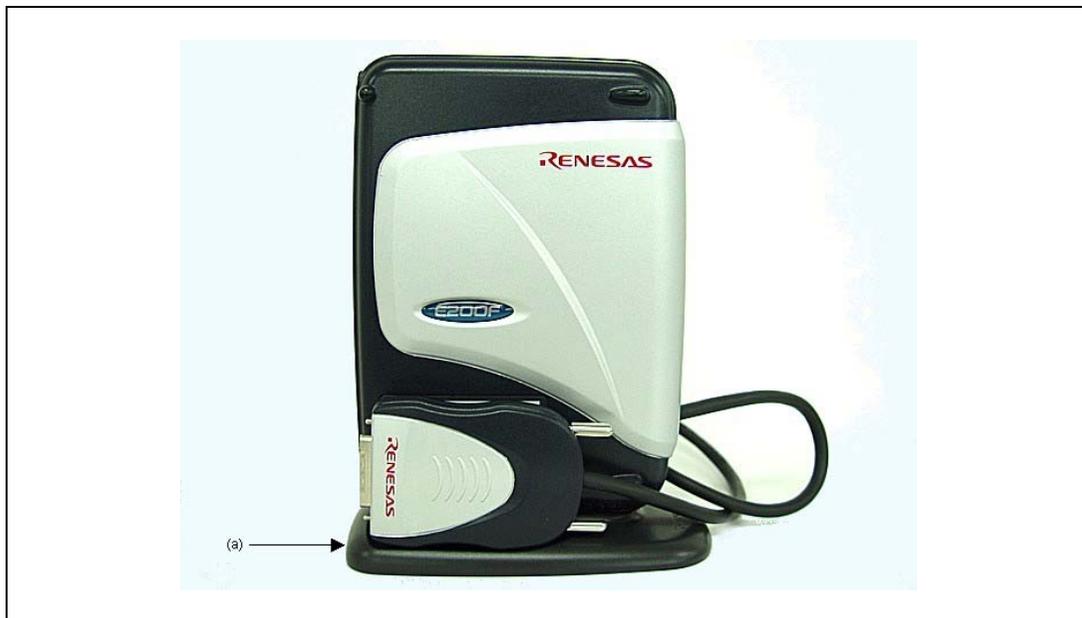


図 1.13 E200F 先端プローブの収納 ( 型名 : R0E0200F0EMU00 )

(a) 縦置き用ベース部 : 縦置きで使用する場合のベース筐体です。エミュレータ未使用時に先端プローブを収納できます。

(10) トレースユニット (オプション) 上面の構成

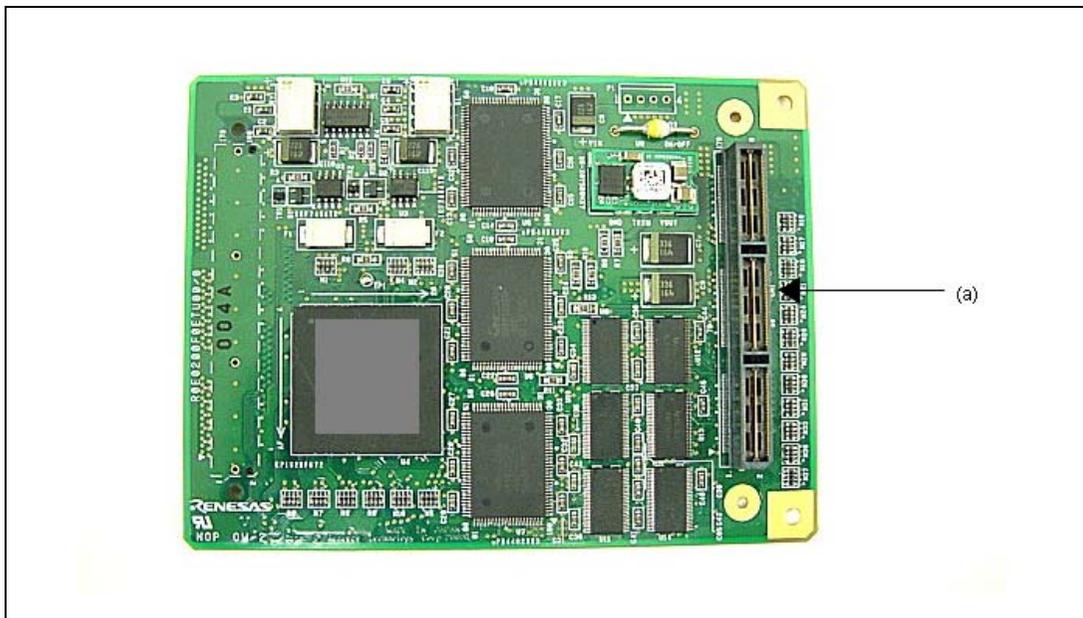


図 1.14 E200F トレースユニット (オプション) の上面

(a) トレースケーブル用コネクタ : トレースボード側のトレースケーブル接続用コネクタです。必ず付属品のトレースケーブルを使用してください。

## (11) トレースユニット (オプション) 裏面の構成

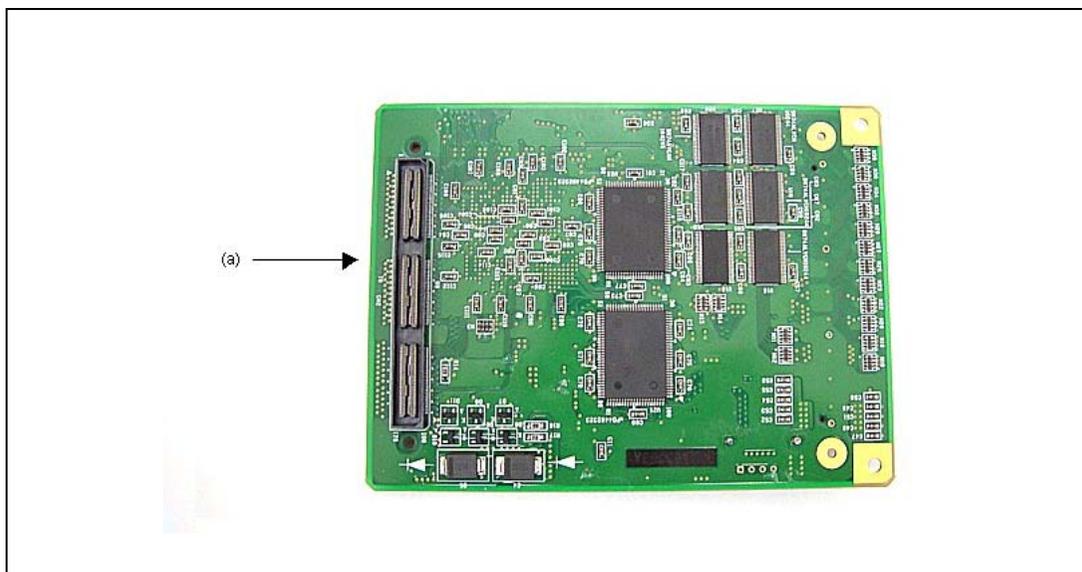


図 1.15 E200F トレースユニット (オプション) の裏面

- (a) 外部バストレース用ユーザ  
インタフェースコネクタ : 外部バストレースのユーザインタフェース用コネクタです。ユーザシステム上に  
実装した専用コネクタと接続します。

(12) 拡張プロフィールユニット (オプション) 裏面の構成

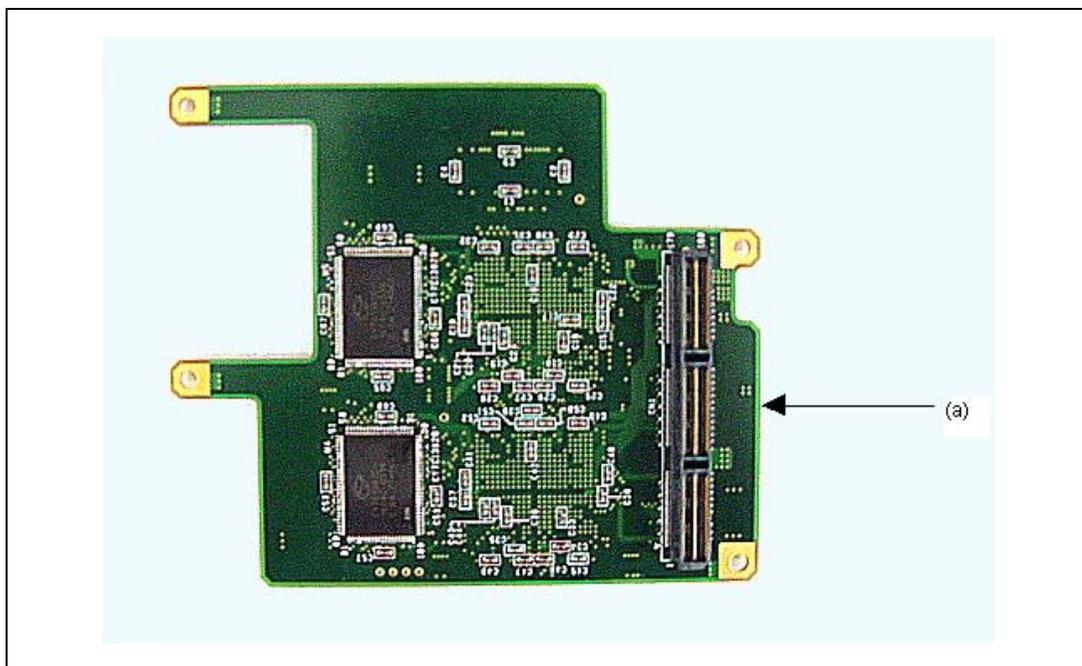


図 1.16 E200F 拡張プロフィールユニット (オプション) の裏面

(a) 拡張プロフィールユニット側  
オプションコネクタ : 拡張プロフィールユニットとメインユニットの接続インタフェース用コネクタ  
です。メインユニットのオプション用コネクタと接続します。

## 1.3 エミュレータ機能

本章では、E200F エミュレータの機能を紹介します。

E200F 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、「5 デバッグ」または「6 チュートリアル」をご参照ください。

### 1.3.1 機能概要

E200F エミュレータの機能概要を表 1.1 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 1.1 E200F 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	<ul style="list-style-type: none"> <li>• 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</li> <li>• リセットエミュレーション</li> <li>• Step 機能               <ul style="list-style-type: none"> <li>シングル Step 機能 (1Step : 1 命令)</li> <li>ソースレベル Step 機能 (1Step : ソース 1 行)</li> </ul> </li> <li>Step Over 機能 ( サブルーチン内はブレイクしない) 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)</li> </ul>
2	リセット機能	<ul style="list-style-type: none"> <li>• ブレイク中、High-performance Embedded Workshop からデバイスへパワーオンリセット発行</li> </ul>
3	イベント検出機能	<ul style="list-style-type: none"> <li>• オンチップイベント検出チャンネル (13 個)</li> <li>• AUD イベント検出チャンネル (8 個)</li> <li>• 外部バスイベント検出チャンネル (6 個)</li> <li>• その他のイベント検出               <ul style="list-style-type: none"> <li>実行時間イベント検出 (1 個)</li> <li>外部プローブイベント検出 (1 個)</li> </ul> </li> <li>その他、I/O アナライズ結果によるイベント検出機能</li> </ul>
4	トレース機能	<ul style="list-style-type: none"> <li>• オンチップイベント検出による内蔵トレース機能</li> <li>• AUD イベント検出による AUD トレース機能</li> <li>• 内蔵トレースデータのメモリ出力機能</li> <li>外部バスイベント検出による外部バストレース機能</li> <li>• その他、I/O アナライザによるトレース取得機能</li> </ul>

## 1 製品概要

5	ブ레이크機能	<ul style="list-style-type: none"> <li>• PC ブ레이크ポイント条件成立によりブ레이크する機能（1000 箇所）</li> <li>• オンチップイベント検出によりブ레이크する機能</li> <li>• AUD イベント検出によりブ레이크する機能</li> <li>• 外部バスイベント検出によりブ레이크する機能</li> <li>• その他のイベント検出によりブ레이크する機能</li> <li>• トレースバッファのオーバフローによりブ레이크する機能</li> <li>• 強制ブ레이크機能</li> </ul>
6	パフォーマンス測定機能	<ul style="list-style-type: none"> <li>• 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</li> <li>• AUD イベントチャンネルにより、Point to Point, Point to Range, Range to Range の実行時間、実行回数などを測定する機能</li> <li>• 外部バスのパフォーマンス機能により、Range, Point to Point, Range to Range の実行時間、実行回数などを測定する機能</li> <li>• 関数ごとに経過サイクル数などを測定し、Go 終了時に一覧表示する機能</li> <li>• ユーザが指定したアドレス範囲内の関数について、実行時間、実行回数などを測定し、Go 終了時に一覧表示する機能</li> </ul>
7	周辺 I/O アナライザ	特定の周辺 I/O について、発生した事象をトレース取得、またはモニタする機能。どの周辺 I/O をサポートしているかはデバイスによって異なります。本機能については、「SHxxxx ご使用時の補足説明」を参照してください。
8	メモリアクセス機能	<ul style="list-style-type: none"> <li>• RAM へのダウンロード</li> <li>• フラッシュメモリへのダウンロード</li> <li>• 1 行アセンブル</li> <li>• 逆アセンブル</li> <li>• メモリリード</li> <li>• メモリライト</li> <li>• ユーザプログラム実行中の変数内容の表示自動更新</li> <li>• FILL</li> <li>• サーチ</li> <li>• ムーブ</li> <li>• コピー</li> <li>• モニタ機能（物理論理アドレス）</li> <li>• 貸し出しメモリ機能</li> </ul>
9	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
10	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
11	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。
12	カバレッジ機能	命令実行情報の表示
13	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。

14	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.4.00 ユーザーズマニュアル「リファレンス 5 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はコア名を示します)

<High-performance Embedded Workshop フォルダ>

¥Tools¥Renesas¥DebugComp¥Platform¥E200F¥xxxx¥IOFiles

次の章から、E200F エミュレータの特徴的な機能についてご説明します。

### 1.3.2 イベント検出機能

E200F 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、より複雑なイベント検出機能を持っています。

#### (1) イベントとは

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

E200F エミュレータは、アドレス条件、データ条件、アクセス条件など指定された条件の組み合わせをイベントポイントとして定義できます。イベントポイント条件が成立したとき、イベントが発生します。E200F エミュレータのイベント検出機能は発生したイベントを検出し、ブレーク動作、トレース動作、およびパフォーマンス測定開始/終了動作を制御できます。

#### (2) イベントの種類

E200F エミュレータには、4 種類のイベントがあります。

##### (a) オンチップイベント (Onchip Event)

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。CPU 内部の各種情報によりイベントポイントを設定します。イベント検出チャンネルは 13 個です。イベントポイントは以下の項目の組合せで定義できます。

- アドレス条件
- データ条件
- ASID条件
- バスステート条件
- メモリアクセス先のアドレス条件
- システムバス条件
- LDTLB命令実行
- 分岐条件
- ソフトウェアトレース条件【注】
- イベント発生回数条件

【注】 Trace()関数実行時の PC、指定汎用レジスタをトレース取得します。以降、本機能をソフトウェアトレース機能と呼びます。

イベントが検出されたときの動作として、ブレーク、内蔵トレース取得/取得開始/取得停止、内蔵パフォーマンス測定開始/終了を指定できます。

本機能は、[Event]ウィンドウの[Onchip Event]シートで設定できます。

(b) AUD イベント (AUD Event)

AUD インタフェースから出力されたトレース情報によりイベントポイントを設定します。イベント検出チャンネルは 8 個です。イベントポイントは以下の項目の組み合わせで定義できます。

- 分岐トレースデータ条件
- ウィンドウトレースデータ条件
- システムバストレーサデータ条件
- ソフトウェアトレースデータ条件
- イベント発生回数条件
- イベント発生後の遅延条件

イベント検出されたときの動作として、ブレイク、AUD トレーサ取得/取得開始/取得停止、AUD パフォーマンス測定開始/終了を指定できます。

本機能は、[Event]ウィンドウの[AUD Event]シートで設定できます。

【留意事項】

AUD イベント検出でブレイクする場合、検出からブレイクまで数サイクルの遅れがあります。ユーザシステムでの事象発生からブレイクまでの遅れが問題となるときは、オンチップイベント機能(Onchip Event)を使用してください。

AUD トレーサの取得モードが[Realtime trace]の場合、出力することができなかったデータについては比較できません。

(c) 外部バスイベント (BUS Event)

外部バスや割り込み端子など端子情報によりイベントポイントを設定します。イベント検出チャンネルは最大 6 個です。イベントポイントは以下の項目の組み合わせで定義できます。

- 外部アドレスバス条件
- 外部データバス条件
- 割り込み信号条件
- イベント発生回数条件
- イベント発生後の遅延条件

イベント検出されたときの動作として、ブレイク、外部バストレーサ取得/取得開始/取得停止を指定できます。

本機能は、[Event]ウィンドウの[BUS Event]シートで設定できます。

【留意事項】

1. 外部バスイベント検出でブレイクする場合、検出からブレイクまで数サイクルの遅れがあります。
2. 本機能はオプションです。トレースユニット (オプション) 使用時に有効となります。

### (d) その他のイベント (Other Event)

本機能は、[Event]ウィンドウの[Other Event]シートで設定できます。

- 実行時間イベント

指定した実行時間経過後ブレイクする機能です。イベント検出チャンネルは1個です。

- 外部プローブイベント

イベント検出チャンネルは1個です。プローブケーブル経由の4つの外部プローブ信号を条件としてイベントポイントを定義できます。

イベント検出されたときの動作として、ブレイク、AUDトレース取得/取得開始/取得停止を指定できます。

#### 【留意事項】

検出からブレイクまで数サイクルの遅れがあります。

- 周辺I/Oアナライザによるイベント

周辺I/Oアナライザ機能によっては、イベントを検出できます。

#### 【留意事項】

1. 検出からブレイクまで数サイクルの遅れがあります。
2. 本機能はオプションです。周辺 I/O アナライザユニット (オプション) 使用時に有効となります。

### 1.3.3 トレース機能

E200F エミュレータには大きく分けて2種類のトレース機能があります。

- CPU、システムバスなどの、デバイス内部の情報をトレース取得する機能
- 外部バスなど、デバイス外部の情報をトレース取得する機能

#### (1) トレース機能

##### (a) デバイス内部の情報をトレース取得する機能

[Internal/AUD/Usermemory trace]ウィンドウに、取得したトレース情報を表示します。

トレース取得できる情報は、[Event]ウィンドウの[Onchip Event]シートのチャンネルにおいて、Action 条件としてトレース取得を選択した事象となります。

トレース取得した情報を格納する場所としては、以下3つの方法があります。

- デバイスに内蔵されている小さなトレース専用メモリに格納する機能  
8情報を格納することができます。  
AUD端子をエミュレータに接続しておらず、またトレース用に使用できるメモリがない場合に有効です。  
本機能は、以降「内蔵トレース機能」または「Internal trace」と呼びます。
- AUD端子からリアルタイムに出力する機能  
AUD端子をエミュレータに接続している場合に便利な機能です。大容量のトレースを取得することができます。またE200Fエミュレータでは、AUD端子から出力された情報を元に、ブ레이크などを発生することができます。  
本機能は、以降「AUDトレース機能」と呼びます。  
トレース取得できる情報の数は、分岐元/分岐先の組を1個とすると最大262,144個です。
- 指定したメモリに格納する機能  
取得したトレース情報を、指定したメモリ範囲に格納することができます。  
指定したメモリ範囲にトレース情報を上書きするため、プログラム領域を指定しないでください。  
本機能は、以降「メモリ出力トレース機能」または「Usermemory trace」と呼びます。

## 1 製品概要

---

E200F エミュレータでは、AUD トレース機能を拡張しています。

拡張機能1：Timestamp情報を取得することができます。

拡張機能2：指定したAUDイベントポイント条件および外部ブローイベントポイント条件により、AUD トレースの取得/取得開始/取得終了を制御できます。イベントポイントの設定については、「5.6 イベントポイントを使用する」を参照してください。

### 【留意事項】

トレース取得できる内容は、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。

トレース取得できるデバイス内部の情報には以下 4 種類があります。

- 分岐情報
- 論理アドレスでの範囲内メモリアクセス情報
- ソフトウェアトレース
- システムバスアクセス情報

分岐トレース機能

分岐元、分岐先アドレスとそのソース、分岐種別、アクセスを行ったバスマスタの種別を表示します。

### 【設定方法】

[Event]ウィンドウのCh12(Branch)行をダブルクリックして開く[Branch trace]ダイアログボックスの[Branch trace ]ページ[Branch]グループボックス中のチェックボックスにチェックをつけることで取得する分岐条件が設定できます。

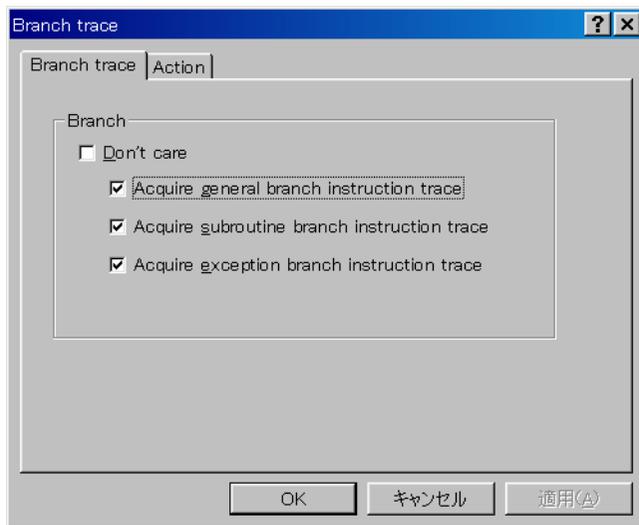


図 1.17 [Branch trace]ダイアログボックス

[Action]ページの[Acquire trace]チェックボックスにチェックをつけることによって分岐トレースが取得できます。

**【留意事項】**

設定を解除する場合は、Ch12(Branch)行を右クリックして開くポップアップメニューから削除を選択してください。

- 範囲内メモリアクセストレース機能

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。

それぞれトレース取得するバスの種類、ASID値、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。

**【設定方法】**

- (i) [Event]ウィンドウのCh5(OA)行またはCh6(OA)行をダブルクリックして[Event condition5]または[Event condition6]ダイアログボックスを開いてください。
- (ii) [Window address]ページの[Don't care]チェックボックスのチェックを外し設定するメモリ範囲を入力してください。

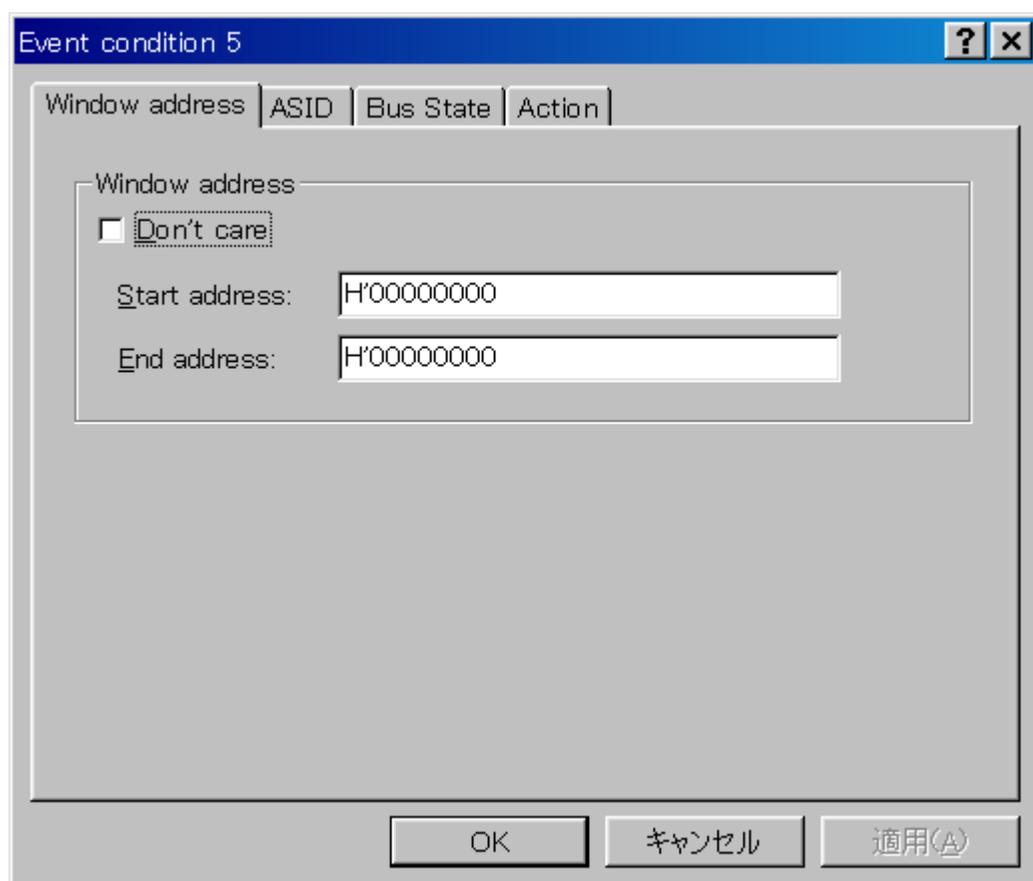


図 1.18 [Window address]ページ

- (iii) [ASID]ページを開き、[Don't care]チェックボックスのチェックを外し設定するASID値を入力してください。ASID値を条件に設定しない場合は[Don't care]チェックボックスをチェックしたままにしてください。
- (iv) [Bus state]ページを開き、設定するバスの種類とバスサイクルを指定してください。

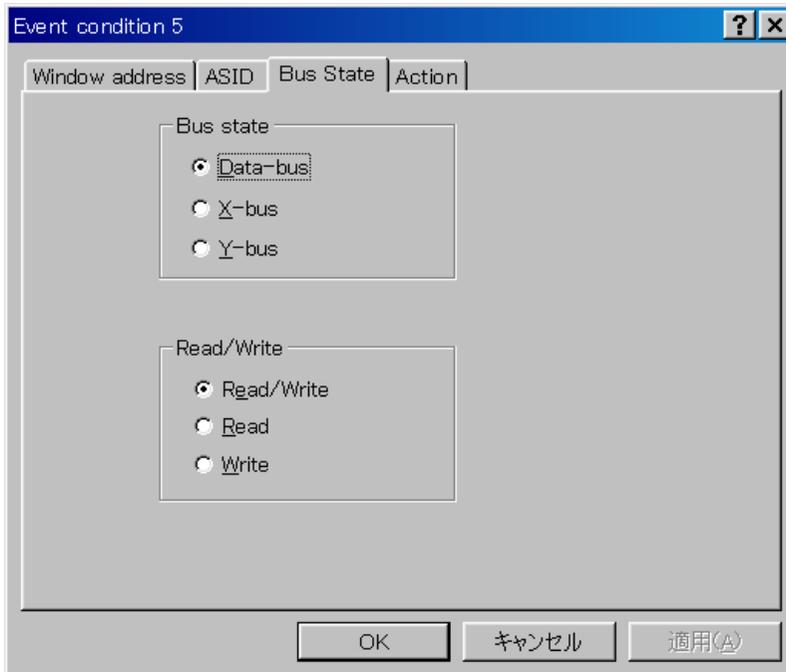


図 1.19 [Bus State]ページ

- (v) [Action]ページの[Acquire trace]チェックボックスにチェックをつけることによって範囲内メモリアクセストレースが取得できます。

**【留意事項】**

設定を解除する場合は、Ch5(OA)または Ch6(OA)行上を右クリックして開くポップアップメニューから削除を選択してください。

- ソフトウェアトレース機能

【留意事項】

本機能はルネサステクノロジ製 SHC/C++コンパイラ(OEM、バンドル販売品を含む)V6.0 よりサポートされます。

ただし、SH4 互換命令以外を出力される場合には SHC/C++コンパイラ(OEM、バンドル販売品を含む)V8.0 以降が必要です。

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 SuperH RISC engine C/C++コンパイラ、アセンブラ最適化リンケージエディタユーザーズマニュアルを参照してください。

ロードモジュールを E200F 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 に対応する汎用レジスタの値と、ソースが表示されます。

ソフトウェアトレース機能を有効にするには、[Event]ウィンドウの software Trace 行をダブルクリックして開く [Software trace]ダイアログボックスで [Acquire Software trace]ラジオボタンをチェックしてください。

【留意事項】

設定を解除する場合は、[Software trace]ダイアログボックスで [Don't care]ラジオボタンをチェックするか、software Trace 行を右クリックして開くポップアップメニューから削除を選択してください。

- システムバスアクセストレース機能

[Event]ウィンドウの Ch8,Ch9 で設定できる機能です。

デバイス内部のシステムバスに関する情報を取得できます。

- 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 1.2 [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	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 ウェイト発生【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 がウェ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。

### (b) デバイス外部の情報をトレース取得する機能

[BUS/MFI trace]ウィンドウに、取得したトレース情報を表示します。

- 外部バストレース機能 (BUS trace)

デバイスの外部バス端子を E200F エミュレータに接続している場合に便利な、大容量のトレース機能です。外部メモリのリード/ライトアクセスが発生した場合、外部バス端子からリアルタイムにトレース情報が出力されません。

また、指定した外部バスイベントポイント条件により、外部バストレースの取得/取得開始/取得終了を制御できます。イベントポイントの設定については、「5.6 イベントポイントを使用する」を参照してください。

外部バストレース機能には、バスサイクルごとに最大 262,144 サイクル分の情報を取得できます。

- その他のトレース機能

オプション機能の、「周辺 I/O アナライザ機能」でトレース取得することができる場合、同じウィンドウでトレース表示結果を見ることができます。

### (2) 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。

- (a) 指定データの検索
- (b) 指定データの抽出
- (c) 指定データをフィルタリングして再表示
- (d) 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。

- (e) ユーザプログラム実行中のトレース設定内容変更

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。

### 1.3.4 ブレーク機能

E200F エミュレータでは、以下の7種類のブレーク機能があります。

#### (1) 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以外の場所には設定できません。

本機能は、[Event]ウィンドウの[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。

#### (2) オンチップイベントブレーク機能

オンチップイベント検出機能で、指定したイベントが検出されたときにブレークする機能です。

本機能は、[Event]ウィンドウの[Onchip Event]シートで、イベント検出後の動作を「ブレーク」とすることによって使用できます。

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[Onchip event]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。

【注】 オンチップイベントブレークの設定はユーザプログラム実行中に変更することができます。

#### (3) AUD イベントブレーク機能

AUD イベント検出機能で、指定したイベントが検出されたときにブレークする機能です。

本機能は、[Event]ウィンドウの[AUD Event]シートで、イベント検出後の動作を「ブレーク」とすることによって使用できます。

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[AUD event]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。

#### (4) 外部バスイベントブレーク機能

外部バスイベント検出機能で、指定したイベントが検出されたときにブレークする機能です。

本機能は、[Event]ウィンドウの[BUS Event]シートで、イベント検出後の動作を「ブレーク」とすることによって使用できます。

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[BUS event]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。

### (5) その他のイベントブレイク機能

- 実行時間イベントブレイク機能
- 外部プローブイベントブレイク機能

その他のイベント検出機能で、指定したイベントが検出されたときにブレイクする機能です。

本機能は、[Event]ウィンドウの[Other Event]シートで、イベント検出後の動作を「ブレイク」とすることによって使用できます。

### (6) トレースブレイク機能

E200F エミュレータ内の AUD トレースバッファ、外部バストレースバッファがいっぱいになったときにブレイクする機能です。

[Internal/AUD/Usermemory acquisition]ダイアログボックスおよび[BUS acquisition]ダイアログボックスで設定できます。

### (7) 強制ブレイク機能

ユーザプログラムを強制的にブレイクする機能です。

## 1.3.5 プローブ機能

E200F エミュレータには、6本の外部プローブを取り付けることができます。

### (1) 入力プローブ(4本)

入力信号をモニタし、指定条件成立によりユーザプログラムをブレイク、AUD トレース開始/終了などができます。

[Event]ウィンドウの[Other Event]シートで設定できます。

### (2) イベント出力プローブ(1本)

AUD イベントの検出により、イベント信号を出力できます。

AUD イベントポイント設定ダイアログボックスの[Action]ページで設定できます。

### (3) GND(1本)

GND 接続用プローブです。

### 1.3.6 周辺 I/O アナライザ機能

E200F エミュレータは、CPU の周辺 I/O 端子信号をアナライザケーブルに接続した場合、周辺 IP のプロトコルを解析して指定した条件によりブレイク、端子信号のトレース開始/終了などができます。

**【留意事項】**

製品によっては、サポートするアナライザ機能が異なります。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。

### 1.3.7 パフォーマンス測定機能

E200F 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 5 種類があります。

**オンチップパフォーマンス測定機能 (Onchip Performance Analysis)**

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。

**【留意事項】**

測定できる項目は、製品によって異なる場合があります。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。

## (a) パフォーマンスの測定条件の設定

パフォーマンスの測定条件の設定は、[Performance Analysis]ダイアログボックス、および PERFORMANCE\_SET コマンドを使用します。[Performance Analysis]ダイアログボックスは、[パフォーマンス解析]ウィンドウ上の設定を行うチャンネル行を選択しマウスの右ボタンを押すと、ポップアップメニューが表示され、[設定]を選択すると表示されます。



図 1.20 [Performance Analysis]ダイアログボックス

## 【留意事項】

コマンドラインシンタックスについては、オンラインヘルプを参照してください。

### 測定開始 / 終了条件指定

測定開始 / 終了条件指定は、[Event]ウィンドウ[Onchip Event]シートの Ch1 ~6、Ch8~12 をダブルクリックして開く、[Event Condition]ダイアログボックスで条件を設定した後[Action]ページで測定開始 / 終了を指定してください。

#### 【留意事項】

1. 測定開始 / 終了条件指定を行わなかった場合は、プログラムの実行で測定を開始し、ブレーク条件成立により測定を終了します。
2. 測定開始 / 終了条件のどちらか一方のみを指定した場合は、パフォーマンス測定を行うことができません。必ず測定開始 / 終了両方の条件指定を行ってください。
3. 測定開始 / 終了条件指定をした場合は、ステップ実行はできません。また、BREAKPOINT/Event Condition ブレーク条件で停止後、そのアドレスから実行を再開する場合、ステップ機能を使用するため動作ができません。BREAKPOINT/Event Condition ブレーク条件設定を解除後に実行を再開してください。
4. 1つのチャン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。

表 1.3 [Action]ページで指定できる条件

項目		意味
PA1	pa1_start_point	設定した Event Condition の条件をパフォーマンスチャンネル 1 の測定開始条件に設定します。
	pa1_end_point	設定した Event Condition の条件をパフォーマンスチャンネル 1 の測定終了条件に設定します。
PA2	pa2_start_point	設定した Event Condition の条件をパフォーマンスチャンネル 2 の測定開始条件に設定します。
	pa2_end_point	設定した Event Condition の条件をパフォーマンスチャンネル 2 の測定終了条件に設定します。
PA3	pa3_start_point	設定した Event Condition の条件をパフォーマンスチャンネル 3 の測定開始条件に設定します。
	pa3_end_point	設定した Event Condition の条件をパフォーマンスチャンネル 3 の測定終了条件に設定します。
PA4	pa4_start_point	設定した Event Condition の条件をパフォーマンスチャンネル 4 の測定開始条件に設定します。
	pa4_end_point	設定した Event Condition の条件をパフォーマンスチャンネル 4 の測定終了条件に設定します。

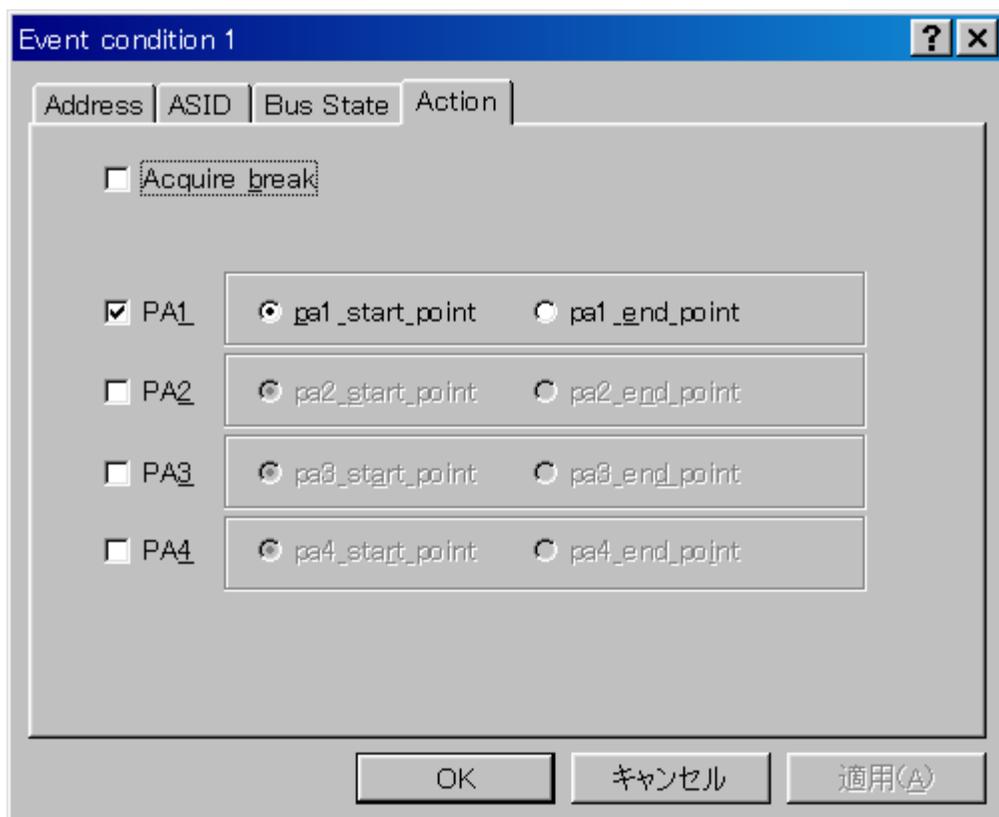


図 1.21 [Action]ページ

【注】 Ch8,9 では PA1、PA2 の設定はできません。

### 測定誤差について

- 測定値は、誤差を含みます。
- ブレーク発生の前後で誤差が生じることがあります。

上記につきましては、表 1.6 も参照してください。

### 測定項目

測定項目は、Ch1～4 毎に [Performance Analysis]ダイアログボックスで行います。最大 4 つの条件を同時に指定可能です。表 1.4 に示します（表 1.4 のオプションは、PERFORMANCE\_SET コマンドの<mode>パラメータです。また、[パフォーマンス解析]ウィンドウの Condition に表示します）。

表 1.4 測定項目 (1)

大分類	分類	測定項目	オプション	備考
無効			なし	測定しない。
CPU 性能	サイクル	経過サイクル数	AC	パワーオンリセット期間を除きます。CPU クロックでカウントします。
		特権モードサイクル数	PM	経過サイクル数中の特権モードサイクル数です。
		SR.BL ビットアサートサイクル数	BL	経過サイクル数中の SR.BL ビット=1 のサイクル数です。
	カウント (命令系)	有効命令発行回数	I	有効命令発行回数 + 2 命令同時実行回数=実行命令数 有効命令数とは、完了した命令数を指します。
		2 命令同時実行回数	2I	有効命令発行回数中の 2 命令同時実行された回数
	カウント (分岐系)	無条件分岐回数	BT	例外発生による分岐以外の無条件分岐回数です。ただし、RTE はカウントされます。
	カウント (例外、割り込み系)	例外受付回数	EA	割り込みを含みます。
		割り込み受け付け回数	INT	NMI を含みます。
		UBC チャンネルヒット回数	UBC	すべての CPU 内チャンネルヒット回数の OR でカウントします。
	ストールサイクル	フルトレースモード・ストールサイクル数(多重カウントあり)	SFM	すべての項目に対して独立にカウントされます。
フルトレースモード・ストールサイクル数(多重カウントなし)		SF	命令実行起因ストールサイクルと同時発生した場合には、本項目はカウントされません。	

表 1.4 測定項目 (2)

大分類	分類	測定項目	オプション	備考
TLB 性能	TLB	命令フェッチ UTLB ミス回数	UMI	命令フェッチによる TLB ミス例外発生回数(EXPEVT セット回数)
		オペランドアクセス UTLB ミス回数	UMO	オペランドアクセスによる TLB ミス例外発生回数(EXPEVT セット回数)
		ITLB ミス回数	IM	有効アクセスに対する ITLB ミス回数 (UTLBのミス/ヒットは考慮しません。)
命令バス性能	命令バス	命令側メモリアクセス回数	MIF	命令フェッチによるメモリアクセス回数 命令フェッチバスでキャンセルされたアクセスはカウントされません。分岐予測中にフェッチされ実際には実行されなかった命令フェッチはカウントされます。 PREFI 命令によるアクセスを含みます。
		命令キャッシュアクセス回数	IC	命令側メモリアクセス回数中の命令キャッシュへのアクセス回数。
		命令キャッシュミス回数	ICM	命令キャッシュアクセスによるキャッシュミス回数。(キャッシュミスにより CPU コア外へアクセスする回数。)
		命令側内蔵メモリアクセス回数 (XY-RAM または L メモリ (O-L メモリ))	XL	命令側メモリアクセス回数中のデバイスにおける XY メモリへのアクセス回数。
		命令側 I-L メモリアクセス回数	ILIF	命令側メモリアクセス回数中のデバイスにおける I-L メモリへのアクセス回数。
		命令側 U メモリアクセス回数	ULF	命令側メモリアクセス回数中のデバイスにおける U メモリへのアクセス回数。

表 1.4 測定項目 (3)

大分類	分類	測定項目	オプション	備考
オペランド バス性能	アクセス	オペランド側メモリ アクセス回数(リード)	MR	オペランドリードによるメモリアクセス回数(オペランドバス上でのロードに相当します)。PREF 命令によるアクセスおよびキャンセルされたアクセスは対象外です。
		オペランド側メモリ アクセス回数(ライト)	MW	オペランドライトによるメモリアクセス回数。(オペランドバス上でのストアに相当します)。キャンセルされたアクセスは対象外です。
		オペランドキャッシュ アクセス回数(リード)	CR	オペランド側メモリアクセス回数(リード)中のオペランドキャッシュリード回数。
		オペランドキャッシュ アクセス回数(ライト)	CW	オペランド側メモリアクセス回数(ライト)中のオペランドキャッシュライト回数。
		オペランド側内蔵メモリ アクセス回数(リード) (XY-RAM または L メモリ (O-L メモリ))	XLR	オペランド側メモリアクセス回数(リード)中の、デバイスにおける XY メモリへのアクセス回数 (XY バス経由/オペランドバス経由、両方含みます。また、MOVX、MOVY が同時に実行された時はリード/ライトによらず 1 カウント UP されます。)
		オペランド側内蔵メモリ アクセス回数(ライト) (XY-RAM または L メモリ (O-L メモリ))	XLW	オペランド側メモリアクセス回数(ライト)中の、デバイスにおける XY メモリへのアクセス回数 (XY バス経由/オペランドバス経由、両方含みます。また、MOVX、MOVY が同時に実行された時はリード/ライトによらず 1 カウント UP されます。)
		オペランド側 I-L メモリ アクセス回数(リード/ ライト)	ILRW	オペランド側メモリアクセス回数(リード/ライト)中の、デバイスにおける I-L メモリへのアクセス回数。

表 1.4 測定項目 (4)

大分類	分類	測定項目	オプション	備考
オペランド バス性能	アクセス	オペランド側 U-RAM アクセス(リード)	UR	オペランド側メモリアクセス回数(リード)中のUメモリアクセス回数 (キャッシュ経由は除きます)
		オペランド側 U-RAM アクセス(ライト)	UW	オペランド側メモリアクセス回数(ライト)中のUメモリアクセス回数 (キャッシュ経由は除きます)
	アクセス ミス	オペランドキャッシュミス 回数(リード)	CMR	オペランドキャッシュアクセス回数(リード)のアクセスによるキャッシュミス回数(キャッシュミスによりコア外へアクセスする回数) PREF 命令によるミスはカウントされません。
		オペランドキャッシュミス 回数(ライト)	CMW	オペランドキャッシュアクセス回数(ライト)のアクセスによるキャッシュミス回数(キャッシュミスによりコア外へアクセスする回数) ライトスルーの設定の場合、カウントされません。 PREF 命令によるミスはカウントされません。
		U-RAM リードバッファミス 回数	UBM	メモリを搭載していないデバイスについて本機能は使用できません。
	ウェイト サイクル	オペランドアクセスウェイト サイクル(リード)	WOR	オペランド側メモリアクセス回数(リード)によるウェイトサイクル数
		オペランドアクセスウェイト サイクル(ライト)	WOW	オペランド側メモリアクセス回数(ライト)によるウェイトサイクル数
		オペランドキャッシュミスウ ェイトサイクル (リード)	WCMR	オペランドキャッシュミス回数(リード)によるウェイトサイクル数(ただし、キャッシュF 競合などによるウェイトサイクル数を含みます)
		オペランドキャッシュミスウ ェイトサイクル (ライト)	WCMW	オペランドキャッシュミス回数(ライト)によるウェイトサイクル数
		オペランド側 I-L メモリ アクセスウェイトサイクル数 (リード)	WILR	オペランド側 I-L メモリアクセス回数(リード)によるウェイトサイクル数

表 1.4 測定項目 (5)

大分類	分類	測定項目	オプション	備考
オペランド バス性能	ウェイト サイクル	オペランド側 I-L メモリ アクセスウェイトサイクル数 (ライト)	WILW	オペランド側 I-L メモリアクセス 回数(ライト)によるウェイトサイ クル数
システム バス性能 (Ch3,4 での み選択可能 です。)	システム バス	リクエスト数	RQ	有効なバス使用サイクル数(セル 数)を System Bus クロックでカウ ントします。
		レスポンス数	RS	有効なバス使用サイクル数(セル 数)を System Bus クロックでカウ ントします。
		リクエストウェイトサイクル	WRQ	発行されたリクエスト(req)に対 し、許可信号(gnt)が発行されな いサイクルを System Bus クロ ックでカウントします。複数のリク エストに対するウェイトが同時 に発生しても 1 とカウントしま す。
		レスポンスウェイトサイクル	WRS	発行されたレスポンス(r_req)に 対し、許可信号(r_gnt)が発行され ないサイクルを System Bus クロ ックでカウントします。複数のレ スポンスに対するウェイトが同 時に発生しても 1 とカウントしま す。

## 1 製品概要

---

以下に代表的な測定項目と測定方法を説明します。

表 1.5 代表的測定内容

代表的測定項目	測定方法
経過時間	経過サイクル数×CPUクロック周期
実行命令数	有効命令発行回数+2命令同時実行回数
割り込み受付回数	例外受付回数
命令フェッチ回数 (キャッシュ、非キャッシュ両方含む)	命令側メモリアクセス回数
命令キャッシュヒット率	$(\text{命令キャッシュアクセス回数} - \text{命令キャッシュミス回数}) / \text{命令キャッシュアクセス回数}$
オペランドアクセス回数 (キャッシュ、非キャッシュ両方含む)	オペランド側メモリアクセス回数(リード)+オペランド側メモリアクセス回数(ライト)
オペランドキャッシュヒット率(リード)	$(\text{オペランドキャッシュアクセス回数(リード)} - \text{オペランドキャッシュミス回数(リード)}) / \text{オペランドキャッシュアクセス回数(リード)}$
オペランドキャッシュヒット率(ライト)	$(\text{オペランドキャッシュアクセス回数(ライト)} - \text{オペランドキャッシュミス回数(ライト)}) / \text{オペランドキャッシュアクセス回数(ライト)}$
オペランドキャッシュヒット率	$(\text{オペランドキャッシュアクセス回数(リード)} + \text{オペランドキャッシュアクセス回数(ライト)} - \text{オペランドキャッシュミス回数(リード)} - \text{オペランドキャッシュミス回数(ライト)}) / (\text{オペランドキャッシュアクセス回数(リード)} + \text{オペランドキャッシュアクセス回数(ライト)})$
System Bus リクエストバス占有率	$(\text{リクエスト数のCPUクロック換算値}) / \text{経過サイクル数}$
System Bus レスポンスバス占有率	$(\text{レスポンス数のCPUクロック換算値}) / \text{経過サイクル数}$

各測定条件については、表 1.6 に示す条件が発生した場合についてもカウントを行います。

表 1.6 パフォーマンス各測定条件においてカウントする場合

測定条件	留意事項
TLB のキャッシング可能ビットの設定により、キャッシングされない場合	キャッシュ可能領域へのアクセスにカウントされます。
キャッシュオンでのカウント	キャッシュ不可領域のアクセスがサイクル数、回数が実際よりも少なく、キャッシュ可能領域、X/Y-RAM、U-RAM エリアへのアクセスは実際よりも多くカウントされることがあります。
分岐回数のカウント	カウンタの値は、2 ずつ増えます。これは、1 回の分岐につき有効なサイクルが 2 サイクルという意味です。

【留意事項】

1. AUD トレース、メモリ出力トレースの Non realtime trace モード中は、ストールの発生状況や実行サイクルが変化するため、正確なカウントが出来ません。
2. カウンタのクロックソースが CPU クロックであるため、スリープモード等で CPU クロックが停止する場合は、カウントもストップします。

**測定結果格納カウンタの拡張設定**

測定結果を格納するカウンタは 32 ビットであり、2 本を接続して 64 ビットカウンタとして使用することも可能です。

64 ビットカウンタを設定するには、[Performance Analysis]ダイアログボックスの Ch1,3 の[Performance Analysis]ダイアログボックス[拡張カウンタ]グループボックスの[有効]チェックボックスをチェックしてください。

(b) 測定結果の表示

測定結果は、[パフォーマンス解析]ウィンドウ、または、PERFORMANCE\_ANALYSIS コマンドで行います。表示結果は 16 進数 (32 ビット) で表示します。

ただし、拡張カウンタを有効にしている場合は 16 進数 (64 ビット) で表示します。

【留意事項】

パフォーマンス測定の結果のカウンタがオーバーフローした場合、"\*\*\*\*\*"を表示します。

(c) 測定結果の初期化

測定結果の初期化は、[パフォーマンス解析]ウィンドウのポップアップメニューで [全てリセット]を選択するか、PERFORMANCE\_ANALYSIS コマンドで INIT を指定してください。

## 1 製品概要

---

### (2) AUD パフォーマンス測定機能 (AUD Performance Analysis)

本機能は、AUD イベント検出システムで指定されたイベント間の実行時間、実行回数などを測定する機能です。タイマの分解能は以下のいずれかの値を設定できます。

20ns, 40ns, 100ns, 400ns

測定可能な最大時間は、分解能 20ns で約 6 時間、分解能 400ns で約 5 日です。

### (3) 外部バスパフォーマンス測定機能 (BUS Performance Analysis)

指定した開始条件と終了条件間の実行時間、実行回数などを測定する機能です。タイマの分解能は以下のいずれかの値を設定できます。

20ns, 50ns, 400ns

測定可能な最大時間は、分解能 20ns で約 6 時間、分解能 400ns で約 5 日です。

### (4) プロファイル機能 (Profile)

E200F エミュレータでは、各関数のパフォーマンスを測定できる「プロファイル機能」があります。関数ごとの所要時間の統計などをとることによって、パフォーマンスの悪い関数を見つけ出すことが容易になります。

測定できる項目は、オンチップパフォーマンス測定機能で測定できる項目と同じです。

【注】 プロファイル機能とオンチップパフォーマンス測定機能は、同時に使用できません。

同時に使用しようとした場合、「Can not use this function」エラーメッセージダイアログボックスが表示されます。

#### 【留意事項】

本機能は、分岐が発生するたびにブレークし、情報を集めてから再度実行しています。

したがって、ユーザプログラムのリアルタイム性は損なわれます。

### (5) リアルタイムプロファイル機能 (Realtime Profile)

指定したアドレス範囲内の各関数のパフォーマンスを測定できます。関数ごとの所要時間の統計などをとることによって、パフォーマンスの悪い関数を見つけ出すことが容易になります。

本機能では、ブレークすることなくパフォーマンス情報を集めています。したがって、ユーザプログラムのリアルタイム性は損なわれません。

指定できるアドレス範囲を以下に示します。

- 拡張プロファイルユニットを接続していないとき 512kB ~ 4MB (512kBで8ブロック)
- 拡張プロファイルユニットを接続しているとき 512kB ~ 12MB (512kBで24ブロック)

### 1.3.8 カバレジ機能

C/C++およびアセンブラレベルでの命令実行情報を表示します。

C0 カバレジを測定できます。

### 1.3.9 メモリアクセス機能

E200F 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またはフラッシュメモリです。

フラッシュメモリへダウンロードする場合、[オプション]メニューの[エミュレータ]を選択して[Configuration]ダイアログボックスを開き、[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。

## 1 製品概要

---

### (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

E200F エミュレータは、プログラムの実行を中断することなくアクセスのあった領域の値をモニタし、ウィンドウ上に表示することができます。論理アドレスでモニタすることができるのは、CPU によるアクセスがあったアドレスです。

### (7) 貸し出しメモリ機能

CS0 エリアにエミュレーション用メモリを割り当てることができます。

### (8) 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、E200F 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。ユーザプログラムの停止時間は、ご使用のホストマシンの性能や JTAG クロックに依存します。

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

ホストマシン	: Pentium®	800MHz
SH7323	: CPU クロック	96MHz
JTAG クロック	: 30MHz	

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 96ms となります。

## 2. ユーザプログラムブレーク中のメモリアクセス

E200F 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。しかし他のメモリアイト操作は RAM 領域に対してのみ可能です。したがって、メモリアイト、BREAKPOINT 等の設定は RAM 領域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、メモリアイト、BREAKPOINT 設定、ダウンロード等の操作は行わないでください。

## 3. ユーザプログラムブレーク中のキャッシュ操作

キャッシュ内蔵デバイスでキャッシュイネーブルの場合、E200F エミュレータは以下の方法でメモリアクセスしています。

メモリアイト時 : キャッシュに書き込み、外部ヘシングルライトを発行します。LRU の更新は行いません。

メモリアイト時 : キャッシュから読み出しを行います。LRU の更新は行いません。

### 1.3.10 スタックトレース機能

E200F エミュレータでは、スタック情報を用いて、現在の PC がある関数がどの関数からコールされているかを表示します。本機能は、Dwarf2 形式のデバッグ情報を持ったロードモジュールをロードした場合のみ使用できません。

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

### 1.3.11 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。

### 1.3.12 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。

## 1.4 使用環境条件

**▲ 注意**

E200F エミュレータを使用する場合、表 1.7、および表 1.8 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 E200F エミュレータを使用した場合、E200F 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 1.7 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 <sup>2</sup> 非動作時 : 最大 4.9m/s <sup>2</sup> 梱包輸送時 : 最大 14.7m/s <sup>2</sup>
4	周囲ガス	腐食性ガスのないこと

表 1.8 動作環境

項番	項目	動作環境
1	ホストマシン	Pentium® 以上 (推奨 1GHz 以上) を搭載し、USB1.1/2.0(Full-Speed)を備えた IBM PC およびその互換機
2	OS	Windows® 2000 および Windows® XP
3	最小稼働メモリ容量	128MB 以上 (推奨 512MB 以上)
4	ハードディスク容量	インストールディスク容量 250MB 以上 (スワップ領域を考慮してメモリ容量の 2 倍以上 (推奨 4 倍以上) の空き容量をご用意ください。)
5	マウスなどのポインティングデバイス	ホストマシン本体に接続可能で Windows® 2000 および Windows® XP に対応している、マウスなどのポインティングデバイス
6	ディスプレイ	モニタ解像度 1024 × 768 以上
7	AC 入力電源	電圧 : AC100V ± 10% 周波数 : 50/60Hz 消費電力 : 48W
8	CD-ROM ドライブ	E200F 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 E200F エミュレータユーザズマニュアルを参照するために必要

---

## 2. セットアップ

---

### 2.1 エミュレータ使用までのフローチャート

E200F 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 2.1 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、E200F 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。

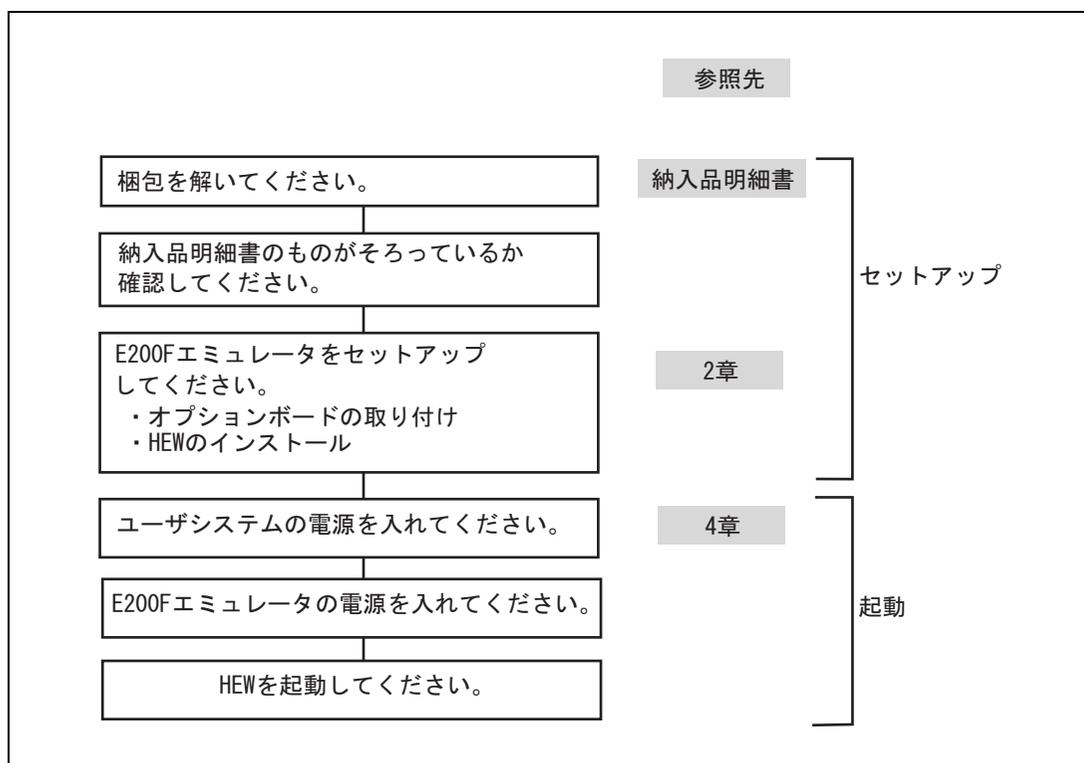


図 2.1 E200F エミュレータ使用フローチャート

## 2.2 デバッガのインストール

### 2.2.1 CD-R について

CD-R のルートディレクトリには E200F 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 2.1 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	E200F エミュレータ用ドライバ	E200F エミュレータ用 USB ドライバです。
Help	E200F 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。
Manual	E200F エミュレータマニュアル	E200F 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。
sot	E200F エミュレータ用故障診断プログラム	3.4 章に使い方についての説明があります。

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。

## 2.3 エミュレータ本体への接続

E200F エミュレータには、各種オプションユニットがあります。

本章では、オプションユニットを E200F 本体に接続する方法についてご説明します。

### 2.3.1 E200F トレースユニットとユーザシステムとの接続

- メインユニットの側面にあるTRACE I/Fのふたを開いてください。
- 図2.2のようにトレースユニットに付属しているトレースケーブルを接続してください。



図 2.2 E200F 側のトレースケーブル接続方法

## 2 セットアップ

---

- トレースユニットとトレースケーブル (CNI側) を接続してください。

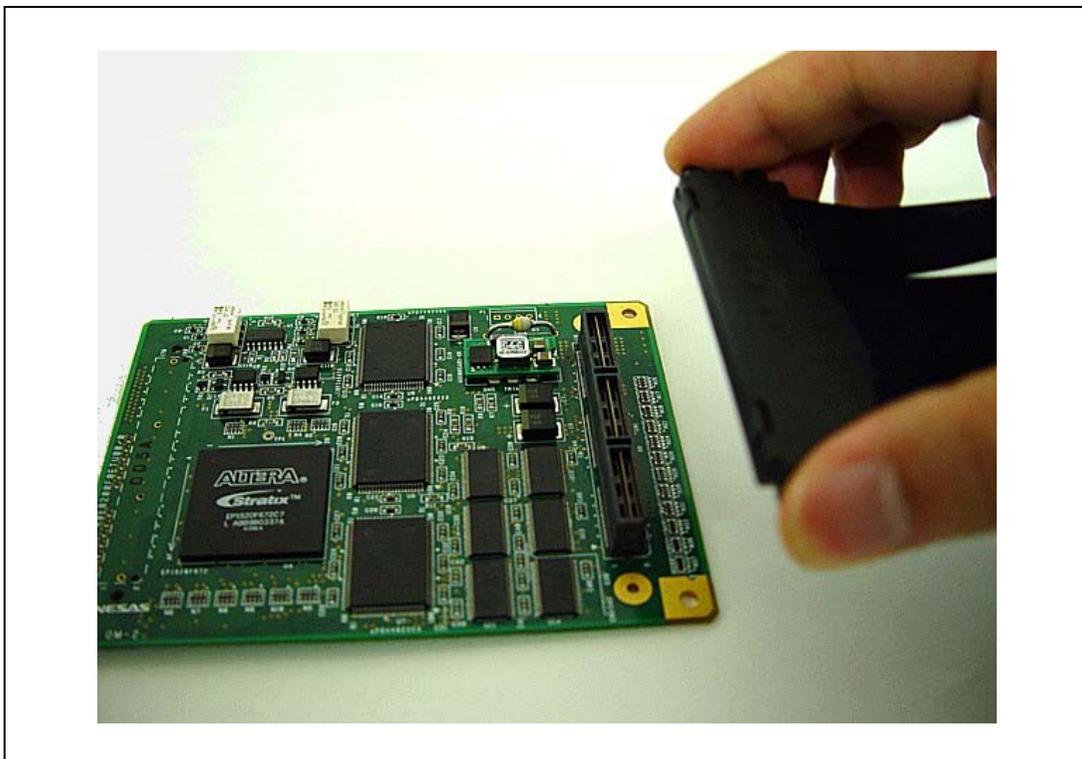


図 2.3 トレースユニット側のトレースケーブル接続方法

- 1ピンの位置に注意してユーザシステムとトレースユニットを接続してください。

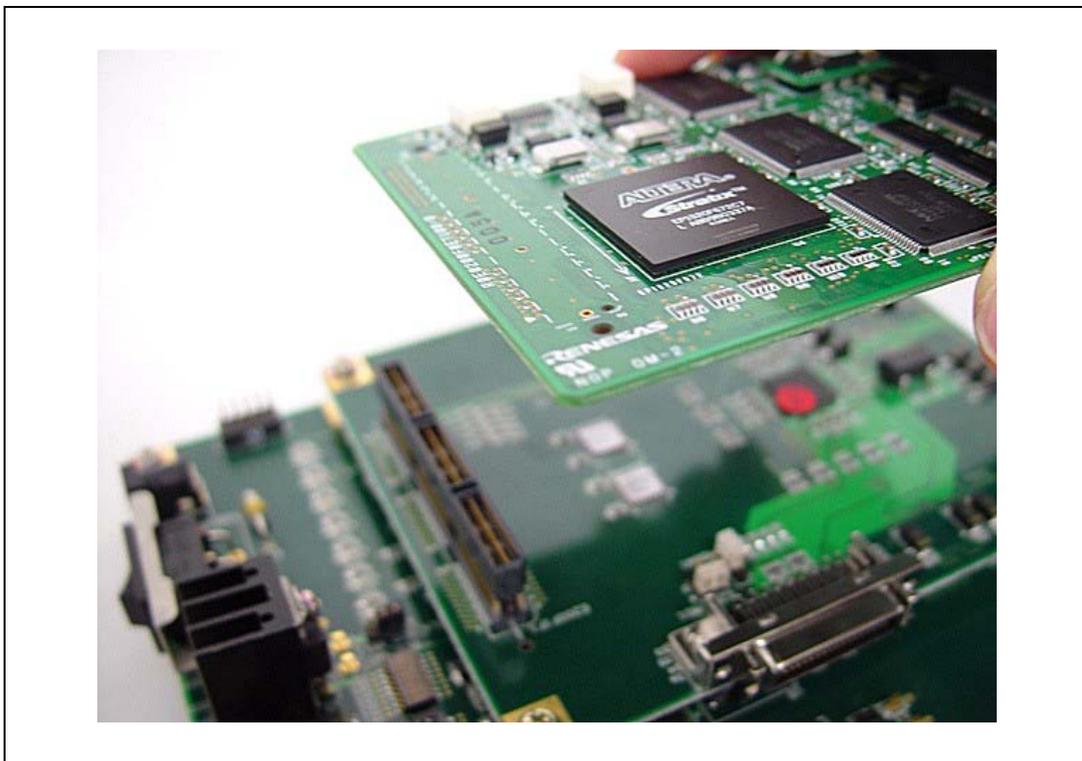


図 2.4 トレースユニット使用時のユーザシステム側のトレースユニット接続方法

### ⚠ 注意

1 ピンの向きに注意して接続してください。

#### 【留意事項】

1. コネクタの信号線の接続先は、サポートデバイスによって異なります。
2. ユーザシステムに接続する際、信号の配線は、別冊の「SHxxxx ご使用時の補足説明」を参照してください。

## 2 セットアップ

### 2.3.2 E200F 周辺 I/O アナライザユニットとユーザシステムとの接続



図 2.5 周辺 I/O アナライザユニット使用時の E200F 側のアナライザケーブル接続方法

#### **▲ 注意**

1 ピンの向きに注意して接続してください。

#### 【留意事項】

1. コネクタの信号線の接続先は、各周辺 I/O アナライザユニットによって異なります。
2. ユーザシステムに接続する際、信号の配線は、別冊の「SHxxxx ご使用時の補足説明」を参照してください。

### 2.3.3 E200F 拡張プロファイルユニットの接続

- 底面のベース筐体のネジを取り外します。



図 2.6 縦置き用ベース筐体のネジ



図 2.7 縦置き用ベース筐体の取り外し

- メインユニットの背面にあるネジ（2ヶ所）を取り外します。



図 2.8 メインユニット筐体のネジ

## 2 セットアップ

---

- 図2.9のように筐体を取り外します。



図 2.9 メインユニット筐体の取り外し

- 拡張プロファイルユニットの1ピンの向きに注意してメインユニットと接続してください。

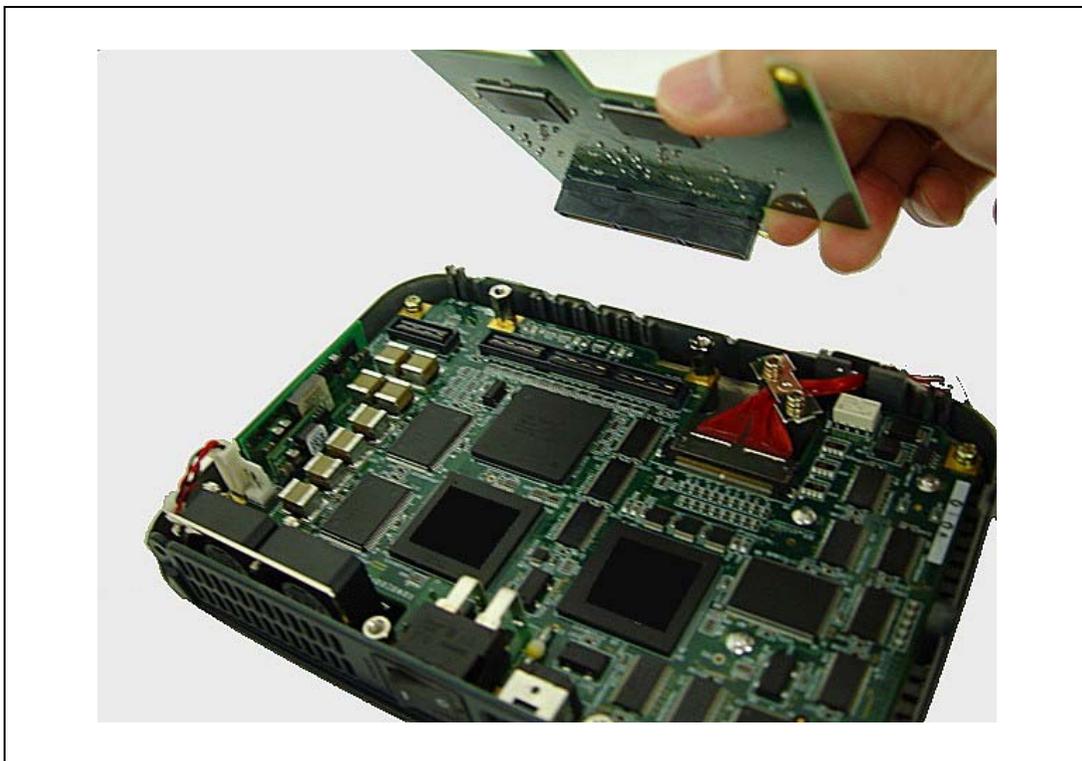


図 2.10 拡張プロファイルユニットとメインユニットの接続

## 2 セットアップ

---

- 付属のネジにより拡張プロファイルユニットを固定してください。

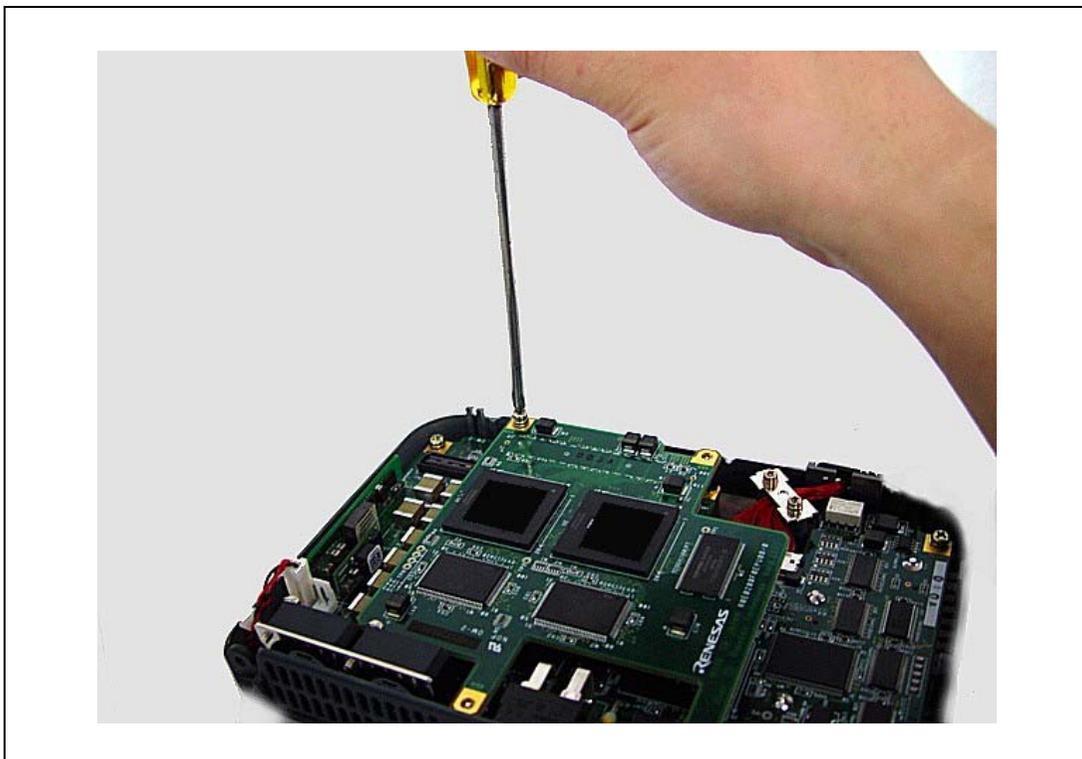


図 2.11 拡張プロファイルユニットのネジ止め

- メインユニットの筐体を元に戻しネジ止め（2ヶ所）してください。
- ベース筐体のネジを締めて固定してください。

## 2.4 ACアダプタの接続

付属の AC アダプタを接続してください。



図 2.12 AC アダプタの接続方法

- "DC IN" と表示のある AC アダプタ DC (+12V) 入力用コネクタに付属品の AC アダプタを接続してください。

### ⚠ 警告

必ず AC アダプタは、付属の E200F 専用 AC アダプタを使用してください。専用の AC アダプタ以外を接続した場合、発煙、発火、およびユーザシステムまたはエミュレータの破壊の危険性があります。

### 2.5 ホストマシンとの接続

E200F エミュレータとホストマシンを接続する方法を説明します。なお、E200F エミュレータ本体における各コネクタの位置は、「1.2 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、検索場所として[検索場所の指定]を選択してください。検索場所は、「<ドライブ>:\DRIVERS」を指定してください。

（<ドライブ>は CD ドライブのドライブ名です。）

【留意事項】

E200F 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。

 <b>警告</b>
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、E200F 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。

E200F エミュレータは、ホストマシンと USB 2.0/1.1 で接続できます。システム構成を図 2.13 に示します。

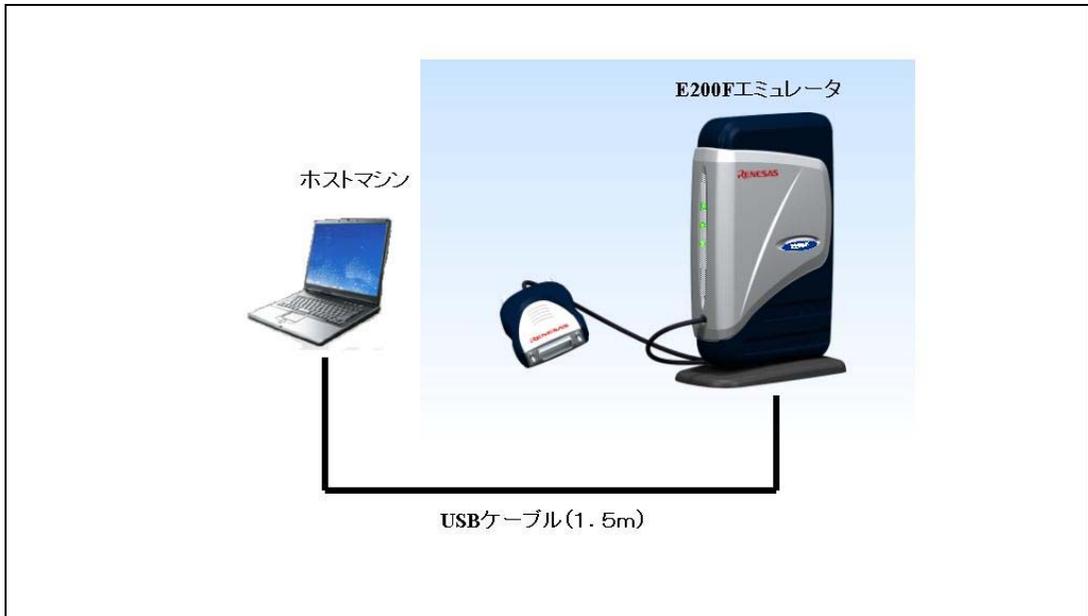


図 2.13 E200F エミュレータのホストマシン接続システム構成

## 2.6 ユーザシステムとの接続

以下に示す手順で E200F 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 E200F エミュレータとユーザシステムを取り外す場合、取り付ける場合も同様の手順で接続してください。

1. E200F エミュレータの電源が OFF になっていることを確認してください。
2. E200F エミュレータの先端プローブコネクタをユーザシステムに接続します。
3. 先端プローブをユーザシステムに確実に固定するために先端プローブ固定用ネジでネジ止めしてください。  
(36ピンコネクタのみ)

- (1) コネクタをユーザシステム上に実装してください。エミュレータが推奨する H-UDI ポートコネクタを表 2.2 に示します。ご使用になる E200F エミュレータにより、先端プローブコネクタ形状が異なります。

表 2.2 推奨コネクタ

	型名	メーカー	仕様	エミュレータ型名
36 ピン コネクタ	7 6 1 4 - 6 0 0 2	住友スリーエム株式会社	36 ピンストレート タイプ (国内推奨)	R0E0200F0EMU00
	2 5 1 4 - 6 0 0 2	3M Limited	36 ピンストレート タイプ (海外推奨)	
38 ピン コネクタ	2 5 7 6 7 0 0 4 - 2	タイコエレクトロニクス アンブ株式会社	38 ピン MICTOR コネクタ	R0E0200F2EMU00

- (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) 36 ピンユーザインタフェースケーブル使用時は、H-UDI ポートコネクタの 2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,33,34,36 ピンを PCB 上で GND に接続してください。38 ピンユーザインタフェースケーブル使用時は、H-UDI ポートコネクタの 5 ピンとコネクタの Ground Bus Leads を GND に接続してください。電気的な GND として使用するほか、E200F 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。
- (4) オプション製品の推奨コネクタおよびピン配置は、別冊の「SHxxxx ご使用時の補足説明」を参照してください。

### 2.6.1 E200F H-UDI/AUD プローブとユーザシステムとの接続

- 図2.14のようにH-UDI/AUDプローブとユーザシステムを接続してください。



図 2.14 E200F とユーザシステムの H-UDI/AUD プローブ接続方法

## 2 セットアップ

---

- 図2.15のようにユーザシステムとH-UDI/AUDプローブを固定用ネジにてネジ止めしてください。



図 2.15 ユーザシステムと H-UDI/AUD プローブの固定方法

## ⚠ 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。MCU 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 MCU 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 2.16 参照)

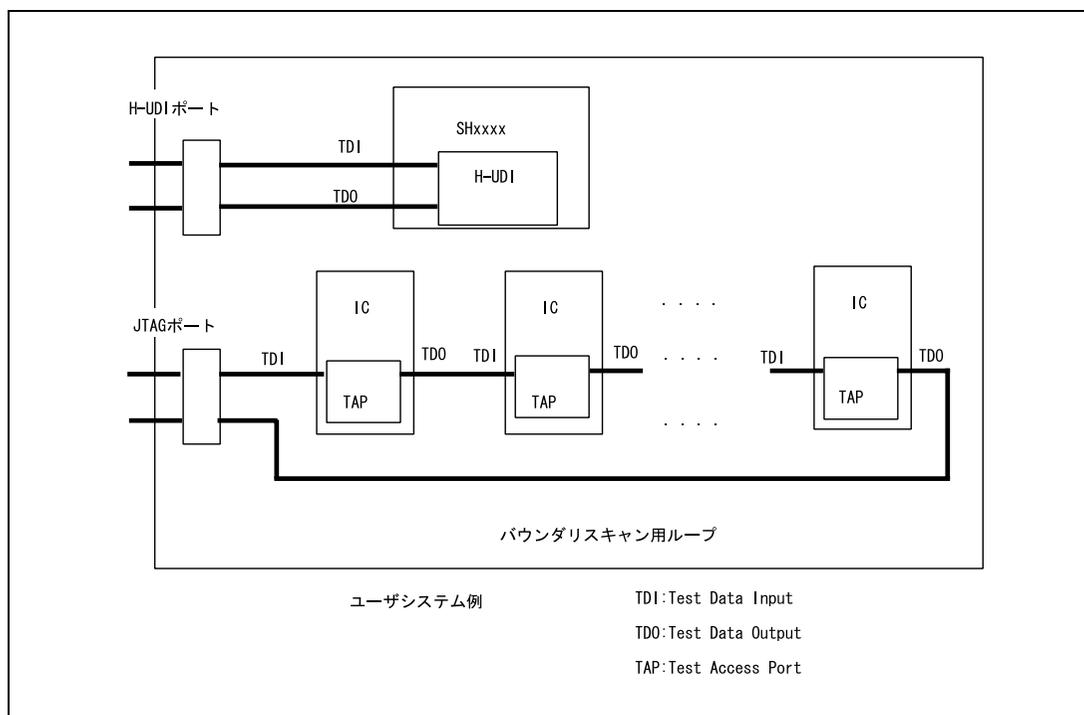


図 2.16 ユーザシステム設計時の注意

## 2 セットアップ

### 2.6.2 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 2.17）

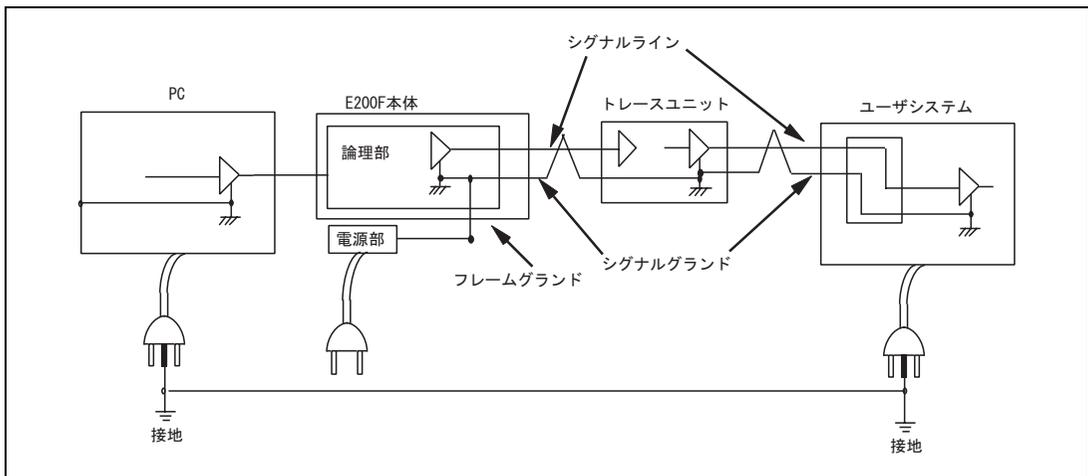


図 2.17 システムグランド系の接続

## 2.7 設定の変更

E200F エミュレータはユーザのデバッグ要求に応じて、エミュレータの機能を柔軟に変更できます。

High-performance Embedded Workshop を起動し、E200F エミュレータを High-performance Embedded Workshop と接続するときに、使用する機能を選択することができます。

High-performance Embedded Workshop 起動時、以下のダイアログボックスが表示されます。

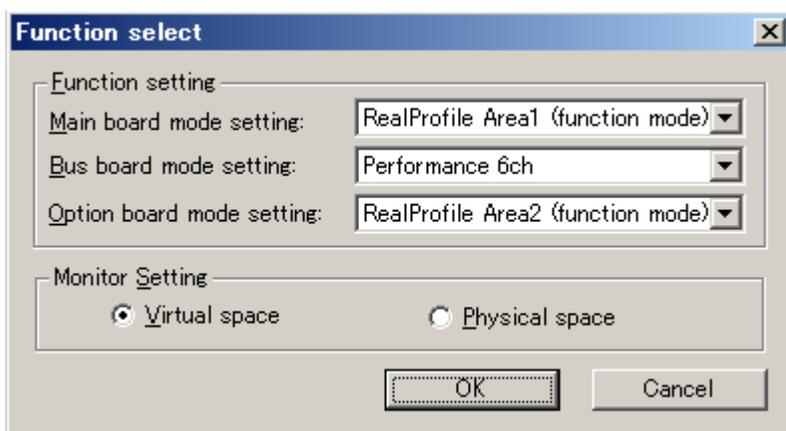


図 2.18 [Function select]ダイアログボックス

次章より、ここで選択できる内容について簡単に示します。

【注】 各機能の詳細につきましては、「1.3 エミュレータ機能」を参照してください。

### 2.7.1 E200F メインユニットのみで実現できる機能の変更方法

[Function select]ダイアログボックスの[Main board mode setting]リストボックスで選択します。

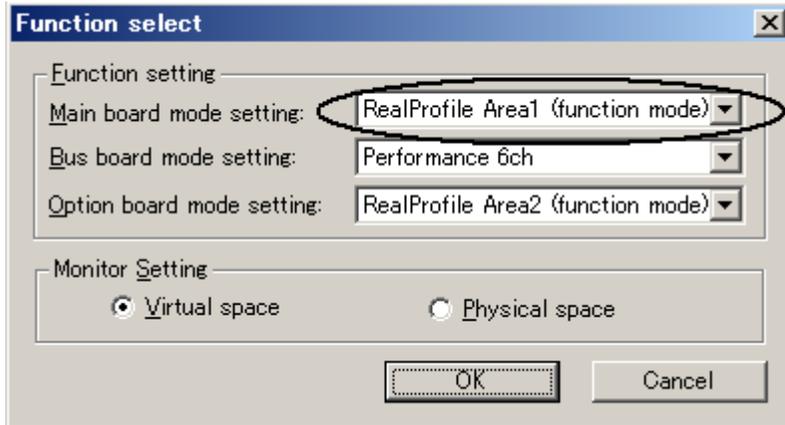


図 2.19 [Function select]ダイアログボックス

- [RealProfile Area1 (function mode)]選択時  
各関数の実行時間累積を計測します。サブルーチン実行時間は含みません。
- [RealProfile Area1 (nest mode)]選択時  
各関数の実行時間累積を計測します。サブルーチン実行時間も含みます。
- [Coverage (4M)]選択時  
4MB空間のC0カバレッジ情報を取得します。
- その他  
製品によっては、周辺I/Oアナライザ機能を選択することができます。

【注】 リアルタイムプロファイラ機能とカバレッジ機能は、拡張プロファイルユニットにより設定できる範囲を増やすことができます。

## 2.7.2 外部バストレースユニット使用時の機能変更方法

[Function select]ダイアログボックスの[Bus board mode setting]リストボックスで選択します。

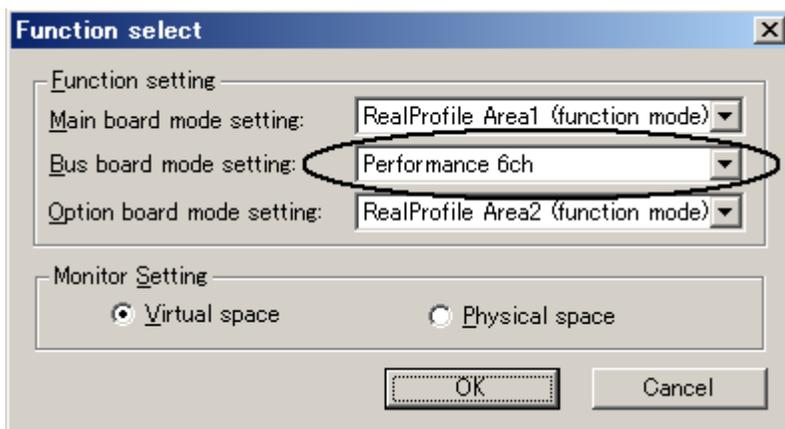


図 2.20 [Function select]ダイアログボックス

- [Trace/break 6ch (Trace 262144 cycles)]選択時  
外部バスイベント検出用チャンネルをブレイクとして使用します。
- [Performance 6ch]選択時  
外部バスイベント検出用チャンネルを実行時間測定用チャンネルとして使用します。
- [Emulation memory (4M, Trace 8192 cycles)]選択時  
外部貸出しメモリ機能 (4Mbyte × 1ブロック) を使用します。

### 【留意事項】

トレースユニットを E200F エミュレータに接続していない場合、本機能はグレー表示となります。

### 2.7.3 拡張プロファイルユニット使用時の機能変更方法

[Function select]ダイアログボックスの[Option board mode setting]リストボックスで選択します。

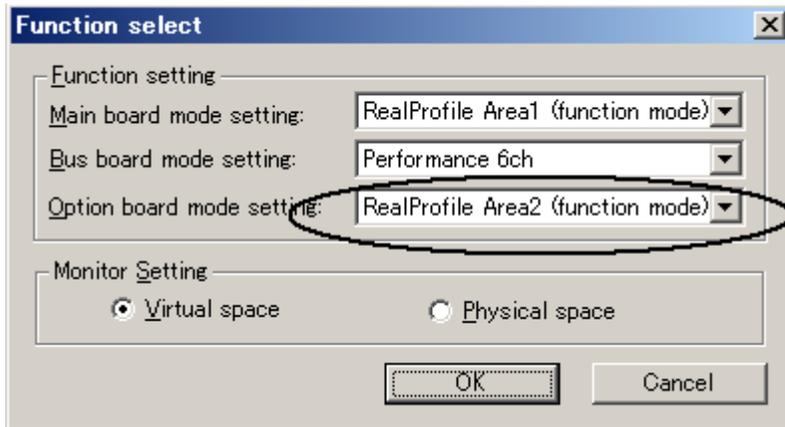


図 2.21 [Function select]ダイアログボックス

- [RealProfile Area2 (function mode)]選択時  
各関数の実行時間累積を計測します。サブルーチン実行時間は含みません。
- [RealProfile Area2 (nest mode)]選択時  
各関数の実行時間累積を計測します。サブルーチン実行時間も含みます。
- [Coverage (8M)]選択時  
8MB空間のC0カバレッジ情報を取得します。

#### 【留意事項】

拡張プロファイルユニットを E200F エミュレータに接続していない場合、本機能はグレー表示となります。

## 2.7.4 モニタ機能の変更

[Function select]ダイアログボックスの[Monitor Setting]グループボックス内に、2つのラジオボタンがあります。

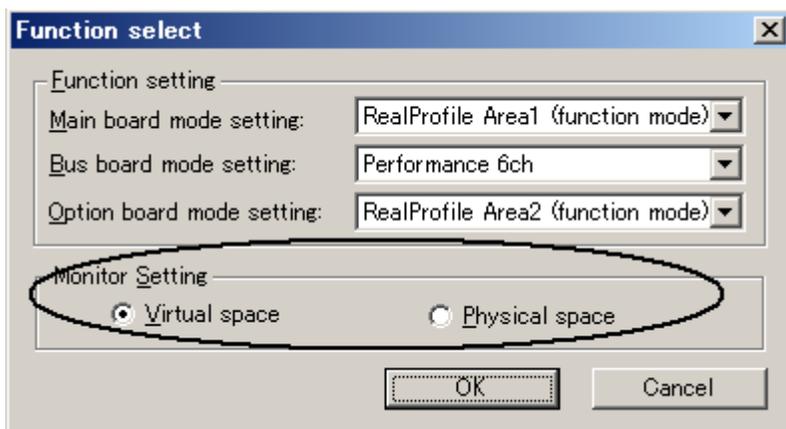


図 2.22 [Function select]ダイアログボックス

- [Virtual space]ラジオボタン  
このラジオボタンを選択した場合、モニタ機能は論理アドレスとして表示します。  
CPUアクセスが発生せずに値が変更になった場合には、変更後の値を表示できません。
- [Physical space]ラジオボタン  
このラジオボタンを選択した場合、モニタ機能は物理アドレスとして表示します。



---

## 3. ハードウェア仕様

---

### 3.1 仕様一覧

外形寸法と質量を以下に表 3.1 に示します。

表 3.1 外形寸法と質量

項番	項目	仕様
1	E200F メインユニット外形寸法	185 × 130 × 45 ( mm )
2	E200F メインユニット質量	321 ( g )
3	E200F トレースユニット外形寸法	120 × 90 × 1.6 ( mm )
4	E200F トレースユニット質量	104 ( g )
5	E200F 拡張プロファイルユニット外形寸法	115 × 89 × 1.6 ( mm )
6	E200F 拡張プロファイルユニット質量	95 ( g )

### 3.2 E200F エミュレータ内インタフェース回路

図 3.1、図 3.2 に E200F エミュレータ内インタフェース回路(36 ピンインタフェース)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。

図 3.3、図 3.4 に E200F エミュレータ内インタフェース回路(38 ピンインタフェース)を示します。

【注】 UVCC を電源とする IC は、3.3V または H-UDI ポートコネクタからの VCC(1.8~3.3V)で駆動します。

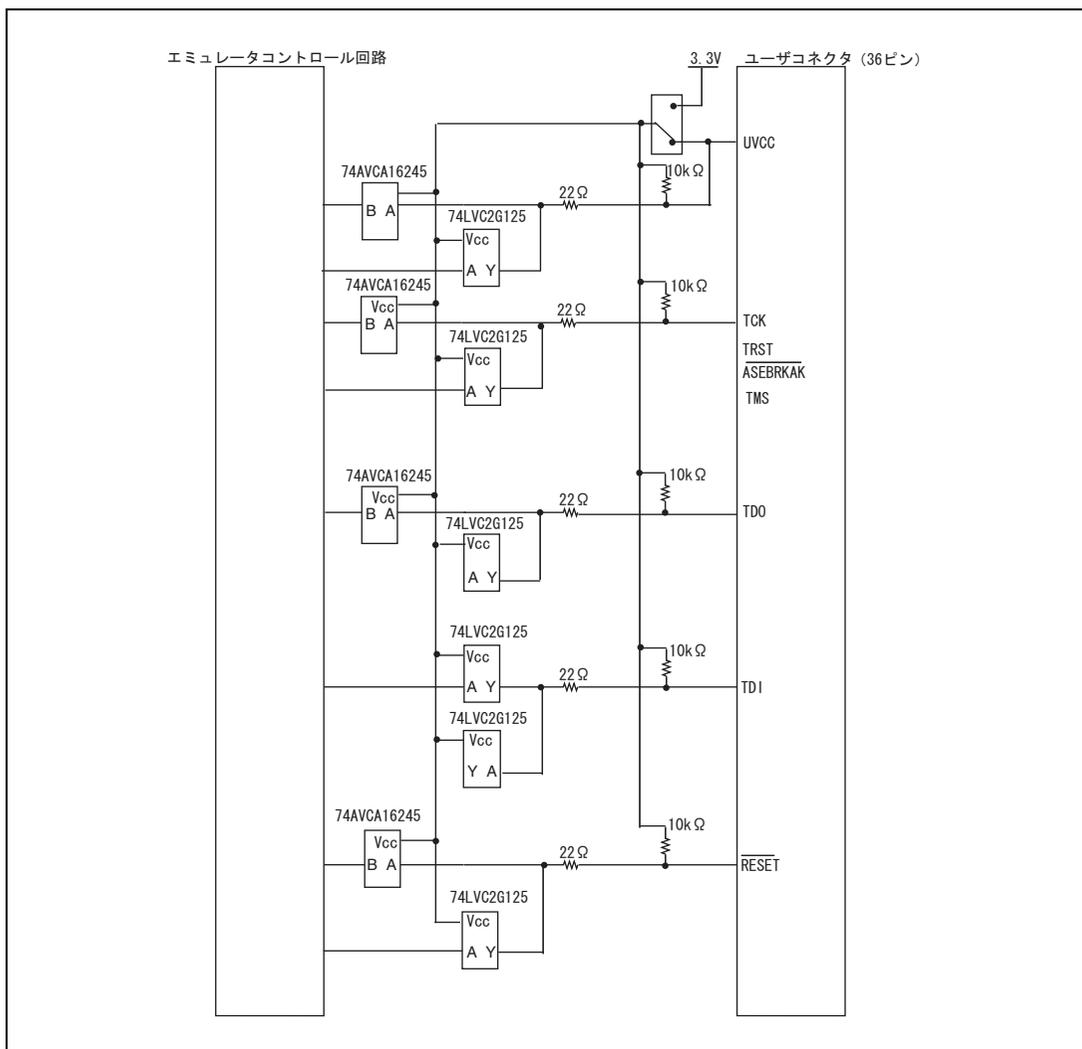


図 3.1 E200F エミュレータ内インタフェース回路(H-UDI 36 ピンインタフェース)

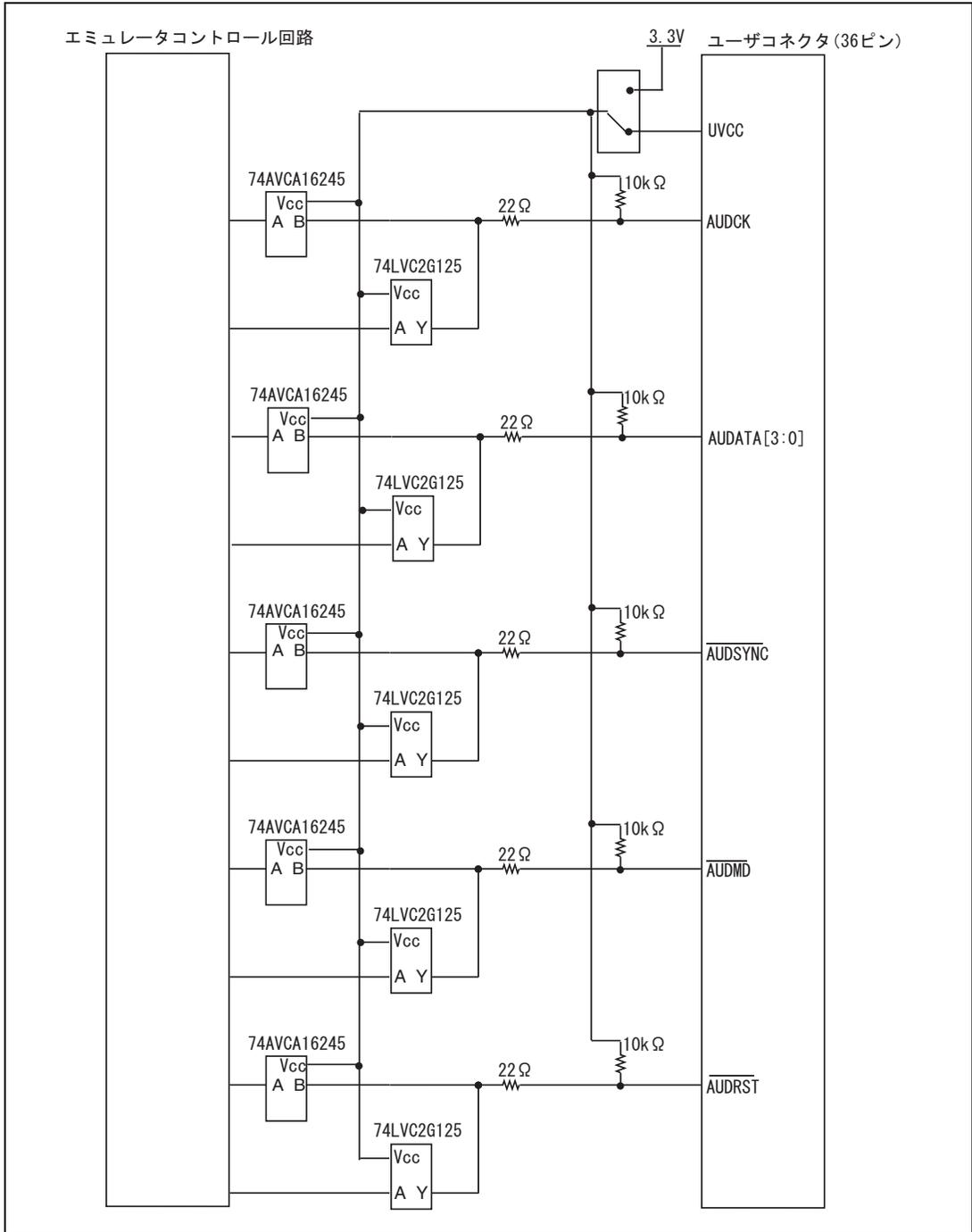


図 3.2 E200F エミュレータ内インタフェース回路(AUD 36ピンインタフェース)

### 3 ハードウェア仕様

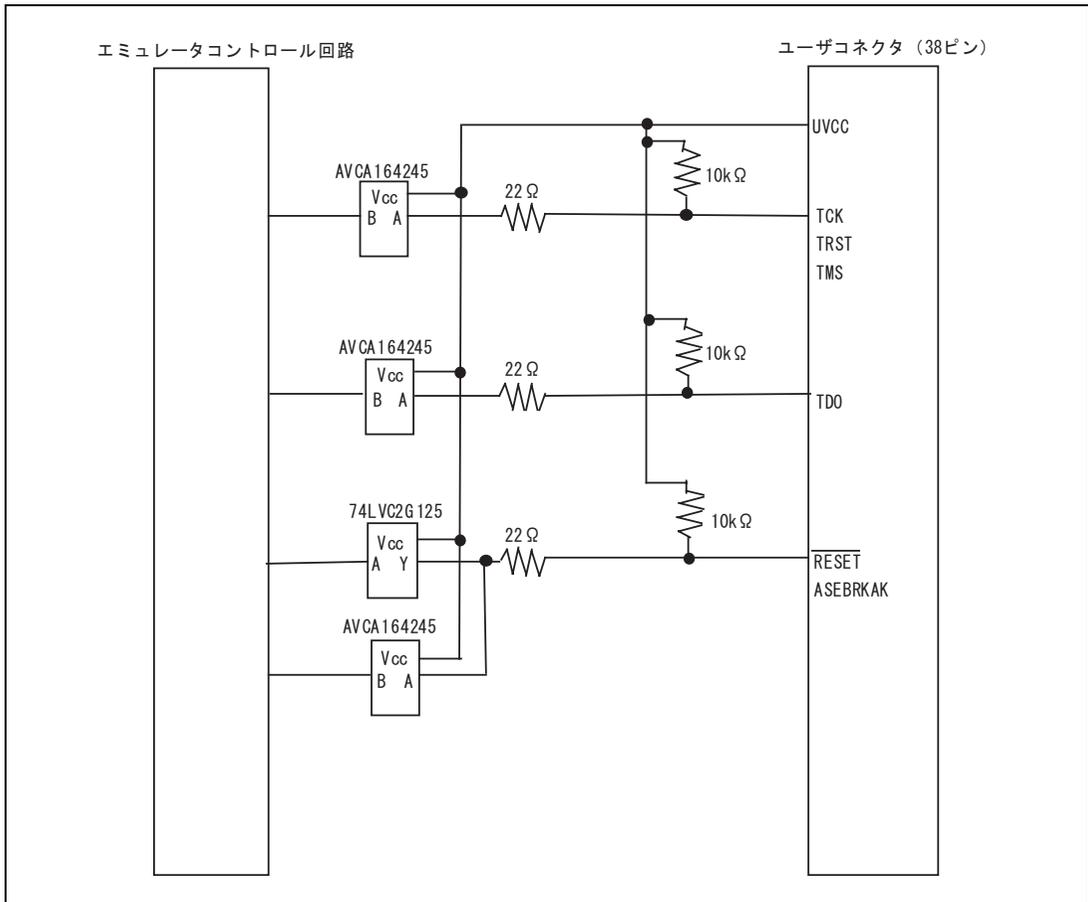


図 3.3 E200F エミュレータ内インタフェース回路(H-UDI 38ピンインタフェース)

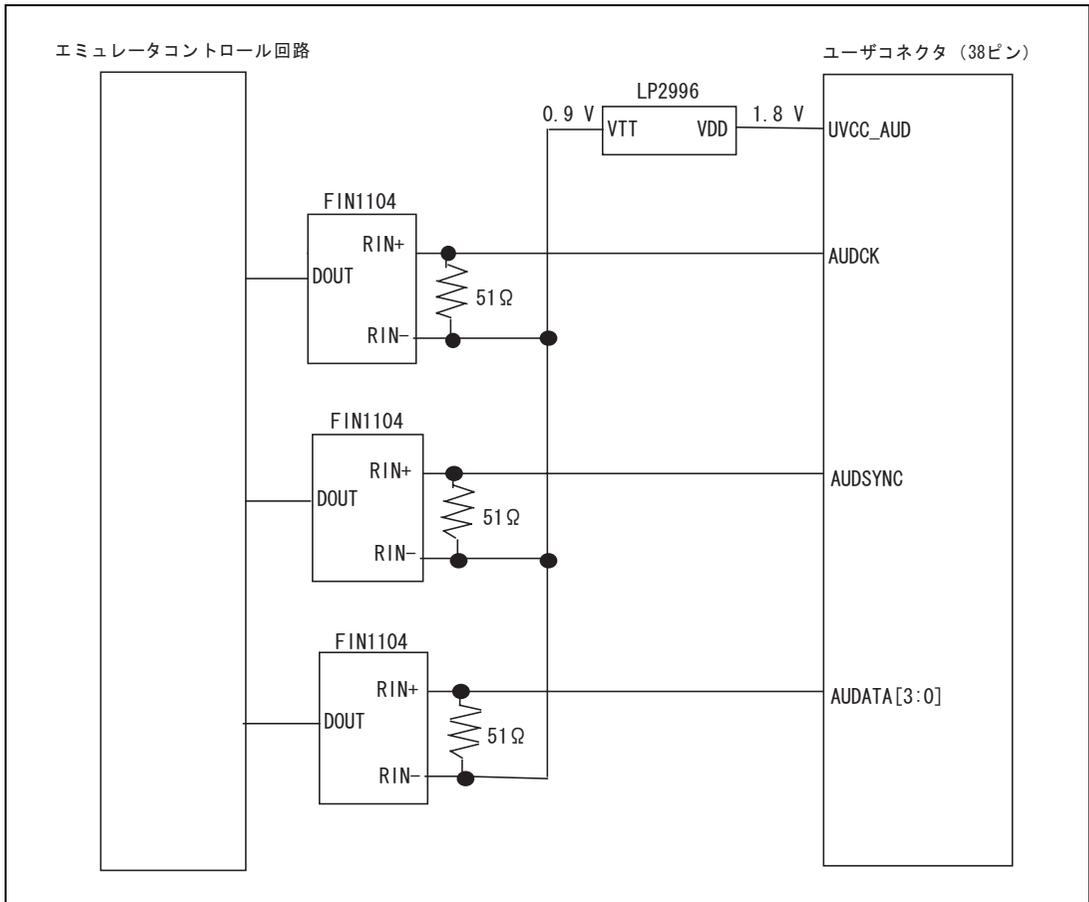


図 3.4 E200F エミュレータ内インタフェース回路(AUD 38 ピンインタフェース)

### 3.3 使用上の注意事項

#### 3.3.1 EMI ノイズの最小化

EMI ノイズを防ぐには、エミュレータをご使用になる前に、図 3.5 に示すようにトレースユニットを筐体に入れてご使用頂くようお願いいたします。

筐体の材質としては、例えば、鉄にニッケルメッキをしたもの、または樹脂の内側にニッケルメッキを行ったものを使用してください。

筐体のサイズは、トレースユニット、先端プローブ及びユーザシステムを入れられるサイズにしてください。



図 3.5 EMI ノイズ対策

EMI ノイズとは電磁妨害(Electrical Magnetic Interference)の略称です。

## 3.4 故障解析

診断のためのセットアップ方法、テスト実施方法、故障解析結果出力方法について説明します。

### 3.4.1 故障診断プログラムをインストールする

この節では、故障診断プログラムを設定する方法を説明します。

(1) ¥SOT¥R0E0200FxE00¥Setup.exe を開く  
CD-R の¥SOT¥R0E0200FxE00 ディレクトリから Setup.exe を実行してください。

R0E0200F0EMU00 の場合は、¥SOT¥R0E0200F0EMU00¥Setup.exe を  
R0E0200F2EMU00 の場合は、¥SOT¥R0E0200F2EMU00¥Setup.exe の setup.exe を実行してください。  
インストールウィザードに従いインストールを行ってください。

### 3.4.2 診断プログラムを実行する

故障診断を実行する方法を説明します。

- (1) ホストコンピュータと E200F エミュレータを接続してください。ユーザシステムとは接続しないでください。
- (2) E200F エミュレータの電源を ON してください。
- (3) [スタート]メニューの[プログラム]から[E200F TM] ->[E200F Fx TM] ( x : 0 または 2 ) を選択してください。
- (4) E200F 診断プログラムが起動します。図 3.6 に示すスタート画面になります。
- (5) COMPONENT を選択します。このとき TARGET は選択しないでください。
- (6) USER TEST MODE ボタンを選択してください。QA TEST MODE は選択しないでください。

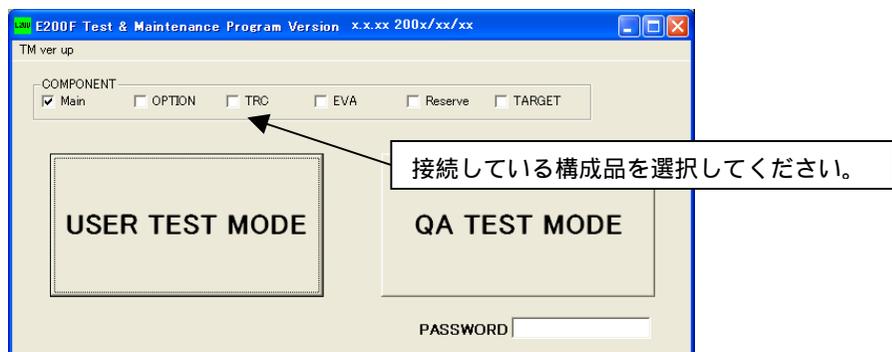


図 3.6 E200F システムオペレーションテストのスタート画面

### 3 ハードウェア仕様

---

- (7) USER TEST MODE に入ると図 3.7 の画面になります。
- (8) UNIT ONLY を選択するとテスト可能な項目が選択されます。
- (9) START ボタンを選択してください。
- (10) FPGA のロードが完了するとテストが開始されます。

【注】 COMPONENT の選択をする場合、TARGET は選択しないでください。  
( TARGET とは、出荷テスト専用の治具のことです。 )

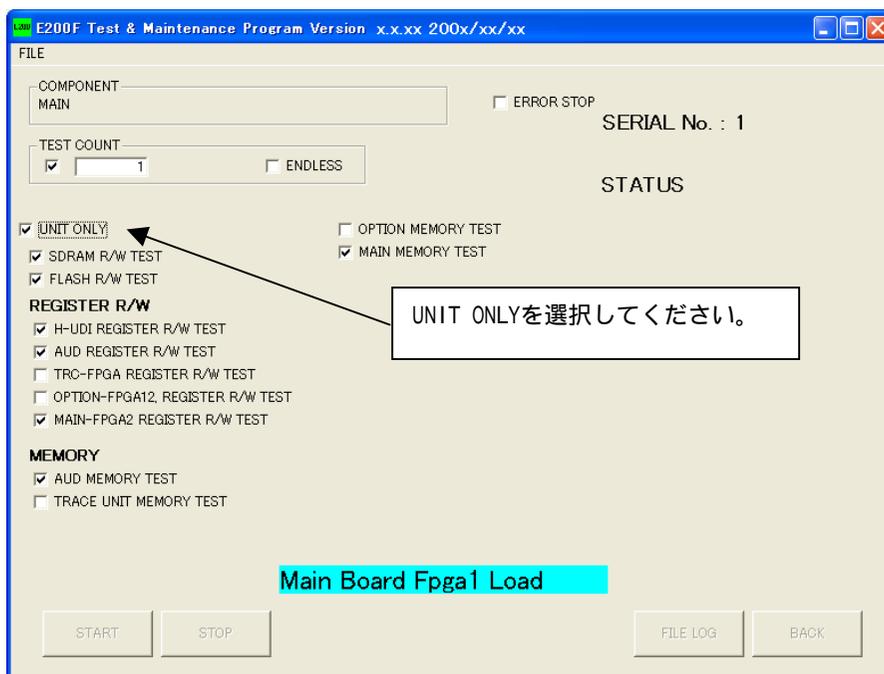


図 3.7 テスト選択画面

【注】 テストの実行中は、USB ケーブルを抜かないでください。

- (11) テストが実行されると STATUS 表示の左隣に Testing と表示されます。
- (12) テストが正常に終了すると Test OK と表示されます。

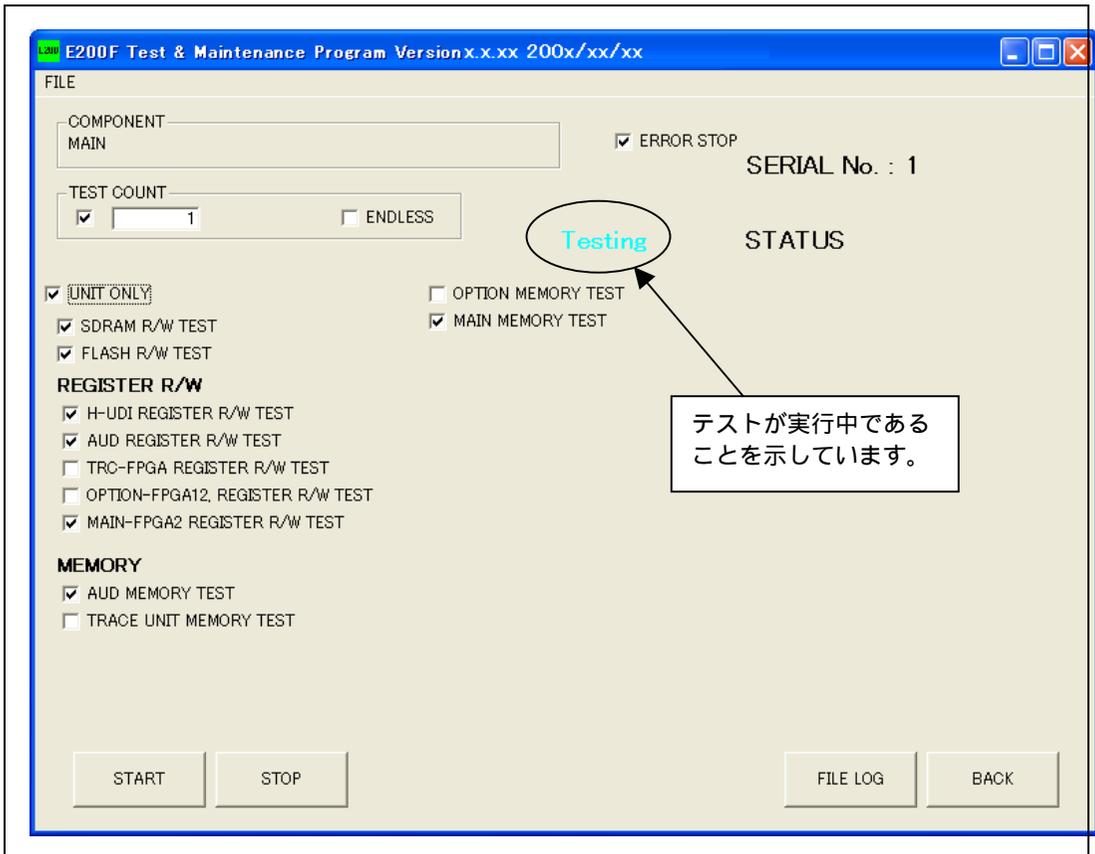


図 3.8 Testing 表示画面

#### 3.4.3 ログファイルを作成する

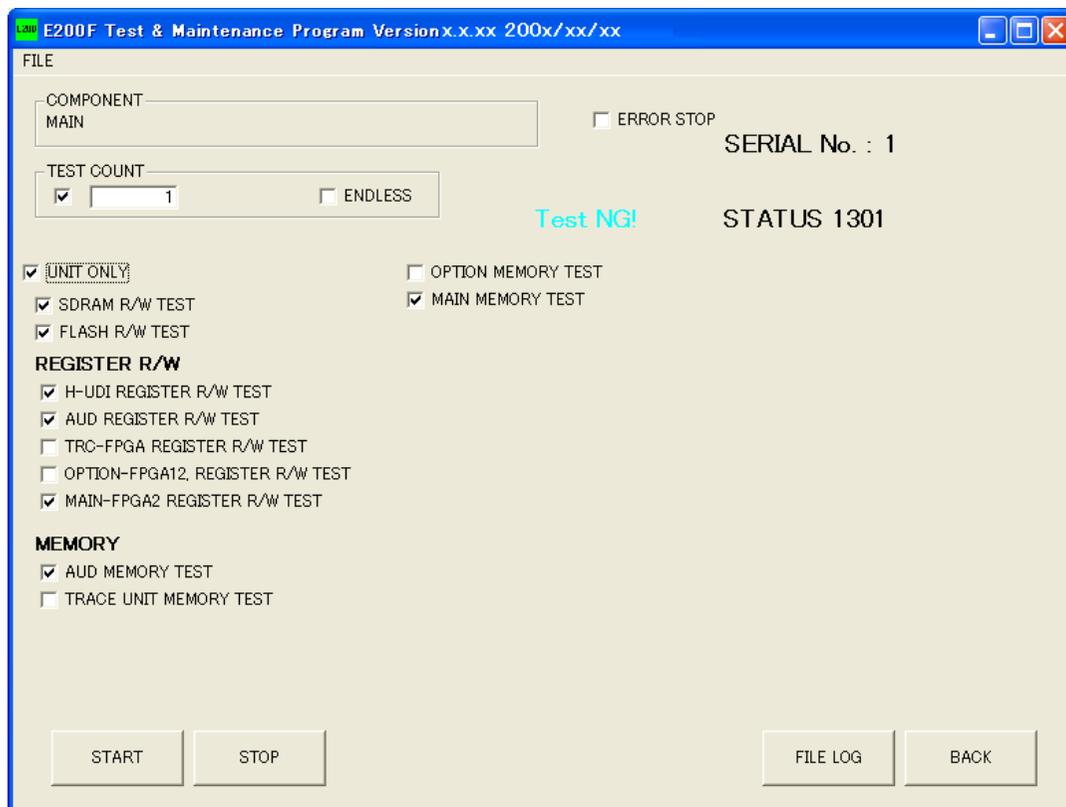


図 3.9 Test NG 画面

図 3.9 のように NG になると Test NG! と表示されます。テストが NG になる場合は、エミュレータの故障の可能性がります。下記の手順で作成された LOG ファイルを担当営業宛に送付してください。

##### LOG ファイルの作成方法

- FILE LOG ボタンを押してください。
- ¥SOT ¥R0E0200FxEMU00 ディレクトリの下に E200FTM.LOG ファイルが作成されます。
- 担当営業宛に LOG ファイルを送付してください。

E200F TM エラーログ例

\*\*\* E200F Emulator T/M ERROR LOG \*\*\*

TM Version x.x.xx

Serial Number 000x

Date 200x/xx/xx

No.	STATUS	NG Address	NG Data	
02	1301	A0000000	7EC0A5F	NG
03	0304	A8000108	FFFFFFFF	NG



---

## 4. デバッグの準備をする

---

### 4.1 システムチェック

次に、ソフトウェアを実行し、E200F エミュレータが正しく接続されていることをチェックします。  
ここでは、製品に添付のチュートリアル用ワークスペースを使用して起動します。

- (1) ホストマシンと E200F エミュレータを接続してください。
- (2) E200F 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) ユーザシステムの電源を入れてください。
- (5) E200F エミュレータの電源を入れてください。
- (6) [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。

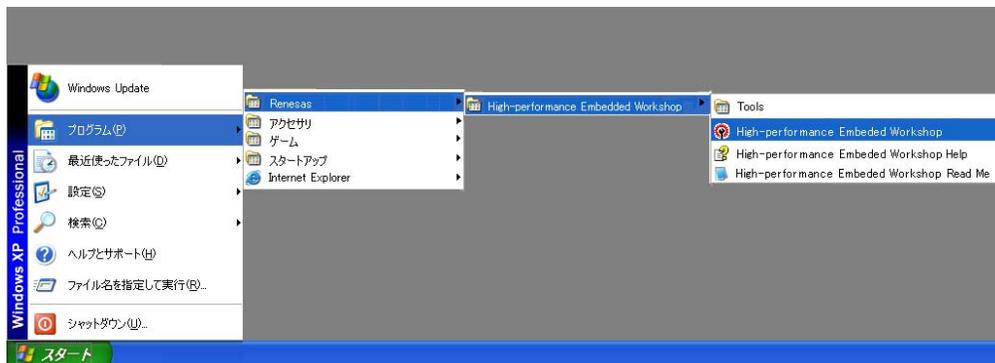


図 4.1 [スタート]メニュー

**【留意事項】**

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

## 4 デバッグの準備をする

---

(7) [ようこそ!]ダイアログボックスが表示されます。

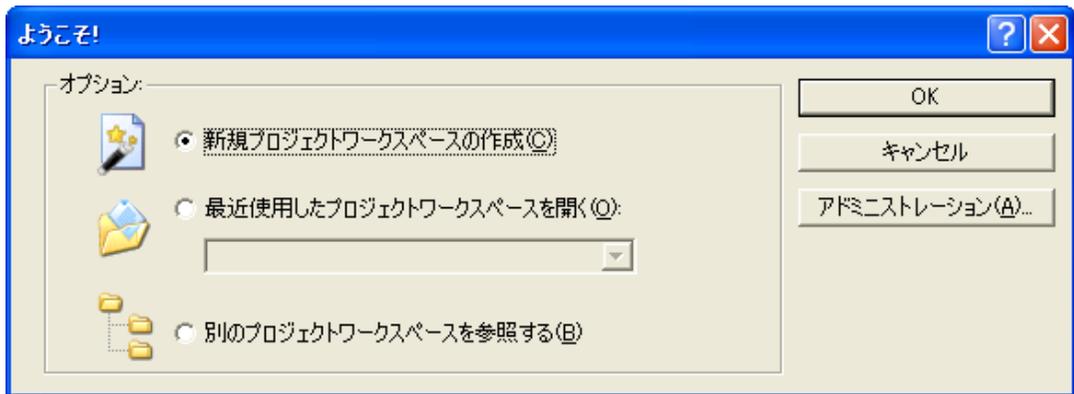


図 4.2 [ようこそ!]ダイアログボックス

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

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インストールドライブ>

¥WorkSpace¥Tutorial¥E200F¥xxxx¥Tutorial

xxxx は対象のデバイス名称を示します。

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。

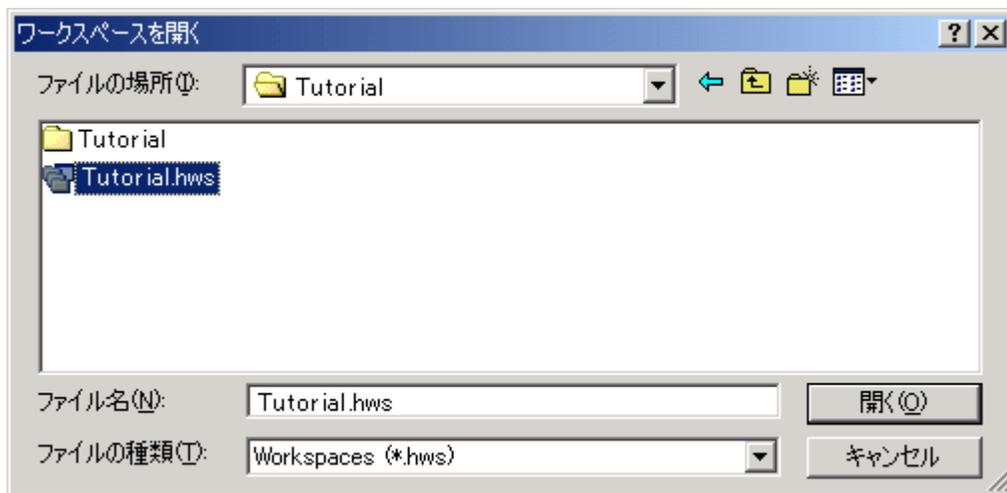


図 4.3 [ワークスペースを開く]ダイアログボックス

(8) [CPU select]ダイアログボックスが表示されます。



図 4.4 [CPU select]ダイアログボックス

ご使用の CPU を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。

以下のダイアログボックスが表示された場合は、ご使用のデバイスに E200F 本体が対応していない可能性があります。別冊の「SHxxxx ご使用時の補足説明 1.1 E200F エミュレータの構成」にて本体型名をご確認ください。

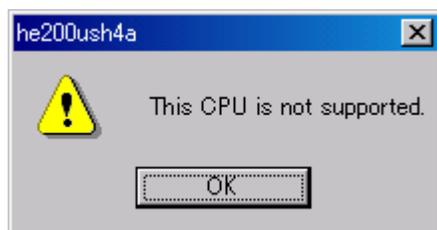


図 4.5 [This CPU is not supported]ダイアログボックス

- (9) [Function select]ダイアログボックスが表示されます。

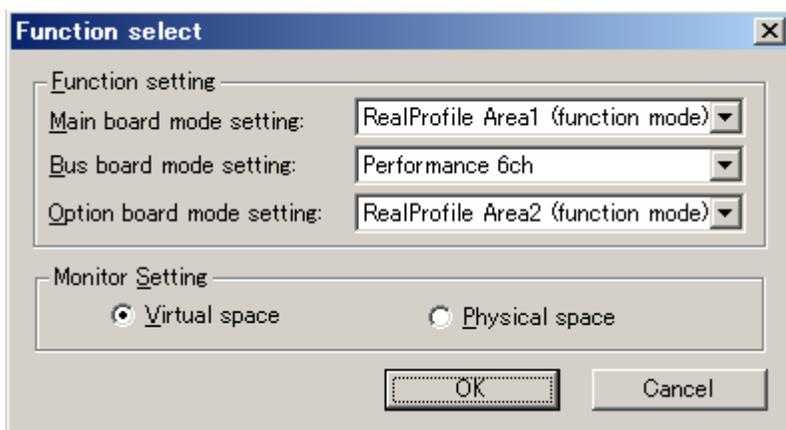


図 4.6 [Function select]ダイアログボックス

ここで、ご使用になる E200F エミュレータの機能を選択してください。  
ここで選択できる内容については、「2.7 設定の変更」を参照してください。

- (10) [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 4.7 [Connecting]ダイアログボックス

- (11) 図 4.7 に示すダイアログボックスが表示されます。

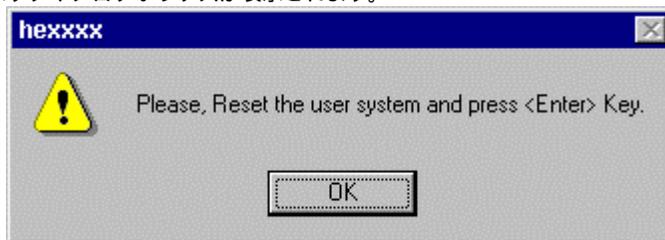


図 4.8 RESET 信号入力要求メッセージのダイアログボックス

- (12) 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。

## 4 デバッグの準備をする

---

(13) 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 4.9 [Can not find /RESET signal]ダイアログボックス

[無視]ボタンをクリックした場合、E200F エミュレータより CPU に内部リセットを発行し、起動を行うことができます。

製品によってはこの方法で起動することができません。別冊の SHxxxx ご使用時の補足説明「2.2 SHxxxx ご使用時のエミュレータ特有機能」をご確認ください。

(14) High-performance Embedded Workshop の[Output]ウィンドウに "Connected" と表示されたら、E200F エミュレータの起動は完了です。

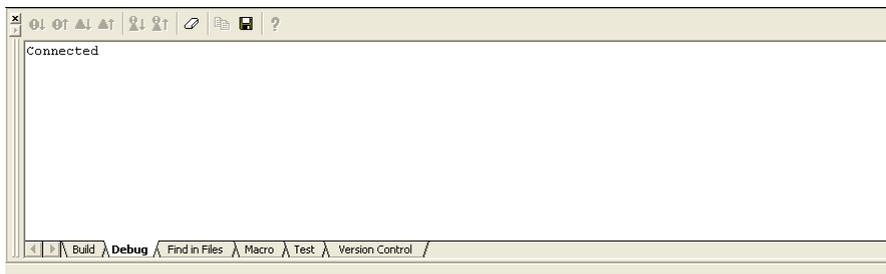


図 4.10 [Output]ウィンドウ

### 【留意事項】

1. E200F 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 4.11 [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、ユーザシステムの電源が入っていないか、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。ユーザシステムの電源と H-UDI ポートコネクタとの結線を確認してください。

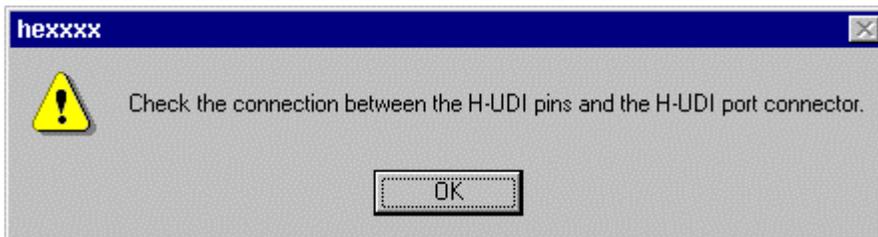


図 4.12 [Check the connection]ダイアログボックス

## 4 デバッグの準備をする

---

- (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 4.13 [COMMUNICATION TIMEOUT ERROR]ダイアログボックス



図 4.14 [INVALID ASERAM FIRMWARE!]ダイアログボックス



図 4.15 [Error JTAG boot]ダイアログボックス

2. その他の要因で E200F 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。

## 4.2 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) ホストマシンと E200F エミュレータ、ユーザシステムを接続し、E200F エミュレータと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。

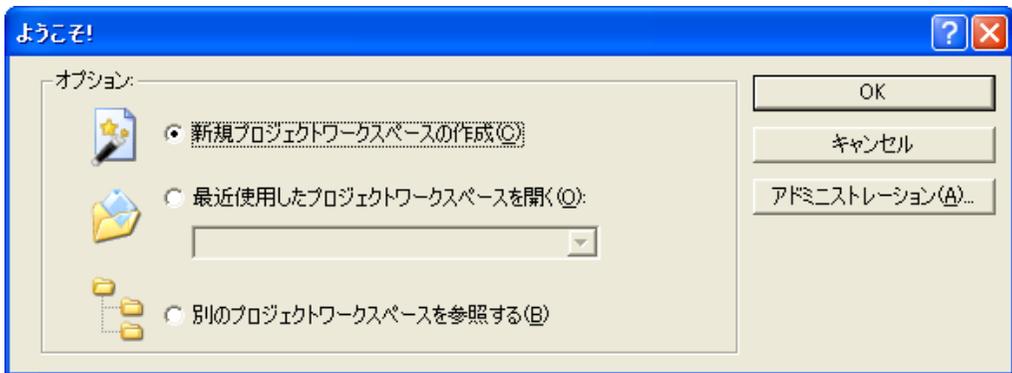


図 4.16 [ようこそ!]ダイアログボックス

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

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

### 4.2.1 新規にワークスペースを作成する場合(ツールチェイン未使用)

- (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。

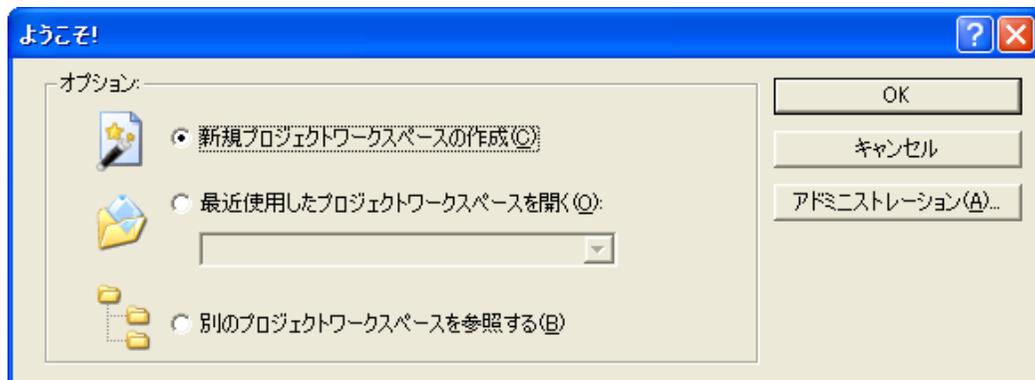


図 4.17 [ようこそ!]ダイアログボックス

(2) Project Generator が開始されます。

ここでは、ツールチェーン用の設定に関する説明は省略します。

ツールチェーンをご購入されていない場合、以下の画面が開きます。



図 4.18 [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あれば、入力する必要はありません。

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

## 4 デバッグの準備をする

(3) 次に、以下の画面が表示されます。

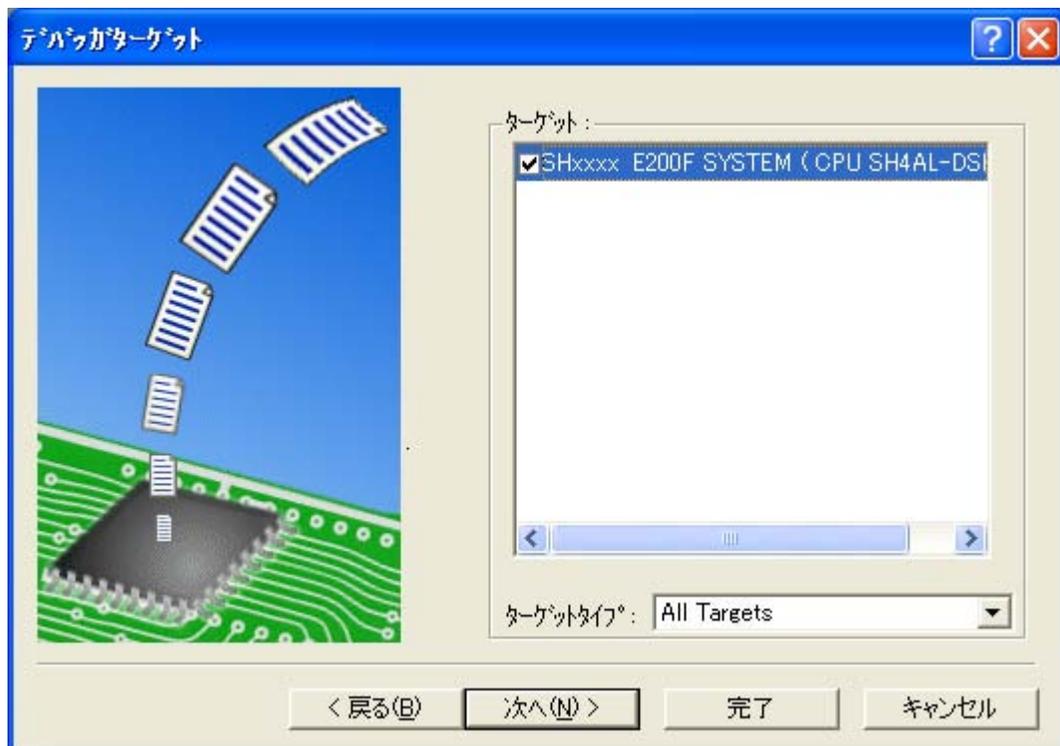


図 4.19 [デバッガターゲット]ダイアログボックス

ここで、該当する E200F 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。



図 4.20 [デバッガオプション]ダイアログボックス

これで E200F 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 E200F エミュレータが接続されます。接続中の操作については、「4.1 システムチェック」を参照してください。

#### 4.2.2 新規にワークスペースを作成する場合(ツールチェイン使用)

- (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。

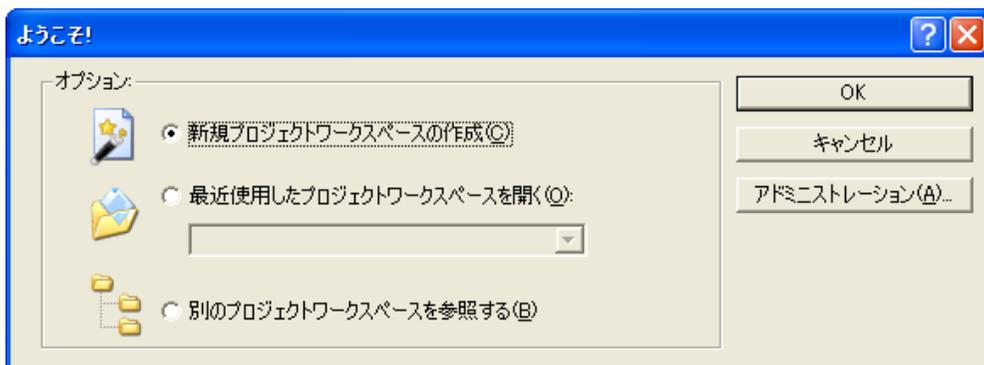


図 4.21 [ようこそ!]ダイアログボックス

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

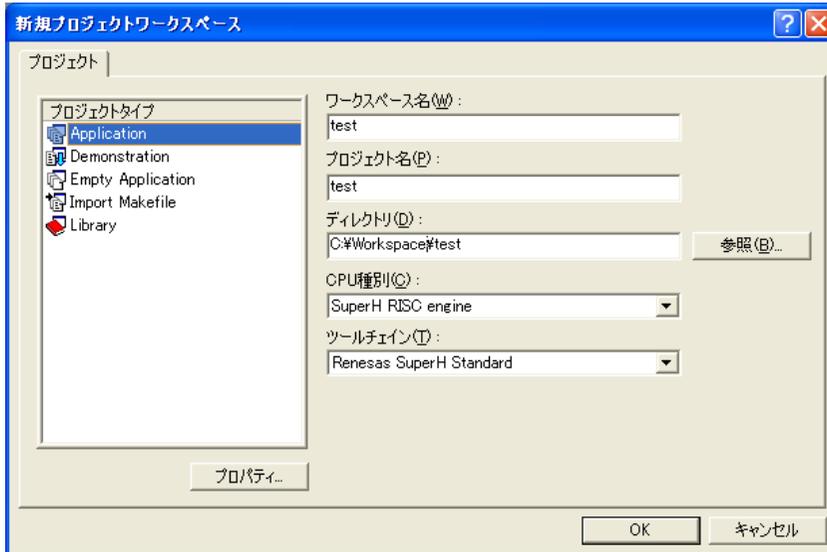


図 4.22 [新規プロジェクトワークスペース]ダイアログボックス

- [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あれば、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。
- [ツールチェーン]ドロップダウンリストボックス  
ツールチェーンをご使用になる場合、該当するツールチェーン名を選択してください。  
使用しない場合、[None]を選択してください。
- [プロジェクトタイプ]リストボックス  
使用したいプロジェクトタイプを選択してください。

**【留意事項】**

E200F 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使用する場合、“Printf 文”を削除してください。

## 4 デバッグの準備をする

- (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 4.23 [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 E200F 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。

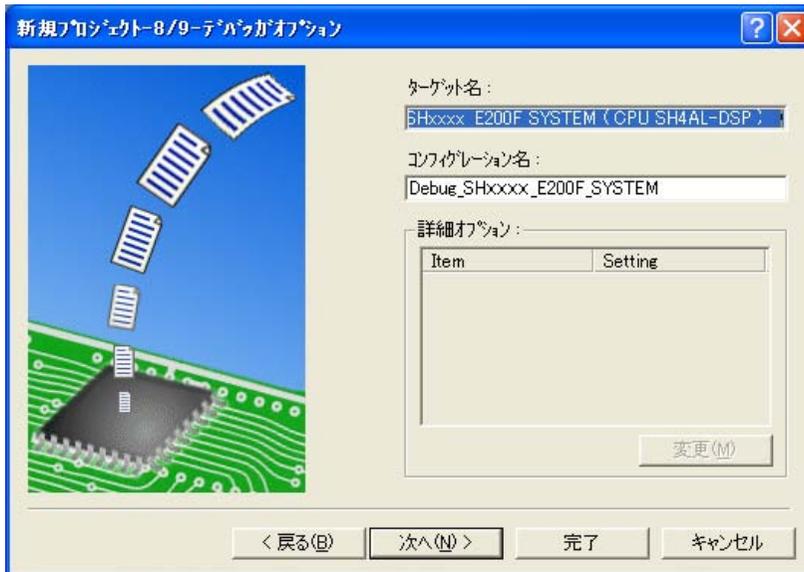


図 4.24 [新規プロジェクト-8/9-デバッガオプション]ダイアログボックス

これで E200F 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。

High-performance Embedded Workshop 起動後、E200F エミュレータを接続してください。

E200F 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。

E200F 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、「4.1 システムチェック」を参照してください。

(a) E200F エミュレータ起動時の設定を行ってから接続する方法

[オプション]メニューの[デバッグの設定]を選択し、[デバッグの設定]ダイアログボックスを開いてください。

ここで、ダウンロードモジュールや起動時に自動的に実行するコマンドチェーンなどを登録することができます。

[デバッグの設定]ダイアログボックスの詳細については、「4.3 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、E200F エミュレータが接続されます。

## 4 デバッグの準備をする

---

### (b) E200F 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

E200F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、E200F エミュレータを簡単に接続できます。

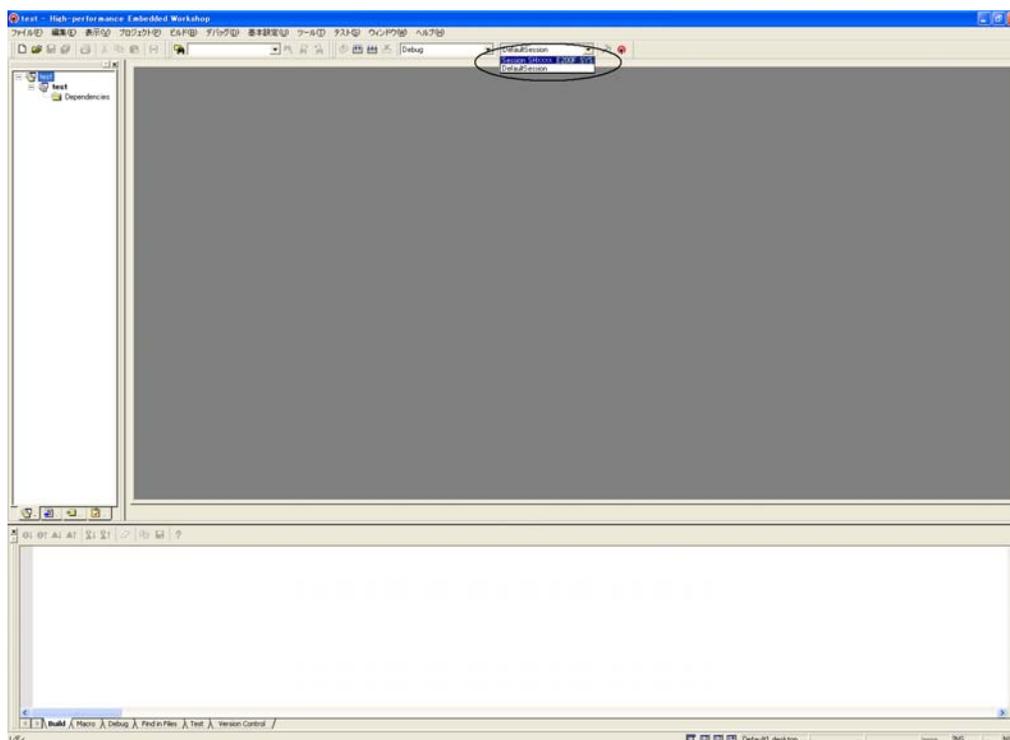


図 4.25 セッションファイルの選択

上記図中の、丸印の中にあるリストボックスから、「図 4.23 [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、E200F 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、E200F エミュレータが自動的に接続されます。

### 4.2.3 既存のワークスペースを指定する場合

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。

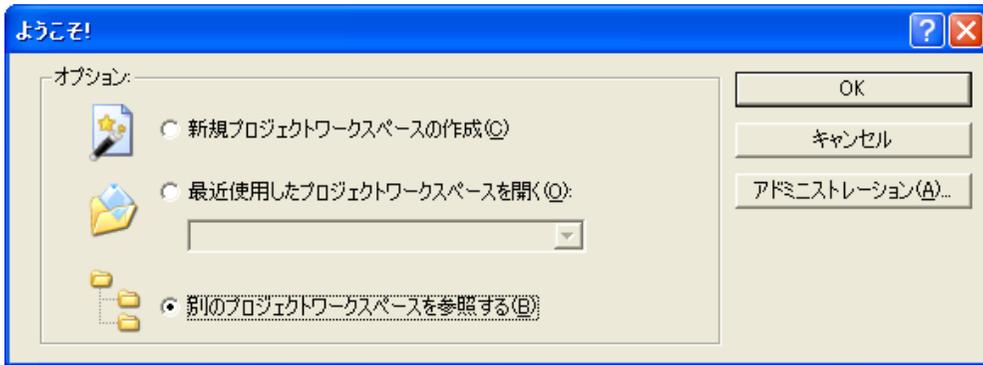


図 4.26 [ようこそ!]ダイアログボックス

[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。

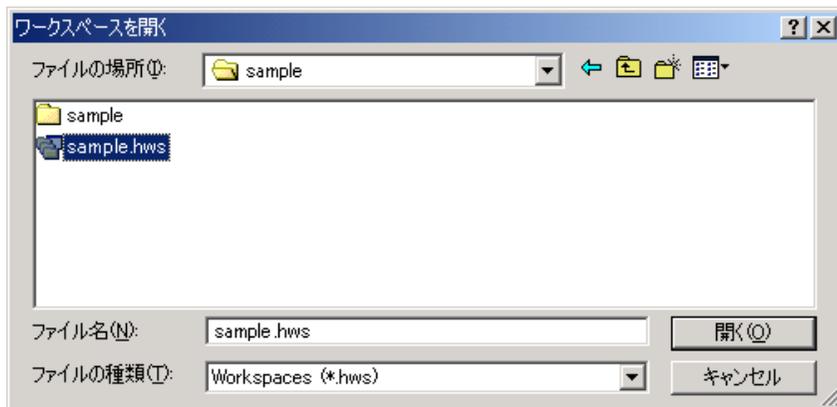


図 4.27 [ワークスペースを開く]ダイアログボックス

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。

### 4.3 エミュレータ起動時の設定

E200F エミュレータの起動時、コマンドチェーンの実行を自動的に行うことができます。  
また、ダウンロードするロードモジュールを複数登録することができます。  
登録したロードモジュールは、ワークスペースウィンドウに表示されます。

[オプション]メニューから[デバッグの設定]を選択してください。

[デバッグの設定]ダイアログボックスが開きます。

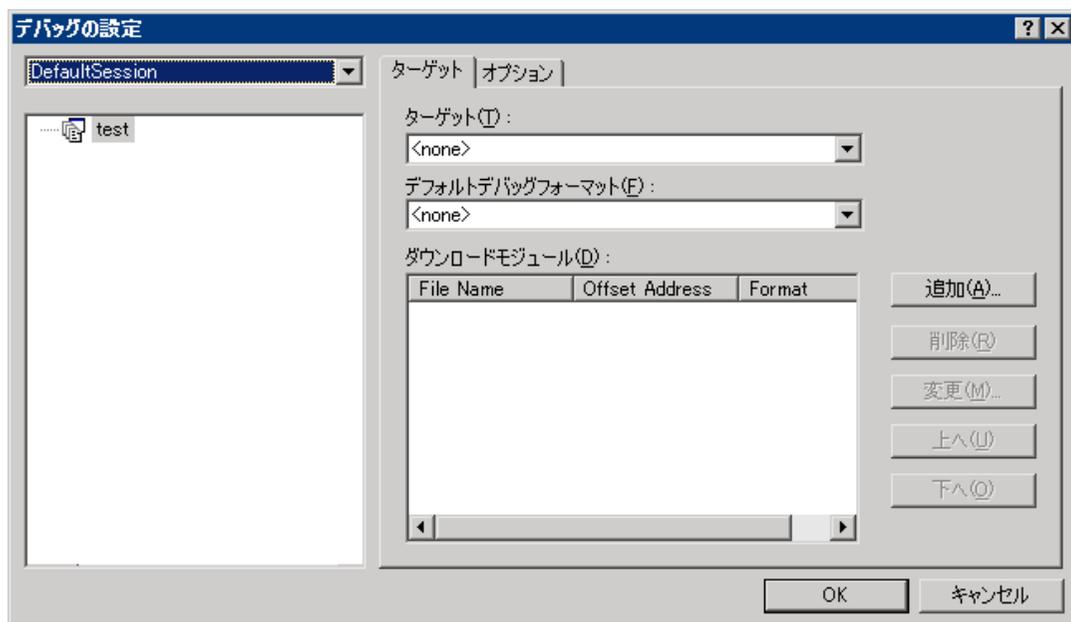


図 4.28 [デバッグの設定]ダイアログボックス ([ターゲット]ページ)

[ターゲット]ドロップダウンリストボックスで接続したい製品名を選択してください。

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

【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、「5.2 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。

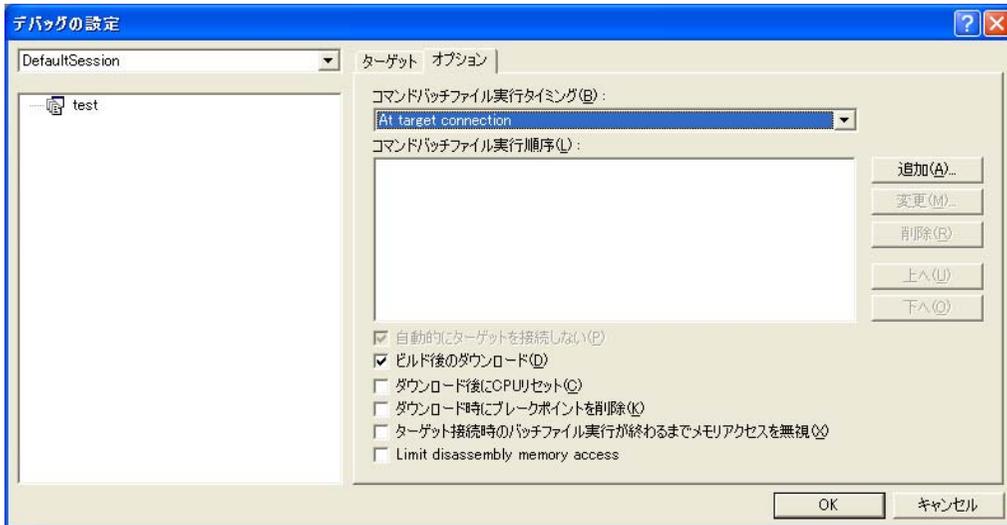


図 4.29 [デバッグの設定]ダイアログボックス ([オプション]ページ)

ここでは、指定したタイミングで自動的に実行するコマンドチェーンを登録します。

指定できるタイミングは以下 3 点です。

- エミュレータ接続時
- ダウンロード直前
- ダウンロード直後

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

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

### 4.4 デバッグセッション

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、デバッグオプションをセッションに保存することもできます。セッションには、デバッグプラットフォーム、ダウンロードするプログラム、各デバッグ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

セッションを選択するには、次の 2 通りの方法があります。

(1) 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 4.30) からセッションを選んでください。



図 4.30 ツールバーの選択

(2) ダイアログボックスから選択する

1. [オプション->デバッグセッション...]を選んでください。[デバッグセッション]ダイアログボックスを表示します(図 4.31)。

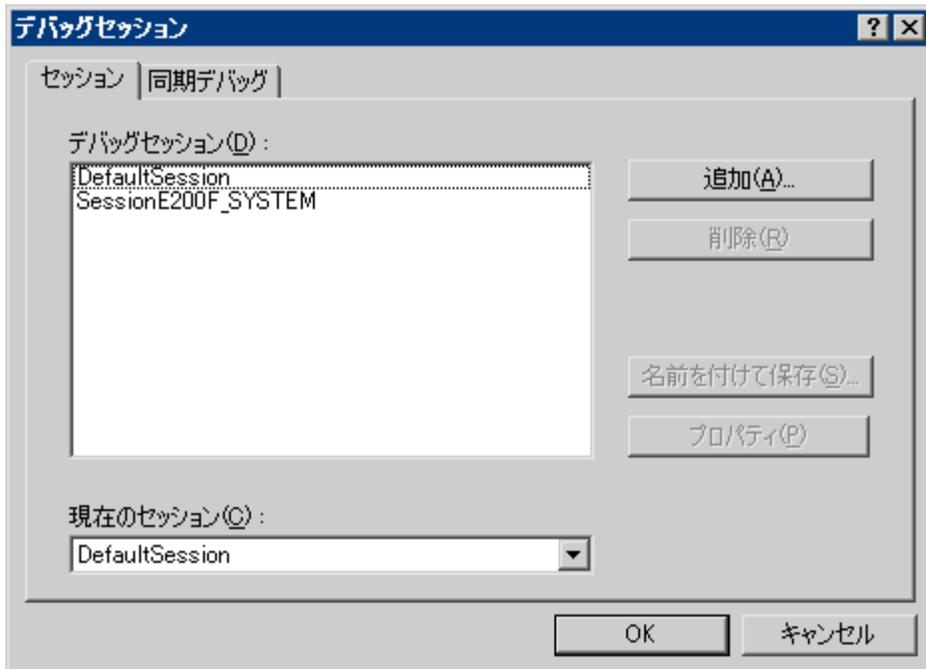


図 4.31 [デバッグセッション]ダイアログボックス

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

### 4.4.2 セッションの追加と削除

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

新しい空のセッションを追加する

1. [オプション->デバッグセッション...]を選んでください。[デバッグセッション]ダイアログボックスを表示します(図 4.31)。
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 4.32)。
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。

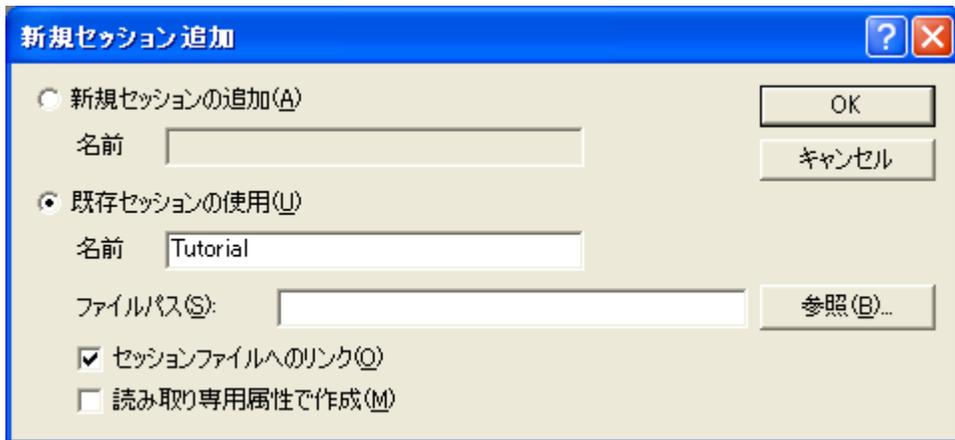


図 4.32 [新規セッション追加]ダイアログボックス

- 既存のセッションを新しいセッションファイルにインポートする
  1. [オプション->デバッグセッション...]を選んでください。[デバッグセッション]ダイアログボックスを表示します(図 4.31)。
  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 4.32)。
  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。
  6. [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。
  7. [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。
  8. [書き込み不可属性でのセッションファイル生成]チェックボックスをチェックした場合、リンクしたセッションファイルをリードオンリーで使用します。
  9. [OK]ボタンをクリックし、[デバッグセッション]ダイアログボックスを閉じてください。
  
- セッションを削除する
  1. [オプション->デバッグセッション...]を選んでください。[デバッグセッション]ダイアログボックスを表示します(図 4.31)。
  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。
  4. 現在のセッションを削除することはできません。
  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。

## 4 デバッグの準備をする

---

- セッションのプロパティを見る

1. [オプション->デバッグセッション...]を選んでください。[デバッグセッション]ダイアログボックスを表示します(図 4.31)。
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 4.33)。

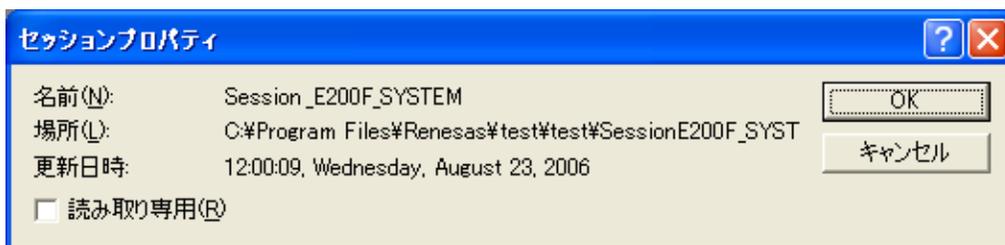


図 4.33 [セッションプロパティ]ダイアログボックス

- セッションをリードオンリーにする

1. [オプション->デバッグセッション...]を選んでください。[デバッグセッション]ダイアログボックスを表示します (図 4.31)。
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 4.33)。
4. [書き込み不可]チェックボックスをチェックしてください。リンクをリードオンリーにします。これは、デバッグ設定ファイルを共有する場合、およびデータを間違えて修正したくない場合に便利です。
5. [OK]ボタンをクリックしてください。

- セッションを別名で保存する

1. [オプション->デバッグセッション...]を選んでください。[デバッグセッション]ダイアログボックスを表示します (図 4.31)。
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 4.34)。
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。

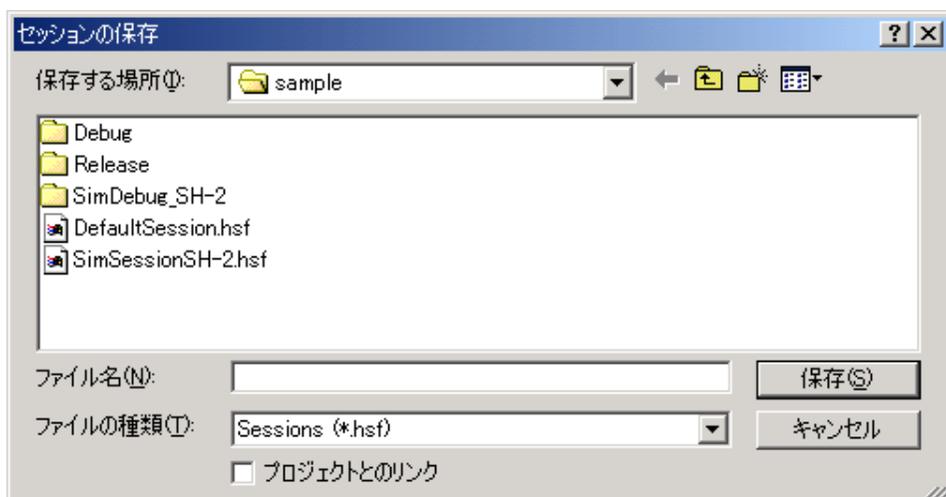


図 4.34 [セッションの保存]ダイアログボックス

### 4.4.3 セッション情報を保存する

セッションを保存するには

[ファイル->セッションの保存]を選んでください。

## 4.5 エミュレータの接続

エミュレータの接続には、以下の方法があります。

(1) E200F エミュレータ起動時の設定を行ってから接続する方法

[オプション]メニューの[デバッグの設定]を選択し、[デバッグの設定]ダイアログボックスを開いてください。ここで、ダウンロードモジュールや起動時に自動的に実行するコマンドチェーンなどを登録することができます。[デバッグの設定]ダイアログボックスの詳細については、「4.3 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、E200F エミュレータが接続されます。

(2) E200F 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

E200F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、E200F エミュレータを簡単に接続できます。

## 4 デバッグの準備をする

---

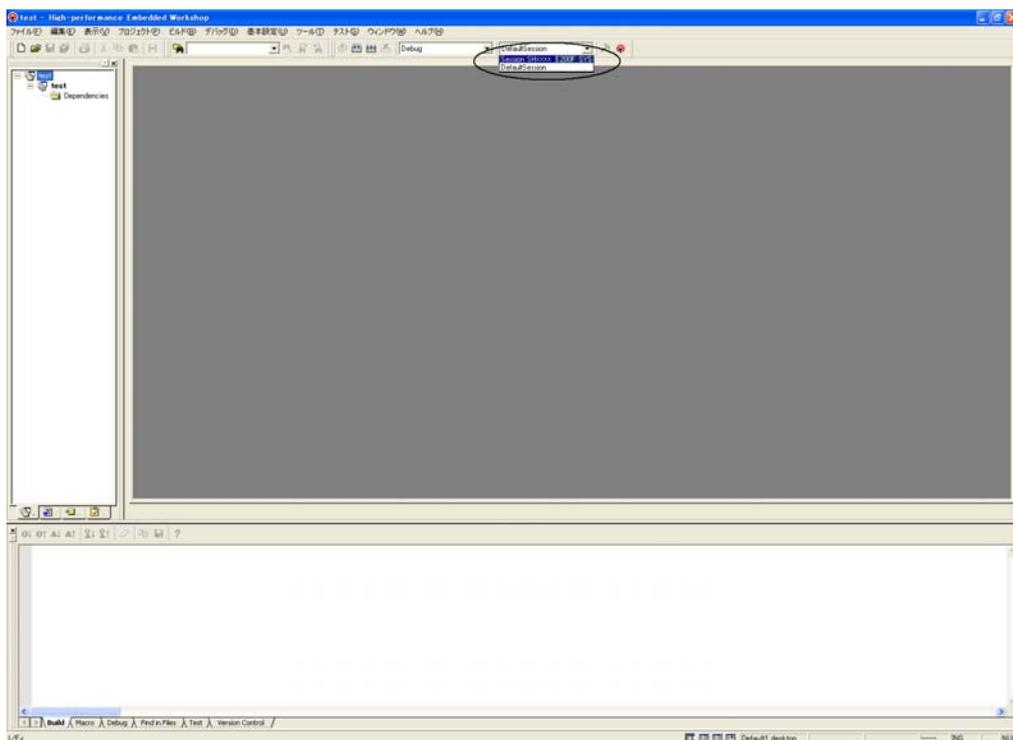


図 4.35 セッションファイルの選択

上記図中の、丸印の中にあるリストボックスから、「図 4.23 [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、E200F 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、E200F 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。

[ビルド->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】 [オプション->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 4.28 [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 E200F エミュレータが選択されている必要があります。

## 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。

(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 4.36 メッセージボックス

## 4.8 アンインストール

ご使用のホストマシンからインストールした E200F エミュレータ用ソフトウェアを削除します。

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

コントロールパネルの「アプリケーションの追加と削除」でもアンインストールは可能ですが、その場合は High-performance Embedded Workshop に含まれるすべてのツール(コンパイラも含む)が削除されますのでご注意ください。

1. High-performance Embedded Workshopを起動してください。
2. [ようこそ!]ダイアログボックスの、[アドミニストレーション...]ボタンをクリックしてください。

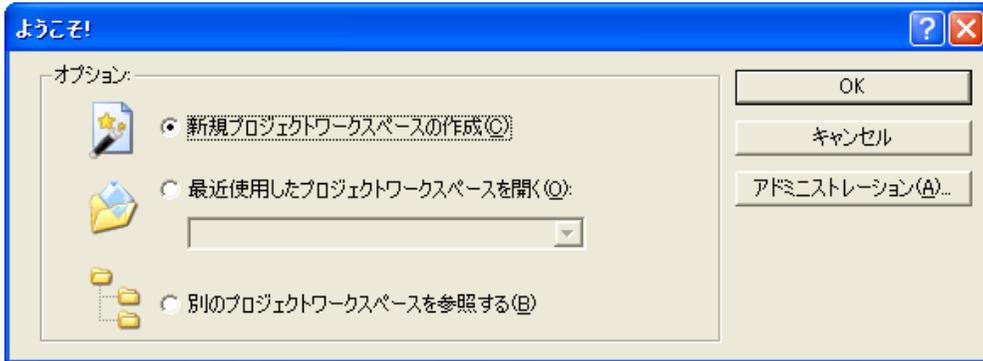


図 4.37 [ようこそ!]ダイアログボックス

3. [ツールアドミニストレーション]ダイアログボックスが開きます。

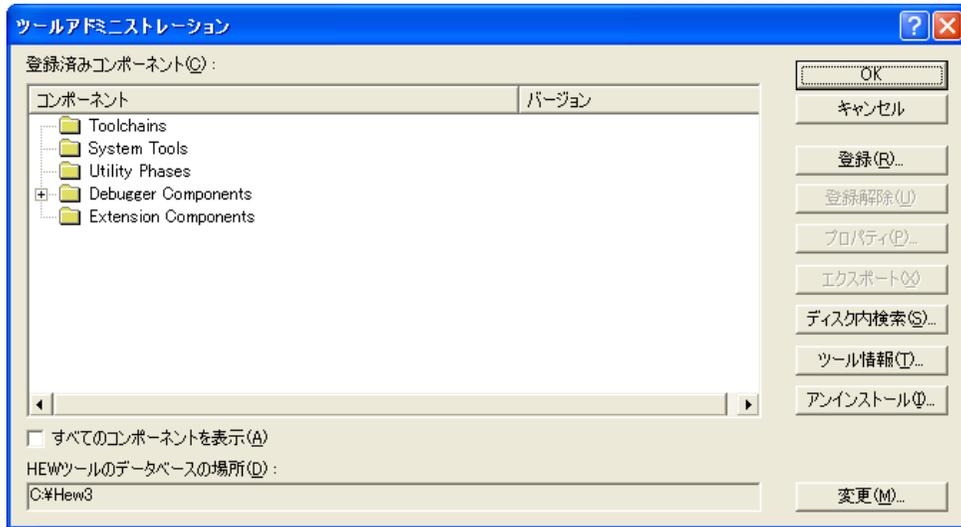


図 4.38 [ツールアドミニストレーション]ダイアログボックス

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



図 4.39 アンインストールする製品の強調表示

#### 4 デバッグの準備をする

---

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

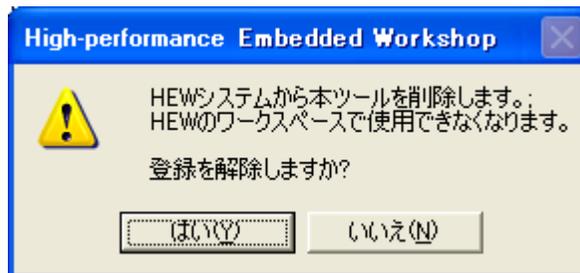


図 4.40 メッセージボックス

これで、High-performance Embedded Workshopへの登録解除は完了です。

次に、ホストマシン内からE200Fエミュレータ用のファイルを削除します。

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

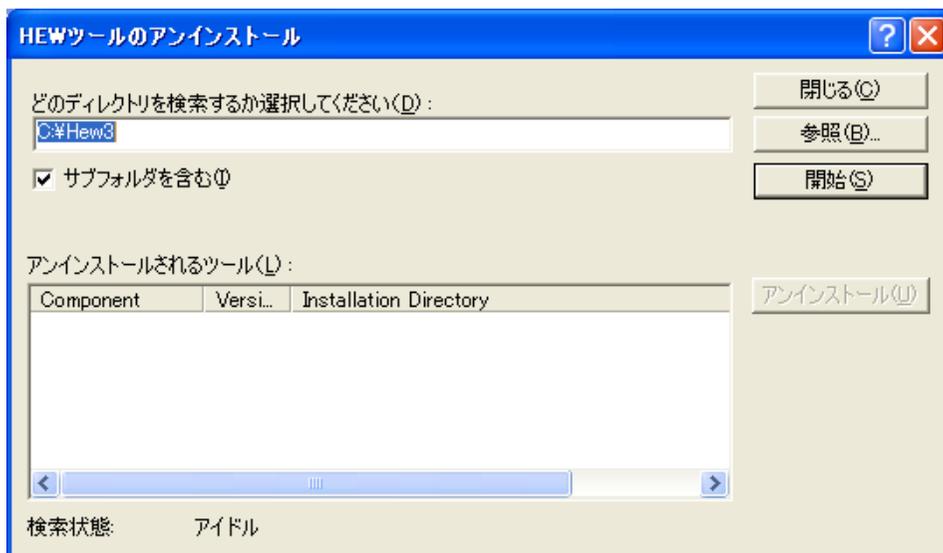


図 4.41 [HEW ツールのアンインストール]ダイアログボックス

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

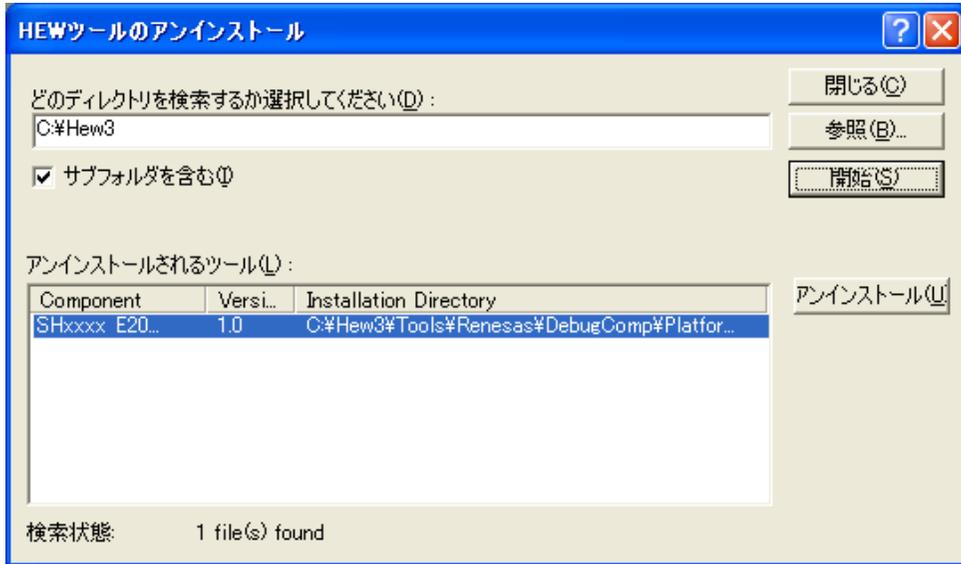


図 4.42 アンインストールする製品の強調表示

アンインストールしたい製品名を強調表示し、[アンインストール]ボタンを押してください。  
これでアンインストールは完了です。

### 注意

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



---

## 5. デバッグ

---

本章では、High-performance Embedded Workshop ユーザーズマニュアル「15 デバッグの使用」に記載されていない本製品特有の機能について説明します。

### 5.1 エミュレーション環境を設定する

この節では、エミュレーションを行うための環境を設定する方法を説明します。

#### 5.1.1 [Configuration]ダイアログボックスを開く

[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。

#### 5.1.2 [General]ページ

[General]ページでは、E200F エミュレータの基本設定を行います。

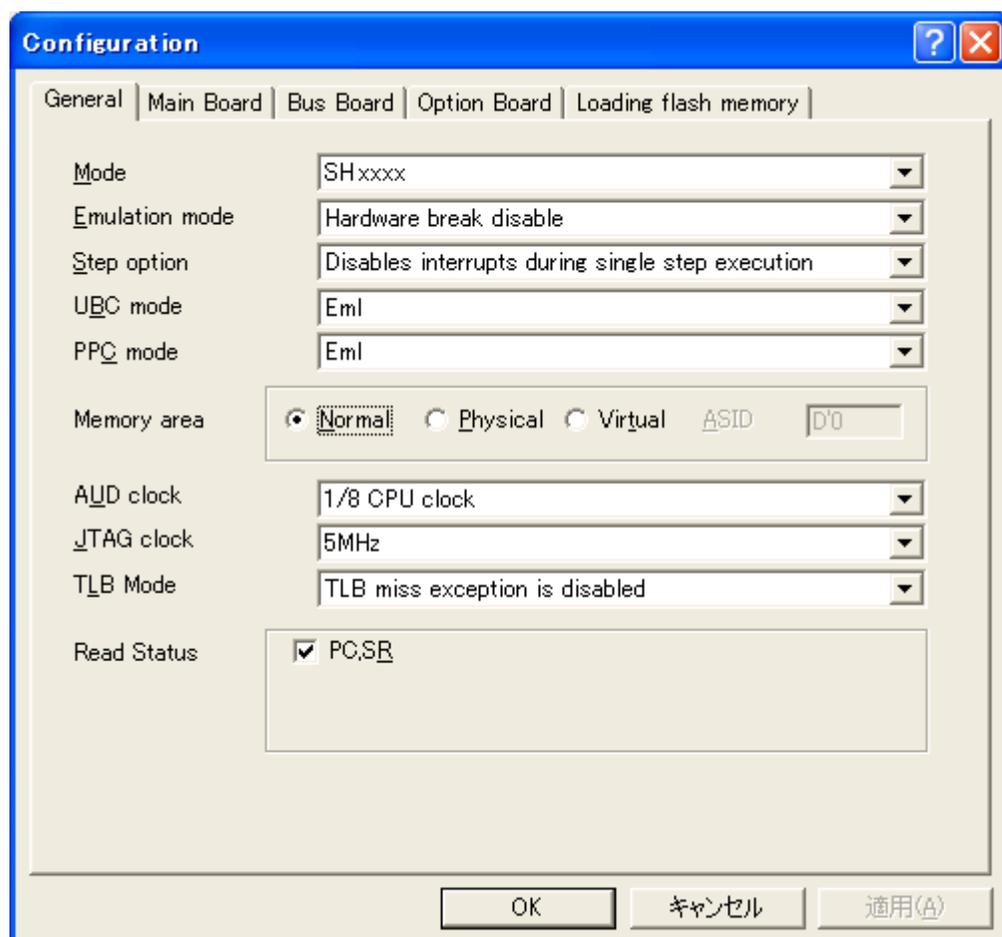


図 5.1 [Configuration]ダイアログボックス ([General]ページ)

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。
	Hardware break disable : E200F エミュレータ本体からのハードウェアブレークを無効にします。AUD Event ブレーク機能、BUS Event ブレーク機能、Other Event ブレーク機能、トレースオーバフローブレーク機能は無効になります。
	Hardware break enable : E200F エミュレータ本体からのハードウェアブレークを有効にします。この場合、ユーザプログラム実行中の Onchip Event 設定機能や物理アドレスのモニタ機能は無効になります。
	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。
[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。
[UBC mode]	UBC モードの設定を行います。
	EML : E200F エミュレータにより、Onchip Event 機能として使用します。
	User : UBC をユーザに開放します。【注 2】
[PPC mode]	PPC モードの設定を行います。
	EML : E200F エミュレータにより、パフォーマンスカウンタとして使用します。
	User : PPC をユーザに開放します。【注 2】
[Memory area]	メモリ区間のアドレス指定方法を設定します。
	Normal : MMU の状態によりアドレスを指定します。
	Physical : 物理アドレスで指定します。
	Virtual : 仮想アドレスで指定します。
[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 機能を使用できる E200F エミュレータを使用しているときのみ必要です。各製品の AUD clock 上限は、別冊の SHxxxx ご使用時の補足説明の「2.2.2 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。各製品の TCK 上限は、別冊の SHxxxx ご使用時の補足説明の「2.2.2 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。
[TLB Mode]	ブレーク中のメモリアクセス例外の許可/禁止を設定します。
	TLB miss exception is disabled : ブレーク中のメモリアクセス例外を禁止します。

## 5 デバッグ

---

TLB miss exception is enabled : ブレーク中のメモリアクセス例外を許可します。

[Read status] ユーザプログラム実行中の PC, SR の値をステータスバーに表示するかを設定します。

- 【注】
1. ブレーク中に発生した割り込みも含まれます。
  2. [UBC mode]で"User"を選択した場合、イベントポイントウィンドウのオンチップシートで設定できる、Ch10(IA\_OA\_R)と Ch11(IA\_OA\_CT\_R)は使用できません。また[PPC mode] で"User"を選択した場合、パフォーマンス解析機能の Ch1,Ch2 とプロファイル機能のオプション 1,2 は使用できません。

【留意事項】

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る場合があります。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。

### 5.1.3 [Main Board]ページ

[Main Board]ページでは、E200F エミュレータメインユニットの基本設定を行います。

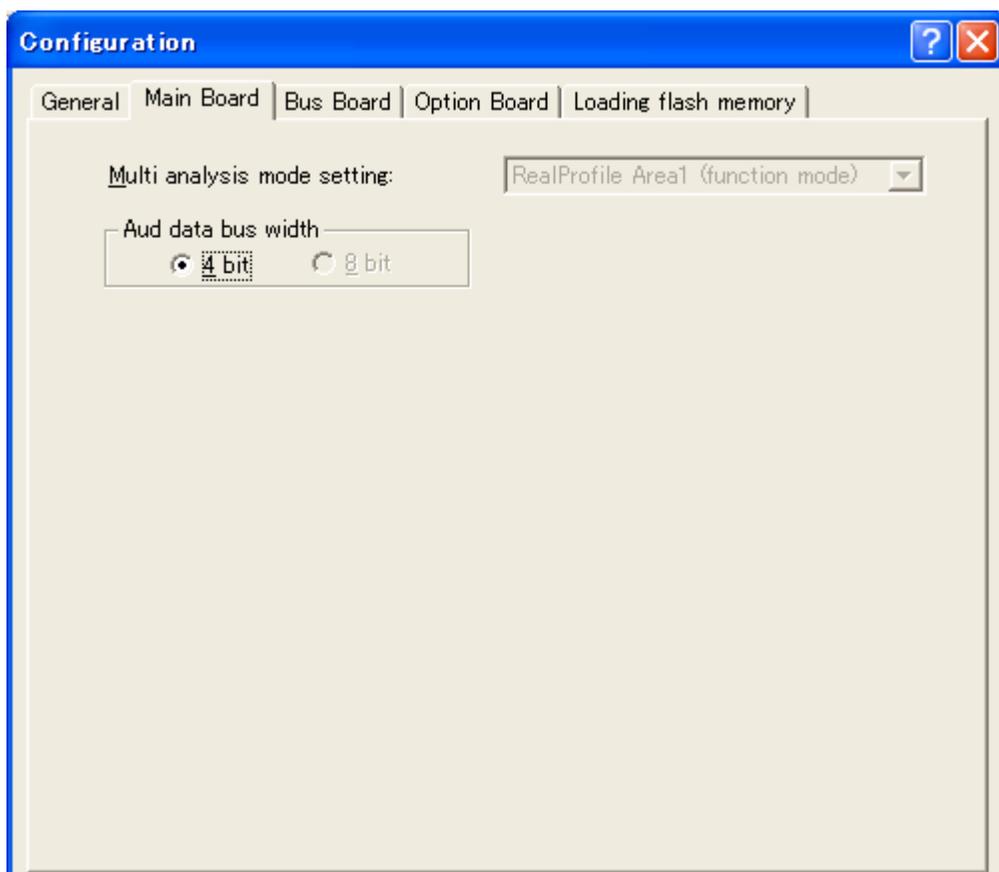


図 5.2 [Configuration]ダイアログボックス ([Main Board]ページ)

## 5 デバッグ

---

設定できる項目は以下の通りです。

[Multi analysis mode setting]	E200F メインユニットの機能を表示します。 ここでは、リアルタイムプロファイル機能か、カバレッジ機能か、I/O アナライザ機能か、どの機能が選択されているのかを表示します。ここでは設定を変更することはできません。変更した場合、E200F エミュレータの再起動が必要です。 RealProfile Area1 (function mode) : リアルタイムプロファイル機能(測定モード:関数モード)が選択されています。 RealProfile Area1 (nest mode) : リアルタイムプロファイル機能(測定モード:ネストモード)が選択されています。 Coverage (4M) : カバレッジ機能が選択されています。 <製品依存> : I/O アナライザ機能を使用します。
[AUD data bus width]	AUD のデータバス幅を設定します。【注】 バス幅は以下のいずれかから選択できます。 4bit, 8bit

【注】 製品によっては、バス幅を固定している場合があります。各製品の仕様は、オンラインヘルプを参照してください。

### 5.1.4 [Bus Board]ページ

[Bus Board]ページでは、E200F エミュレータトレースユニットの基本設定を行います。

**【留意事項】**

トレースユニットを E200F エミュレータに接続していない場合、本ページは表示されません。

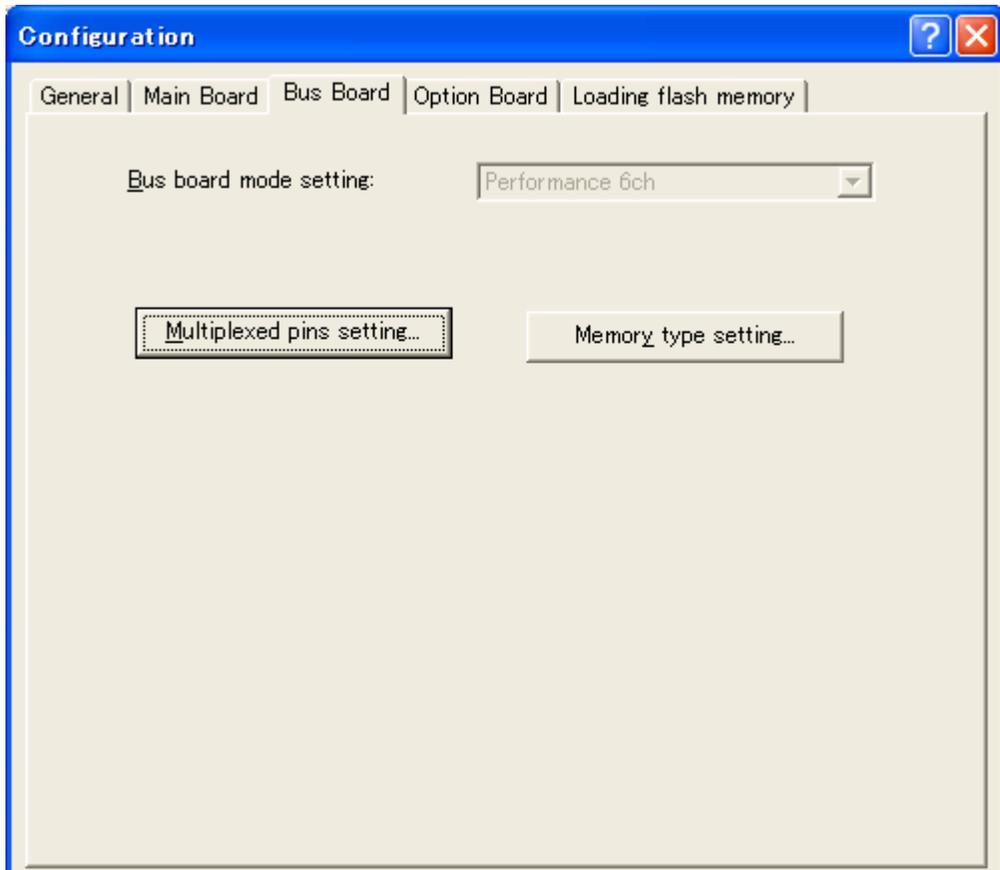


図 5.3 [Configuration]ダイアログボックス ([Bus Board]ページ)

## 5 デバッグ

設定できる項目は以下の通りです。

[Bus board mode setting]      トレースユニットの機能を表示します。

Trace/break 6ch (Trace 262144 cycles)    :

外部バスト्रेस/ブレイク機能 ( イベント検出チャンネル : 6 個 ) が選択されています。トレースできるバスサイクルは 262144 サイクルとなります。

Performance 6ch    :

外部バスパフォーマンス機能 ( 測定チャンネル : 6 個 ) が選択されています。

Emulation memory (4M, Trace 8192 cycles)    :

外部貸し出しメモリ機能 ( 4Mbyte × 1 ブロック ) が選択されています。トレースできるバスサイクルは、8192 サイクルとなります。

[Multiplexed pins setting...]

マルチプレクス端子の使用状態を選択します。

[Memory type setting...]

各エリアにつながっているメモリの種別や接続内容を選択します。バイト制御付き SRAM 以外は [Normal] を選択してください。

[Multiplex pins setting...] ボタンをクリックすると、[Multiplexed pins setting] ダイアログボックスが開きます。マルチプレクス端子の使用状態に従い、端子名を選択してください。

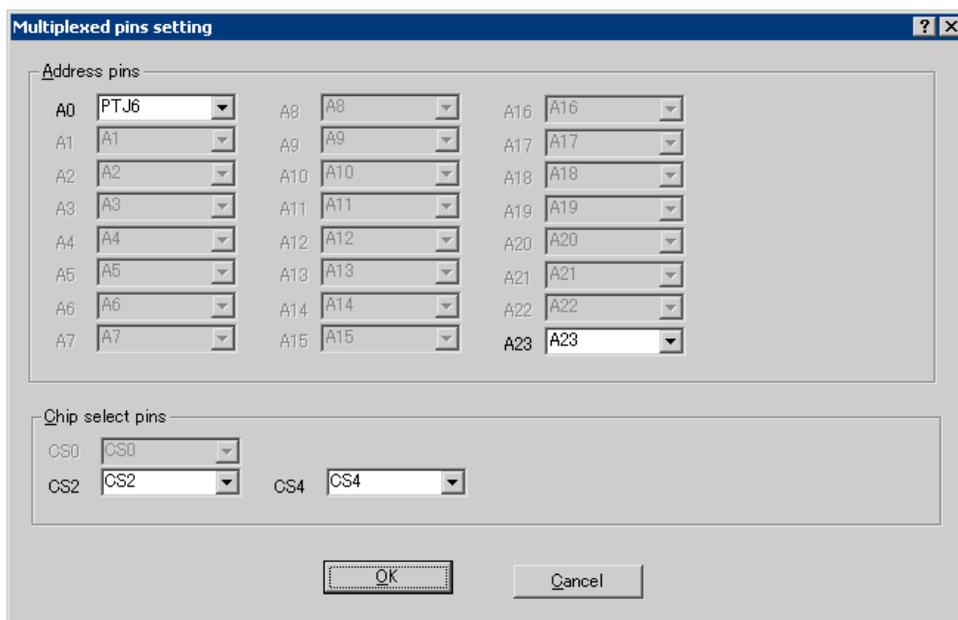


図 5.4 [Multiplexed pins setting] ダイアログボックス

【注】      本ダイアログボックスは、製品によって異なります。

[Memory type setting...]ボタンをクリックすると、[Memory type setting]ダイアログボックスが開きます。ボード上のメモリ種別を設定してください。

トレース取得対象がSDRAMである場合、Row アドレス、Column アドレスのビット数、CAS レイテンシについても本ダイアログボックスで設定してください。

また、データ幅についても、設定する必要がある場合がありますのでご注意ください。

正しく指定されない場合、バストレーシングウィンドウで正しく表示できない可能性があります。

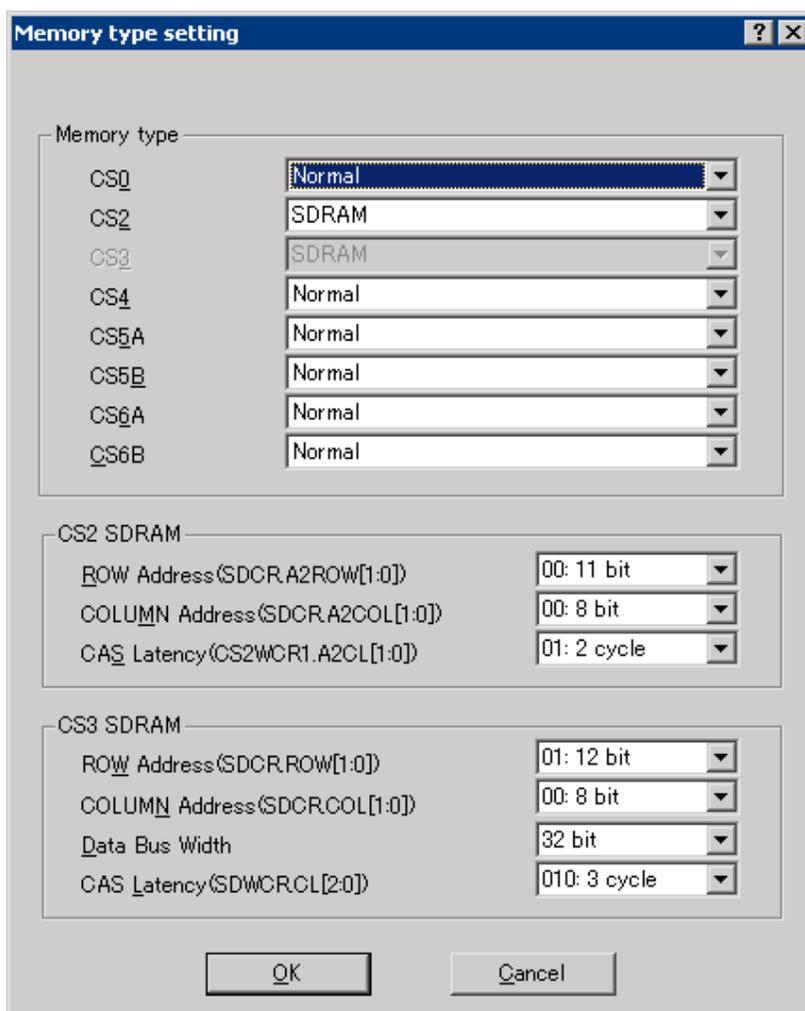


図 5.5 [Memory type setting]ダイアログボックス

【注】 本ダイアログボックスは、製品によって異なります。

### 5.1.5 [Option Board]ページ

[Option Board]ページでは、E200F エミュレータ拡張プロファイルユニットの基本設定を行います。

**【注意事項】**

拡張プロファイルユニットを E200F エミュレータに接続していない場合、本ページは表示されません。

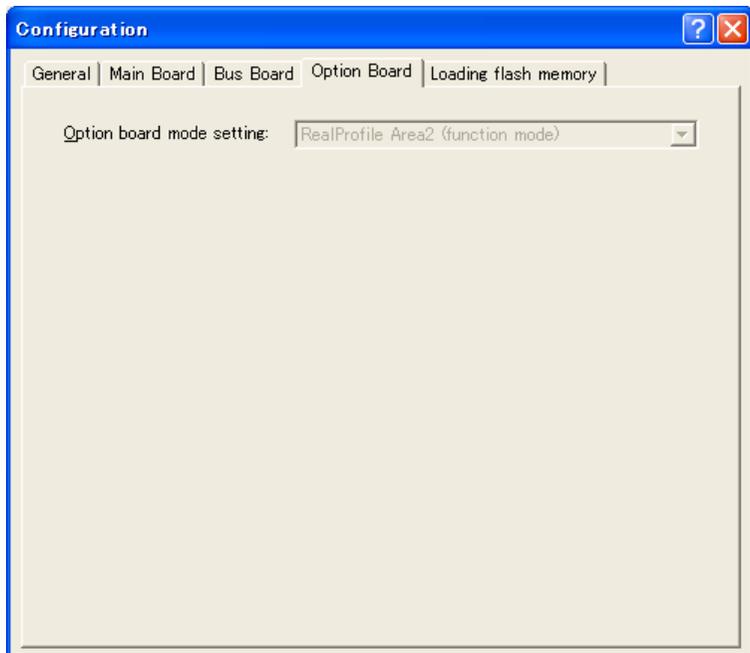


図 5.6 [Configuration]ダイアログボックス ([Option Board]ページ)

設定できる項目は以下の通りです。

- |                                     |   |
|-------------------------------------|---|
| [Option board mode setting]         | 拡張プロファイルユニットの機能を表示します。                    |
| RealProfile Area2 (function mode) : | リアルタイムプロファイル機能 (測定モード: 関数モード) が選択されています。  |
| RealProfile Area2 (nest mode) :     | リアルタイムプロファイル機能 (測定モード: ネストモード) が選択されています。 |
| Coverage (8M) :                     | カバレッジ機能が選択されています。 8MB 領域の測定が可能です。         |

### 5.1.6 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。詳細につきましては、「6.22 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。

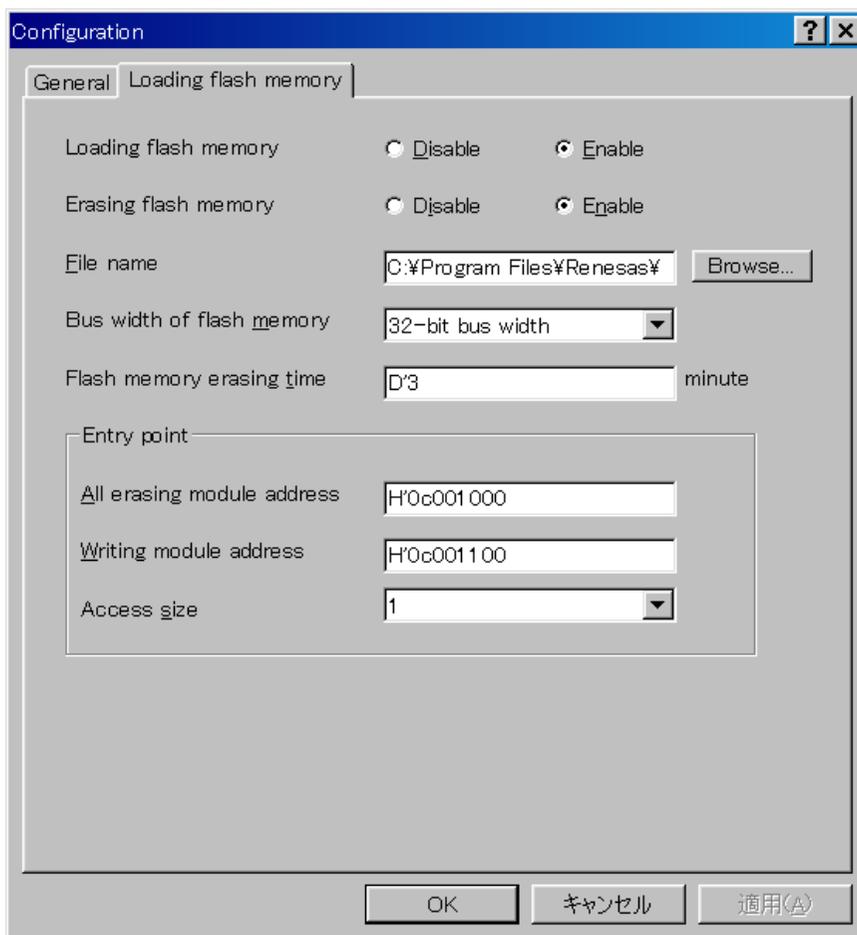


図 5.7 [Configuration]ダイアログボックス ([Loading flash memory]ページ)

## 5 デバッグ

---

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。

### 5.1.7 Memory Mapping ダイアログボックスを開く

[基本設定 エミュレータ メモリリソース...]を選択するか、[Emulator Memory Resource]ツールバーボタンをクリックすると、[Memory Mapping]ダイアログボックスが開きます。

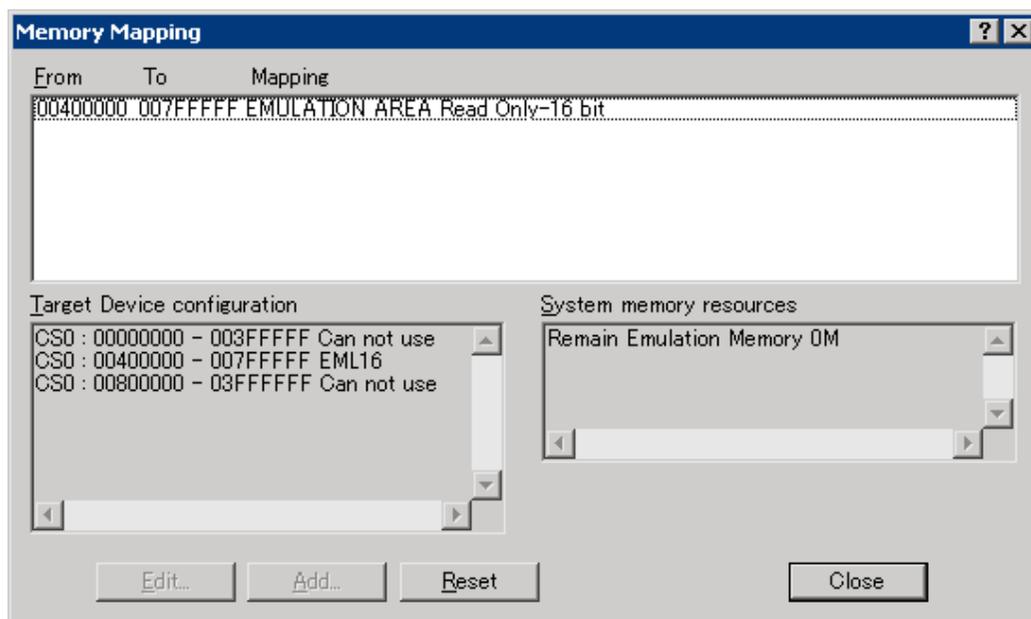


図 5.8 [Memory Mapping]ダイアログボックス

現在のメモリマップを表示し、貸し出しメモリの状態を示しています。

- [Edit...]      メモリマップのアドレス範囲および属性を変更するため[Edit Memory Mapping]ダイアログボックスを開きます。
- [Add...]      エミュレーションメモリを新規に割り付けます。
- [Reset]      メモリマップをデフォルト設定にリセットします。
- [Close]      ダイアログボックスを閉じます。

ターゲットマイコンのメモリマップ情報は、[ステータス]ウィンドウの[Memory]シートに表示されます。

- 【注】
1.    トレースユニットを E200F エミュレータに接続していない場合、本ページでメモリマップの設定はできません。
  2.    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る場合があります。詳細につきましては、オンラインヘルプをご参照ください。
  3.    本ダイアログボックスで表示される項目はご使用のエミュレータにより異なる場合があります。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。

### 5.1.8 メモリマップ設定を変更する

[Memory Mapping]ダイアログボックスで変更したいメモリマップ情報を選択し[Edit...]ボタンをクリックすると、[Edit Memory Mapping]ダイアログボックスが開きます。

また、[Memory Mapping]ダイアログボックスで[Add...]ボタンをクリックすることによっても、本ダイアログボックスが開きます。

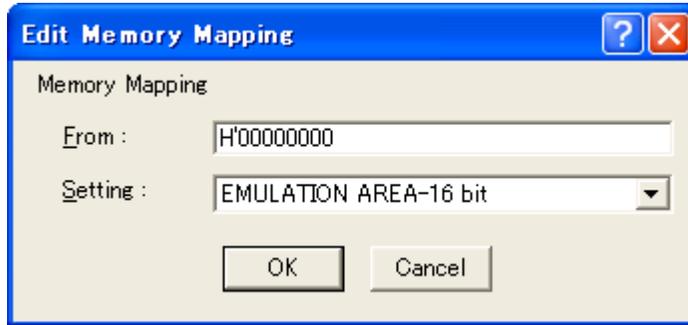


図 5.9 [Edit Memory Mapping]ダイアログボックス

メモリマップのアドレス範囲および属性を変更します。

[From] 範囲の開始アドレスを入力します。

[Setting] マップの属性を設定します。以下のいずれかを選択できます。製品により違います。ヘルプを参照してください。

EMULATION AREA-16 bit : データバス幅を 16bit に設定します。

EMULATION AREA-16 bit : データバス幅を 16bit に設定し、書き込み禁止に設定します。

Read Only

USER AREA : ユーザ空間に指定します。

【注】 マップ設定できる単位は 4Mbyte 固定です。

## 5.2 プログラムをダウンロードする

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

- 【注】 ブレークが起ると、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バ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。

### 5.2.1 プログラムをダウンロードする

デバッグするロードモジュールをダウンロードします。

プログラムのダウンロードは、[デバッグ->ダウンロード]からロードモジュールを選択するか、[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 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、バス幅の設定がユーザシステムに合っているかを十分にご確認ください。

## 5.2.2 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。

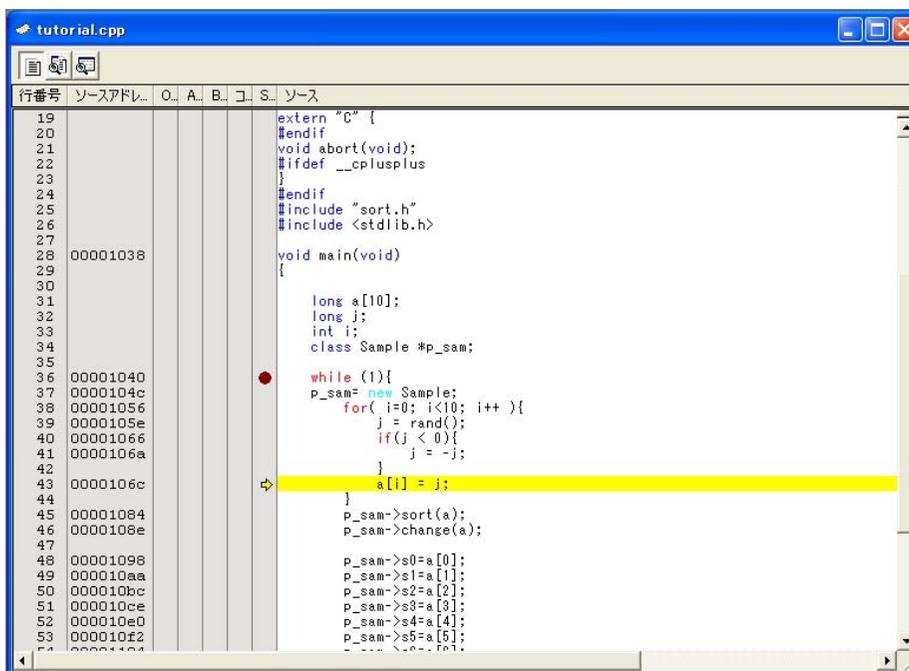


図 5.10 [エディタ]ウィンドウ

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	(ソースアドレスカラム)	アドレス情報
2 列目	(Onchip event カラム)	オンチップイベント情報
3 列目	(AUD event カラム)	AUD イベント情報
4 列目	(BUS event カラム)	外部バスイベント情報
5 列目	(コードカバレジカラム)	カバレジ情報
6 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

### 【留意事項】

トレースユニットを E200F エミュレータに接続していない場合、BUS event カラムは表示されません。

### ソースアドレスカラム

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

### Onchip event カラム

Onchip event カラムには下記を表示します。

- オンチップイベント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オンチップチャンネルの本数分設定可能です。

Onchip 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。

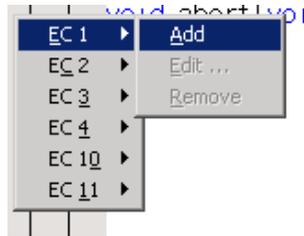


図 5.11 Onchip event カラムポップアップメニュー

#### 【留意事項】

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

### AUD event カラム

AUD event カラムには下記を表示します。

- ⓑ AUD ブレークを設定しています。
- ⓑ AUD シーケンシャルブレークを設定しています。
- ⓐ AUD トレース取得を設定しています。
- ⓐ AUD トレーススタートを設定しています。
- ⓐ AUD トレースストップを設定しています。
- ⓐ AUD シーケンシャルトレースストップを設定しています。
- ⓐ AUD パフォーマンススタートを設定しています。
- ⓐ AUD パフォーマンスストップを設定しています。

AUD event カラムをダブルクリックすることによって、上記のブレーク用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。

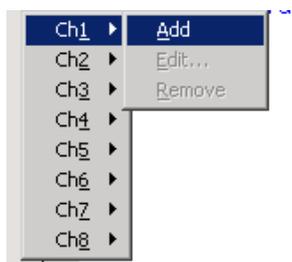


図 5.12 AUD event カラムポップアップメニュー

## 5 デバッグ

---

### 【注意事項】

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

### BUS event カラム

BUS event カラムには下記を表示します。

-  外部バスブレイクを設定しています。
-  外部バスシーケンシャルブレイクを設定しています。
-  外部バストレース取得を設定しています。
-  外部バストレーススタートを設定しています。
-  外部バストレースストップを設定しています。
-  外部バストレースシーケンシャルストップを設定しています。

BUS event カラムをダブルクリックすることによって、上記のブレイク用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 5.13 BUS event カラムポップアップメニュー

### 【注意事項】

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

### S/W ブレイクポイントカラム

S/W ブレイ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 ブレイクポイントを設定している
-  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。
5. 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。
6. [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。

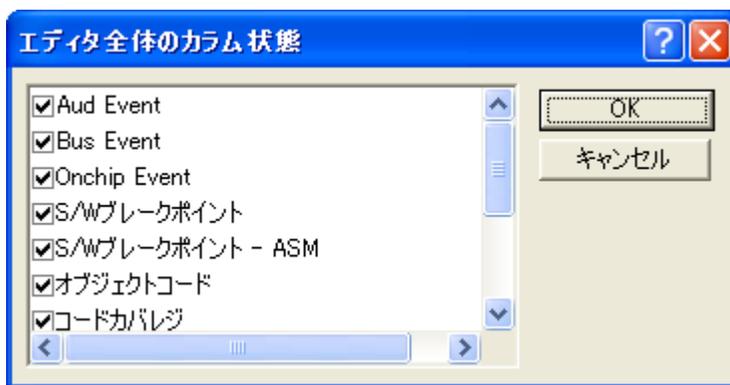


図 5.14 [エディタ全体のカラム状態]ダイアログボックス

⇒1つのソースファイルでカラムをオフにするには

削除したいカラムのあるソースファイルを開き、[エディタ]ウィンドウを右クリックしてください。

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

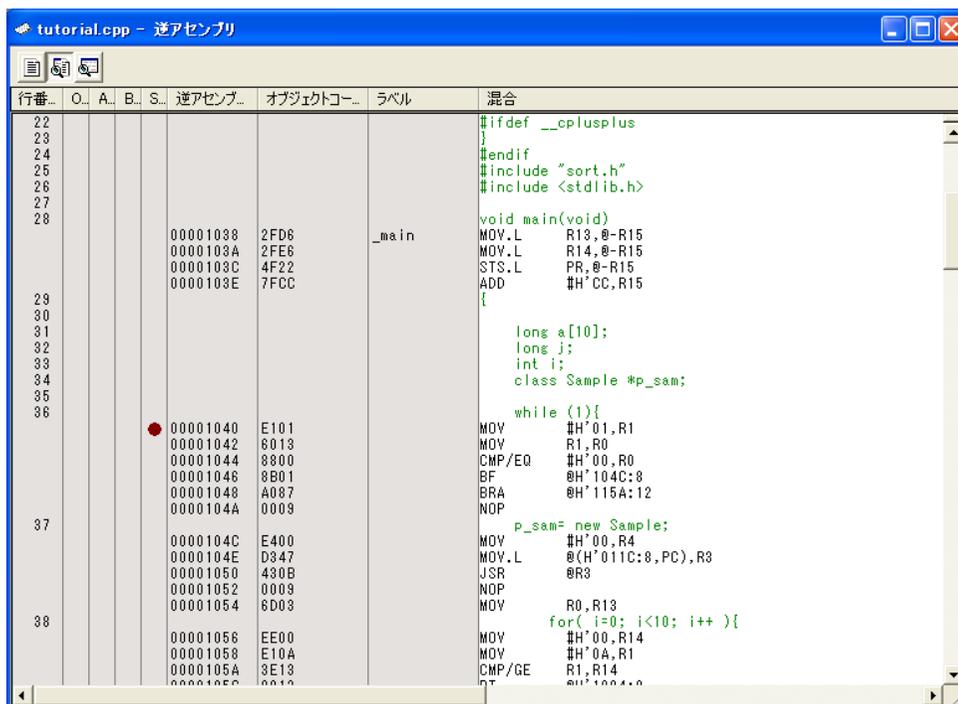
### 5.2.3 アセンブリ言語コードを表示する

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

ソースファイルが存在しなくてもアセンブリ言語レベルでコードを表示したい場合は、[表示]->[逆アセンブリ...]を選択するか、[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SEMBルモニック(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、[逆アセンブリ]ウィンドウのポップアップメニューから[混合表示]を選択すると、ソースとコードの両方を表示することができます。以下は[混合表示]を選択した場合の表示例です。



行番...	O...	A...	B...	S...	逆アセンブ...	オブジェクトコー...	ラベル	混合
22								#ifdef __cplusplus
23								}
24								#endif
25								#include "sort.h"
26								#include <stdlib.h>
27								
28								void main(void)
					00001038	2FD6	_main	MOV.L R13,0-R15
					0000103A	2FE6		MOV.L R14,0-R15
					0000103C	4F22		STS.L PR,0-R15
					0000103E	7FCC		ADD #H'CC,R15
29								{
30								long a[10];
31								long j;
32								int i;
33								class Sample *p_sam;
34								
35								while (1){
36					00001040	E101		MOV #H'01,R1
					00001042	6013		MOV R1,R0
					00001044	8800		CMP/EQ #H'00,R0
					00001046	8B01		BF @H'104C:8
					00001048	A087		BRA @H'115A:12
					0000104A	0009		NDP
37								p_sam= new Sample;
					0000104C	E400		MOV #H'00,R4
					0000104E	D347		MOV.L @H'011C:8,PC),R3
					00001050	430B		JSR @R3
					00001052	0009		NDP
					00001054	6D03		MOV R0,R13
38								for( i=0; i<10; i++){
					00001056	EE00		MOV #H'00,R14
					00001058	E10A		MOV #H'0A,R1
					0000105A	3E13		CMP/GE R1,R14
					0000105C	0010		BT @H'1004:0

図 5.15 [逆アセンブリ]ウィンドウ

## 5.2.4 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。

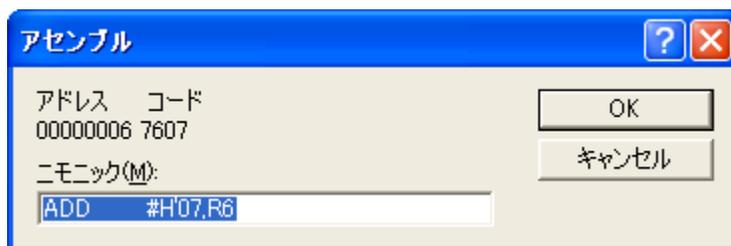


図 5.16 [アセンブル]ダイアログボックス

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、[二モニック]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、メモリの実際の機械語コードからディスアセンブルします。メモリの内容を修正すると、ダイアログボックス（および[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。

### 5.2.5 特定のアドレスを見る

[逆アセンブリ]ウィンドウを使って作成したプログラムを見ているとき、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。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 5.17 [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。

### 5.2.6 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場所では、式も入力することができます。先頭にハッシュ文字を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。

## 5.3 リアルタイムにメモリ内容を表示する

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

モニタ対象のアドレスを物理アドレスとするか、論理アドレスとするかについては、起動時に選択できます。

物理アドレスとした場合、[Configuration]ダイアログボックスの[General]シート内の[Emulation mode]リストボックスで、[Hardware break enable]を選択している場合はモニタできません。

### 5.3.1 [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。

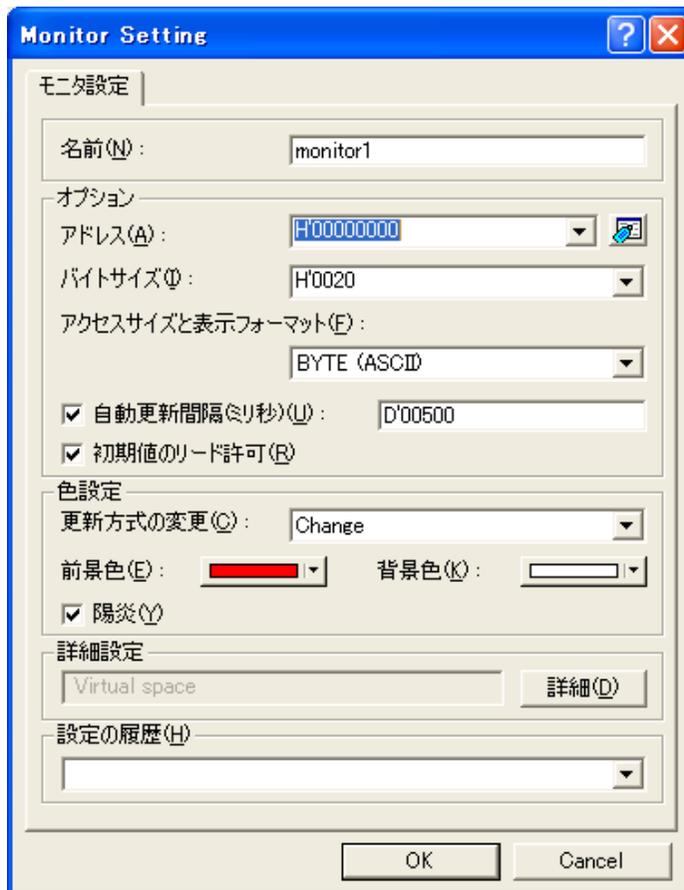


図 5.18 [Monitor Setting]ダイアログボックス

## 5 デバッグ

[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)
[更新方法の変更]	モニタを行う先頭アドレスを設定します。 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]	モニタ対象のアドレスが物理アドレスであるか、論理アドレスであるかを表示します。
[設定の履歴]	前回の設定内容呼び出します。

【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。

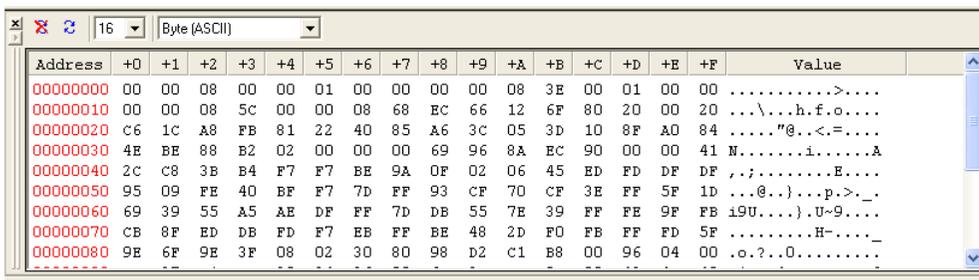


図 5.19 [モニタ]ウィンドウ

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。

### 5.3.2 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。

### 5.3.3 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。

### 5.3.4 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。

## 5.3.5 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]ウィンドウに登録されている変数の値は、ユーザプログラム実行中に更新できます。

[ウォッチ]ウィンドウのポップアップメニューの、[Auto Update]を選択してください。

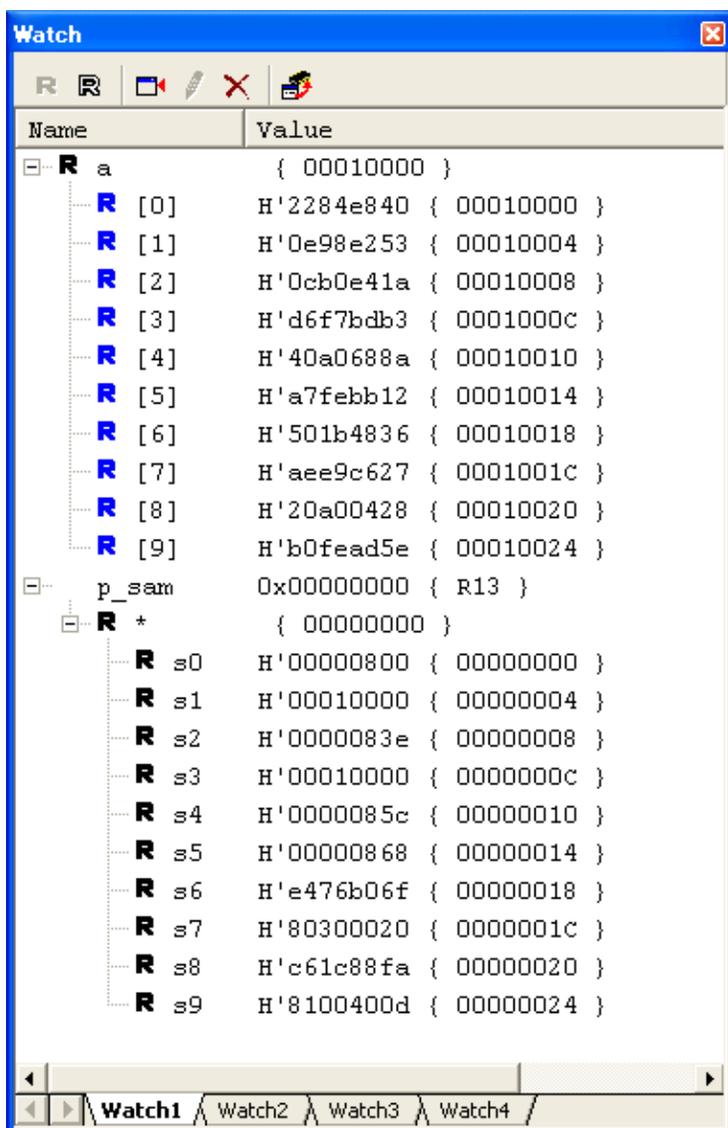


図 5.20 [Watch]ウィンドウ

R マークのある変数は、ユーザプログラム実行中に更新できることを示します。

E200F エミュレータでは、ユーザプログラム実行中にメモリ内容を読み出して[ウォッチ]ウィンドウに登録した変数の内容を更新する実現手段として下記 2 種類の方法があります。

7. Monitor機能を使用して、ユーザプログラムを停止せずに値の更新を行う。  
E200FエミュレータのMonitor機能を使用して、該当する変数の値を更新します。
8. ユーザプログラムを一旦ブレイクし、メモリ内容を読み出して値の更新を行う。

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

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

- |     |   |
|-----|---|
| 青色部 | Monitor 機能を使用したデータリードにより値の更新を行うことを示します。 |
| 黒色部 | 通常データリードにより値の更新を行うことを示します。              |

- 【注】
1. 本機能は変数ごと、また構造体であれば指定された構造体一括/要素ごとに設定できます。
  2. [ウォッチ]ウィンドウを閉じたりスクロールすると、設定情報が解除されます。
  3. レジスタに割付けられている変数には設定することができません。
  4. リアルタイムウォッチ機能は、AUD モニタ機能を使用して実現しています。したがって、AUD モニタ機能と[ウォッチ]ウィンドウのリアルタイム更新機能をあわせて、4 領域分のモニタが可能です。

### 5.3.6 [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ことで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。

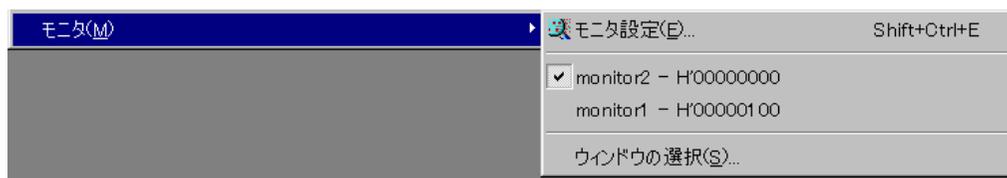


図 5.21 モニタ設定リスト

### 5.3.7 [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。

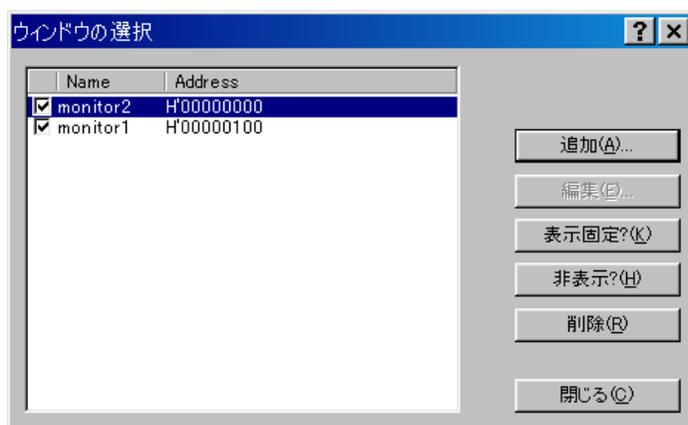


図 5.22 [ウィンドウの選択]ダイアログボックス

## 5.4 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[ステータス]ウィンドウを表示します。

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

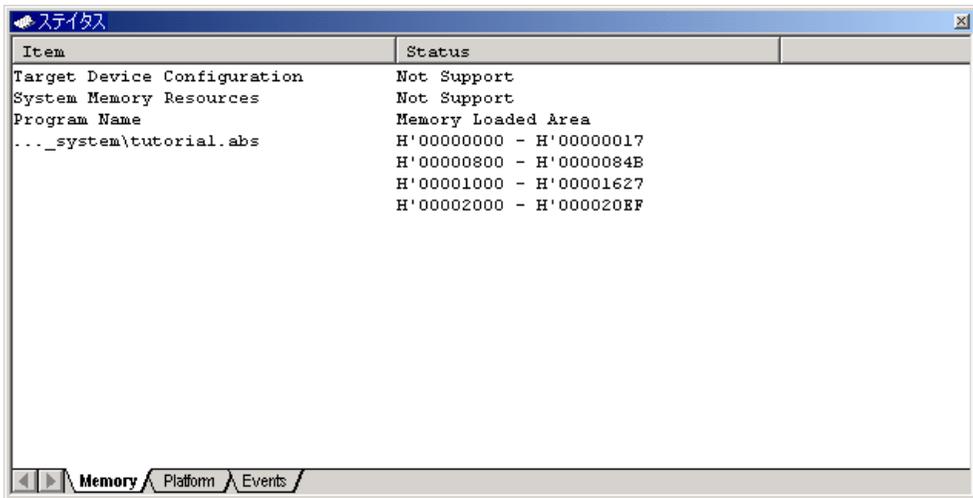


図 5.23 [ステータス]ウィンドウ

[ステータス]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。貸し出しメモリ機能使用時は、[System Memory Resources]項目に現在割り付け可能である貸し出しメモリのバイト数を表示します。
- [Platform]シート  
CPU種別および動作モードなど、エミュレータのステータス情報、実行状態および実行統計情報を含んでいます。
- [Events]シート  
リソース情報およびブレークポイント等のイベント情報に関する情報を含んでいます。

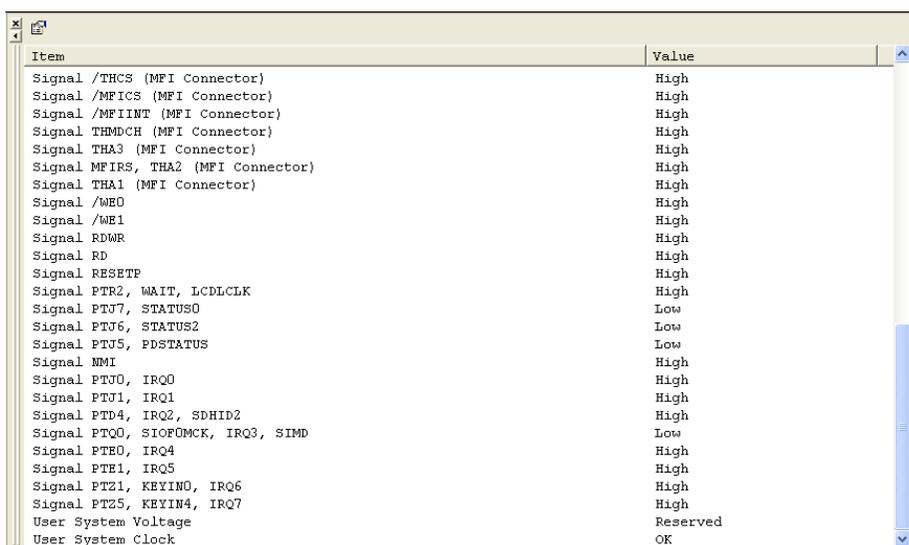
## 5.5 エミュレータの情報を定期的に読み出し表示する

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

【注】 Extended Monitor 機能はユーザシステムや CPU から出力される信号をモニタするため、ユーザプログラムの実行に影響を与えることはありません。

### 5.5.1 [Extended Monitor]ウィンドウを開く

[Extended Monitor]ウィンドウを開くには、[表示->CPU->拡張モニタ]を選択するか、[拡張モニタ]ツールバーボタンをクリックします。



Item	Value
Signal /THCS (MFI Connector)	High
Signal /MPICS (MFI Connector)	High
Signal /MPIINT (MFI Connector)	High
Signal THMDCH (MFI Connector)	High
Signal THA3 (MFI Connector)	High
Signal MFIRS, THA2 (MFI Connector)	High
Signal THA1 (MFI Connector)	High
Signal /WE0	High
Signal /WE1	High
Signal EDWR	High
Signal FD	High
Signal RESETP	High
Signal PTR2, WAIT, LCDCLK	High
Signal PTJ7, STATUS0	Low
Signal PTJ6, STATUS2	Low
Signal PTJ5, PDSTATUS	Low
Signal MMI	High
Signal PTJ0, IRQ0	High
Signal PTJ1, IRQ1	High
Signal PTD4, IRQ2, SDHID2	High
Signal PTQ0, SIOFUMCK, IRQ3, SIMD	Low
Signal PTE0, IRQ4	High
Signal PTE1, IRQ5	High
Signal PTZ1, KEYIN0, IRQ6	High
Signal PTZ5, KEYIN4, IRQ7	High
User System Voltage	Reserved
User System Clock	OK

図 5.24 [Extended Monitor]ウィンドウ

【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。

## 5.5.2 表示項目を選択する

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



図 5.25 [拡張モニタコンフィギュレーション]ダイアログボックス

[Extended Monitor]ウィンドウに表示する各項目を設定できます。

### 5.6 イベントポイントを使用する

E200F 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 1000 ポイントまで設定できます。

#### 5.6.2 イベントポイントとは

イベントポイント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。E200F エミュレータは、4 種類のイベントポイントを設定できます。

##### (1) オンチップイベントポイント (Onchip Eventpoint)

CPU 内部の各種情報によりイベントポイントを設定します。

イベント検出チャンネルは 13 個です。

イベント検出後の動作として、ブレーク、内蔵トレース取得/取得開始/取得停止、内蔵パフォーマンス測定開始/終了を指定できます。

複数の Onchip Eventpoint を組み合わせることにより、より複雑なシーケンシャル条件設定が可能です。

本機能は、[Event]ウィンドウの[Onchip Event]シートで設定できます。

【注】 設定できる内容は、製品ごとに異なる場合があります。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) AUD イベントポイント (AUD Eventpoint)

AUD インタフェースから出力されたトレース情報によりイベントポイントを設定します。

イベント検出チャンネルは 8 個です。

イベント検出後の動作として、ブレーク、AUD トレース取得/取得開始/取得停止、AUD パフォーマンス測定開始/終了を指定できます。

複数の AUD Eventpoint を組み合わせることにより、より複雑なシーケンシャル条件設定が可能です。

本機能は、[Event]ウィンドウの[AUD Event]シートで設定できます。

(3) 外部バスイベントポイント (BUS Eventpoint)

CPUの外部バスや割り込み端子など端子情報によりイベントポイントを設定します。

イベント検出チャンネルは最大6個です。

イベント検出されたときの動作として、ブレイク、外部バストレース取得/取得開始/取得停止を指定できます。

複数のBUS Eventpointを組み合わせてより複雑なシーケンシャル条件設定が可能です。

本機能は、[Event]ウィンドウの[BUS Event]シートで設定できます。

- 【注】
1. トレースユニットをE200Fエミュレータに接続していない場合、本機能はサポートしません。
  2. E200Fエミュレータのトレースユニットの機能を入れ替えることにより、イベント検出チャンネルの機能が変わります。詳細につきましては、「5.1.4 [Bus Board]ページ」を参照してください。

(4) その他のイベントポイント (Other Eventpoint)

本機能は、[Event]ウィンドウの[Other Event]シートで設定できます。

(a) 実行時間イベントポイント

プログラムの実行時間を条件として、イベントポイントを設定します。

イベント検出チャンネルは1個です。

イベント検出後の動作として、ブレイクを指定できます。

(b) 外部プローブイベントポイント

プローブケーブル経由の4つの外部プローブ信号を条件として、イベントポイントを設定します。

イベント検出チャンネルは1個です。

イベント検出後の動作として、ブレイク、AUDトレース取得/取得開始/取得停止を指定できます。

(c) その他のイベントポイント

I/Oアナライザ機能によっては、イベント条件が設定できるものがあります。

### 5.6.3 [Event]ウィンドウを開く

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

[Event]ウィンドウには、5枚のシートがあります。

- [Breakpoint]シート  
PCブレイクポイントの設定内容を表示します。また、PCブレイクポイントの設定、変更および解除を行うことができます。
- [Onchip Event]シート  
オンチップイベントチャンネルの設定内容を表示、設定します。
- [AUD Event]シート  
AUDイベントチャンネルの設定内容を表示、設定します。
- [BUS Event]シート  
外部バスイベントチャンネルの設定内容を表示、設定します。
- [Other Event]シート  
その他のイベントチャン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。

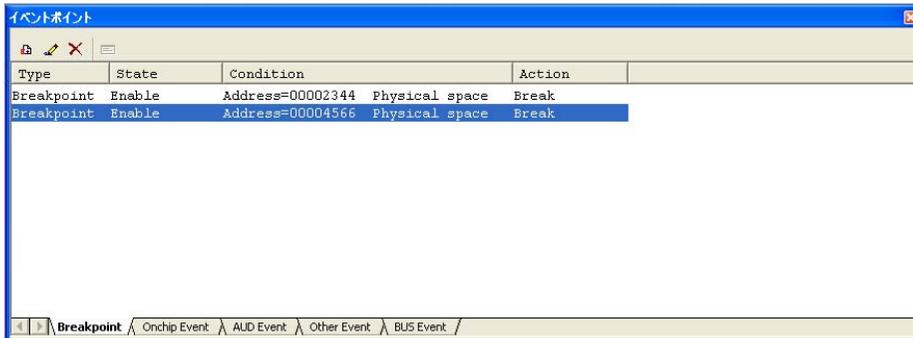


図 5.26 [Event]ウィンドウ ([Breakpoint]シート)

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

ポップアップメニューから[追加...]を選択するか、または本ウィンドウに表示されているPC ブレークポイントを選択し、ポップアップメニューから[編集...]を選択すると、[Breakpoint]ダイアログボックスを表示します。

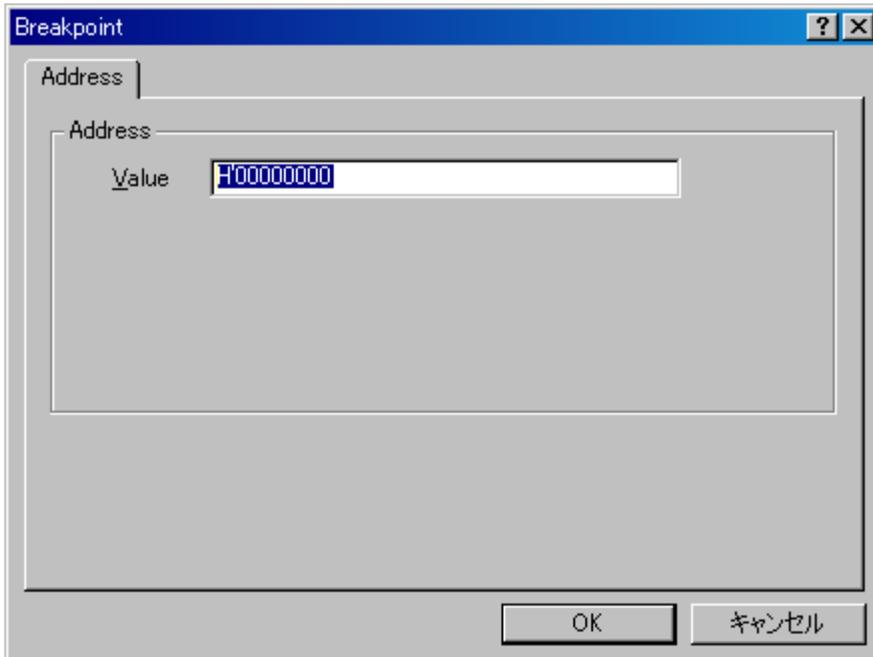


図 5.27 [Breakpoint]ダイアログボックス

本ダイアログボックスより、PC ブレークポイントのアドレス条件を設定します。

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 1000 個まで設定できます。

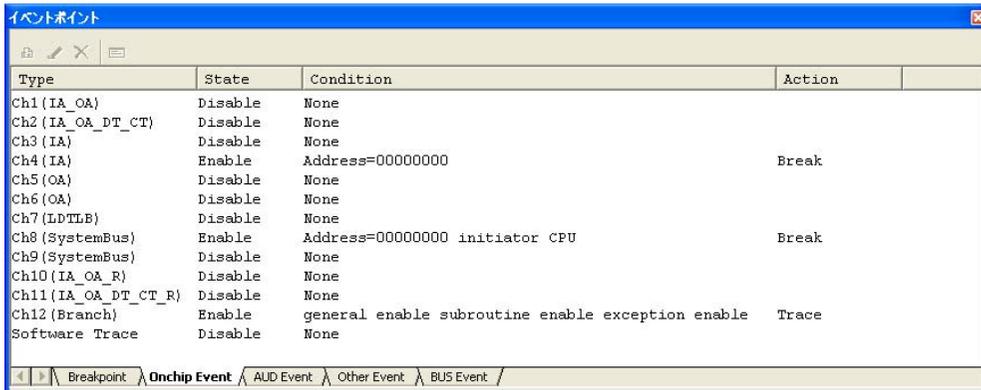
設定できる内容は製品によって異なります。詳しくは、各製品のオンラインヘルプを参照してください。

[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。

指定したブレークポイント条件は、[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。

## 5.6.5 Onchip Eventpoint を設定する

[Onchip Event]シートでは Onchip Eventpoint の設定内容の表示、変更ができます。



Type	State	Condition	Action
Ch1 (IA_OA)	Disable	None	
Ch2 (IA_OA_DT_CT)	Disable	None	
Ch3 (IA)	Disable	None	
Ch4 (IA)	Enable	Address=00000000	Break
Ch5 (OA)	Disable	None	
Ch6 (OA)	Disable	None	
Ch7 (LDTLB)	Disable	None	
Ch8 (SystemBus)	Enable	Address=00000000 initiator CPU	Break
Ch9 (SystemBus)	Disable	None	
Ch10 (IA_OA_R)	Disable	None	
Ch11 (IA_OA_DT_CT_R)	Disable	None	
Ch12 (Branch)	Enable	general enable subroutine enable exception enable	Trace
Software Trace	Disable	None	

図 5.28 [Event]ウィンドウ ([Onchip Event]シート)

イベント検出チャンネル本数や設定できる内容は製品によって異なりますので、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] Onchip イベントチャンネル番号とタイプを表示します。

[State] 該当イベントポイントの有効/無効を示します。

Enable : 有効

Disable : 無効

[Condition] イベントポイントが成立する条件を表示します。表示内容はチャンネルにより異なります。

[Action] イベントポイント条件成立時の動作を表示します。表示内容はチャンネルにより異なります。

本ウィンドウでイベントチャンネルをダブルクリックするか、またはイベントチャンネルを選択しポップアップメニューから[編集...]を選択すると、[Event condition x]ダイアログボックスが開き、イベントポイント条件を変更することができます。

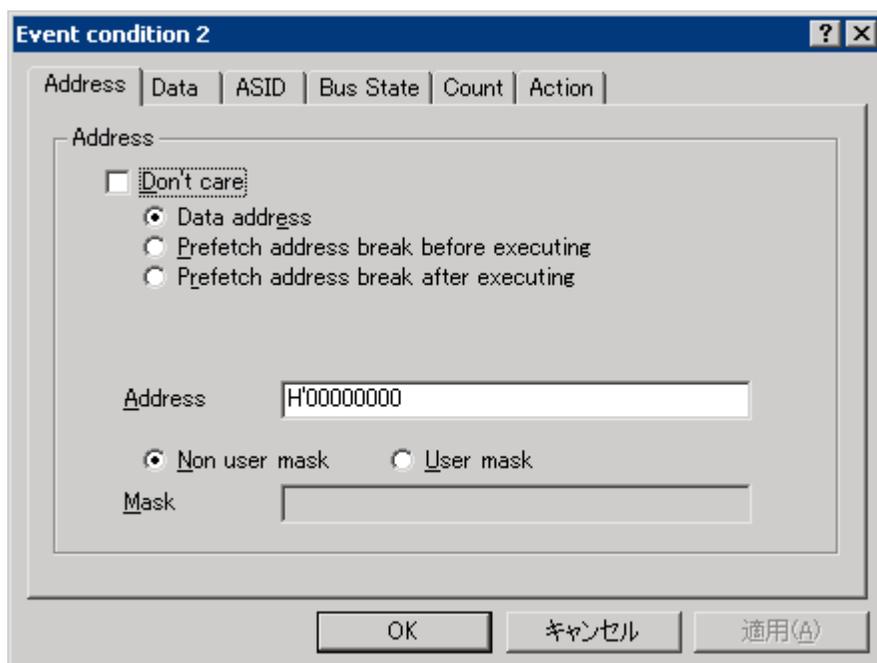


図 5.29 [Event condition x]ダイアログボックス ([Address]ページ)

【注】 [Event condition x]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。

表 5.1 に Event Condition の条件の内容を示します。

表 5.1 Event Condition の条件

項番	Event Condition 条件	説明
1	アドレスバス条件 (Address)	デバイスのアドレスバスまたはプログラムカウンタの値が一致したときにブレイクします。
2	データバス条件 (Data)	デバイスのデータバスの値が一致したときにブレイクします。バイト、ワード、ロングアクセスのデータサイズを指定できます。
3	バスステート条件 (Bus State)	バスステート条件には、次の 2 つの条件設定があります。 Bus State 条件 : デバイスのデータバス、X バス、Y バスアドレスバスのいずれかの値が一致したときにブレイクまたはトレース取得します。 Read/Write 条件 : 指定したリード/ライト条件と一致したときにブレイクまたはトレース取得します。
4	ウィンドウアドレス条件	指定したメモリ範囲内のデータをアクセスしたときにブレイク、またはトレース取得を行います。
5	システムバス	システムバス上のアドレス、データなどが一致したときにブレイク、またはトレース取得を行います。
6	LDTLB 命令ブレイク条件	デバイスが LDTLB 命令を実行したときにブレイクします。
7	カウント	設定した条件が、指定した回数分成立したときにブレイクします。
8	分岐条件 (Branch trace)	デバイスが設定した条件で分岐したときにブレイク、またはトレース取得を行います。(デフォルトではトレース取得が有効となっています。)
9	ソフトウェアトレース	ソフトウェアトレースを取得するかどうかを選択します。
10	Action	ブレイク、トレース、パフォーマンス開始/終了条件の設定など条件が一致したときの動作を選択します

## 5 デバッグ

以下の表に、Ch 1～Ch12 およびソフトウェアトレースチャンネルで設定できる条件の組み合わせについて説明します。

表 5.2 Event Condition の条件設定用のダイアログボックス

チャンネル	機能										
	アドレス バス条件 (Address)	データ バス 条件 (Data)	ASID 条件 (ASID)	バス ステ ート 条件 (Bus State)	ウィン ドウア ドレス 条件 (Window address)	システ ムバス	LDTLB 命令 ブレーク	カウン ト 条件 (Count)	分岐 条件 (Branch Trace)	Software Trace	Action
Ch1(IA_OA)		x			x	x	x	x	x	x	(B・P)
Ch2 (IA_OA_DT_CT)					x	x	x		x	x	(B・P)
Ch3(IA)		x		x	x	x	x	x	x	x	(B・P)
Ch4(IA)		x		x	x	x	x	x	x	x	(B・P)
Ch5(OA)	x	x				x	x	x	x	x	(B・T・ P)
Ch6(OA)	x	x				x	x	x	x	x	(B・T・ P)
Ch7(LDTLB)	x	x	x	x	x	x		x	x	x	ブレーク 固定
Ch8(SystemBus)		x	x	x	x		x	x	x	x	(B・T・ P)
Ch9(SystemBus)		x	x	x	x		x	x	x	x	(B・T・ P)
Ch10(IA_OA_R)		x			x	x	x	x	x	x	(B・P)
Ch11 (IA_OA_DT_CT_R)					x	x	x		x	x	(B・P)
Ch12(Branch)	x	x	x	x	x	x	x	x		x	(B・T・ P)
Software Trace	x	x	x	x	x	x	x	x	x		トレース 固定

【注】 は、ダイアログボックスで設定できることを表します。

×は、設定できないことを表します。

Action 項目の

B は、ブレーク設定ができることを表します。

T は、トレース設定ができることを表します。

P は、パフォーマンス開始/終了条件の設定ができることを表します。

## (1) シーケンシャル設定

E200F エミュレータは、Event Condition のシーケンシャル設定をすることができます。

表 5.3 シーケンシャルブレイク条件

	分類	ブレイク条件	説明	
[CPU Sequential Event]ページ	2 Channel Sequential	Ch 2 -> 1	Event Condition 2,1 の順番で条件が成立したときにプログラムを停止します。 Ch2,1 に break 条件が設定されている必要があります。	
		Ch 4 -> 3	Event Condition 4,3 の順番で条件が成立したときにプログラムを停止します。 Ch4,3 に break 条件が設定されている必要があります。	
		Ch 6 -> 5	Event Condition 6,5 の順番で条件が成立したときにプログラムを停止します。 Ch6,5 に break 条件が設定されている必要があります。	
		Ch 11 -> 10	Event Condition 11,10 の順番で条件が成立したときにプログラムを停止します。 Ch11,10 に break 条件が設定されている必要があります。	
	Many Channel Sequential	Ch 3 -> 2 -> 1	Event Condition 3,2,1 の順番で条件が成立したときにプログラムを停止します。 Ch3,2,1 に break 条件が設定されている必要があります。	
		Ch 4 -> 3 -> 2 -> 1	Event Condition 4,3,2,1 の順番で条件が成立したときにプログラムを停止します。 Ch4,3,2,1 に break 条件が設定されている必要があります。	
		Ch 5 -> 4 -> 3 -> 2 -> 1	Event Condition 5,4,3,2,1 の順番で条件が成立したときにプログラムを停止します。 Ch5,4,3,2,1 に break 条件が設定されている必要があります。	
		Ch 6 -> 5 -> 4 -> 3 -> 2->1	Event Condition 6,5,4,3,2,1 の順番で条件が成立したときにプログラムを停止します。 Ch6, 5,4,3,2,1 に break 条件が設定されている必要があります。	
		Ch 10 -> 6 -> 5 -> 4 -> 3 -> 2 -> 1	Event Condition 10,6,5,4,3,2,1 の順番で条件が成立したときにプログラムを停止します。 Ch10,6, 5,4,3,2,1 に break 条件が設定されている必要があります。	
		Ch 11 -> 10 -> 6 -> 5 -> 4 -> 3 -> 2->1	Event Condition 11,10,6,5,4,3,2,1 の順番で条件が成立したときにプログラムを停止します。 Ch11,10,6, 5,4,3,2,1 に break 条件が設定されている必要があります。	
	CPU Extend		[CPU Sequential Extend]ページを展開します。ここでは任意の組み合わせでシーケンシャル設定を行うことができます。 詳しくは本章の「(2)シーケンシャルブレイクの拡張設定」を参照してください。	
	[System Bus Sequential Event]ページ	System Bus Sequential Event	Ch 9 -> 8	Event Condition 9,8 番で条件が成立したときにプログラムを停止します。 Ch9,8 に break 条件が設定されている必要があります。

	分類	ブレーク条件	説明
[System Bus Sequential Event]ページ		Ch 8 -> 9	Event Condition 8,9 番で条件が成立したときにプログラムを停止します。 Ch8,9 に break 条件が設定されている必要があります。
	System Bus Extend		[System Bus Sequential Extend]ページを展開します。ここでは任意の組み合わせでシーケンシャル設定を行うことができます。 詳しくは本章の「(2)シーケンシャルブレークの拡張設定」を参照してください。

## (2) シーケンシャルブレーク拡張設定

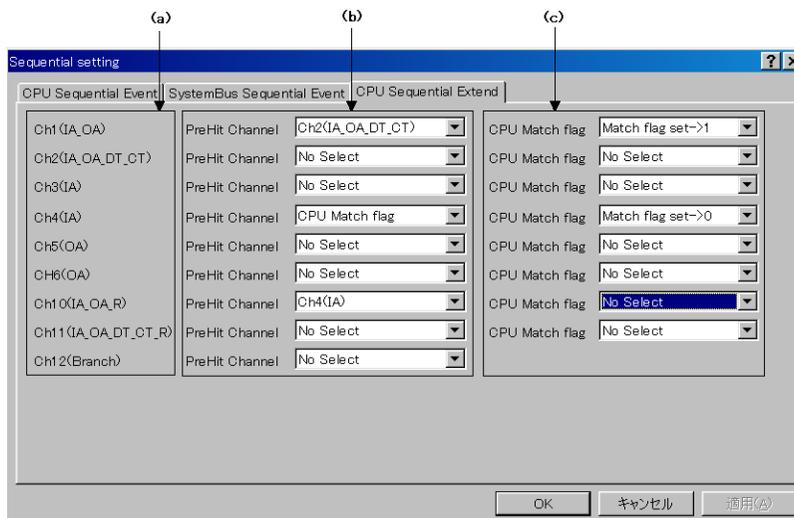


図 5.30 [CPU Sequential Extend]ページ

(a) 条件を設定するチャンネル名を表します。

(b) 条件を設定するチャンネルの前に成立させる条件を選択します。

チャンネル名を選択した場合は、ここで選択したチャンネルが既に条件成立していることを必要条件とします。

CPU Match flagを選択した場合は、CPU Match flagがセットされていることを必要条件とします。

ここで選択されたチャンネルでの条件選択ではブレークは発生しません。

(c) 条件が成立した場合に、CPU Match flagをセットまたはクリアします。

プログラムがブレークするとCPU Match flagは初期化されます。

各チャンネルのブレーク条件設定は、[Event Condition]ダイアログボックスより設定を行ってください。

[System Bus Sequential Extend]ページでも同様です。

## 5 デバッグ

### (3) シーケンシャルブレーク拡張設定の使用例

製品添付のチュートリアルプログラムを例に説明します。

チュートリアルプログラムについては、「6 チュートリアル」を参照してください。

Event Condition 条件を次のように設定します。

#### 1. Ch 1

アドレスH ' 00001084をPrefetch address break after executing条件が成立した時にブレークする。

#### 2. Ch 2

アドレスH ' 0000106cをPrefetch address break after executing条件が成立した時にブレークする。

#### 3. Ch 4

アドレスH ' 000010aaをPrefetch address break after executing条件が成立した時にブレークする。

#### 4. Ch10

アドレスH ' 000010e0をPrefetch address break after executing条件が成立した時にブレークする。

【注】 この時その他のチャンネルは設定しないでください。

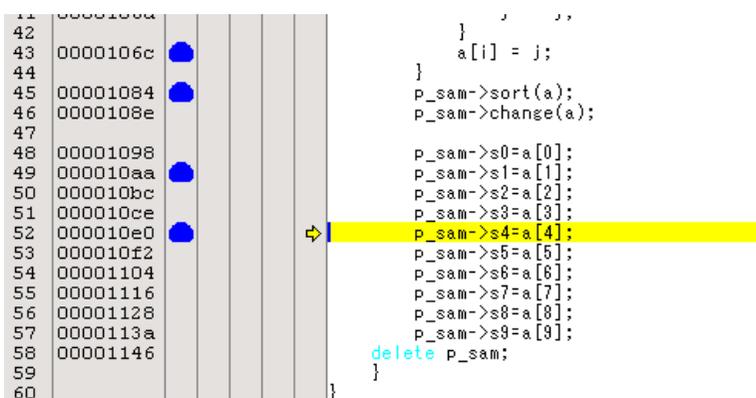
#### 5. [CPU Sequential Extend]ページを図5.30のように設定します。

次に、プログラムカウンタ、スタックポインタ (PC=H ' 00000800、R15=H ' 00010000) を[レジスタ]ウィンドウに設定して、[Go]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。

Ch 10 の条件まで、プログラムを実行して停止します。

この時 Ch2 -> 1 -> 4 -> 10 の順で条件が成立しています。



```
41 00001004
42
43 0000106c
44
45 00001084
46 0000108e
47
48 00001098
49 000010aa
50 000010bc
51 000010ce
52 000010e0
53 000010f2
54 00001104
55 00001116
56 00001128
57 0000113a
58 00001146
59
60

}

}
a[i] = j;
}
}
p_sam->sort(a);
p_sam->change(a);

p_sam->s0=a[0];
p_sam->s1=a[1];
p_sam->s2=a[2];
p_sam->s3=a[3];
p_sam->s4=a[4];
p_sam->s5=a[5];
p_sam->s6=a[6];
p_sam->s7=a[7];
p_sam->s8=a[8];
p_sam->s9=a[9];
delete p_sam;
}
```

図 5.31 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)

## 5.6.6 AUD Eventpoint を設定する

[AUD Event]シートでは AUD Eventpoint の設定内容の表示、変更ができます。



図 5.32 [Event]ウィンドウ ([AUD Event]シート)

8本のイベント検出チャンネルより、8個のイベントポイントを設定できます。

【注】 AUD イベントポイント条件は、AUD 端子から出力するチップ内の情報により設定するため Onchip Eventpoint のトレース取得条件を設定する必要があります。AUD イベントポイント条件に応じて、オンチップイベントチャンネル Ch5 (OA), Ch6 (OA), Ch7 (SystemBus), Ch8 (SystemBus), Ch12 (Branch)または Software Trace のトレース取得条件を設定してください。

シート内に表示する項目は以下の通りです。

[Type]	AUD イベントチャンネル番号とタイプを表示します。
	Normal : 標準のイベントチャンネル
	Delay : デレイ条件設定できるイベントチャンネル
	Reset : AUD シーケンシャルイベントのリセットポイントとして設定できるイベントチャンネル
[State]	該当イベントポイントの有効/無効を示します。
	Enable : 有効
	Disable : 無効
[Condition]	イベントポイントが成立する条件を表示します。表示内容はチャンネルにより異なります。
[Action]	イベントポイント条件成立時の動作を表示します。表示内容はチャンネルにより異なります。

本ウィンドウでイベントチャンネルをダブルクリックするか、またはイベントチャンネルを選択しポップアップメニューから[編集...]を選択すると、[Chx]ダイアログボックスを表示します。

[Chx]ダイアログボックスは、[General]ページ、[Branch]ページ、[Window]ページ、[Software]ページ、[SystemBus]ページ、[Count]ページ、[Delay]ページ、[Action]ページにより構成されています。

各ページで設定された条件を組み合わせたものを、イベントポイントの検出条件として設定します。

(1) [General]ページ

AUD Eventpoint 条件設定に使用する AUD トレース情報を指定します。

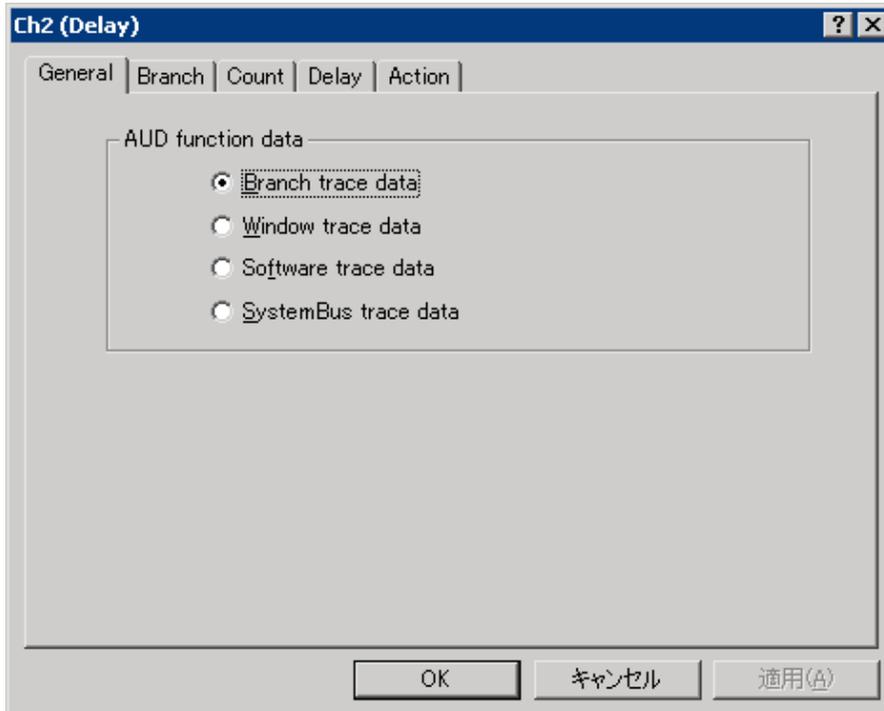


図 5.33 [Chx (Delay)]ダイアログボックス ([General]ページ)

- |                        |                                     |
|------------------------|-------------------------------------|
| [Branch trace data]    | 分岐トレース情報に基づき、イベントポイント条件を設定します。      |
| [Window trace data]    | ウィンドウトレース情報に基づき、イベントポイント条件を設定します。   |
| [Software trace data]  | ソフトウェアトレース情報に基づき、イベントポイント条件を設定します。  |
| [SystemBus trace data] | システムバスのトレース情報に基づき、イベントポイント条件を設定します。 |

## (2) [Branch]ページ

分岐トレースのタイプ条件およびアドレス条件を指定します。

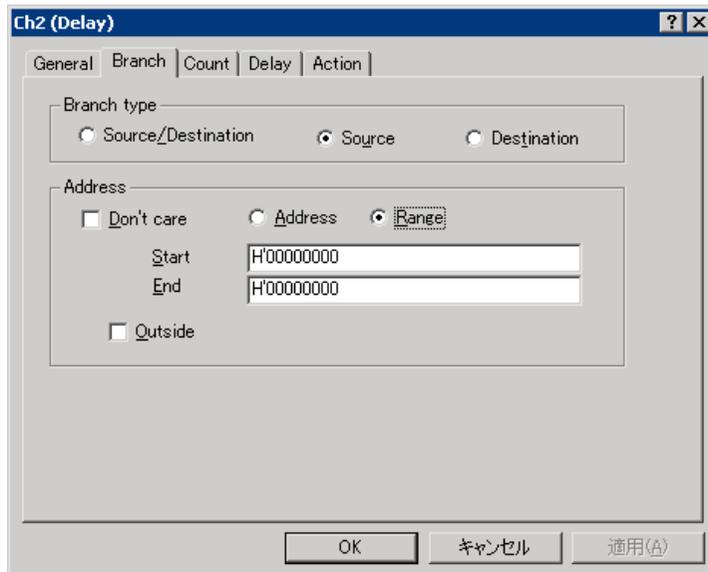


図 5.34 [Chx (Delay)]ダイアログボックス ([Branch]ページ)

[Branch type] 分岐のタイプを指定します。

- Source/Destination : 分岐のタイプを指定しません。
- Source : 分岐元のアドレスを条件として設定します。
- Destination : 分岐先のアドレスを条件として設定します。

[Address] アドレス条件を設定します。

- Don't care : アドレス条件を設定しません。
- Address : 単一アドレスを設定します。
- Range : アドレス範囲を設定します。
- Start : 単一アドレスまたはアドレス範囲の開始アドレスを指定します。
- End : アドレス範囲の終了アドレスを指定します。
- Outside : 単一アドレスまたはアドレス範囲に設定された値以外を条件として設定します。

- 【注】
1. [General]ページで[Branch trace data]を指定したときのみ、本ページが表示されます。
  2. AUD イベントチャンネル Ch1, Ch2 のみ、アドレス範囲を指定できます。
  3. 単一アドレスを指定したとき、[Start]にマスクアドレスを入力できます。
  4. アドレス範囲を指定したとき、[Start],[End]にマスクアドレスを入力できません。

(3) [Window]ページ

ウィンドウトレースのアドレス条件およびデータ条件を指定します。

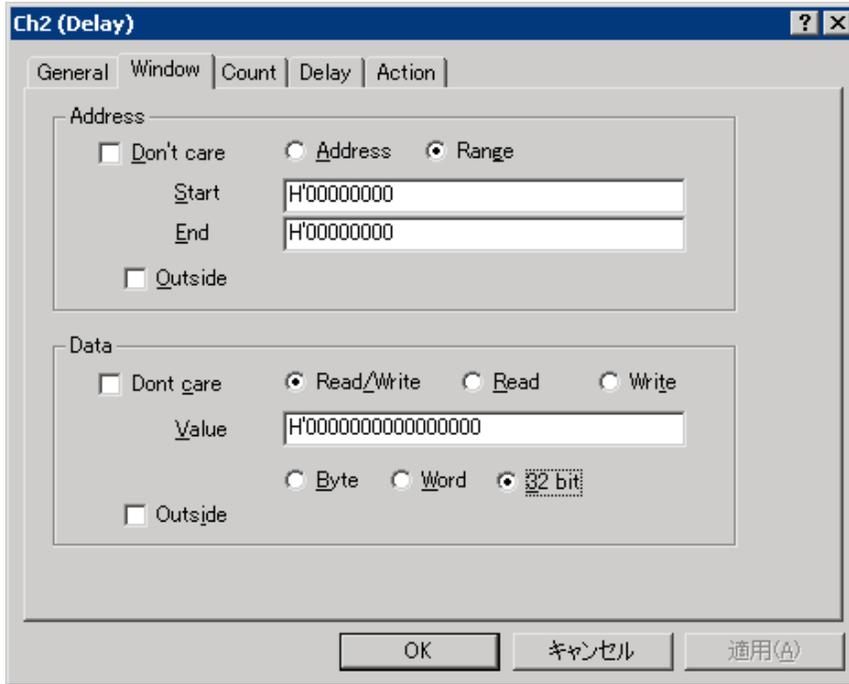


図 5.35 [Chx (Delay)]ダイアログボックス ([Window]ページ)

- [Address] アドレス条件を設定します。
- Don't care : アドレス条件を設定しません。
  - Address : 単一アドレスを設定します。
  - Range : アドレス範囲を設定します。
  - Start : 単一アドレスまたはアドレス範囲の開始アドレスを指定します。
  - End : アドレス範囲の終了アドレスを指定します。
  - Outside : 単一アドレスまたはアドレス範囲に設定された値以外を条件として設定します。
- [Data] データ条件を設定します。
- Don't care : データ条件を設定しません。
  - Read/Write : リード、ライトサイクルを条件にします。
  - Read : リードサイクルを条件にします。
  - Write : ライトサイクルを条件にします。
  - Value : データバスの値を設定します。(マスクデータを入力できます)
  - Byte : バイトアクセスを条件にします。
  - Word : ワードアクセスを条件にします。
  - 32 bit : 32ビットアクセスを条件にします。
  - Outside : Value に設定されたデータ値以外を条件として設定します。

- 【注】
1. [General]ページで[Window trace data]を指定したときのみ、本ページが表示されます。
  2. AUD イベントチャンネル Ch1,Ch2 のみ、アドレス範囲を指定できます。
  3. 単一アドレスを指定したとき、[Start]にマスクアドレスを入力できます。
  4. アドレス範囲を指定したとき、[Start],[End]にマスクアドレスを入力できません。

## (4) [Software]ページ

ソフトウェアトレースのアドレス条件およびデータ条件を指定します。

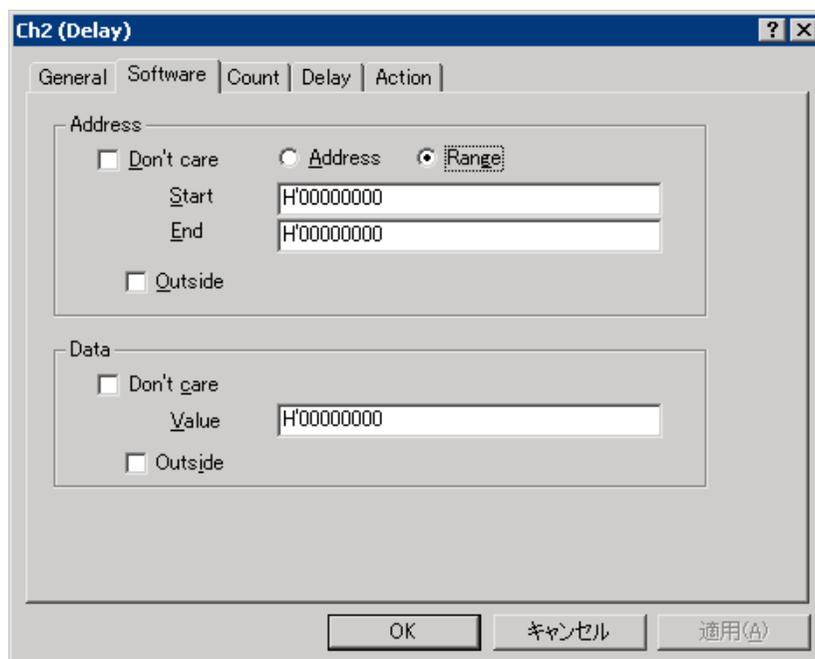


図 5.36 [Chx (Delay)]ダイアログボックス ([Software]ページ)

[Address] アドレス条件を設定します。

- Don't care : アドレス条件を設定しません。
- Address : 単一アドレスを設定します。
- Range : アドレス範囲を設定します。
- Start : 単一アドレスまたはアドレス範囲の開始アドレスを指定します。
- End : アドレス範囲の終了アドレスを指定します。
- Outside : 単一アドレスまたはアドレス範囲に設定された値以外を条件として設定します。

[Data] データ条件を設定します。

- Don't care : データ条件を設定しません。
- Value : データバスの値を設定します。(マスクデータを入力できます)
- Outside : Value に設定されたデータ値以外を条件として設定します。

- 【注】
1. [General]ページで[Software trace data]を指定したときのみ、本ページが表示されます。
  2. AUD イベントチャンネル Ch1, Ch2 のみ、アドレス範囲を指定できます。
  3. 単一アドレスを指定したとき、[Start]にマスクアドレスを入力できます。
  4. アドレス範囲を指定したとき、[Start],[End]にマスクアドレスを入力できません。

## (5) [SystemBus]ページ

システムバストレースのアドレス条件およびデータ条件を指定します。

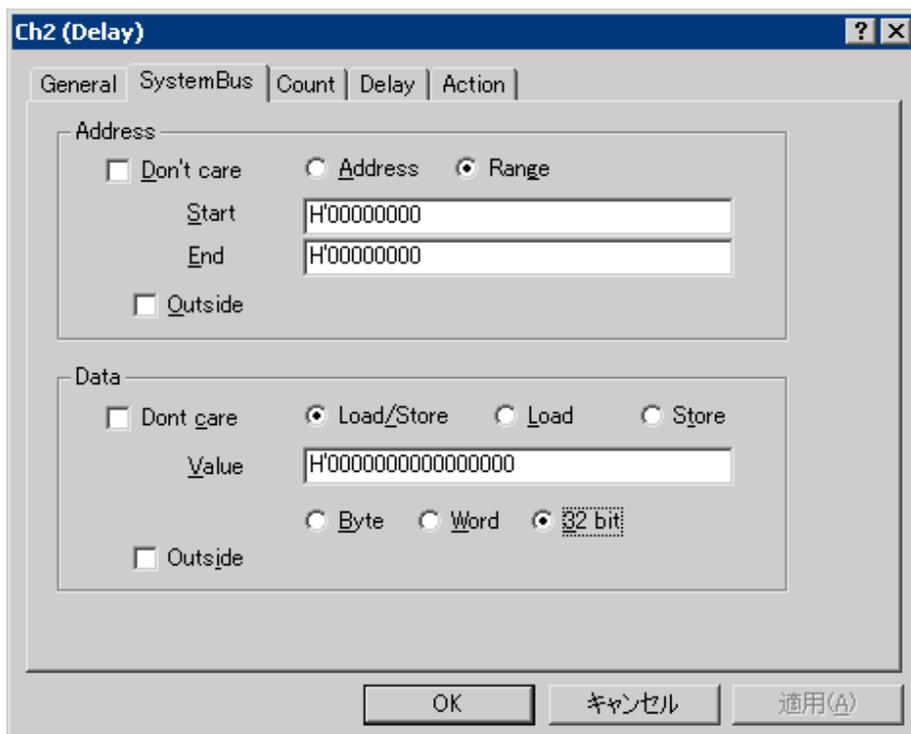


図 5.37 [Chx (Delay)]ダイアログボックス ([SystemBus]ページ)

## 5 デバッグ

---

[Address]	アドレス条件を設定します。
Don't care	: アドレス条件を設定しません。
Address	: 単一アドレスを設定します。
Range	: アドレス範囲を設定します。
Start	: 単一アドレスまたはアドレス範囲の開始アドレスを指定します。
End	: アドレス範囲の終了アドレスを指定します。
Outside	: 単一アドレスまたはアドレス範囲に設定された値以外を条件として設定します。
[Data]	データ条件を設定します。
Don't care	: データ条件を設定しません。
Load/Store	: ロード、ストアサイクルを条件にします。(サポートしません)
Load	: ロードサイクルを条件にします。(サポートしません)
Store	: ストアサイクルを条件にします。
Value	: データバスの値を設定します。(マスクデータを入力できます)
Byte	: バイトアクセスを条件にします。
Word	: ワードアクセスを条件にします。
32 bit	: 32 ビットアクセスを条件にします。
Outside	: Value に設定されたデータ値以外を条件として設定します。

- 【注】
1. [General]ページで[SystemBus trace data]を指定したときのみ、本ページが表示されます。
  2. AUD イベントチャンネル Ch1,Ch2 のみ、アドレス範囲を指定できます。
  3. 単一アドレスを指定したとき、[Start]にマスクアドレスを入力できます。
  4. アドレス範囲を指定したとき、[Start],[End]にマスクアドレスを入力できません。
  5. システムバスのデータ情報は、ロードサイクル(Load)のとき AUD 端子から出力しません。データ条件はストアサイクル(Store)のみ設定できます。

- (6) [Count]ページ  
成立回数条件を指定します。

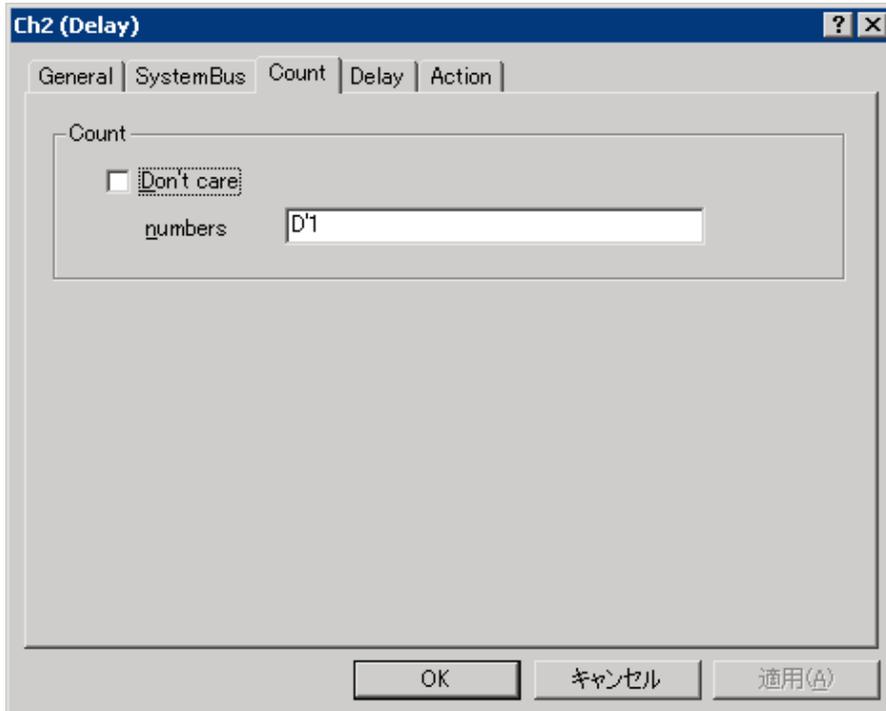


図 5.38 [Chx (Delay)]ダイアログボックス ([Count]ページ)

- [Don't care] 成立回数を設定しません。  
[numbers] 成立回数条件の値を設定します。 D'1 ~ D'65535 の値を設定することができます。

【注】 [Action]ページで[Trace get]を選択したとき、本ページは表示されません。



## (8) [Action]ページ

条件成立後の動作を指定します。

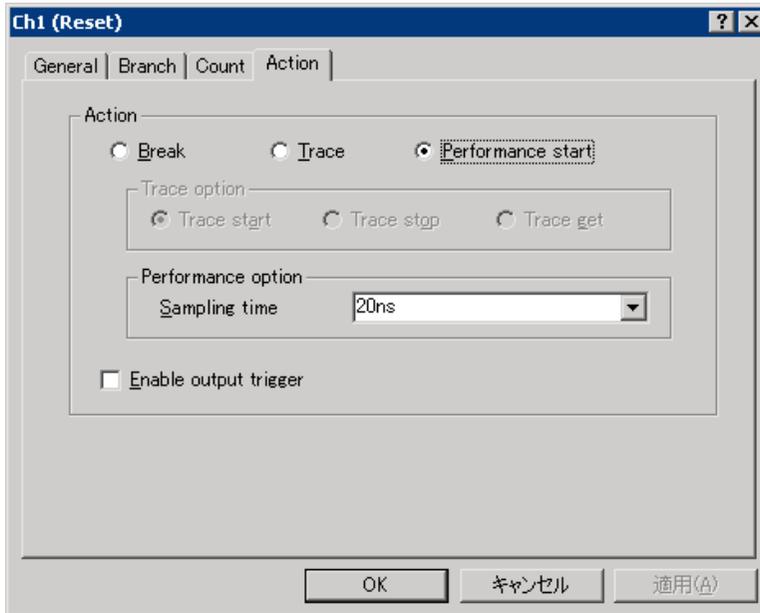


図 5.40 [Chx (Delay)]ダイアログボックス ([Action]ページ)

[Break]	条件が成立したとき、ブレイクします。
[Trace]	<p>選択すると[Trace option]が有効になり、AUD トレース動作を設定します。</p> <p>Trace start : 条件が成立したとき、AUD トレースを開始します。</p> <p>Trace stop : 条件が成立したとき、AUD トレースを停止します。</p> <p>Trace get : 条件が成立したとき、AUD トレースを取得します。</p>
[Performance start]	<p>条件が成立したとき、AUD パフォーマンス測定を開始または終了します。</p> <p>選択すると[Performance option]が有効になり、パフォーマンス測定の時間間隔を指定できます。</p> <p>Sampling time : AUD パフォーマンス測定の時間間隔を指定します。</p> <p>以下のいずれかを選択できます。</p> <p>20ns, 40ns, 100ns, 400ns</p>
[Enable output trigger]	条件が成立したとき、トリガを出力するかを指定します。

- 【注】
- AUD パフォーマンス測定は 2 個の AUD イベントチャンネルを、それぞれ測定開始/測定停止として使用します。イベントチャンネルをパフォーマンス測定として指定したとき、関連チャンネルもパフォーマンス測定として指定してください。(関連チャンネル: Ch1 ~ Ch2, Ch3 ~ Ch4, Ch5 ~ Ch6, Ch7 ~ Ch8)
  - AUD イベントチャンネル Ch1, Ch3, Ch5, Ch7 のみ、[Performance option]が表示されます。

## (9) [Sequential AUD Event]ダイアログボックス

シーケンシャル AUD イベントは、指定した順番で AUD Eventpoint 条件がすべて成立したとき発生するイベントです。

AUD イベントチャンネル Ch1 をリセットポイントとして指定できます。リセットポイントを通過すると、これまでに成立したイベントポイント条件を無効にし、新たに最初のイベントポイント条件からチェックを始めます。

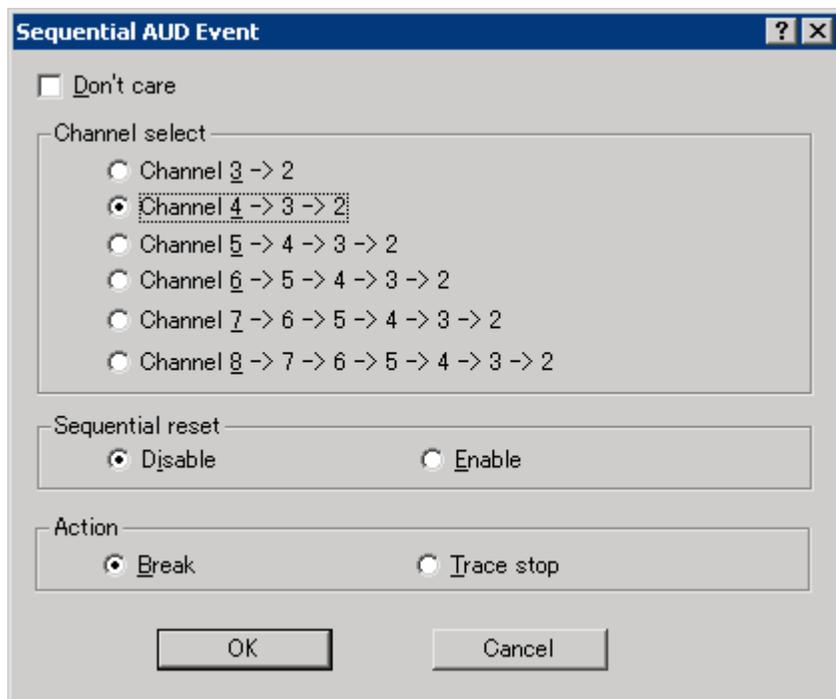


図 5.41 [Sequential AUD Event]ダイアログボックス

[Don't care]	Sequential AUD Event 条件を設定しません。
[Channel select]	Sequential AUD Event が成立する順番を指定します。 Channel 3->2 : AUD イベントチャンネル 3->2 の順番で条件成立すると、Sequential AUD Event が発生します。 Channel 4->3->2 : AUD イベントチャンネル 4->3->2 の順番で条件成立すると、Sequential AUD Event が発生します。 Channel 5->4->3->2 : AUD イベントチャンネル 5->4->3->2 の順番で条件成立すると、Sequential AUD Event が発生します。 Channel 6->5->4->3->2 : AUD イベントチャンネル 6->5->4->3->2 の順番で条件成立すると、Sequential AUD Event が発生します。 Channel 7->6->5->4->3->2 : AUD イベントチャンネル 7->6->5->4->3->2 の順番で条件成立すると、Sequential AUD Event が発生します。 Channel 8->7->6->5->4->3->2 : AUD イベントチャンネル 8->7->6->5->4->3->2 の順番で条件成立すると、Sequential AUD Event が発生します。
[Sequential reset]	AUD イベントチャンネル Ch1 をリセットポイントとして使用するかを選択します。 Disable : リセットポイントとして使用しません。 Enable : リセットポイントとして使用します。
[Action]	Sequential AUD Event を検出した後の動作を指定します。 Break : Sequential AUD Event を検出した後、ブレークします。 Trace stop : Sequential AUD Event を検出した後、AUD トレースストップします。

【注】 Sequential AUD Event 条件を設定すると、[Channel select]で選択された AUD イベントチャンネルでイベントポイント条件を編集するとき、[Action]ページは変更不可になります。[Action]ページの設定を変更するには、Sequential AUD Event 条件を解除してから行ってください。

## 5.6.7 BUS Eventpoint を設定する

[BUS Event]シートでは BUS Eventpoint の設定内容の表示、変更ができます。

Type	State	Condition	Action
Ch1 (Reset)	Enable	address=00001234 to H'5678	Break
Ch2 (Delay)	Disable	None	
Ch3 (Normal)	Enable	data=H'FFFF word read nmi=high irq=×1×01××0	Break
Ch4 (Normal)	Disable	None	
Ch5 (Normal)	Enable	address=00001111 (tutorial.cpp/54)	Trace get
Ch6 (Normal)	Disable	None	

図 5.42 [Event]ウィンドウ ([BUS Event]シート)

6本のイベント検出チャンネルより、6個のイベントポイントを設定できます。

- 【注】
1. トレースユニットを E200F エミュレータに接続していない場合、本機能はサポートしません。
  2. E200F エミュレータのトレースユニットのデバッグ機能を再設定することにより、イベント検出チャンネルの数が変わります。詳細につきましては、「5.1.4 [Bus Board]ページ」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]	外部バスイベントチャンネル番号とタイプを表示します。
	Normal : 標準のイベントチャンネル
	Delay : デレイ条件設定できるイベントチャンネル
	Reset : 外部バスシーケンシャルイベントのリセットポイントとして設定できる イベントチャンネル
[State]	該当イベントポイントの有効/無効を示します。
	Enable : 有効
	Disable : 無効
[Condition]	イベントポイントが成立する条件を表示します。表示内容はチャンネルにより異なります。
[Action]	イベントポイント条件成立時の動作を表示します。表示内容はチャンネルにより異なります。

本ウィンドウでイベントチャンネルをダブルクリックするか、またはイベントチャンネルを選択しポップアップメニューから[編集...]を選択すると、[Chx]ダイアログボックスを表示します。

[Chx]ダイアログボックスは、[Address]ページ、[Data]ページ、[Interrupt]ページ、[Count]ページ、[Delay]ページ、[Action]ページにより構成されています。

各ページで設定された条件を組み合わせせたものを、イベントポイントの検出条件として設定します。

- (1) [Address]ページ  
アドレス条件を指定します。

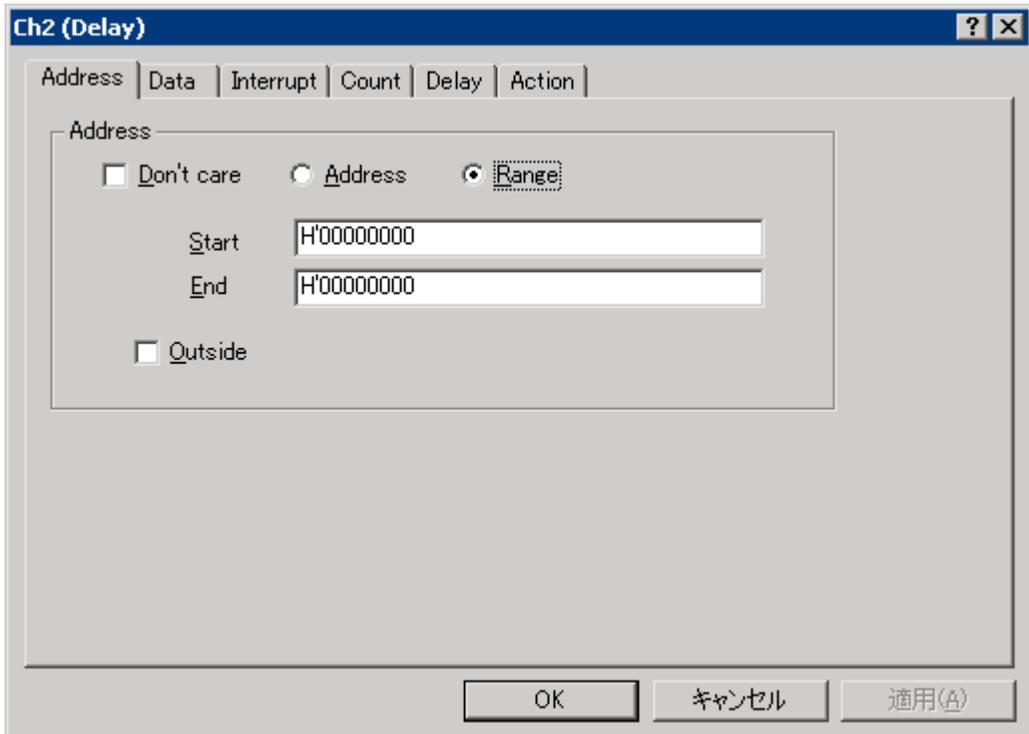


図 5.43 [Chx (Delay)]ダイアログボックス ([Address]ページ)

- [Address] アドレス条件を設定します。
- Don't care : アドレス条件を設定しません。
  - Address : 単一アドレスを設定します。
  - Range : アドレス範囲を設定します。
  - Start : 単一アドレスまたはアドレス範囲の開始アドレスを指定します。  
(マスクアドレスを入力できます)
  - End : アドレス範囲の終了アドレスを指定します。  
(マスクアドレスを入力できます)
  - Outside : 単一アドレスまたはアドレス範囲に設定された値以外を条件として設定します。

- (2) [Data]ページ  
 データバス条件を指定します。

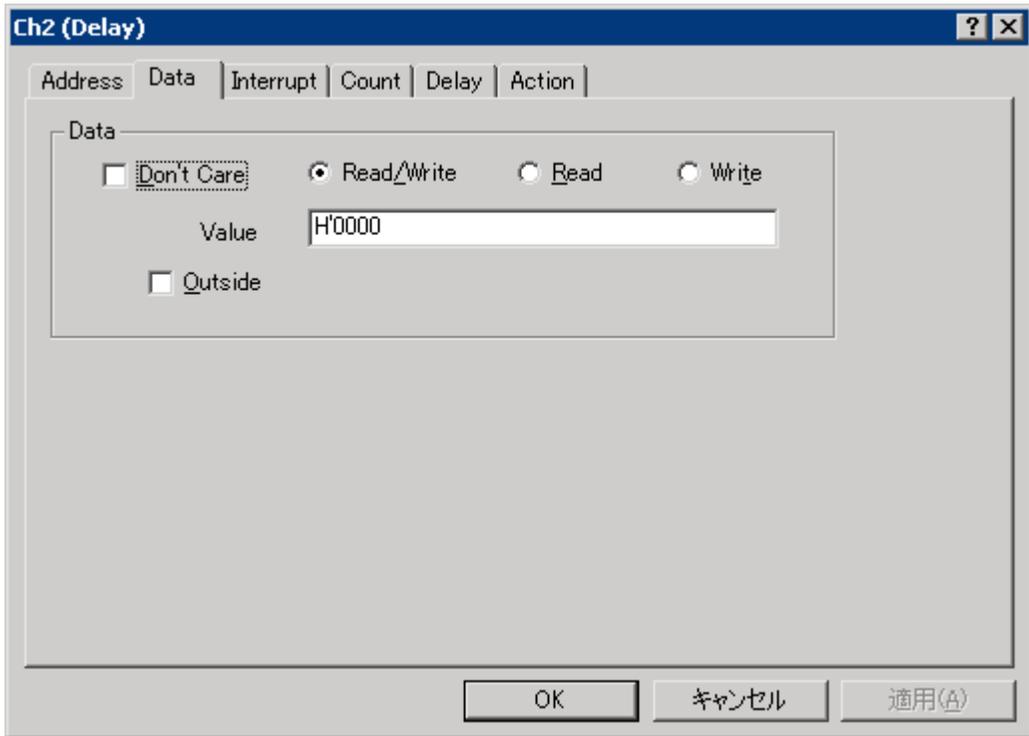


図 5.44 [Chx (Delay)]ダイアログボックス ([Data]ページ)

[Data] データ条件を設定します。

- Don't care : データ条件を設定しません。
- Read/Write : リード、ライトサイクルを条件にします。
- Read : リードサイクルを条件にします。
- Write : ライトサイクルを条件にします。
- Value : データバスの値を設定します。(マスクデータを入力できます)
- Outside : Value に設定されたデータ値以外を条件として設定します。

## (3) [Interrupt]ページ

NMI と外部割り込みの信号条件を指定します。

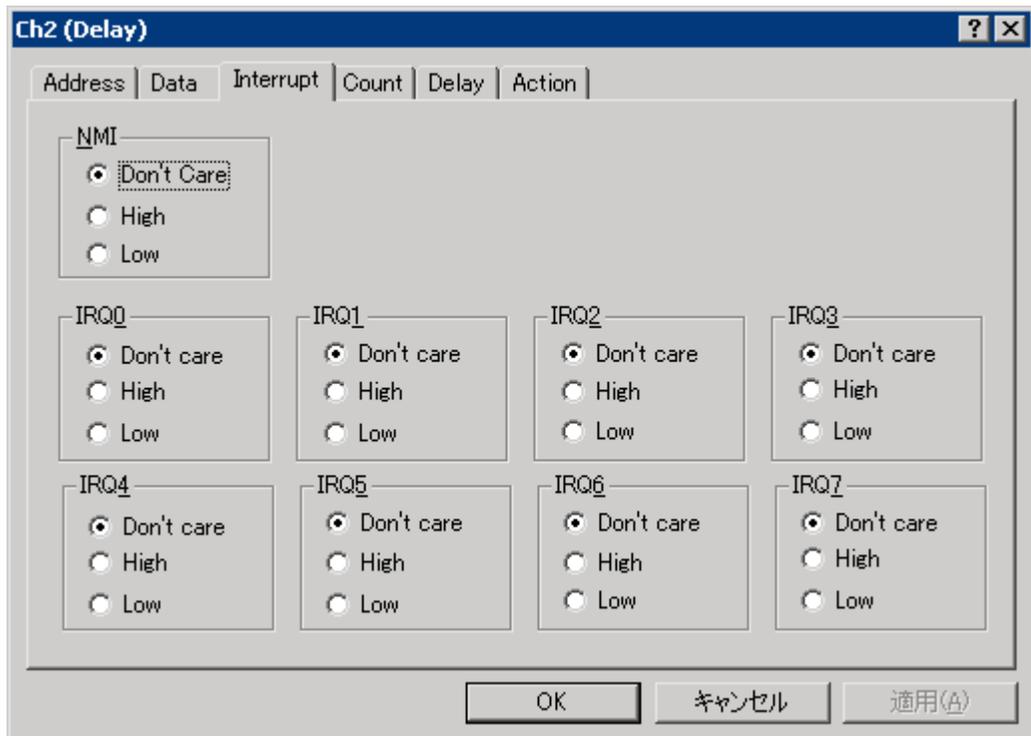


図 5.45 [Chx (Delay)]ダイアログボックス ([Interrupt]ページ)

- [Don't care] 信号条件を設定しません。  
 [High] 信号が High レベルの場合を条件成立とします。  
 [Low] 信号が Low レベルの場合を条件成立とします。

【注】 サポートデバイスによっては、外部割り込み信号が異なります。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。

- (4) [Count]ページ  
成立回数条件を指定します。

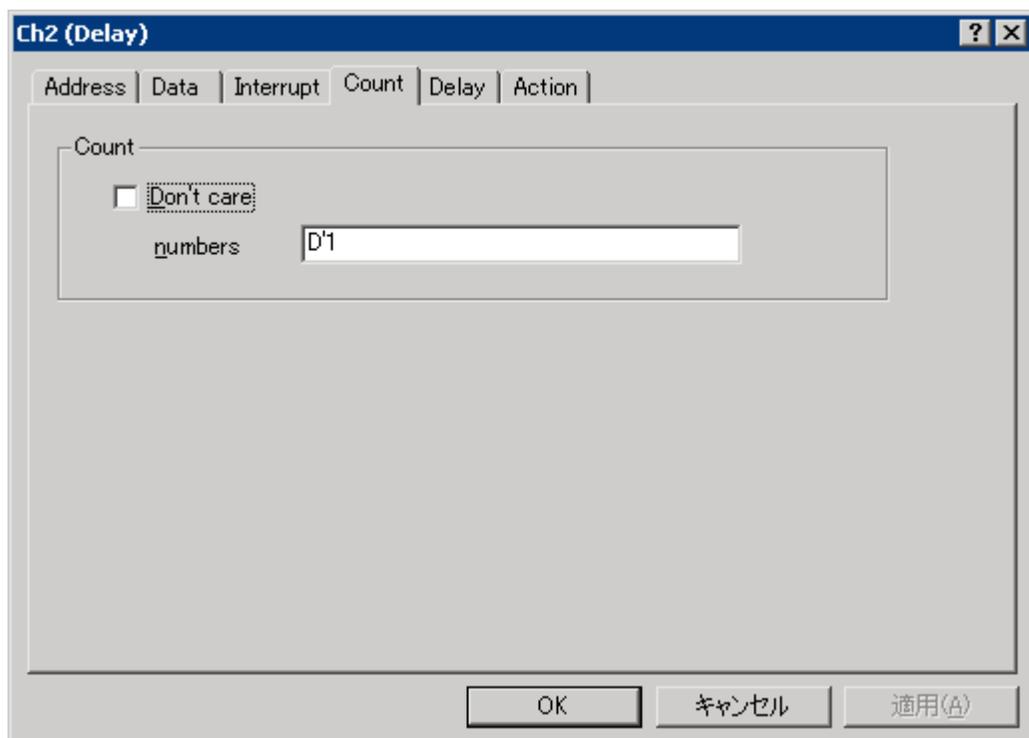


図 5.46 [Chx (Delay)]ダイアログボックス ([Count]ページ)

- [Don't care] 成立回数を設定しません。  
[numbers] 成立回数条件の値を設定します。  
D'1 ~ D'65535 の値を設定することができます。

【注】 [Action]ページで[Trace get]を選択したとき、本ページは表示されません。

## (5) [Delay]ページ

イベント検出してから外部バストレースストップまでのディレイサイクルを設定します。

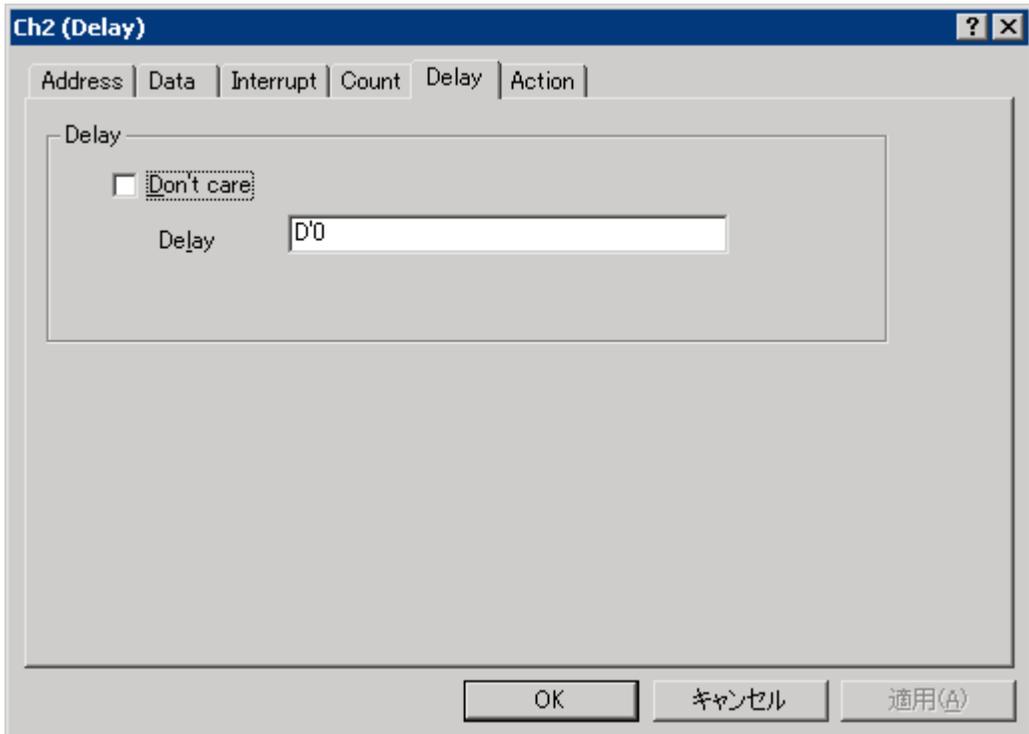


図 5.47 [Chx (Delay)]ダイアログボックス ([Delay]ページ)

[Don't care]	ディレイ条件を設定しません。
[Delay]	ディレイサイクルの数を設定します。 D'0 ~ D'262143 の値を設定できます。

【注】 外部バスイベントチャンネル Ch2、かつ[Action]ページで[Trace stop]を選択したときのみ、本ページが表示されます。

(6) [Action]ページ

条件成立後の動作を指定します。

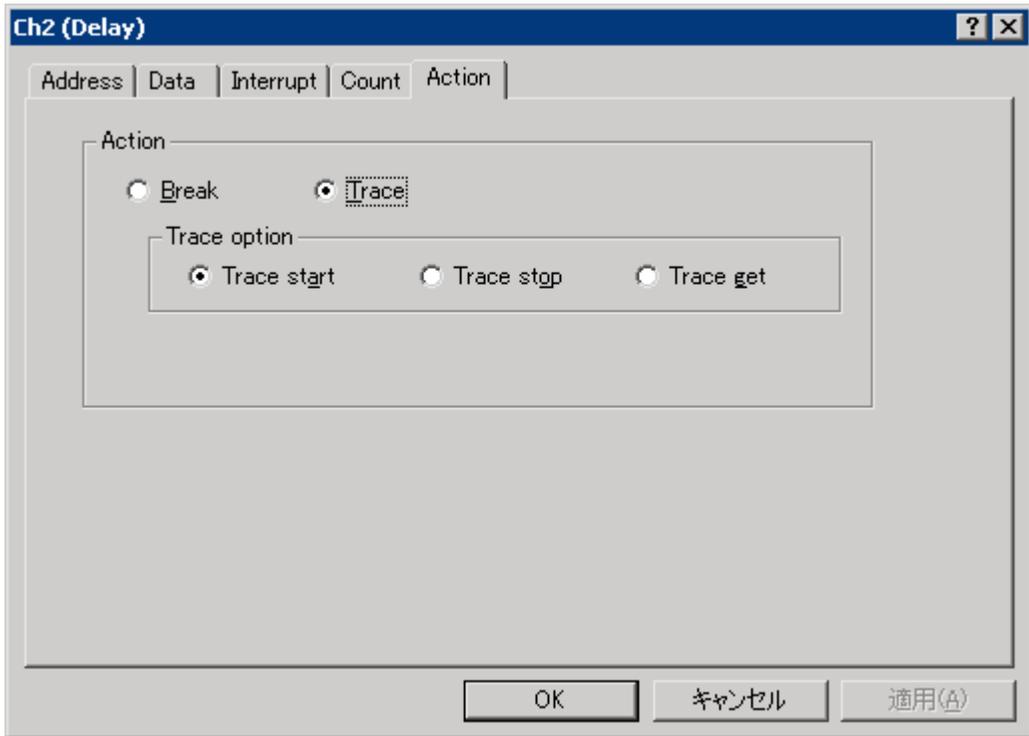


図 5.48 [Chx (Delay)]ダイアログボックス ([Action]ページ)

[Break] 条件が成立したとき、ブレイクします。

[Trace] 選択すると[Trace option]が有効になり、外部バストレース動作を設定します。

Trace start : 条件が成立したとき、外部バストレースを開始します。

Trace stop : 条件が成立したとき、外部バストレースを停止します。

Trace get : 条件が成立したとき、外部バストレースを取得します。

## (7) [Sequential BUS Event]ダイアログボックス

シーケンシャル BUS イベントは、指定した順番で BUS Eventpoint 条件がすべて成立したとき発生するイベントです。

BUS イベントチャンネル Ch1 をリセットポイントとして指定できます。リセットポイントを通過すると、これまでに成立したイベントポイント条件を無効にし、新たに最初のイベントポイント条件からチェックを始めます。

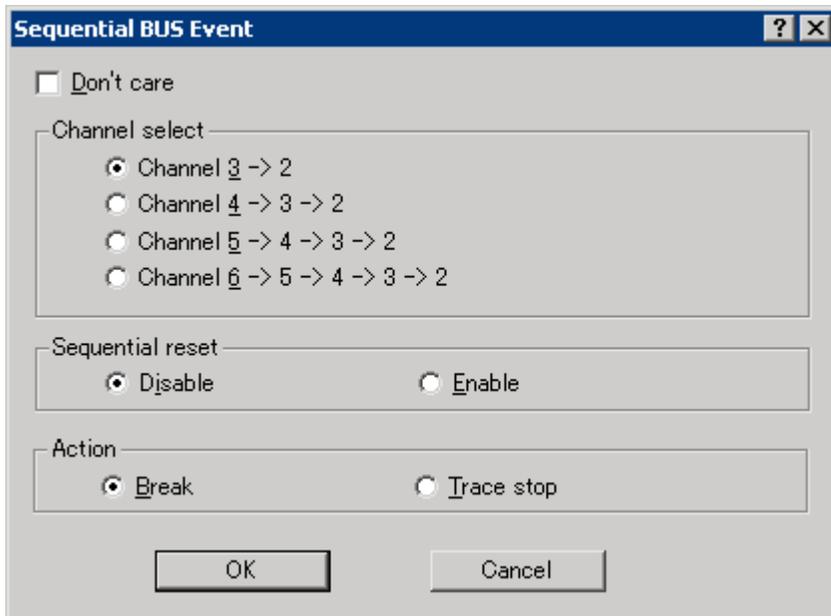


図 5.49 [Sequential BUS Event]ダイアログボックス

[Don't care]	Sequential BUS Event 条件を設定しません。
[Channel select]	Sequential BUS Event が成立する順番を指定します。 Channel 3->2 : BUS イベントチャンネル 3->2 の順番で条件成立すると、Sequential BUS Event が発生します。 Channel 4->3->2 : BUS イベントチャンネル 4->3->2 の順番で条件成立すると、Sequential BUS Event が発生します。 Channel 5->4->3->2 : BUS イベントチャンネル 5->4->3->2 の順番で条件成立すると、Sequential BUS Event が発生します。 Channel 6->5->4->3->2 : BUS イベントチャンネル 6->5->4->3->2 の順番で条件成立すると、Sequential BUS Event が発生します。
[Sequential reset]	BUS イベントチャンネル Ch1 をリセットポイントとして使用するかを選択します。 Disable : リセットポイントとして使用しません。 Enable : リセットポイントとして使用します。
[Action]	Sequential BUS Event を検出した後の動作を指定します。 Break : Sequential BUS Event を検出した後、ブレイクします。 Trace stop : Sequential BUS Event を検出した後、外部バストレースストップします。

【注】 Sequential BUS Event 条件を設定すると、[Channel select]で選択された BUS イベントチャンネルでイベントポイント条件を編集するとき、[Action]ページは変更不可になります。[Action]ページの設定を変更するには、Sequential BUS Event 条件を解除してから行ってください。

シーケンシャル設定時は、シーケンシャル設定を行っているチャンネル以外は条件が成立しません。

## 5.6.8 Other Eventpoint を設定する

[Other Event]シートでは Other Eventpoint の設定内容の表示、変更ができます。

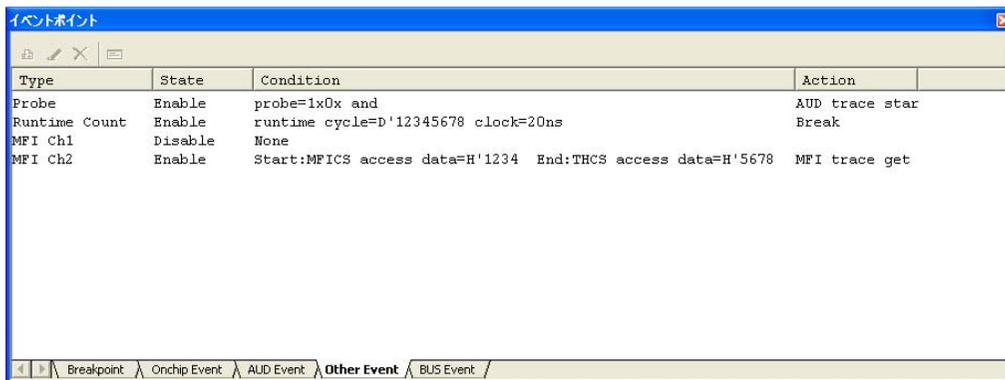


図 5.50 [Event]ウィンドウ ([Other Event]シート)

シート内に表示する項目は以下の通りです。

ご使用になっているオプションユニットの状況により、表示内容が追加になる場合があります。

[Type]	その他のイベントチャンネルを表示します。
	Probe : 外部プローブイベントチャンネル
	Runtime Count : 実行時間イベントチャンネル
[State]	該当イベントポイントの有効/無効を示します。
	Enable : 有効
	Disable : 無効
[Condition]	イベントポイントが成立する条件を表示します。表示内容はチャンネルにより異なります。
[Action]	イベントポイント条件成立時の動作を表示します。表示内容はチャンネルにより異なります。

本ウィンドウでイベントチャンネルをダブルクリックするか、またはイベントチャンネルを選択しポップアップメニューから[編集...]を選択すると、条件設定用ダイアログボックスを表示します。条件設定用ダイアログボックスの表示内容はイベントチャンネルにより異なります。

(1) [Probe]ダイアログボックス

[Probe]イベントチャンネルをダブルクリックすると表示します。

[Probe]ダイアログボックスは、[Condition]ページ、[Action]ページにより構成されています。

(a) [Condition]ページ

外部プローブケーブル経由の4つの外部プローブ信号条件を指定します。

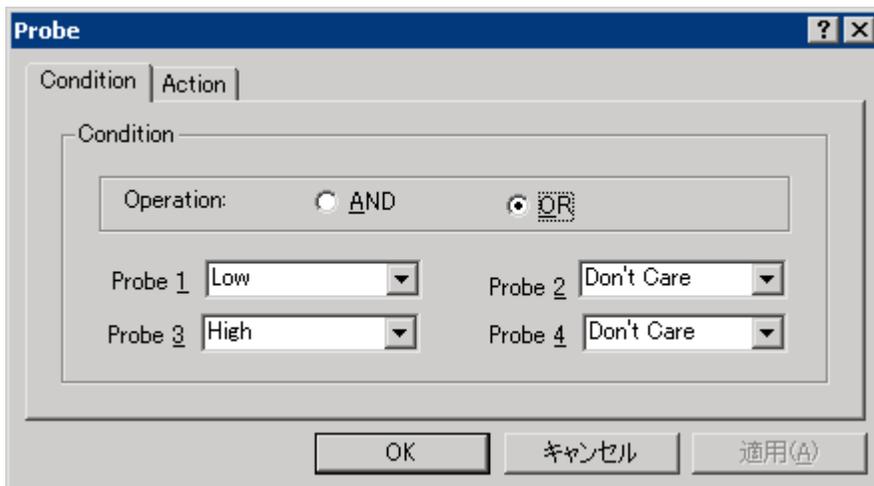


図 5.51 [Probe]ダイアログボックス ([Condition]ページ)

- [Operation]            プローブ信号条件の組み合わせ方法を指定します。
- AND                    : プロブ信号条件の AND 演算により成立を判定します。
- OR                     : プロブ信号条件の OR 演算により成立を判定します。
- [Probe 1 ~ 4]        プロブ信号条件を指定します。
- Don't care            : プロブ信号条件を指定しません。
- High                   : プロブ信号が High レベルの場合を条件成立とします。
- Low                    : プロブ信号が Low レベルの場合を条件成立とします。

(b) [Action]ページ

条件成立後の動作を指定します。

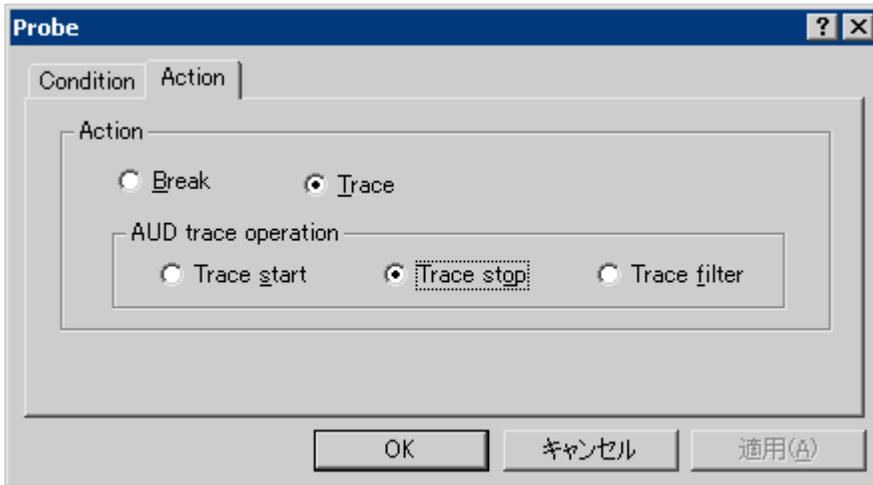


図 5.52 [Probe]ダイアログボックス ([Action]ページ)

[Break]

条件が成立したとき、ブレイクします。

[Trace]

選択すると[AUD trace option]が有効になり、AUD トレース動作を設定します。

Trace start : 条件が成立したとき、AUD トレースを開始します。

Trace stop : 条件が成立したとき、AUD トレースを停止します。

Trace filter : 条件が成立したとき、AUD トレース取得のフィルタ条件とします。

(2) [Runtime Count]ダイアログボックス

[Runtime Count]イベントチャンネルをダブルクリックすると表示します。

[Runtime Count]ダイアログボックスは、[Condition]ページ、[Action]ページにより構成されています。

(a) [Condition]ページ

ユーザプログラムの実行時間を指定します。

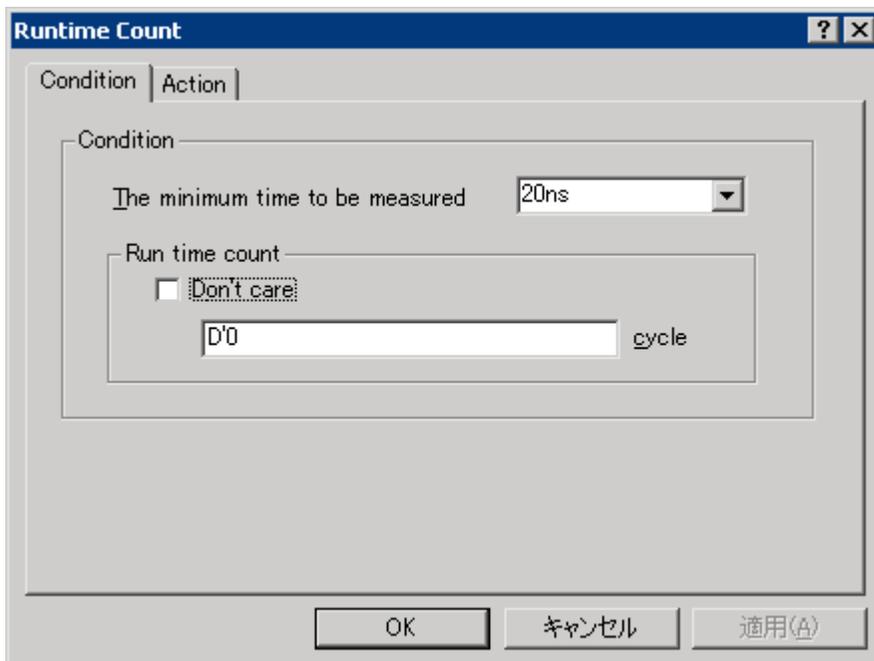


図 5.53 [Runtime Count]ダイアログボックス ([Condition]ページ)

[The minimum time to be measured]

実行時間測定用タイムの分解能を指定します。

以下のいずれかを選択できます。

20ns, 40ns, 100ns, 400ns

[Runtime count]

実行時間条件を指定します。

Don't care : 実行時間条件を設定しません。

Cycle : 測定サイクルの数を入力します。D'0 ~ D'1099511627775 の値を設定できます。

(b) [Action]ページ

条件成立後の動作を指定します。

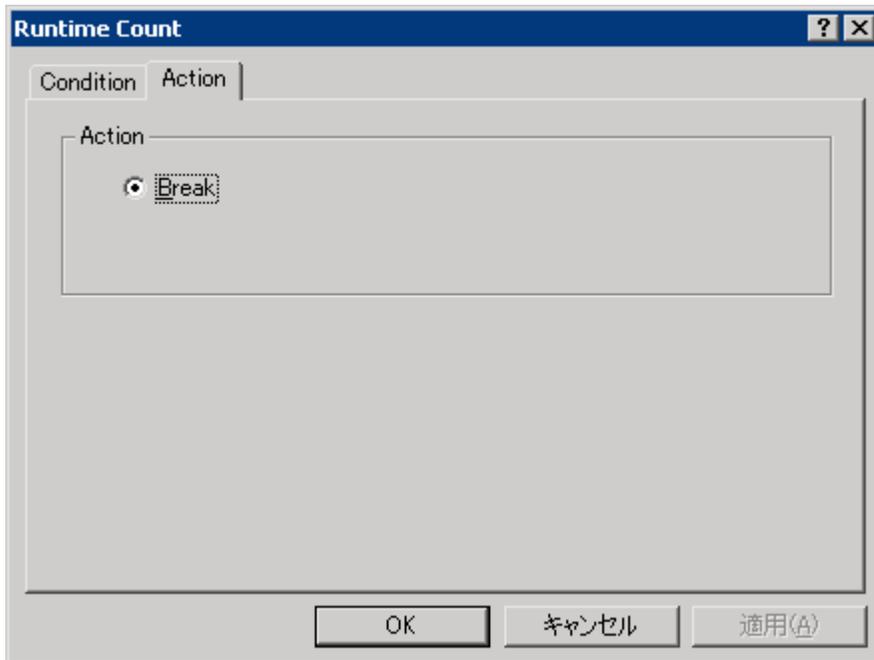


図 5.54 [Runtime Count]ダイアログボックス ([Action]ページ)

[Break]

条件が成立したとき、ブレイクします。

### 5.6.9 ブレークポイント/イベントポイントの編集

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

### 5.6.10 ブレークポイント/イベントポイントを有効にする

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

### 5.6.11 ブレークポイント/イベントポイントを無効にする

ブレークポイント/イベントポイントを選択後、ポップアップメニューから[無効]を選択すると、選択しているブレークポイント/イベントポイントを無効にします。無効にした場合は、指定条件が一致しても条件成立しません。

### 5.6.12 ブレークポイント/イベントポイントを削除する

ブレークポイント/イベントポイントを選択後、ポップアップメニューから[削除]を選択すると、選択しているブレークポイント/イベントポイントを削除します。ブレークポイント/イベントポイントを削除しないで、詳細情報は保持したまま、条件が一致しても成立させないようにするには、[無効]オプションを使用します。(「5.6.11 ブレークポイント/イベントポイントを無効にする」参照)

### 5.6.13 ブレークポイント/イベントポイントをすべて削除する

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

### 5.6.14 ブレークポイント/イベントポイントのソース行を表示する

ブレークポイント/イベントポイントを選択後、ポップアップメニューから[ソースを表示]を選択すると、ブレークポイント/イベントポイントのある[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ブレークポイント/イベントポイントを1個選択しているときのみ有効となります。

## 5.7 トレース情報を見る

E200F エミュレータでは、デバイス内部の情報をトレース取得する機能とデバイス外部の情報をトレース取得する機能があります。

内蔵トレース (Internal trace)、AUD トレース (AUD trace)、ユーザメモリに出力するトレース (Usermemory trace)、および外部バストレース (BUS trace) の 4 種類のトレース情報を取得できます。製品によっては周辺 I/O トレース情報を取得することができます。

Internal trace、AUD trace および Usermemory trace のトレース情報は [Internal/AUD/Usermemory trace] ウィンドウに表示します。BUS trace のトレース情報は [BUS] ウィンドウに表示します。

Trace 機能の説明は、「1.3.3 トレース機能」を参照してください。

### 5.7.1 [Internal/AUD/Usermemory trace] ウィンドウを開く

[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  をクリックすると、[Trace Window Type] ダイアログボックスが開きます。



図 5.55 [トレースウィンドウの選択] ダイアログボックス

[Internal/AUD/Usermemory trace] を選択し [OK] ボタンをクリックすると、[Internal/AUD/Usermemory trace] ウィンドウが表示されます。

## 5 デバッグ

### (1) 内蔵トレース情報 (Internal trace) を取得する

[Internal/AUD/Usermemory trace]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[Internal/AUD/Usermemory 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[Internal 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。

E200F エミュレータはトレース情報の取得条件を設定しない場合、デフォルトで内蔵トレース機能でトレース情報を取得します。

取得したトレース情報は[Internal/AUD/Usermemory trace]ウィンドウに表示します。

PTR	IP	Master	Type	BranchType	B R. Address	D.. P. P.	Instruction	P. T.. T.. Source
-000013	-d*000006	CPU	BRANCH	GENERAL	00002114		BF	@H'20F8:8
+000012			DESTINATION		000020F8		MOV	R15, R1
-000011	-d*000005	CPU	BRANCH	GENERAL	00002114		BF	@H'20F8:8
-000010			DESTINATION		000020F8		MOV	R15, R1
-000009	-d*000004	CPU	BRANCH	GENERAL	00002114		BF	@H'20F8:8
-000008			DESTINATION		000020F8		MOV	R15, R1
-000007	-d*000003	CPU	BRANCH	GENERAL	00002114		BF	@H'20F8:8
-000006			DESTINATION		000020F8		MOV	R15, R1
-000005	-d*000002	CPU	BRANCH	GENERAL	00002114		BF	@H'20F8:8
-000004			DESTINATION		000020F8		MOV	R15, R1
-000003	-d*000001	CPU	BRANCH	GENERAL	00002114		BF	@H'20F8:8
-000002			DESTINATION		000020F8		MOV	R15, R1
-000001	-d*000000	CPU	BRANCH	SUBROUTINE	0000211A		FS	
+000000			DESTINATION		00001098		MOV	R15, R2

図 5.56 [Internal/AUD/Usermemory trace]ウィンドウ (Internal trace の場合)

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。

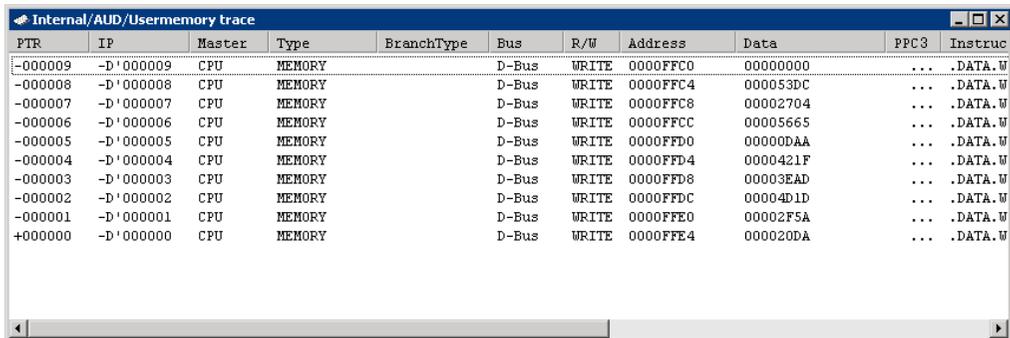
[PTR]	トレースバッファ内ポインタ (最後に実行した命令が +0 となります)
[IP]	取得したトレース情報数
[Master](Bus Master)	アクセスを行ったバスマスタの種類
[Type]	トレース情報種別
	BRANCH : 分岐元
	DESTINATION : 分岐先
	MEMORY : メモリアクセス
	S_TRACE : Trace(x)関数実行したことを示す
	LOST : トレース情報が失われたことを示す (リアルタイムモード時のみ)
[Branch Type]	分岐種別(分岐トレース取得時のみ)
	GENERAL : 一般分岐
	SUBROUTINE : サブルーチン分岐
	EXCEPTION : 例外分岐
[Bus]	ど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
[PPC3]	内蔵パフォーマンスカウンタ Ch3 の出力
[PPC4]	内蔵パフォーマンスカウンタ Ch4 の出力
[Instruction]	命令モニタ
[Source]	C/C++またはアセンブラソース
[Label]	ラベル情報

## (2) AUD トレース情報 (AUD trace) を取得する

[Internal/AUD/Usermemory trace]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[Internal/AUD/Usermemory 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 trace]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。

**【注】** AUD トレース機能は AUD 端子から出力するチップ内の情報により実現するため、Onchip Eventpoint のトレース取得条件を設定する必要があります。AUD トレース取得条件に応じて、オンチップイベントチャネル Ch5 (OA), Ch6 (OA), Ch7 (SystemBus), Ch8 (SystemBus), Ch12 (Branch)または Software Trace のトレース取得条件を設定してください。



FTR	IP	Master	Type	BranchType	Bus	R/W	Address	Data	PPC3	Instruc
-000009	-D'000009	CPU	MEMORY		D-Bus	WRITE	0000FFC0	00000000	...	.DATA.W
-000008	-D'000008	CPU	MEMORY		D-Bus	WRITE	0000FFC4	000053DC	...	.DATA.W
-000007	-D'000007	CPU	MEMORY		D-Bus	WRITE	0000FFC8	00002704	...	.DATA.W
-000006	-D'000006	CPU	MEMORY		D-Bus	WRITE	0000FFCC	00005665	...	.DATA.W
-000005	-D'000005	CPU	MEMORY		D-Bus	WRITE	0000FFD0	00000DAA	...	.DATA.W
-000004	-D'000004	CPU	MEMORY		D-Bus	WRITE	0000FFD4	0000421F	...	.DATA.W
-000003	-D'000003	CPU	MEMORY		D-Bus	WRITE	0000FFD8	00003EAD	...	.DATA.W
-000002	-D'000002	CPU	MEMORY		D-Bus	WRITE	0000FFDC	00004D1D	...	.DATA.W
-000001	-D'000001	CPU	MEMORY		D-Bus	WRITE	0000FFE0	00002F5A	...	.DATA.W
+000000	-D'000000	CPU	MEMORY		D-Bus	WRITE	0000FFE4	000020DA	...	.DATA.W

図 5.57 [Internal/AUD/Usermemory trace]ウィンドウ (AUD trace の場合)

## 5 デバッグ

---

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]	AUD 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)
[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
[PPC3]	内蔵パフォーマンスカウンタ Ch3 の出力
[PPC4]	内蔵パフォーマンスカウンタ Ch4 の出力
[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
[Timestamp-Difference]	タイムスタンプの差分値

【注】 表示する内容は製品によって異なりますので、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。

### (3) ユーザメモリに出力したトレース情報 (Usermemory trace) を取得する

[Internal/AUD/Usermemory trace]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[Internal/AUD/Usermemory 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[Usermemory trace]を選択するとユーザメモリに出力したトレース情報を取得します。

表示する項目は、内蔵トレース情報を取得するときの表示項目と同じです。詳細につきましては、「5.7.1 (1) 内蔵トレース情報 (Internal trace) を取得する」を参照してください。

## (4) トレース情報取得条件/取得モードを設定する

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イベントポイントにて実現し、トレース情報の取得開始、取得終了、取得を制御します。

イベントポイントについては、「5.6 イベントポイントを使用する」を参照してください。

トレース情報の取得モードは、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 [Internal/AUD/Usermemory acquisition]ダイアログボックスで設定します。

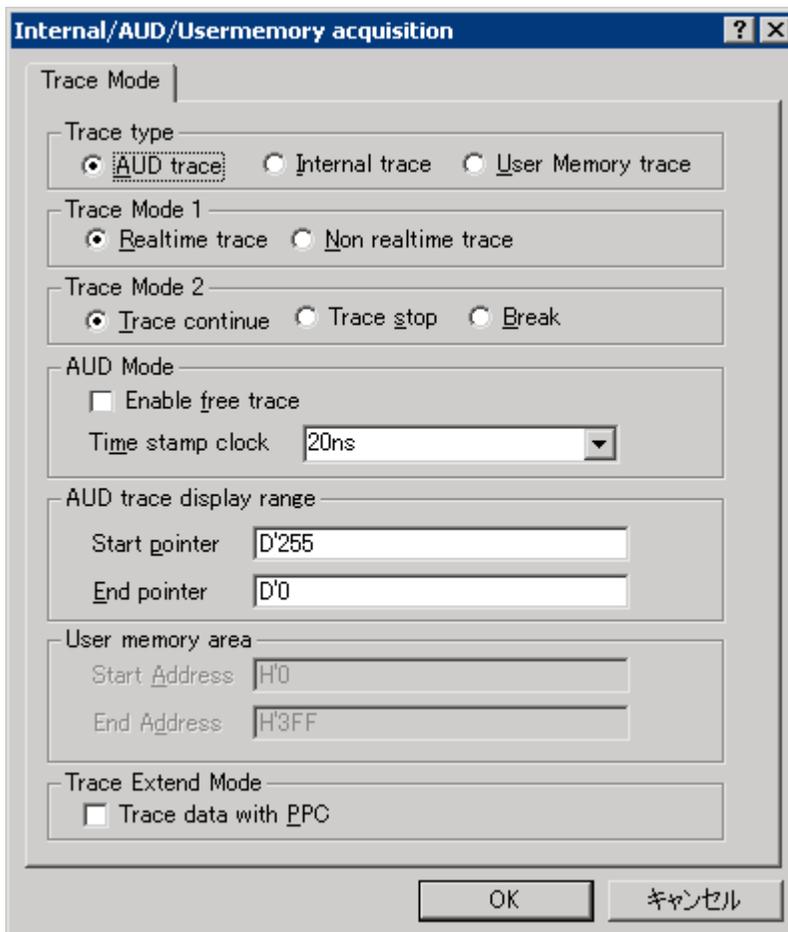


図 5.58 [Internal/AUD/Usermemory acquisition]ダイアログボックス

## 5 デバッグ

---

[Trace type]	トレース機能の種類を選択します。 AUD trace : AUD トレース機能を選択します。 Internal trace : 内蔵トレース機能を選択します。 User Memory trace : ユーザメモリ出力トレース機能を選択します。
[Trace Mode 1]	トレース情報が連続して発生した場合の動作モードを選択します。 Realtime trace : 一部のトレース情報が出力されません。 Non realtime trace : トレース情報が出力されるまで CPU が待たされます。
[Trace Mode 2]	E200F エミュレータの AUD トレースバッファが Full になったときの動作を設定します。 Trace continue : 古いトレース情報に上書きして、常に最新の情報を取得します。 Trace stop : 以後のトレース情報を取得しません。 Break : ブレークします。
[AUD Mode]	AUD トレース機能を使用するときのモードを設定します。 Enable free trace : チェックすると、AUD イベントポイントの設定を無視し、トレース情報をすべて取得します。 Time stamp clock : タイムスタンプ用タイマの分解能を選択します。 以下のいずれかを選択できます。 20ns, 40ns, 100ns, 400ns
[AUD trace display range]	AUD トレース機能を使用するときのトレース表示範囲を設定します。 Start pointer : 表示範囲の先頭ポインタ値を数値で入力します。 End pointer : 表示範囲の終了ポインタ値を数値で入力します。
[User memory area]	ユーザメモリ出力トレース機能を使用するときのトレース表示範囲を設定します。 Start Address : 表示範囲の先頭アドレスを設定します。 End Address : 表示範囲の終了アドレスを設定します。
[Trace Extend Mode]	トレース取得項目として、内蔵パフォーマンスカウンタ Ch3, Ch4 を含めるかを設定します。 Trace data with PPC : チェックすると、内蔵パフォーマンスカウンタ Ch3, Ch4 を含めます。

- 【注】
1. [Internal trace]を選択した場合、[Trace Mode 1], [Trace Mode 2], [AUD Mode], [AUD trace display range], [User memory area]項目は無効になります。
  2. [AUD trace]を選択したときのみ、[Trace Mode 2], [AUD Mode], [AUD trace display range]が有効になります。
  3. [User Memory trace]を選択したときのみ、[User memory area]が有効になります。

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。

## 5.7.2 [BUS/MFI trace]ウィンドウを開く

[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すると、[Trace Window Type]ダイアログボックスが開きます。

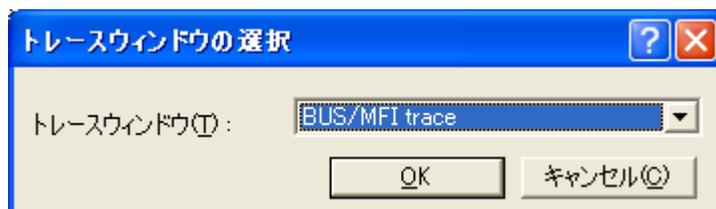


図 5.59 [トレースウィンドウの選択]ダイアログボックス

[BUS/MFI trace]を選択し[OK]ボタンをクリックすると、[BUS/MFI trace]ウィンドウが表示されます。

- 【注】
1. トレースユニットを E200F エミュレータに接続していない場合、外部バストレース機能はサポートしません。
  2. E200F エミュレータのトレースユニットの機能を入れ替えることにより、外部バストレース機能で取得できるバスサイクル数が変わります。詳細につきましては、「5.1.4 [Bus Board]ページ」を参照してください。
  3. 製品によっては、本ウィンドウに BUS トレースだけでなく、他のトレース機能も表示することができます。例えば SH-Mobile 系デバイスでは、BUS/MFI Trace と表示します。  
この場合、MFI 経由のアクセスについてトレース取得した結果を表示します。  
各製品の仕様については、オンラインヘルプを参照してください。

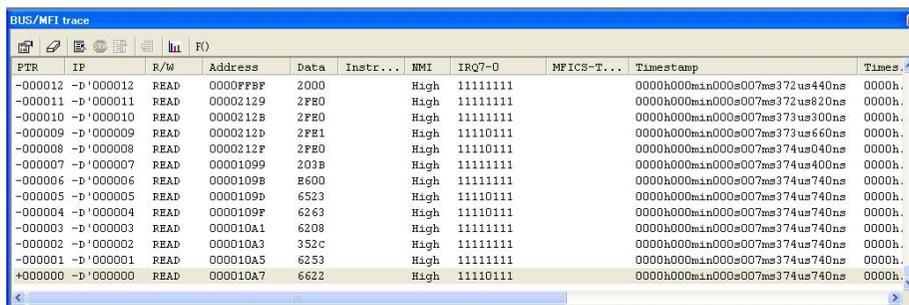
## 5 デバッグ

### (1) 外部バストレース情報 (BUS/MFI trace) を取得する

[BUS/MFI trace]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[BUS 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[BUS/MFI trace]を選択すると外部バストレース機能を使用してトレース情報を取得します。

E200F エミュレータはトレース情報の取得条件を設定しない場合、デフォルトで外部バストレース機能でトレース情報を取得します。

取得したトレース情報は[BUS/MFI trace]ウィンドウに表示します。



PTR	IP	R/W	Address	Data	Instr...	NMI	IRQ7-0	MFICS-T...	Timestamp	Times
-000012	-D '000012	READ	0000FFFF	2000		High	11111111		0000h000msn000e007ms372us440ns	0000h.
-000011	-D '000011	READ	00002129	2FE0		High	11111111		0000h000msn000e007ms372us820ns	0000h.
-000010	-D '000010	READ	0000212B	2FE0		High	11111111		0000h000msn000e007ms373us300ns	0000h.
-000009	-D '000009	READ	0000212d	2FE1		High	11110111		0000h000msn000e007ms373us660ns	0000h.
-000008	-D '000008	READ	0000212F	2FE0		High	11110111		0000h000msn000e007ms374us040ns	0000h.
-000007	-D '000007	READ	00001099	203B		High	11111111		0000h000msn000e007ms374us400ns	0000h.
-000006	-D '000006	READ	0000109B	E60B		High	11111111		0000h000msn000e007ms374us740ns	0000h.
-000005	-D '000005	READ	0000109D	6523		High	11110111		0000h000msn000e007ms374us740ns	0000h.
-000004	-D '000004	READ	0000109F	6263		High	11110111		0000h000msn000e007ms374us740ns	0000h.
-000003	-D '000003	READ	000010A1	6208		High	11111111		0000h000msn000e007ms374us740ns	0000h.
-000002	-D '000002	READ	000010A3	352C		High	11111111		0000h000msn000e007ms374us740ns	0000h.
-000001	-D '000001	READ	000010A5	6253		High	11111111		0000h000msn000e007ms374us740ns	0000h.
+000000	-D '000000	READ	000010A7	6622		High	11110111		0000h000msn000e007ms374us740ns	0000h.

図 5.60 [BUS/MFI trace]ウィンドウ (BUS trace の場合)

表示する項目は以下の通りです。

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。

[PTR]	BUS トレースバッファ内のポインタ (最後に取得されたサイクルが+0 となります)
[IP]	取得したトレース情報数
[R/W]	発生したアクセスサイクルが、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]	データバスの値
[NMI]	NMI 入力の状態
[IRQ5-0]	6 本の IRQ 入力の状態
[Instruction]	命令モニタ
[Time stamp]	バスサイクルのタイムスタンプ
[Timestamp-Difference]	タイムスタンプの差分値

## (2) トレース情報取得条件/取得モードを設定する

トレースバッファは有限であるため、バッファがフル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イベントポイントにて実現し、トレース情報の取得開始、取得終了、取得を制御します。

イベントポイントについては、「5.6 イベントポイントを使用する」を参照してください。

トレース情報の取得モード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[BUS acquisition]ダイアログボックスで設定します。

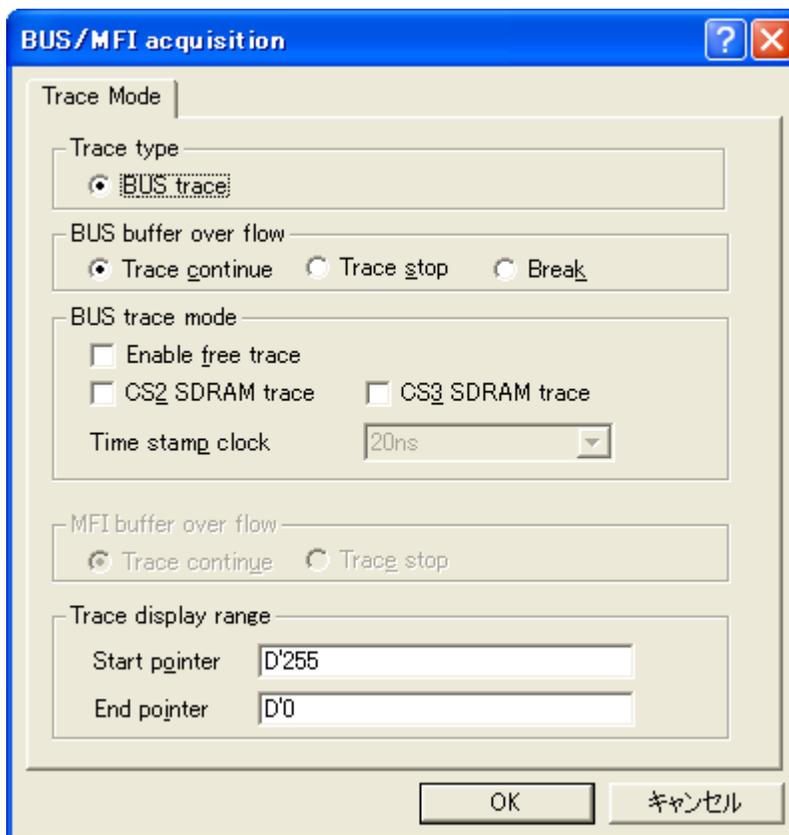


図 5.61 [BUS acquisition]ダイアログボックス

## 5 デバッグ

---

- [BUS buffer over flow] 外部バストレースバッファがフルになったときの動作を設定します。
- Trace continue : 古いトレース情報に上書きして、常に最新の情報を取得します。
  - Trace stop : 以後のトレース情報を取得しません。
  - Break : ブレークします。
- [BUS trace mode] 外部バストレース機能を使用するときのモードを設定します。
- Enable free trace : チェックすると、外部バスフリートレースを許可します。
  - CS2 SDRAM trace : チェックすると、CS2 エリアを SDRAM と認識してトレース解析します。
  - CS3 SDRAM trace : チェックすると、CS3 エリアを SDRAM と認識してトレース解析します。
  - Time stamp clock : タイムスタンプ用タイマの分解能を選択します。  
以下のいずれかを選択できます。  
20ns, 40ns, 100ns, 400ns
- [Trace display range] トレース表示範囲を設定します。
- Start pointer : 表示範囲の先頭ポインタ値を数値で入力します。
  - End pointer : 表示範囲の終了ポインタ値を数値で入力します。

### 5.7.3 Trace カラムを非表示にする

[Trace Window]ウィンドウ内の不要なカラムは非表示にすることができます。

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

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

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。

### 5.7.4 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。

[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 5.4 [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。
[Bus]	バス種別を指定します。
[R/W]	アクセスサイクルの種類を指定します。

【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 5.62 [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 5.63 [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。

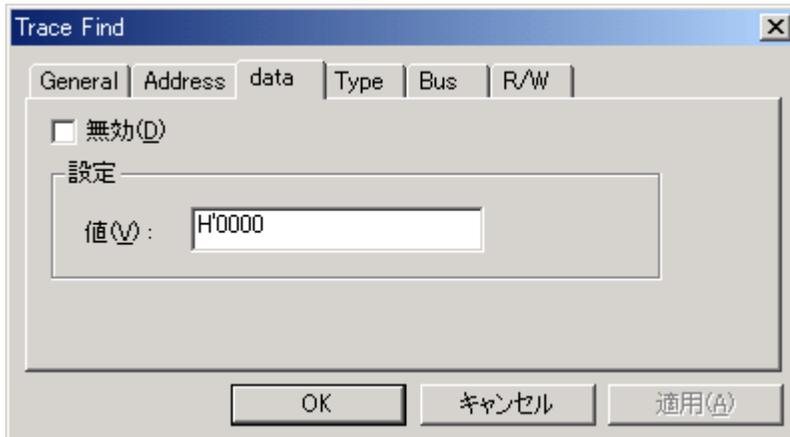


図 5.64 [Trace Find]ダイアログボックス([data]ページ)

[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 5.65 [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

アクセスした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 5.66 [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) [Bus]ページ

バス状態を指定します。

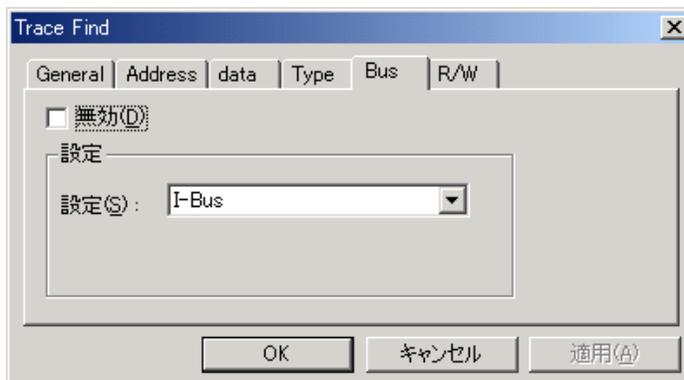


図 5.67 [Trace Find]ダイアログボックス([Bus]ページ)

[無効]

チェックすると、バス条件を検出しません。

[設定]

指定したバス条件を検出します。

[設定]

バス条件を選択します。 ([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。

### 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。[Trace Window]ウィンドウに表示しているトレース情報をテキストファイルとして保存します。保存する範囲を、[PTR]の範囲によって指定することができます(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、[Trace Window]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[Trace Window]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は[ソーストリム]メニューを選択すると取り除いた空白を元に戻します。

### 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。

トレース取得は、自動的に再開します。

### 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。

[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 5.5 [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。
[Bus]	バス種別を指定します。
[R/W]	アクセスサイクルの種類を指定します。

【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[Trace Window]ウイ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 5.68 [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。                              |
| [開始ポイント]        | フィルタを開始する PTR の値を入力します。                    |
| [終了ポイント]        | フィルタを終了する PTR の値を入力します。                    |

【注】 フィルタ範囲入力時、[開始ポイント]オプションにフィルタを終了する PTR の値、[終了ポイント]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。

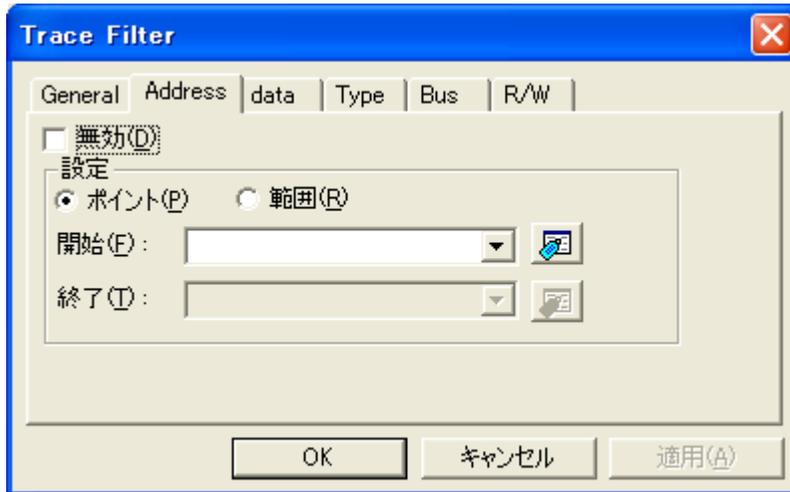


図 5.69 [Trace Filter]ダイアログボックス([Address]ページ)

[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 5.70 [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]	アドレス範囲を指定します。 ([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。

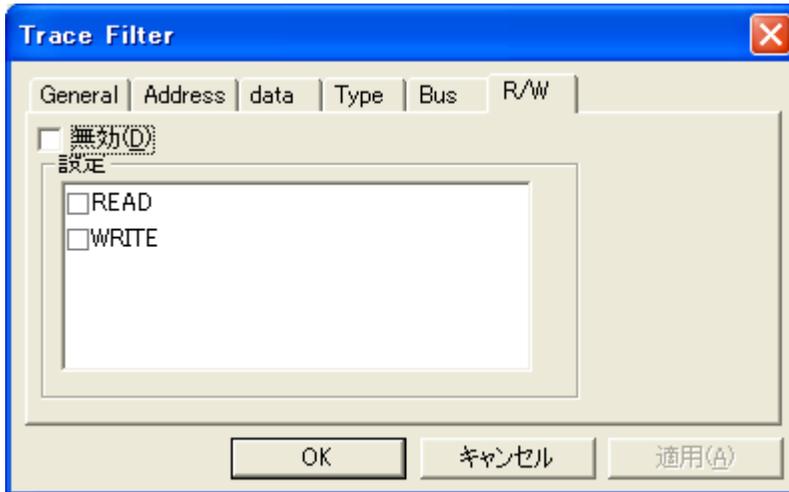


図 5.71 [Trace Filter]ダイアログボックス([R/W]ページ)

[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

アクセスした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 5.72 [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) [Bus]ページ

バス状態を指定します。

タイムスタンプ取得時は無効です。



図 5.73 [Trace Filter]ダイアログボックス([Bus]ページ)

[無効]

チェックすると、バス条件を検出しません。

[設定]

指定したバス条件を検出します。 ([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 5.74 [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削除します。
[閉じる]	ダイアログボックスを閉じます。 このとき、[結果]リストのすべての結果は消去されます。

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

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

【注】 本エミュレータでは[PTR]項目のみ範囲で指定可能です。それ以外の項目は単一の文字列で指定してください。

統計情報の解析における文字列の判定は[Trace Window]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 5.75 [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。

E200F エミュレータでは、オンチップパフォーマンス測定 ( Onchip Performance Analysis )、AUD パフォーマンス測定 ( AUD Performance Analysis )、外部バスパフォーマンス測定 ( BUS Performance Analysis ) の 3 種類の Performance Analysis 機能を持っています。

Performance Analysis 機能の説明は、「1.3.7 パフォーマンス測定機能」を参照してください。

### 5.8.1 [Onchip Performance Analysis]ウィンドウを開く

[Onchip Performance Analysis]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。

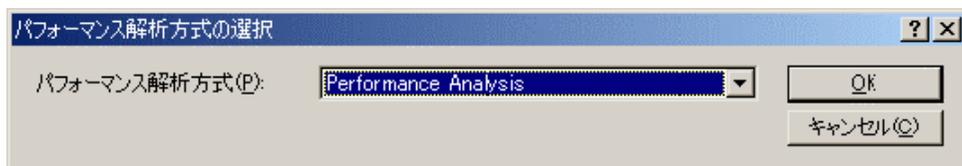


図 5.76 [パフォーマンス解析方式の選択]ダイアログボックス

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

Channel	Condition	Result
Ch1	AC	0002E097
Ch2	BT	00000032
Ch3	DISABLE	00000000
Ch4	DISABLE	00000000

図 5.77 [パフォーマンス解析]ウィンドウ ( Onchip Performance )

Onchip Performance Analysis 機能はマイコン内蔵のパフォーマンス測定機能により測定するため、リアルタイム性は損なわれません。

【注】 オンチップパフォーマンス測定の測定条件、チャネル本数は製品によって異なります。

本ウィンドウで測定チャネルをダブルクリックするか、または測定チャネルを選択しポップアップメニューから[Set...]を選択すると、[Performance Analysis]ダイアログボックスが開き、測定条件を変更することができます。



図 5.78 [Performance Analysis]ダイアログボックス

[Performance Analysis Chx]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。

## 5.8.2 [AUD Performance Analysis]ウィンドウを開く

[AUD Performance Analysis]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。

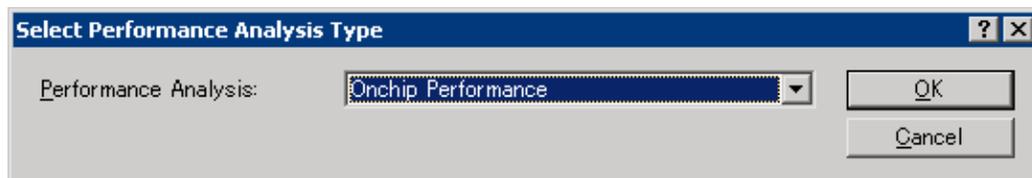
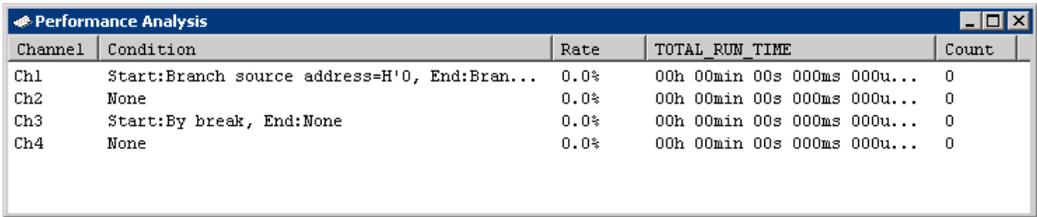


図 5.79 [パフォーマンス解析方式の選択]ダイアログボックス

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



Channel	Condition	Rate	TOTAL RUN TIME	Count
Ch1	Start:Branch source address=H'0, End:Bran...	0.0%	00h 00min 00s 000ms 000u...	0
Ch2	None	0.0%	00h 00min 00s 000ms 000u...	0
Ch3	Start:By break, End:None	0.0%	00h 00min 00s 000ms 000u...	0
Ch4	None	0.0%	00h 00min 00s 000ms 000u...	0

図 5.80 [パフォーマンス解析]ウィンドウ (AUD Performance)

AUD Performance Analysis 機能は E200F エミュレータのメインユニット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。

AUD Performance Analysis の測定チャンネルの開始条件/終了条件は、それぞれ 1 本の AUD イベントチャンネルを使用します。4 本の測定チャンネルは 8 本の AUD イベントチャンネルを使用します。

**【注】** AUD パフォーマンス機能は AUD 端子から出力するチップ内の情報により実現するため、Onchip Eventpoint のトレース取得条件を設定する必要があります。AUD パフォーマンス測定条件に応じて、オンチップイベントチャンネル Ch5 (OA), Ch6 (OA), Ch7 (SystemBus), Ch8 (SystemBus), Ch12 (Branch)または Software Trace のトレース取得条件を設定してください。

AUD パフォーマンス機能の測定開始アドレス通過後、測定停止アドレスを通過する前にブレイク発生した場合、次の再実行時は継続して測定を行いません。

本ウィンドウで測定チャンネルをダブルクリックするか、または測定チャンネルを選択しポップアップメニューから[Set...]を選択すると、[Performance Analysis Aud Channel x]ダイアログボックスが開き、測定条件を変更することができます。

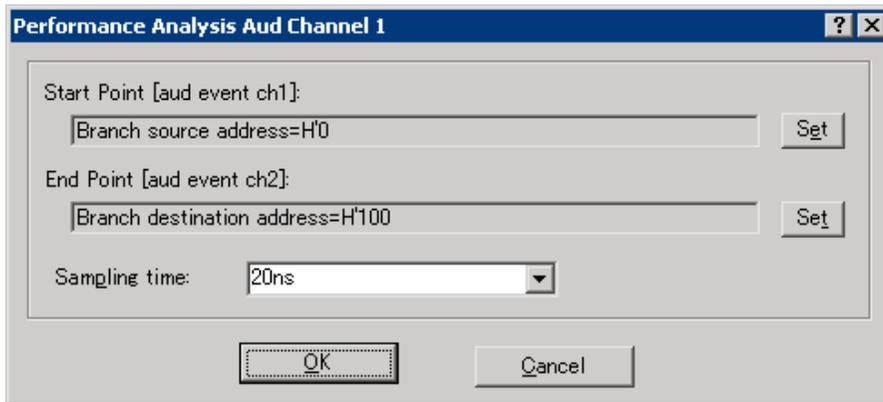


図 5.81 [Performance Analysis Aud Channel x]ダイアログボックス

[Start Point [aud event chx]]	測定チャネルの開始ポイント条件を表示します。
[End Point [aud event chx]]	測定チャネルの終了ポイント条件を表示します。
[Set]	開始ポイントまたは終了ポイントに対応する、AUD イベントチャネルの設定用ダイアログボックスを表示します。
[Sampling time]	測定用タイムの分解能を選択します。 以下のいずれかを選択できます。 20ns, 40ns, 100ns, 400ns

[Set]ボタンをクリックすると、対応する AUD イベントチャネルの設定用ダイアログボックスが開き、測定開始/終了条件を変更できます。

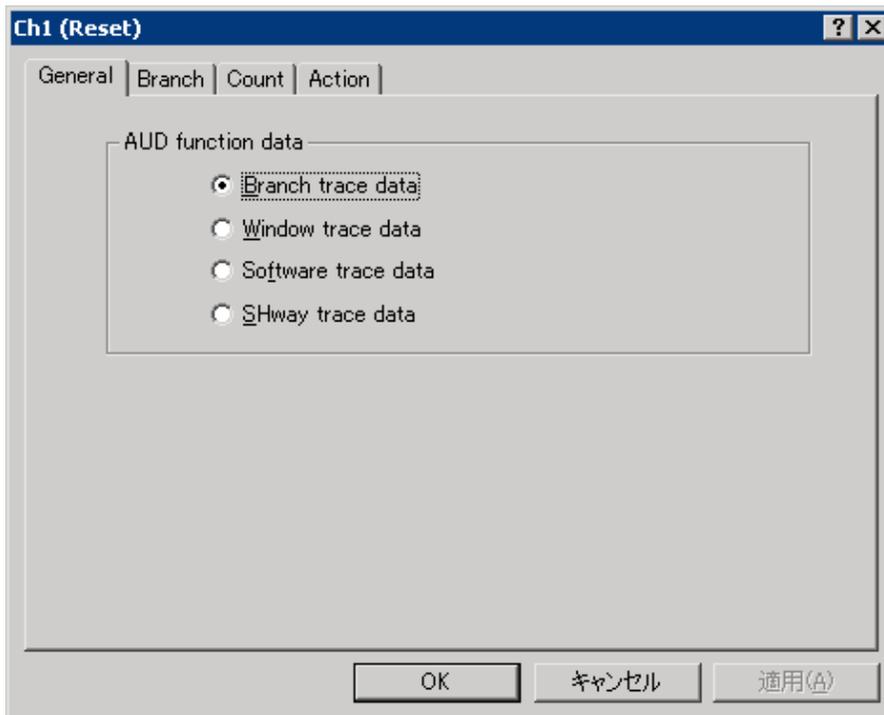


図 5.82 AUD Performance Analysis 測定条件編集 (AUD イベント設定用ダイアログボックス)

- 【注】
1. AUD Performance Analysis 機能を使用するとき、各チャンネルの[Action]ページで[Performance start/stop]を指定してください。
  2. 測定条件を変更すると、変更内容は[Event]ウィンドウの[AUD Event]シートにも反映します。

### 5.8.3 [BUS Performance Analysis]ウィンドウを開く

[BUS Performance Analysis]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。

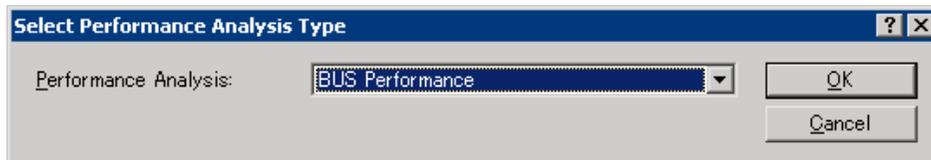


図 5.83 [パフォーマンス解析方式の選択]ダイアログボックス

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

Channel	Condition	Rate	TOTAL_RUN_TIME	Count
Ch1	RtoR all address=H'0 to H'100 count D'6 read	0.0%	00h 00min 00s 000ms 000us 000ns	0
Ch2	address=H'0 to H'300 read			
Ch3	PtoP all address=H'0 to H'0	0.0%	00h 00min 00s 000ms 000us 000ns	0
Ch4	PtoP all address=H'2000 to H'3000 write	0.0%	00h 00min 00s 000ms 000us 000ns	0
Ch5	None	0.0%	00h 00min 00s 000ms 000us 000ns	0
Ch6	None	0.0%	00h 00min 00s 000ms 000us 000ns	0

図 5.84 [パフォーマンス解析]ウィンドウ (BUS Performance)

BUS Performance Analysis 機能は E200F エミュレータのトレースユニット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。

測定用途に応じて 3 つの測定モードを選択できます。

[外部バスパフォーマンス測定モード]

測定モード	測定内容	測定用途
指定範囲内測定 (Range)	指定した範囲内の実行時間および実行回数を測定します。	関数の処理時間のうち関数内から呼び出す子関数の処理時間を除いた処理時間を測定する場合などに使用します。
指定アドレス間測定 (PtoP)	指定したアドレス間の実行時間および実行回数を測定します。	関数の処理時間を測定する場合などに使用します。
指定範囲間測定 (RtoR)	指定範囲から別の指定範囲までの実行時間および実行回数を測定します。	アセンブリプログラムなど、サブルーチンを連続して配置したプログラムにおいて、連続するサブルーチンのうちのいずれかが呼ばれてから別の連続するサブルーチンのいずれかが呼ばれるまでの実行時間を測定する場合などに使用します。

測定チャンネルは最大 8 本まで使用できます。

ただし、指定範囲間測定モードを選択したとき、開始条件と終了条件はそれぞれ 1 本の測定チャンネルを使用するため、1 つの測定項目につき 2 本のチャンネルを使用します。

- 【注】
1. トレースユニットを E200F エミュレータに接続していない場合、外部バス機能はサポートしません。
  2. E200F エミュレータのトレースユニットの機能を入れ替えることにより、測定チャンネルの数が変わります。詳細につきましては、「5.1.4 [Bus Board]ページ」を参照してください。

本ウィンドウで測定チャンネルをダブルクリックするか、または測定チャンネルを選択しポップアップメニューから[Set...]を選択すると、[Performance Analysis Bus Channel x]ダイアログボックスが開き、測定条件を変更することができます。

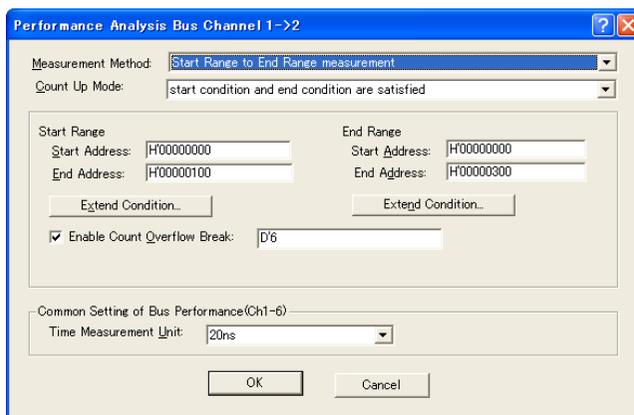


図 5.85 [Performance Analysis Bus Channel x]ダイアログボックス

[Measurement Method]	測定モードを指定します。
	Specified Range measurement : 指定範囲内の測定。
	Start Point to End Point measurement : 指定アドレス間の測定。
	Start Range to End Range measurement : 指定範囲間の測定。
[Count Up Mode]	実行回数のカウントアップ方法を指定します。
	start condition and end condition are satisfied : 開始条件が成立した後に終了条件が成立したときカウントアップします。
	end condition is satisfied : 終了条件が成立する度に、カウントアップします。
[Start Range]	測定条件または、指定範囲間測定モード時の測定開始条件を設定します。
	Start Address : 開始アドレスを入力します。
	End Address : 終了アドレスを入力します。
	Extend Condition : 拡張条件を設定します。
[End Range]	指定範囲間測定モード時の測定終了条件を設定します。
	Start Address : 開始アドレスを入力します。
	End Address : 終了アドレスを入力します。
	Extend Condition : 拡張条件を設定します。
[Enable Count Overflow Break]	チェックすると、実行回数が設定した回数を超えるとブレークします。
[Time measurement Unit]	測定用タイマの分解能を指定します。 以下のいずれかを選択できます。 20ns, 50ns, 400ns

## 5 デバッグ

---

- 【注】
1. [End Range]項目は、[Start Range to End Range measurement]を選択したときのみ表示します。
  2. [Enable Count Overflow Break]は測定チャンネル 1 を選択したときのみ表示します。

[Extend Condition...]ボタンをクリックすると、拡張条件設定用ダイアログボックスが開きます。バスステート条件を設定できます。測定条件として範囲を選択している場合は、リードサイクル固定となります。

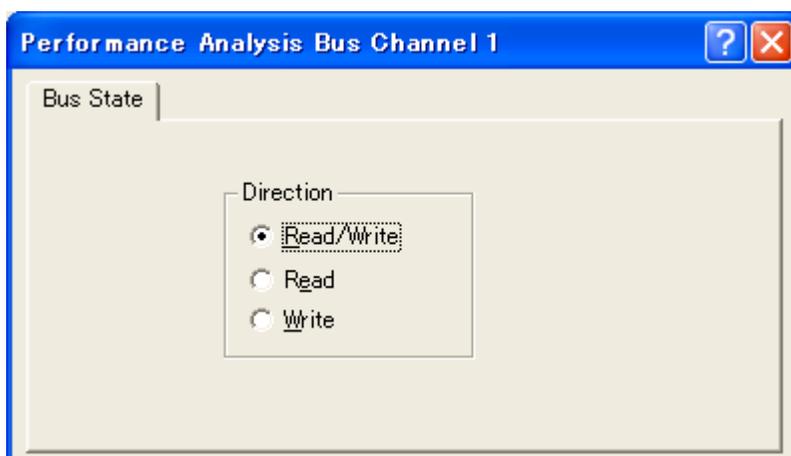


図 5.86 BUS Performance Analysis 拡張条件設定用ダイアログボックス

### 5.8.4 カラムを非表示にする

[Performance Analysis]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。

### 5.8.5 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。

### 5.8.6 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。

### 5.8.7 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。

## 5.9 プロファイル情報を見る

プロファイル機能は、アプリケーションプログラムの実行パフォーマンスを関数単位に測定します。アプリケーションプログラム中の性能劣化の原因となっている場所および要因を調査することができます。

High-performance Embedded Workshop はプロファイルデータの参照方法、参照目的に応じて、3つのウィンドウでプロファイル測定結果を表示します。

### 5.9.1 スタック情報ファイル

プロファイル機能は、最適化リンカ (Ver.7.0 以降) が出力するスタック情報ファイル (拡張子 ".SNI") を読み込むことができます。このファイルには、ソースファイル上の (静的な) 関数呼び出し関係の情報が入っています。

High-performance Embedded Workshop がスタック情報ファイルを読み込むことで、ユーザアプリケーションが未実行 (プロファイルデータの測定を行う前) でも、関数の呼び出し関係を表示できるようになります。(ただし、[プロファイル] ウィンドウのポップアップメニューで [表示設定->未実行関数を表示しない] をチェックしている場合を除きます。)

High-performance Embedded Workshop がスタック情報ファイルを読み込まない場合、プロファイル機能で表示するデータは、プロファイルデータ測定中に実行した関数についてのみに なります。

リンカでスタック情報ファイルを生成するには、[ビルド-> SuperH Risc engine Standard Toolchain...] を選択して、[Standard Toolchain] ダイアログボックスを開き、[最適化リンカ] タブで [カテゴリ] リストボックスを [その他] に指定し、[スタック情報ファイル(sni)出力] チェックボックスをチェックしてください。

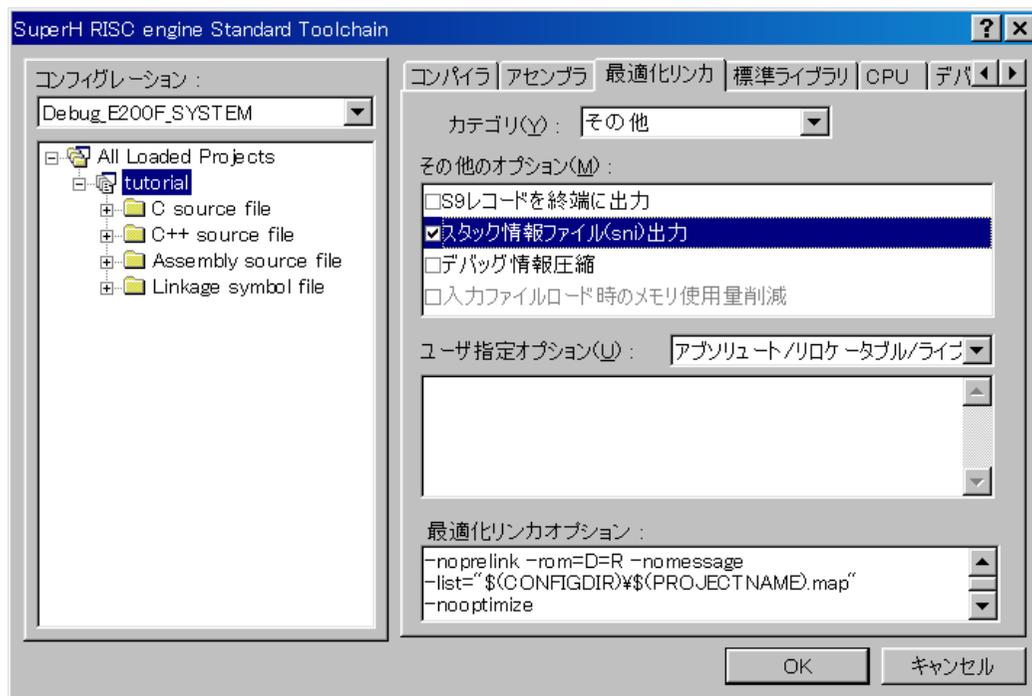


図 5.87 [Standard Toolchain]ダイアログボックス(1)

## 5.9.2 プロファイル情報ファイル

プロファイル情報ファイルを作成するためには、アプリケーションプログラムのプロファイルデータを測定後に、[プロファイル]ウィンドウの Pop-up メニューで[プロファイル情報の保存...]メニューオプションを選択し、ファイル名を指定します。

プロファイル情報ファイルには、関数の呼び出し回数とグローバル変数のアクセス回数の情報が入っています。最適化リンカ(Ver.7.0以降)は、プロファイル情報ファイルを読み込み、関数および変数の配置を実際のプロگرام動作状況に合わせた配置に最適化する機能を持っています。

プロファイル情報ファイルをリンカに入力するには、[Standard Toolchain]ダイアログの[最適化リンカ]タブで[カテゴリ]リストボックスを[最適化]に指定し、[プロファイル情報]チェックボックスをチェックして、プロファイル情報ファイル名を指定してください。



図 5.88 [Standard Toolchain]ダイアログボックス(2)

なお、[プロファイル情報]チェックボックスを有効にするには、[設定]リストボックスを[最適化なし]以外に設定する必要があります。

## 5.9.3 スタック情報ファイルのロード

スタック情報ファイルを読み込むかどうかは、ロードモジュールロード時に表示する、確認のメッセージボックスで指定できます。メッセージボックスの[OK]ボタンをクリックするとスタック情報ファイルをロードします。

確認のメッセージボックスは、次の場合に表示します。

- スタック情報ファイルが存在するとき
- [オプション]ダイアログボックス(メインメニューの[ツール->オプション...]を選択すると開きます)の[確認]タブ(図5.89)で[スタック情報ファイルをロードします (SNI ファイル)]チェックボックスをチェックしている場合

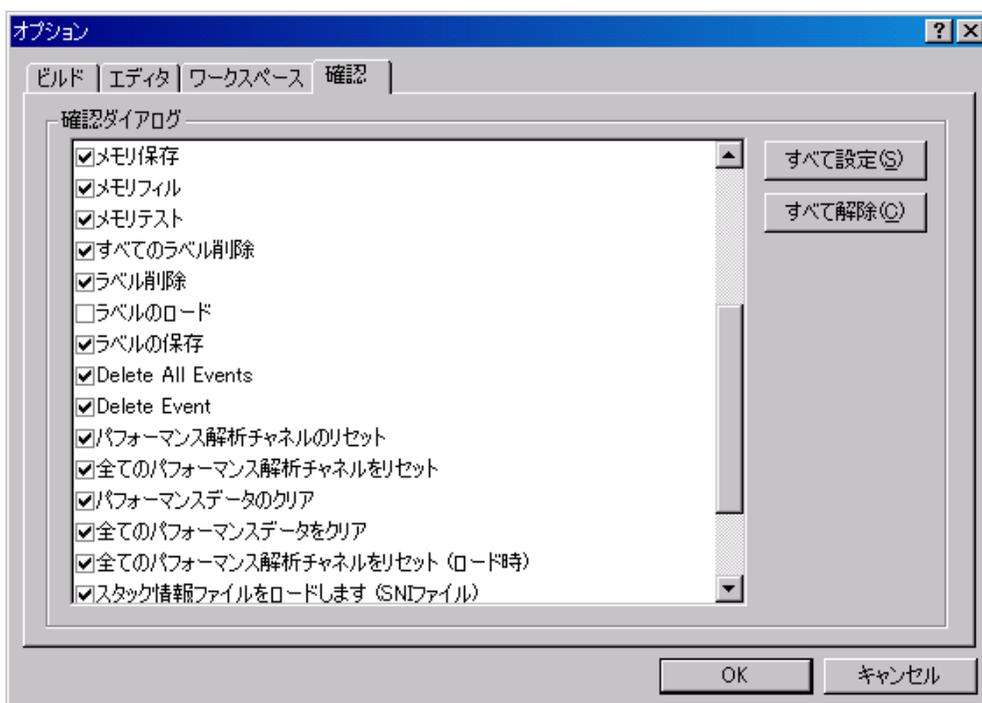


図 5.89 [オプション]ダイアログボックス

#### 5.9.4 プロファイルを有効にする

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

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

#### 5.9.5 測定方法を指定する

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

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

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

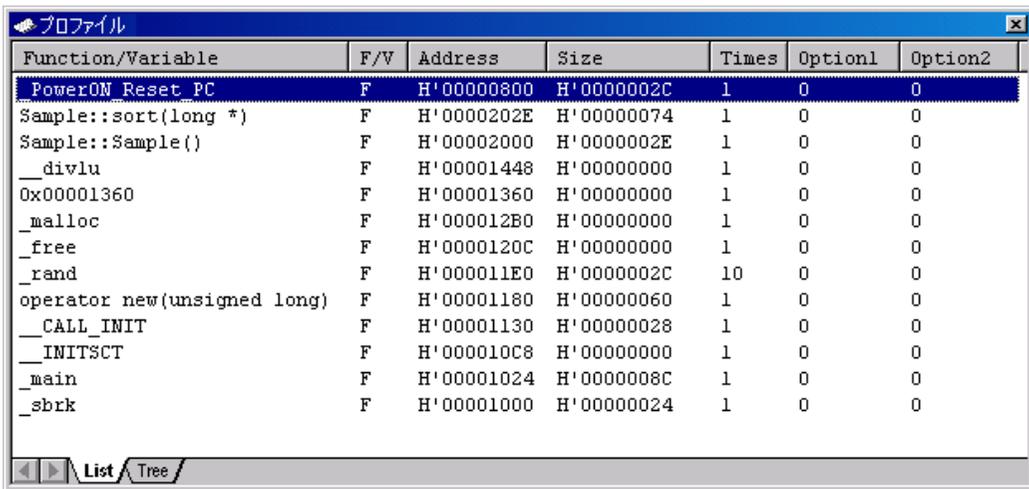
#### 5.9.6 ユーザプログラムを実行し結果を確認する

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

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

## 5.9.7 [List]シート

関数とグローバル変数をリスト表示し、各関数/変数のプロファイルデータを表示します。



Function/Variable	F/V	Address	Size	Times	Option1	Option2
PowerON_Reset_PC	F	H'00000800	H'0000002C	1	0	0
Sample::sort(long *)	F	H'0000202E	H'00000074	1	0	0
Sample::Sample()	F	H'00002000	H'0000002E	1	0	0
_divlu	F	H'00001448	H'00000000	1	0	0
0x00001360	F	H'00001360	H'00000000	1	0	0
_malloc	F	H'000012B0	H'00000000	1	0	0
_free	F	H'0000120C	H'00000000	1	0	0
_rand	F	H'000011E0	H'0000002C	10	0	0
operator new(unsigned long)	F	H'00001180	H'00000060	1	0	0
__CALL_INIT	F	H'00001130	H'00000028	1	0	0
__INITSCT	F	H'000010C8	H'00000000	1	0	0
_main	F	H'00001024	H'0000008C	1	0	0
_sbrk	F	H'00001000	H'00000024	1	0	0

図 5.90 [List]シート

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

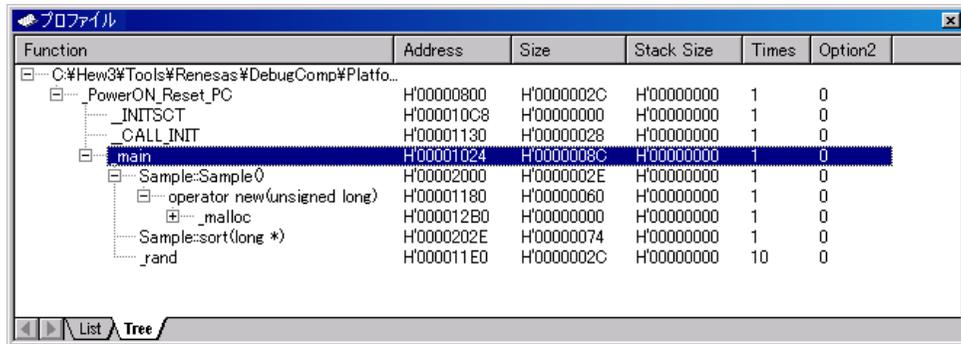
[Function/Variable]列または[Address]列をダブルクリックすると、該当するアドレスに対応したソースプログラムを表示します。

ウィンドウ内でマウスの右ボタンをクリックするとポップアップメニューを表示します。このポップアップメニューは、「5.9.8 [Tree]シート」を参照してください。

## 5.9.8 [Tree]シート

関数の呼び出し関係を表示し、各呼び出し位置におけるプロファイルデータを表示します。

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



Function	Address	Size	Stack Size	Times	Option2
PowerON_Reset_PC	H'00000800	H'0000002C	H'00000000	1	0
_INIT\$CT	H'000010C8	H'00000000	H'00000000	1	0
_CALL_INIT	H'00001130	H'00000028	H'00000000	1	0
main	H'00001024	H'0000008C	H'00000000	1	0
Sample:Sample0	H'00002000	H'0000002E	H'00000000	1	0
operator new(unsigned long)	H'00001180	H'00000060	H'00000000	1	0
_malloc	H'000012B0	H'00000000	H'00000000	1	0
Sample:sort(long *)	H'0000202E	H'00000074	H'00000000	1	0
_rand	H'000011E0	H'0000002C	H'00000000	10	0

図 5.91 [Tree]シート

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

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

### (1) ソースファイル表示

選択している行の該当アドレスに対応したソースプログラムまたは逆アセンブルを表示します。

### (2) チャート表示

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

### (3) 有効

プロファイルデータ収集のオン・オフを切り替えます。プロファイルデータ測定がONのとき、メニューテキストの左にチェックマークを表示します。

### (4) 関数呼び出しをトレースしない

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

[プロファイル]ウィンドウの[Tree]シートで関数呼び出し関係を表示するためには、本メニューをチェックせずにプロファイルデータを測定してください。また、測定結果のプロファイル情報ファイルを使用して、最適化リソースエディタによる最適化を行う場合も、本メニューをチェックしないでください。

### (5) 検索...

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

### (6) データ検索...

[データ検索]ダイアログボックスを表示します。



図 5.92 [データ検索]ダイアログボックス

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

### (7) データクリア

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

### (8) プロファイル情報の保存...

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

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

### (9) テキスト形式で保存...

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

## (10) 表示設定

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

## (a) 関数と変数を表示

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

## (b) 関数を表示

[Function/Variable]列で、関数のみを表示します。

## (c) 変数を表示

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

## (d) 未実行関数を表示しない

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

## (e) 子関数の実行結果を含んで表示

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

## (11) プロパティ...

測定項目の設定を行います。

## 5.9.9 プロファイル - チャートウィンドウ

[プロファイル - チャート]ウィンドウは、特定の関数に着目した関数の呼び出し関係を表示します。本ウィンドウは、着目する関数を中心に表示し、その左側には着目した関数を呼び出した関数、右側には、着目している関数が呼び出した関数を、それぞれ表示します。また、各呼び出しを行った回数も表示します。

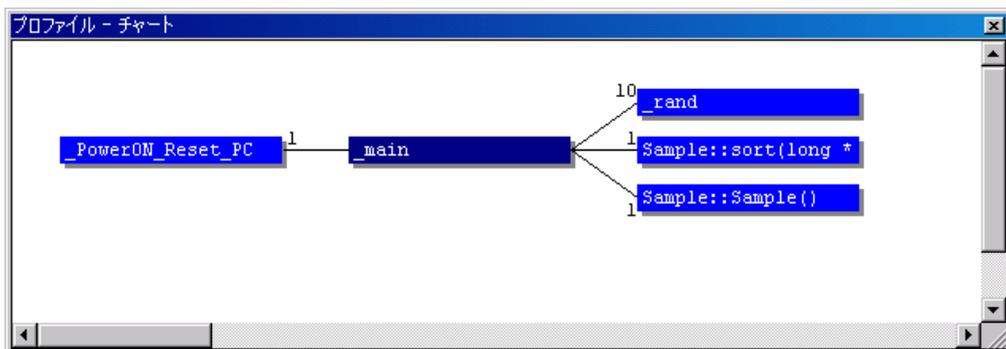


図 5.93 [プロファイル - チャート]ウィンドウ

### 5.9.10 表示データの種類および用途

プロファイル機能から下記情報を得ることができます。

#### (1) Address

関数を配置しているメモリ上の位置を知ることができます。アドレス順にソート表示することにより、メモリ上の配置イメージで関数とグローバル変数を並べることができます。

#### (2) Size

サイズ順にソート表示すれば、サイズが小さくて頻繁に呼び出している関数を見つけることができます。そのような関数があれば inline 関数にすることで、関数呼び出しのオーバーヘッドを減らせる場合があります。

また、サイズの大きい関数を実行すると、更新するキャッシュのサイズが大きくなります。このような、キャッシュミスの原因となり得る関数を頻繁に呼び出していないかを容易に確認できます。

#### (3) Stack Size

関数呼び出しのネストが深い場合、関数呼び出し経路をたどり、その経路上の全関数のスタックサイズを合計することで、おおよそのスタック使用量を見積もれます。

#### (4) Times

呼び出し（アクセス）回数順にソート表示すれば、頻繁に呼び出している関数や頻繁にアクセスしている変数を容易に調べることができます。

#### (5) プロファイルデータ

CPU により、さまざまなデータを測定できます。各 CPU で測定するデータは、パフォーマンス測定機能で測定できる項目と同じです。詳細はオンラインヘルプを参照してください。

### 5.9.11 プロファイル情報ファイルを作成する

プロファイル情報ファイルを作成する場合は、ポップアップメニューの[プロファイル情報の保存...]メニューオプションを選択します。[プロファイル情報の保存]ダイアログボックスを表示します。ファイル名を選択して[保存]ボタンを押すと、選択したファイルにプロファイル情報を書き込みます。[全て保存]ボタンを押すと、全てのファイルにプロファイル情報を書き込みます。

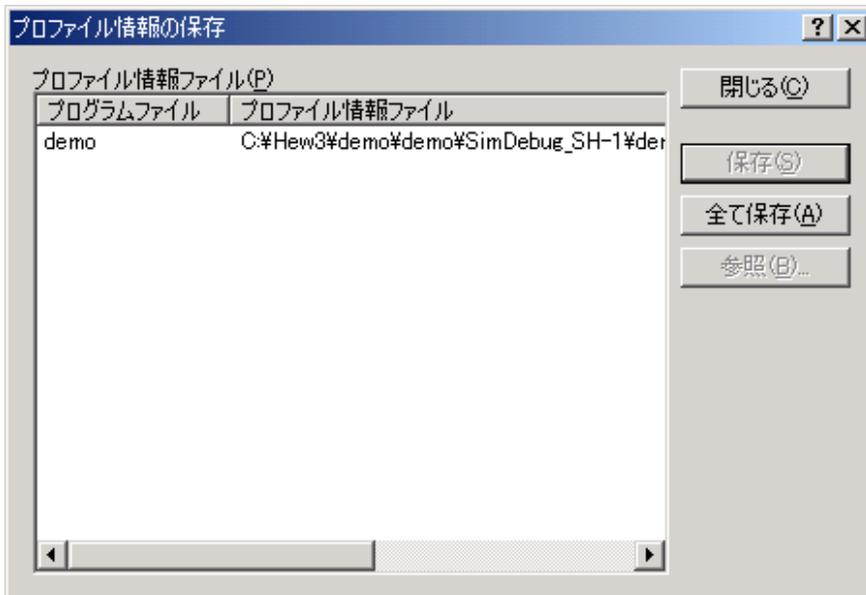


図 5.94 [プロファイル情報の保存]ダイアログボックス

### 5.9.12 注意事項

#### (1) 誤差について

プロファイル機能は、ユーザプログラム実行を内部的にブレークし、測定データを収集して再度ユーザプログラムを実行しています。

そのため、測定項目がブレーク時と再実行時に発生した場合もカウントします。

したがって、プロファイル測定値には誤差を含むことになります。

本機能の測定値はあくまでも目安と考えてください。

#### (2) プロファイル機能使用中に使用できなくなる機能について

##### 1. オンチップパフォーマンス測定機能

プロファイル機能イネーブル時には、オンチップパフォーマンス測定機能は使用できません。

##### 2. STEP機能

プロファイル機能イネーブル時、STEP機能は使用しないでください。プロファイルデータを正常に測定できません。

##### 3. 内蔵トレース機能

プロファイル機能イネーブル時、内部的に内蔵トレースのモード選択を全項目選択状態とするため、内蔵トレースのモード選択は無効になります。

内蔵トレースは、プロファイル機能イネーブル時には使用しないでください。

##### 4. Halt機能

内蔵トレース、AUDトレース、外部バストレース、MFIトレースのHalt機能は、プロファイル機能イネーブル時には使用しないでください。

##### 5. ユーザプログラム実行中のメモリアクセス

プロファイル機能イネーブル時、ユーザプログラム実行中のメモリアクセスは禁止です。

##### 6. プロファイル機能を使用する場合は、分岐命令発生時にブレークが発生します。このため、リアルタイム性がなくなります。また、ブレーク発生時にE200Fエミュレータ用ファームウェアに制御が戻るため、E200Fエミュレータ用ファームウェアからユーザプログラムへ復帰したときの分岐命令の実行が[Trace Window]ウィンドウに表示されることがあります。この場合の表示は、\*\*EML\*\*と表示します。

##### 7. プロファイル機能を使用している場合は、Break Condition 3の機能は使用できません。

##### 8. プロファイル機能使用時は、[Configuration]ダイアログボックス[Emulation mode]でHardware break enableを設定しないでください。

9. プロファイル機能使用時は、PC、SRの値のステータスバー表示は不定値となります。

(3) その他の注意事項

1. プロファイル機能を使用すると、ユーザプログラム実行を内部的にブレークします。そのため、AUD Performance Analysis, BUS Performance Analysis, Realtime Profileの測定結果には誤差を含むことになります。
2. プロファイル機能を使用すると、前にパフォーマンス測定機能にて設定していた内容や測定したデータは消去されます。
3. プロファイル機能は、内部的にブレークを発生させて実現しています。そのため、ユーザプログラム実行開始から終了までに要する時間が長くなります。ユーザプログラムの停止時間は、ご使用のホストマシンの性能やJTAGクロックに依存します。

参考値として、以下の環境でのユーザプログラム実行時間を示します。

環境

ホストマシン : Pentium® III 850MHz  
メモリ : 256Mbyte  
SH7323 : TCKクロック10MHz  
OS : Windows® 2000  
実行プログラム : ネストコール10,000回

- (i) プロファイル機能を使用しない場合 : 1秒未満
  - (ii) プロファイル機能を、子関数を含まない設定で使用した場合 : 63秒
  - (iii) プロファイル機能を、子関数を含む設定で使用した場合 : 72秒
4. プロファイル機能を使用すると、トレース、設定を内部的に変更し、[Internal trace]モードにします。[AUD trace]モードに変更しないと、E200Fエミュレータの機能のうち一部が使用できなくなりますので、ご注意ください。

### 5.10 リアルタイムプロファイル情報を見る

リアルタイムプロファイル機能は、アプリケーションプログラムの指定範囲内の実行パフォーマンスを関数単位に測定します。アプリケーションプログラム中の性能劣化の原因となっている場所および要因を調査することができます。

リアルタイムプロファイル機能は、E200F エミュレータのメインユニットと拡張プロファイルユニットのプロファイラ測定回路によりパフォーマンス測定するため、リアルタイム性は損なわれません。

E200F エミュレータのリアルタイムプロファイル測定モードには、以下 2 モードあります。

- Functionモード  
関数実行時間の累積表示に、サブルーチン実行時間を含まない機能です。
- Nestモード  
関数実行時間の累積表示に、サブルーチン実行時間を含む機能です。

E200F エミュレータ起動時に表示される、[Function select]ダイアログボックスで、どちらのモードを使用するかを決定してください。

【注】 本機能には、以下の制限事項があります。

(1) リアルタイムプロファイル機能全般に対する制限事項

(a) 測定対象領域について

E200F エミュレータでは、512KB を一単位とし、最大 24 ブロックのエリアの関数全てについてのプロファイル情報を取得できます。

E200F エミュレータのハードウェアで、8 ブロック分の測定用メモリを最大 3 つ持っており、そのメモリを用いてリアルタイムプロファイル機能を実現しています。

したがって、以下の制限事項がありますので、ご注意ください。

- それぞれのブロックは、互いに隣接したアドレス領域を設定することができますが、8ブロック境界にまたがるアドレス範囲となる関数は設定できません。設定した場合、警告メッセージを表示します。その場合、正しい測定はできません。

(b) インライン展開

コンパイラの最適化によりインライン展開となった関数については、[リアルタイムプロファイル]ウィンドウに表示されません。

(c) 再帰関数

再帰関数の実行時間は正しく測定できますが、実行回数は 1 回となります。

## (d) AUD トレースについて

本機能は、AUD トレースで出力されるデータを使用しています。

したがって、リアルタイムトレースモードで使用された場合、トレースデータを消失することがあり、この場合は正常に測定できません。

このような場合には、ノンリアルタイムトレースモードで使用されることを推奨します。

## (e) オーバレイプログラムについて

オーバレイにより、入れ替えが発生するプログラムについては測定できません。

## (2) Function モード使用時の制限事項

## (a) テイルコール

関数呼び出しが以下の図のようなテイルコールの場合、呼び出された関数からの戻り値は、呼び出し元の関数の戻りアドレスになります。この場合、呼び出された関数の実行時間、実行回数は正しく測定できません。

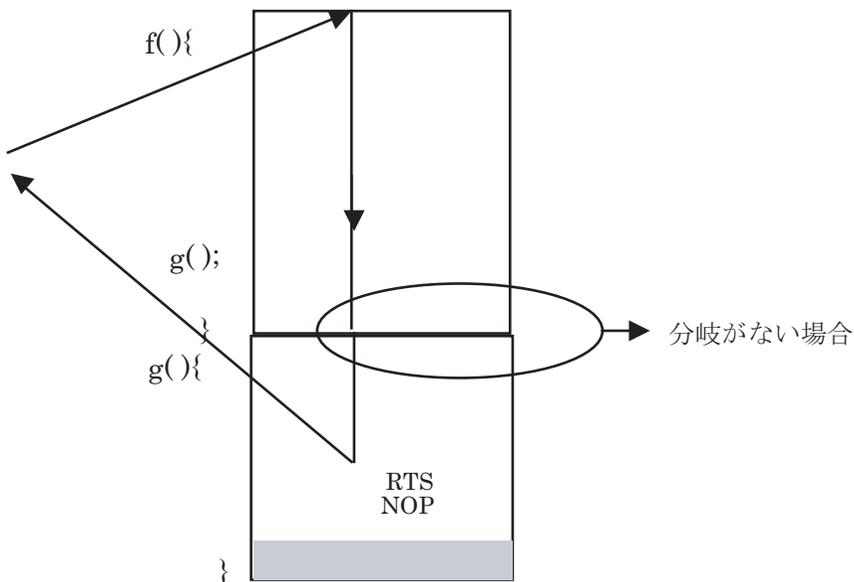


図 5.95 テイルコール (Function モード)

(b) 測定範囲内の Go 実行開始アドレス、ブレークアドレスと、測定可能範囲の関係

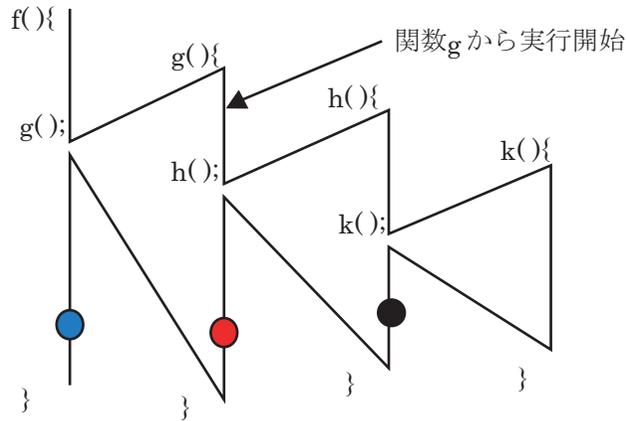


図 5.96 測定可能範囲 (Function モード)

黒丸の箇所でブレークした場合の測定可能範囲：関数h,kの実行時間、実行回数  
 赤丸の箇所でブレークした場合の測定可能範囲：関数h,kの実行時間、実行回数  
 青丸の箇所でブレークした場合の測定可能範囲：関数h,kの実行時間、実行回数  
 関数gの実行時間、回数は測定できません。

上記のようになっております。

実行開始した関数内でブレークすることを推奨します。

上位関数に戻った場合にも、実行を開始した関数の実行回数は測定できません。

(3) Nest モード使用時の制限事項

(a) テイルコール

5. 関数呼び出しが以下の図のようなテイルコールの場合、呼び出された関数からの戻り値は、呼び出し元の関数の戻りアドレスになります。この場合、呼び出された関数の実行時間は正しく測定できません。実行回数は正しく測定できます。

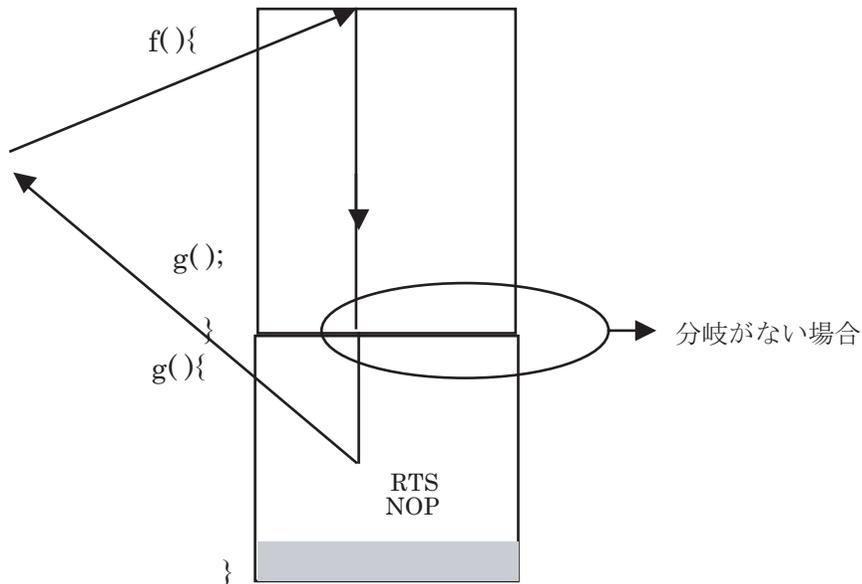


図 5.97 テイルコール (Nest モード)

6. テイルコールで呼び出された関数から、別の関数をテイルコールにて呼び出された場合において制限があります。連続してテイルコールが発生するような場合、3段階まで正しく測定できます。

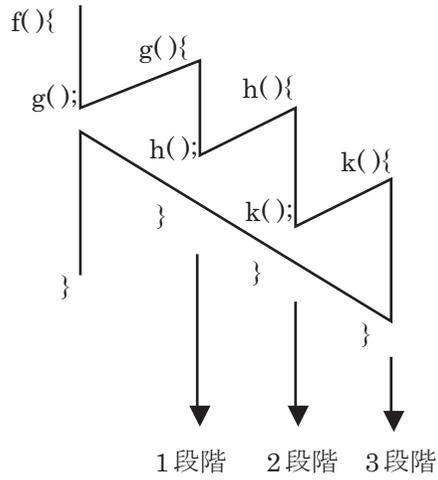


図 5.98 呼び出し制限

## (b) 関数のネストについて

測定対象内にある最上位関数からの関数コールが 32 段階を超えた場合、正常に測定できなくなります。  
この場合、警告メッセージを表示します。

## (c) 測定対象外である関数からの呼び出しについて

測定対象外にある関数から、測定対象内にある関数を呼び出された場合、呼び出された関数が呼び出し元の関数に正しく戻れない場合については、正しい測定ができません。

また、正しく戻った場合でも、戻り先アドレスから 3 命令以内に別の関数をコールした場合は正しく測定ができません。

測定対象領域外関数からユーザプログラムを実行し、測定対象領域内関数に遷移します。その後、測定対象領域外の別の関数に分岐します。

## (d) 測定範囲内の Go 実行開始アドレス、ブレークアドレスと、測定可能範囲の関係

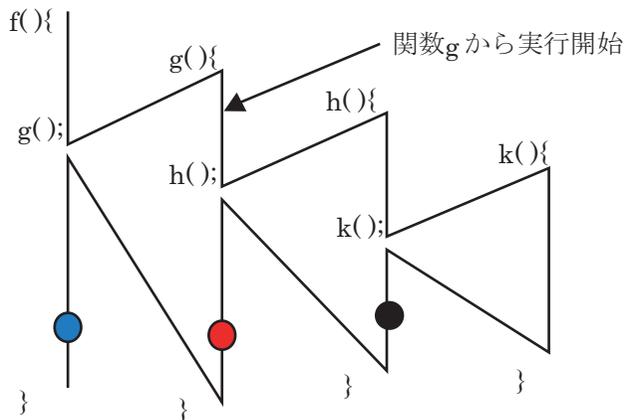


図 5.99 測定可能範囲 (Nest モード)

黒丸の箇所でブレークした場合の測定可能範囲：関数kの実行時間、実行回数。

赤丸の箇所でブレークした場合の測定可能範囲：関数h,kの実行時間、実行回数。

青丸の箇所でブレークした場合の測定可能範囲：関数h,kの実行時間、実行回数。

上記のようになっております。

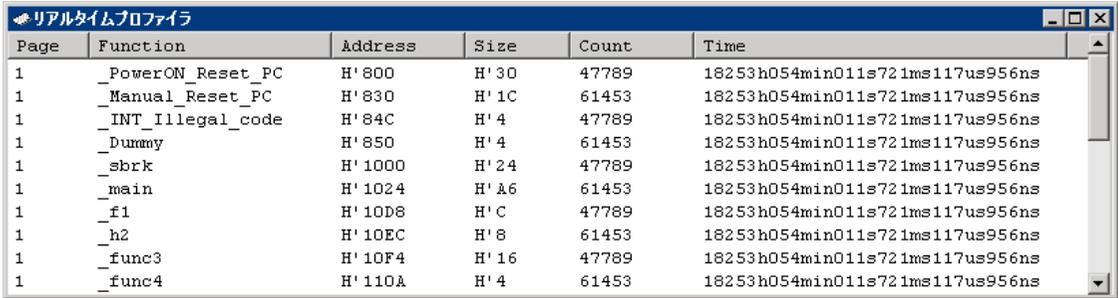
実行開始した関数内でブレークすることを推奨します。

上位関数に戻った場合にも、実行を開始した関数の実行回数は測定できません。

## 5.10.1 [リアルタイムプロファイル]ウィンドウを開く

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

カラムヘッダをクリックすることにより、表示内容をその項目の昇順、降順で並べ替えることができます。



Page	Function	Address	Size	Count	Time
1	_PowerON_Reset_PC	H'800	H'30	47789	18253h054min011s721ms117us956ns
1	_Manual_Reset_PC	H'830	H'1C	61453	18253h054min011s721ms117us956ns
1	_INT_Illegal_code	H'84C	H'4	47789	18253h054min011s721ms117us956ns
1	_Dummy	H'850	H'4	61453	18253h054min011s721ms117us956ns
1	_sbrk	H'1000	H'24	47789	18253h054min011s721ms117us956ns
1	_main	H'1024	H'A6	61453	18253h054min011s721ms117us956ns
1	_f1	H'10D8	H'C	47789	18253h054min011s721ms117us956ns
1	_h2	H'10EC	H'8	61453	18253h054min011s721ms117us956ns
1	_func3	H'10F4	H'16	47789	18253h054min011s721ms117us956ns
1	_func4	H'110A	H'4	61453	18253h054min011s721ms117us956ns

図 5.100 [リアルタイムプロファイル]ウィンドウ

リアルタイムプロファイル機能から下記情報を得ることができます。

## (1) Address

関数を配置しているメモリ上の位置を知ることができます。

## (2) Size

サイズ順にソート表示すれば、サイズが小さくて頻繁に呼び出している関数を見つけることができます。そのような関数があれば inline 関数にすることで、関数呼び出しのオーバーヘッドを減らせる場合があります。

また、サイズの大きい関数を実行すると、更新するキャッシュのサイズが大きくなります。このような、キャッシュミスの原因となり得る関数を頻繁に呼び出していないかを容易に確認できます。

## (3) Count

関数をコールした回数を表示します。

## (4) Time

実行時間の累積を表示します。

### 5.10.2 測定範囲を指定する

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。ポップアップメニューから[測定範囲を追加]を選択すると[EDIT]ダイアログボックスが開き、リアルタイムプロファイルの測定範囲を指定できます。

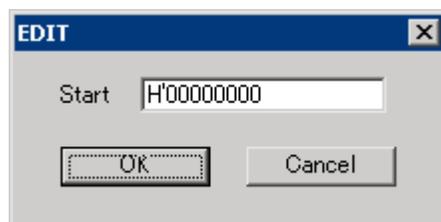


図 5.101 [EDIT]ダイアログボックス

E200F エミュレータに拡張プロファイルユニットが接続している場合、合計 6M バイト範囲内のプロファイル情報を測定できます。拡張プロファイルユニットを接続していない場合、合計 2M バイト範囲内のプロファイル情報を測定できます。

E200F エミュレータでは、512KB を一単位とし、8 ブロックのエリアの関数全てについてのプロファイル情報を取得できます。

各ブロックに指定するアドレスは、隣接している必要はありません。

拡張プロファイルユニットを接続している場合は、512KB のブロックが 16 ブロック加わり、合計 24 ブロック分のエリアを測定できます。

各ブロックの測定データは、リアルタイムプロファイルウィンドウの各ページに表示されます。

表示するページを切り替える場合、リアルタイムプロファイルウィンドウのポップアップメニューから[ページを選択]を選択することによって開く[Select page]ダイアログボックスで切り替えたいページを選択して[OK]ボタンをクリックします。

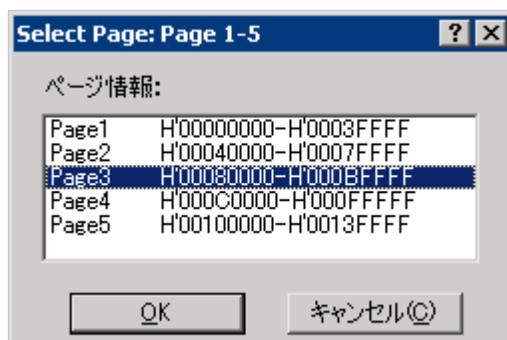


図 5.102 [Select page]ダイアログボックス

### 5.10.3 測定を開始する

ユーザプログラムを実行すると測定を開始します。

ユーザプログラムを停止すると、測定結果を[リアルタイムプロファイル]ウィンドウに表示します。

### 5.10.4 測定結果をクリアする

ポップアップメニューから[全てのデータをクリア]を選択すると、[Count]カラム、[Time]カラムの測定結果をクリアします。

### 5.10.5 測定範囲を削除する

ポップアップメニューから[削除]を選択すると、指定している測定範囲をすべて削除し、測定結果をクリアします。

### 5.10.6 測定時間の最小単位を設定する

E200F エミュレータでは、測定時間の最小単位を 20ns、40ns、100ns、400ns に変更できます。

20ns を選択した場合、最長 3 時間まで測定できます。

40ns を選択した場合、最長 6 時間まで測定できます。

100ns を選択した場合、最長 15 時間まで測定できます。

400ns を選択した場合、最長 61 時間まで測定できます。

変更する場合、リアルタイムプロファイルウィンドウのポップアップメニューから[設定]を選択し、[Properties]ダイアログボックスを開いてください。

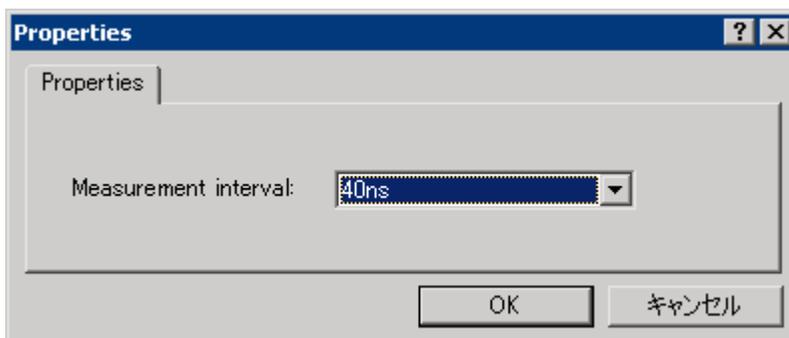


図 5.103 [Properties]ダイアログボックス

【注】 リアルタイムプロファイル機能を使用する場合は、必ず以下の設定を行ってください。

- ・トレースウィンドウのポップアップメニューの[設定]メニューを選択することによって開く[Acquisition]ダイアログボックス

[Trace type] : [AUD Trace]を選択してください。

[Trace Extend Mode] : [Trace data with PPC]チェックボックスのチェックを外してください。

- ・イベントポイントウィンドウのCh12

[Branch event]シート : 全ての分岐条件にチェックをつけてください。

[Action]シート : [Acquire trace]にチェックをつけてください。

これらは E200F エミュレータ起動時のデフォルト設定です。

### 5.11 コードカバレッジを測定する

E200F エミュレータでは、ユーザが指定したアドレス範囲についてコードカバレッジ情報 (C0 カバレッジ) を収集し、結果を表示します。

- 【注】
1. 遅延スロットの実行情報は、測定できません。
  2. コードカバレッジ機能を使用する場合は、必ず以下の設定を行ってください。
    - ・トレースウィンドウのポップアップメニューの[設定]メニューを選択することによって開く [Acquisition] ダイアログボックス
      - [Trace type] : [AUD Trace] を選択してください。
      - [Trace Extend Mode] : [Trace data with PPC] チェックボックスのチェックを外してください。
    - ・イベントポイントウィンドウの Ch12
      - [Branch event] シート : 全ての分岐条件にチェックをつけてください。
      - [Action] シート : [Acquire trace] にチェックをつけてください。これらは E200F エミュレータ起動時のデフォルト設定です。
  3. 本機能は、AUD トレースで出力されるデータを使用しています。  
したがって、リアルタイムトレースモードで使用された場合、トレースデータを消失することがあり、この場合は正常にコードカバレッジ情報収集ができません。このような場合には、ノンリアルタイムトレースモードで使用されることを推奨します。E200F エミュレータ起動時のデフォルト設定はリアルタイムトレースモードです。

### 5.11.1 コードカバレジウィンドウを開く

[表示->コード->カバレジ->コードカバレジ...]を選択するか[コードカバレジ]ツールバーボタンをクリックしてください。

エミュレータ起動後、初めてカバレジ機能を使用する場合は、以下のダイアログボックスが開きます。

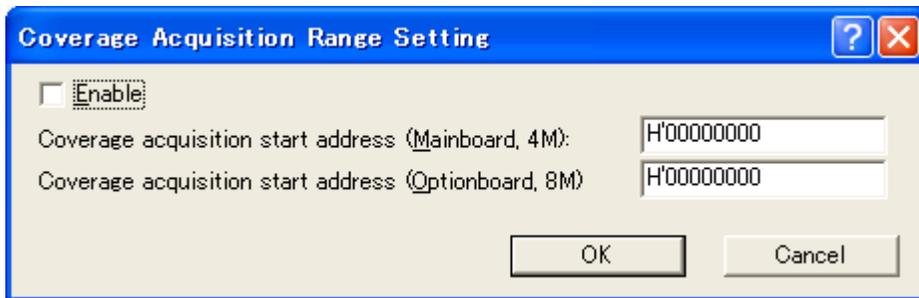


図 5.104 [カバレジ取得範囲設定]ダイアログボックス

[有効]	このチェックボックスをチェックすると、コードカバレジ情報の収集を有効にします。
[Coverage acquisition start address (Mainboard, 4M)]	測定開始アドレスを指定します。測定範囲は 4MB です。
[Coverage acquisition start address (Optionboard, 8M)]	測定開始アドレスを指定します。測定範囲は 8MB です。本機能は拡張プロファイルユニットを接続している場合に使用できます。
[OK]	指定した条件を設定して、ダイアログボックスを閉じます。
[キャンセル]	指定した条件を設定しないで、ダイアログボックスを閉じます。

本ダイアログボックスは、エミュレータの起動後 2 回目以降は表示しません。本ダイアログボックスを開く場合は、[コードカバレジ]ウィンドウのポップアップメニューから[ハードウェア設定...]を選択してください。

[カバレジ取得範囲設定]ダイアログボックスを閉じると、次に[カバレジウィンドウを開く]ダイアログボックスが開きます。

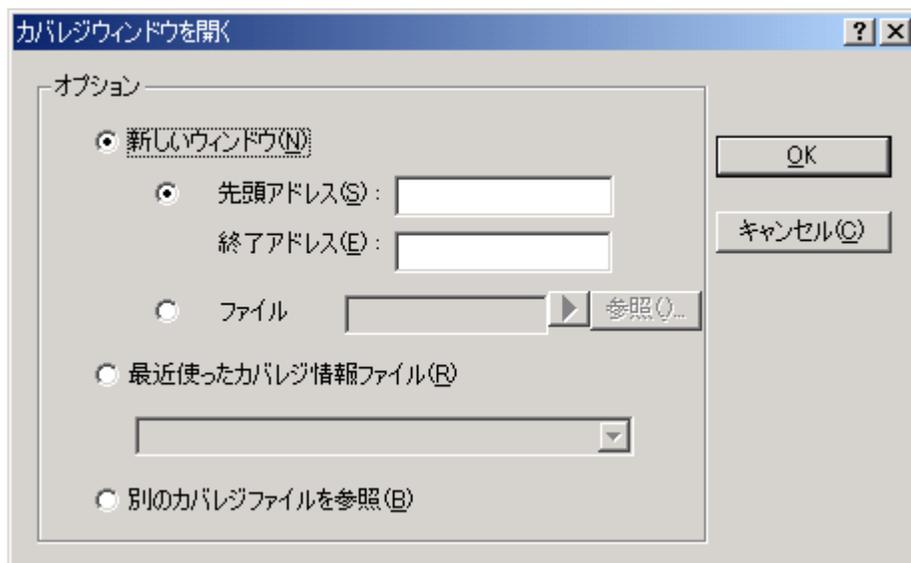


図 5.105 [カバレッジウィンドウを開く]ダイアログボックス

[カバレッジウィンドウを開く]ダイアログボックスでは、カバレッジウィンドウに表示したい範囲を設定します。

[新しいウィンドウ]	新たなカバレッジ範囲を設定します。
[先頭アドレス]	カバレッジ情報表示の開始アドレスを指定します。（接頭辞省略時は 16 進で入力）
[終了アドレス]	カバレッジ情報表示の終了アドレスを指定します。（接頭辞省略時は 16 進で入力）
[ファイル]	現在のプロジェクト中の".C"または".CPP"を型名に持つソースファイルを指定します。 これにより指定ファイル内に存在する関数をカバレッジ範囲として設定します。 ファイル型名を省略した場合は、".C"を補います。".C"または".CPP"以外のファイル型名を持つファイルは指定できません。 ブレースフォルダまたは[参照...]ボタンが利用できます。
[最近使ったカバレッジ情報ファイル]	最近保存されたファイルが 4 個まで表示されます。リストの中から選択してください。
[別のカバレッジファイルを参照]を参照]	リストに無いファイルを参照できます。

本ダイアログボックスで[OK]ボタンをクリックすると、[コードカバレッジ]ウィンドウを表示します。

アドレス範囲を指定した場合とソースファイルを指定した場合は、ウィンドウの表示形式が異なります。

**【注】** 1 枚の[カバレッジウィンドウ]で開けるアドレス範囲は 2MB までとなります。測定範囲全体を確認される場合は、複数のウィンドウに分けて表示してください。

## (1) コードカバレッジウィンドウ (アドレス指定)

Range	Statistic	Executed	Address	Assembler	Source
00000800- 00000FFF	-	0	00000800	MOV.L @ (H'0040:8,PC), R15	void PowerON_Rese
		0	00000802	MOV.L @ (H'0044:8,PC), R2	set_vbr ({voj
		0	00000804	MOV #H'10, R6	
		0	00000806	SUB R6, R2	
		0	00000808	LdC R2, VBR	
		0	0000080A	MOV.L @ (H'0040:8,PC), R3	_CALL_INIT();
		0	0000080C	JSR @R3	
		0	0000080E	NOP	
		0	00000810	STS DSR, R2	set_cr (SR_Inj
		0	00000812	MOV.L @ (H'003C:8,PC), R14	

図 5.106 [コードカバレッジ]ウィンドウ (アドレス指定)

[コードカバレッジ]ウィンドウはスプリッターで2分割されています。

## (a) 左側ウィンドウ

カバレッジ範囲とカバレッジ統計情報を表示します。

表示する項目は以下の通りです。

[Range]	アドレス範囲
[Statistic]	C0 カバレッジ値 (パーセント表示)

[Statistic]カラムのパーセントは、左側ウィンドウのポップアップメニューで[パーセンテージ]を選択すると表示されます。

Range	Statistic
00002000- 000020FF	-

パーセンテージ(P)

図 5.107 [コードカバレッジ]ウィンドウのパーセント表示

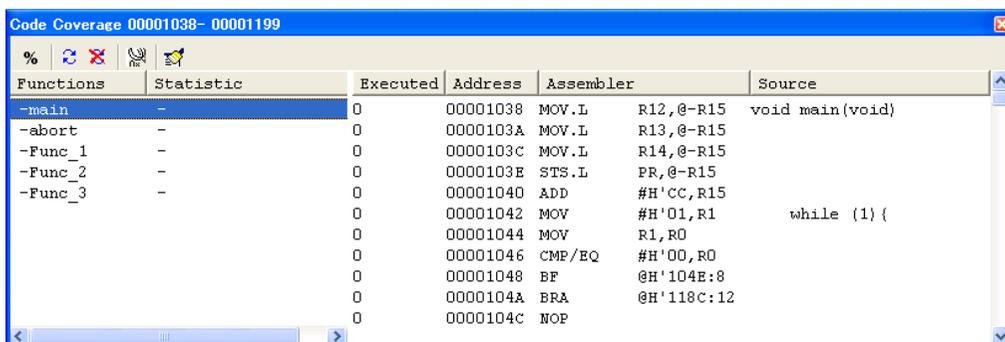
## (b) 右側ウィンドウ

C/C++およびアセンブラレベルでのカバレッジ情報を表示します。

表示する項目は以下の通りです。

[Executed]	1 : 命令を実行した 0 : 命令を実行していない
[Address]	命令アドレス
[Assembler]	逆アセンブル表示
[Source]	C/C++またはアセンブラソース

## (2) コードカバレッジウィンドウ (ソースファイル指定)



Functions	Statistic	Executed	Address	Assembler	Source
-main	-	0	00001038	MOV.L R12,@-R15	void main(void)
-abort	-	0	0000103A	MOV.L R13,@-R15	
-Func_1	-	0	0000103C	MOV.L R14,@-R15	
-Func_2	-	0	0000103E	STS.L PR,@-R15	
-Func_3	-	0	00001040	ADD #H'CC,R15	
		0	00001042	MOV #H'01,R1	while (1){
		0	00001044	MOV R1,R0	
		0	00001046	CMP/EQ #H'00,R0	
		0	00001048	BF @H'104E:8	
		0	0000104A	BRA @H'118c:12	
		0	0000104C	NOP	

図 5.108 [コードカバレッジ]ウィンドウ (ソースファイル指定)

[コードカバレッジ]ウィンドウはスプリッターで2分割されています。

## (a) 左側ウィンドウ

カバレッジ範囲とカバレッジ統計情報を表示します。

表示する項目は以下の通りです。

[Functions]	カバレッジ対象の関数
[Statistic]	C0 カバレッジ値 (パーセント表示)

[Statistic]カラムのパーセントは、左側ウィンドウのポップアップメニューで[パーセンテージ]を選択すると表示されます。(アドレス指定の場合と同様)

カラムタブをクリックすることで関数名またはC0カバレッジ値を降順または昇順に並べ替えることができます。

## (b) 右側ウィンドウ

左側のウィンドウでダブルクリックにより選択された関数について、C/C++およびアセンブラレベルでのカバレッジ情報を表示します。

表示する項目は以下の通りです。

[Executed]	1: 命令を実行した 0: 命令を実行していない
[Address]	命令アドレス
[Assembler]	逆アセンブル表示
[Source]	C/C++またはアセンブラソース

### 5.11.2 ソースファイルを表示する

ポップアップメニューから[ソースファイル表示]を選択すると、[エディタ]ウィンドウを開いて、[コードカバレッジ]ウィンドウ上のカーソル位置のアドレスに対応するソースファイルを表示します。

### 5.11.3 表示アドレスを変更する

ポップアップメニューから[表示アドレス...]を選択すると、[表示アドレス]ダイアログボックスを表示します。



図 5.109 [表示アドレス]ダイアログボックス

本ダイアログボックスで[コードカバレッジ]ウィンドウの表示アドレスを変更します。

## 5.11.4 カバレッジ表示範囲を変更する

## (1) カバレッジ表示範囲をアドレスで指定した場合

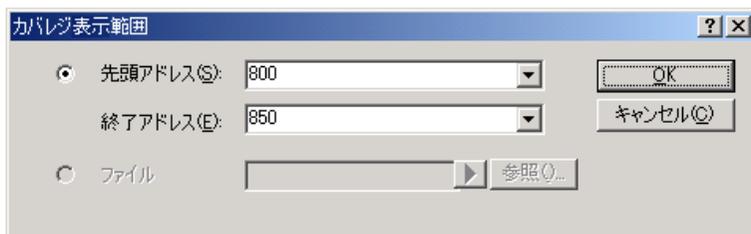


図 5.110 [カバレッジ表示範囲]ダイアログボックス (アドレス指定)

本ダイアログボックスで命令実行情報取得の条件を変更します。

下記項目を指定できます。

[開始アドレス]	先頭アドレス (接頭辞省略時は 16 進で入力)
[終了アドレス]	終了アドレス (接頭辞省略時は 16 進で入力)

[OK]ボタンをクリックすると、カバレッジ表示範囲を変更します。

## (2) カバレッジ表示範囲をソースファイルで指定した場合



図 5.111 [カバレッジ表示範囲]ダイアログボックス (ソースファイル指定)

本ダイアログボックスで命令実行情報取得の条件を変更します。

下記項目を指定できます。

[ファイル] 現在のプロジェクト中の ".C" または ".CPP" を型名に持つソースファイルを指定します。これにより指定ファイル内に存在する関数をカバレッジ範囲として設定します。ファイル型名を省略した場合は、".C" を補います。".C" または ".CPP" 以外のファイル型名を持つファイルは指定できません。ブレースフォルダまたは[参照...]ボタンが利用できます。

[OK]ボタンをクリックすると、カバレッジ測定範囲を変更します。

### 5.11.5 カバレジ情報をクリアする

#### (1) 指定した範囲のカバレジ情報をクリアする

ポップアップメニューから[カバレジ範囲クリア]を選択すると、[カバレジ範囲クリア]ダイアログボックスを表示します。



図 5.112 [カバレジ範囲クリア]ダイアログボックス

クリアする範囲の開始アドレスと終了アドレスを指定します。

[OK]ボタンをクリックすると、指定した範囲をクリアします。

#### (2) すべてのカバレジ情報をクリアする

ポップアップメニューから[カバレジ全クリア]を選択すると、[カバレジ情報を保存]ダイアログボックスを表示します。

### 5.11.6 カバレジ情報をファイルに保存する

ポップアップメニューから[保存...]を選択すると、[カバレジ情報を保存]ダイアログボックスを表示します。



図 5.113 [カバレジ情報を保存]ダイアログボックス

保存するカバレジ情報ファイルの場所と名前を指定します。ブレースフォルダまたは[参照...]ボタンが使用できません。

ファイル拡張子の入力を省略すると、ファイル拡張子として ".COV" を自動的に付加します。

ファイル拡張子として、".COV" および ".TXT" 以外を入力するとエラーメッセージを出力します。

### 5.11.7 カバレッジ情報をファイルからロードする

ポップアップメニューから[ロード...]を選択すると、[カバレッジ情報ロード]ダイアログボックスを表示します。



図 5.114 [カバレッジ情報ロード]ダイアログボックス

ロードするカバレッジ情報ファイルの場所と名前を指定します。ブレースフォルダまたは[参照...]ボタンが使用できます。

ロードできるファイル拡張子は ".COV" のみです。その他のファイル拡張子を入力するとエラーメッセージを出力します。

【注】 カバレッジ測定範囲をソースファイルで指定した場合は、保存した ".COV" ファイルをロードすることはできません。

### 5.11.8 最新の情報に更新する

ポップアップメニューから[最新の情報に更新]を選択すると、[コードカバレッジ]ウィンドウの内容を最新に更新します。

### 5.11.9 情報の更新を抑止する

ポップアップメニューの[表示固定]をチェックすると、プログラム実行停止時にウィンドウの更新を行いません。これにより、カバレッジ情報取得のためのエミュレータへのアクセスを抑止することができます。

### 5.11.10 保存確認ダイアログボックス

- (1) コードカバレッジ情報をクリアするとき、または[コードカバレッジ]ウィンドウを閉じるとき確認のダイアログボックスを表示します。

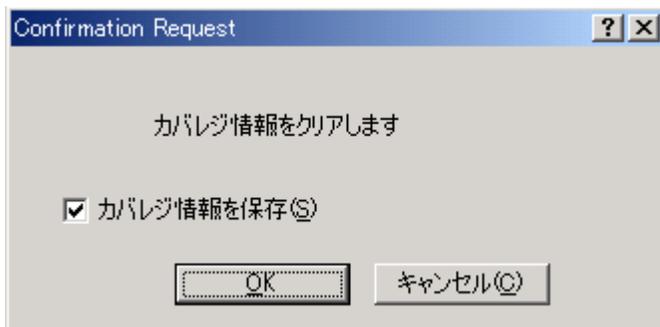


図 5.115 [Confirmation Request]ダイアログボックス

[カバレッジ情報を保存]チェックボックスをチェックすると、カバレッジ情報をクリアする前にカバレッジデータをファイルに保存することができます。

[Ok]ボタンをクリックすると、カバレッジ情報をクリアします。

- (2) [ファイル->セッションの保存]を選択したとき  
[コードカバレッジ]ウィンドウを開いている場合、開いているウィンドウ数分の[カバレッジ情報を保存]ダイアログボックスを開きます。

それぞれのウィンドウのデータを別々またはまとめて保存することができます。

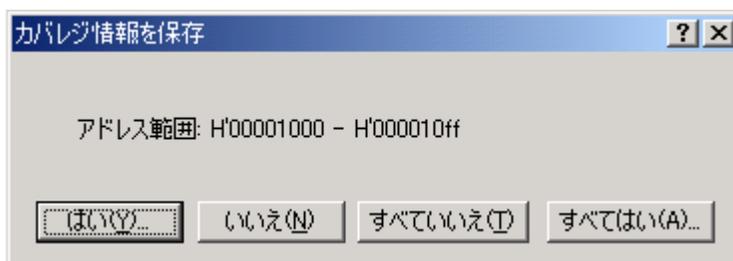


図 5.116 [カバレッジ情報を保存]ダイアログボックス

[すべていいえ]ボタンをクリックすると、すべてのカバレッジ情報を保存しないでダイアログボックスを閉じます。  
[すべてはい]ボタンをクリックすると、すべての[コードカバレッジ]ウィンドウデータを1個のファイルに保存します。

## 5.11.11 エディタウィンドウへのコードカバレッジ結果表示

命令実行済のソース行に対応するデバッグカラムを強調表示することで[エディタ]ウィンドウにもカバレッジ結果を表示します。[コードカバレッジ]ウィンドウでカバレッジに関する設定を変更すると、対応するデバッグカラムの表示も更新します。

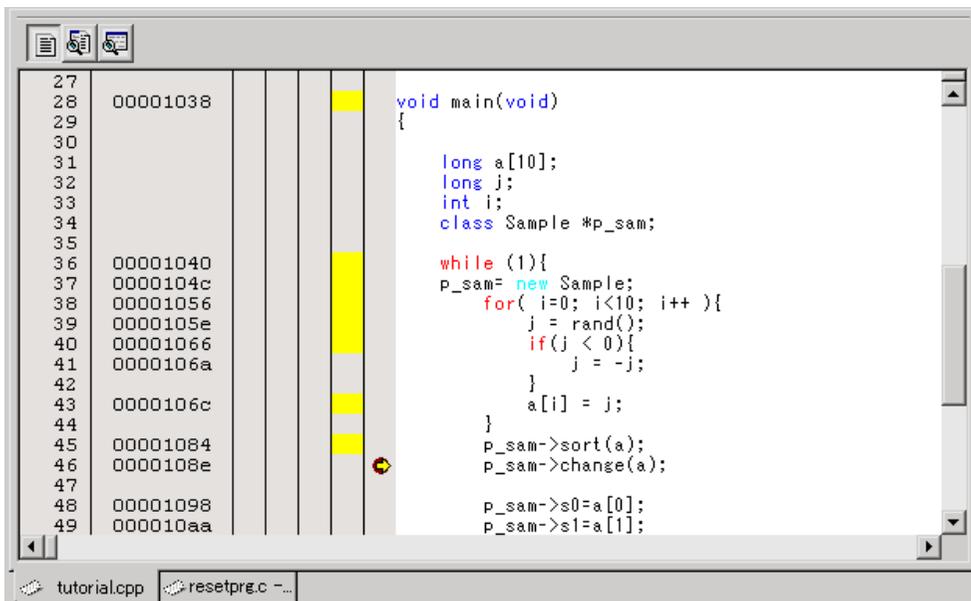


図 5.117 [Code Coverage]カラム

## 5.12 複数デバッグプラットフォームを同期動作させる

High-performance Embedded Workshop では複数のデバッグプラットフォームを同期動作させることができます。

複数デバッグプラットフォームの同期動作は High-performance Embedded Workshop から High-performance Embedded Workshop を起動することで実現しています。ここで起動する側の High-performance Embedded Workshop をマスタ、起動された側の High-performance Embedded Workshop をスレーブと呼びます。

スレーブ High-performance Embedded Workshop を起動するには、[ツール->従属 HEW の起動...]を選択するか、[従属 HEW の実行]ツールバーボタン  をクリックします。

スレーブ High-performance Embedded Workshop はマスタ High-performance Embedded Workshop と同様の操作が可能です。

マスタ High-performance Embedded Workshop での下記動作はスレーブ High-performance Embedded Workshop に通知されます。これによりスレーブ High-performance Embedded Workshop をマスタ High-performance Embedded Workshop と同期させて動作させることができます。

- リセット後実行
- 実行
- プログラムの停止

【注】 マスタ High-performance Embedded Workshop から複数のスレーブ High-performance Embedded Workshop を起動することはできませんが、スレーブ High-performance Embedded Workshop のネスト(スレーブ High-performance Embedded Workshop からスレーブ High-performance Embedded Workshop を起動する)はできません。

また、[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]より別の High-performance Embedded Workshop を起動することで2台のエミュレータで別々のデバッグを行うことができます。

### 5.12.1 2台のエミュレータを区別する方法

E200F エミュレータを2台、USB コネクタに接続してください。

次に、チュートリアル用ワークスペース等を使用して、High-performance Embedded Workshop を起動してください。

以下のメッセージが表示されます。



図 5.118 ドライバ選択メッセージ

OK ボタンを押してください。以下のウィンドウが表示されます。

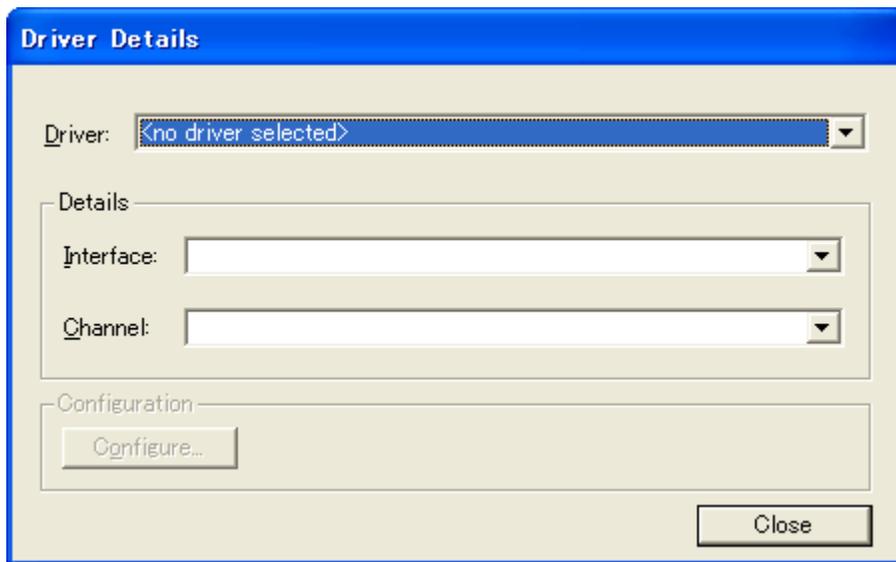


図 5.119 [Driver Details]ダイアログボックス(1)

[Driver] ドロップダウンリストボックスで、「Renesas E-Series USB Driver」を選択し、[Channel] ドロップダウンリストボックスをドロップダウンしてください。

以下のように、[Channel]ドロップダウンリストボックスに2台分のチャンネル情報が表示されます。

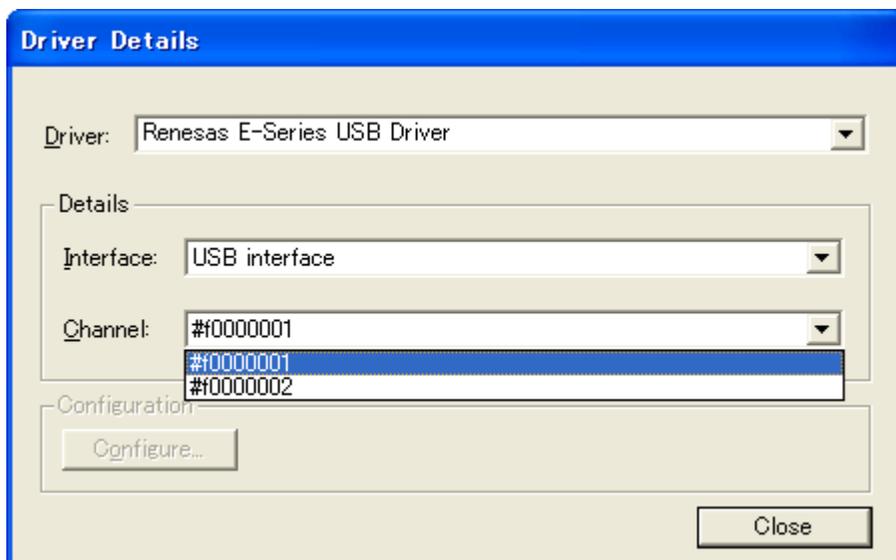


図 5.120 [Driver Details]ダイアログボックス(2)

この情報が、エミュレータのシリアル番号情報です。

High-performance Embedded Workshop 起動時には、マスタ CPU 側に接続されているエミュレータのシリアル番号を[Channel]ドロップダウンリストボックスから選択してください。起動処理は、通常と同様です。

スレーブ High-performance Embedded Workshop 起動時には、スレーブ CPU 側に接続されているエミュレータのシリアル番号を[Channel]ドロップダウンリストボックスから選択して起動してください。



---

## 6. チュートリアル

---

### 6.1 はじめに

E200F エミュレータの主な機能を紹介するために、チュートリアルプログラムを提供しています。このプログラムを用いて説明します。

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

チュートリアルプログラムでは、以下の処理を行います。

main 関数でソートするランダムデータを生成します。

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

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

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

#### 【留意事項】

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 E200F 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。

## 6.2 High-performance Embedded Workshop の起動

「4.1 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。

## 6.3 E200F 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、E200F 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できるE200Fエミュレータを使用してい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「2.2.2 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。

## 6.4 [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、[オプション]メニューから[エミュレータ]を選択し、さらに[システム...]を選択してください。[Configuration]ダイアログボックスが表示されます。

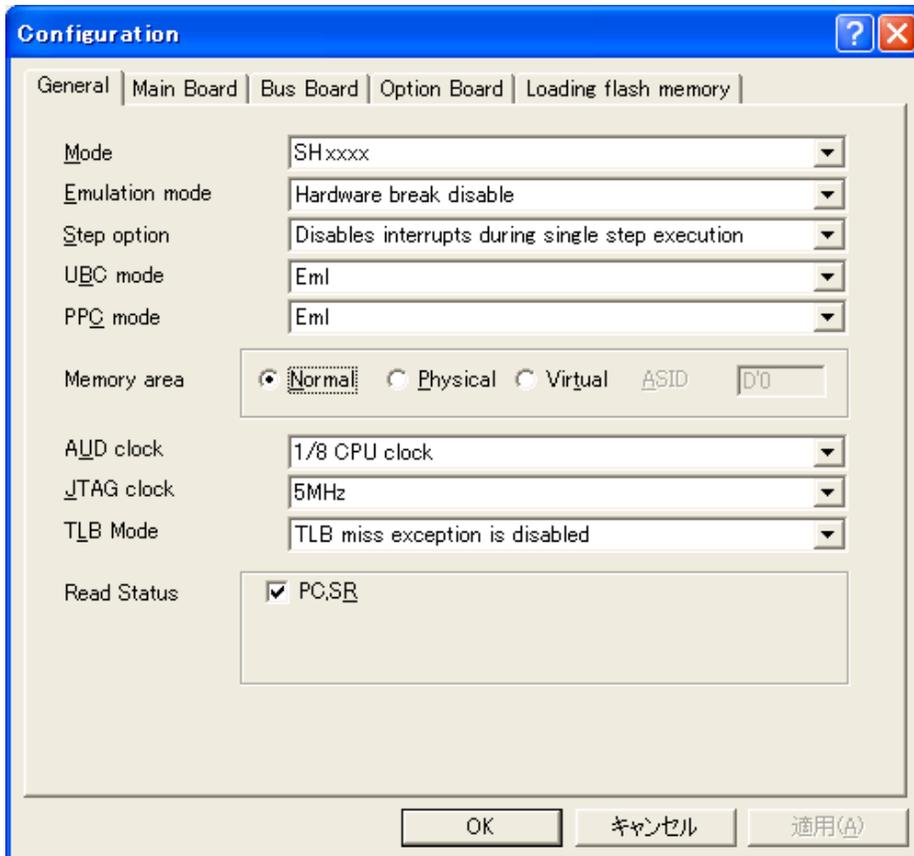


図 6.1 [Configuration]ダイアログボックス

[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。

## 6.5 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 MPU 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、[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" を入力してください。

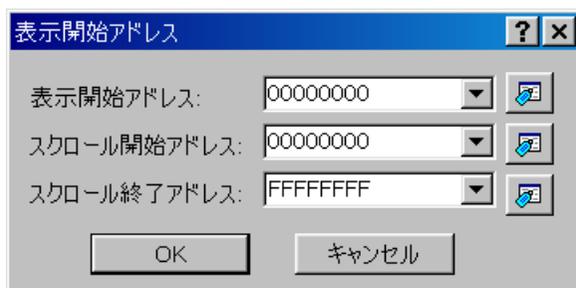


図 6.2 [表示開始アドレス]ダイアログボックス

[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。

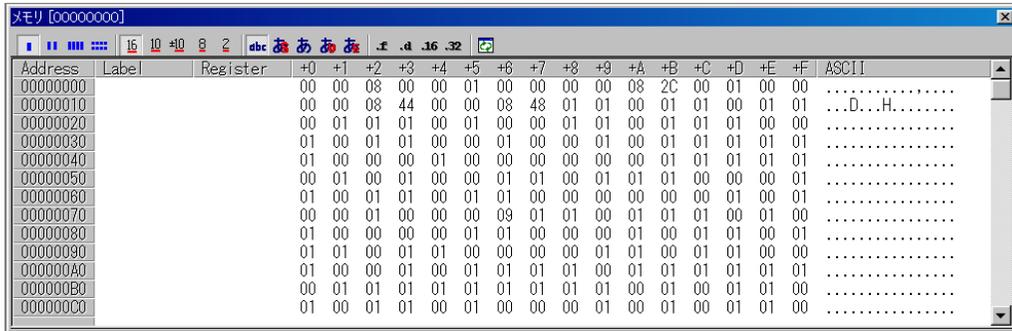


図 6.3 [メモリ]ウィンドウ

[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。

表示データサイズを変更する場合等、メモリウィンドウの詳しい使い方については、High-performance Embedded Workshop ユーザーズマニュアルを参照してください。

## 6.6 チュートリアルプログラムのダウンロード

### 6.6.1 チュートリアルプログラムをダウンロードする

デバッグしたいオブジェクトプログラムをダウンロードできます。

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

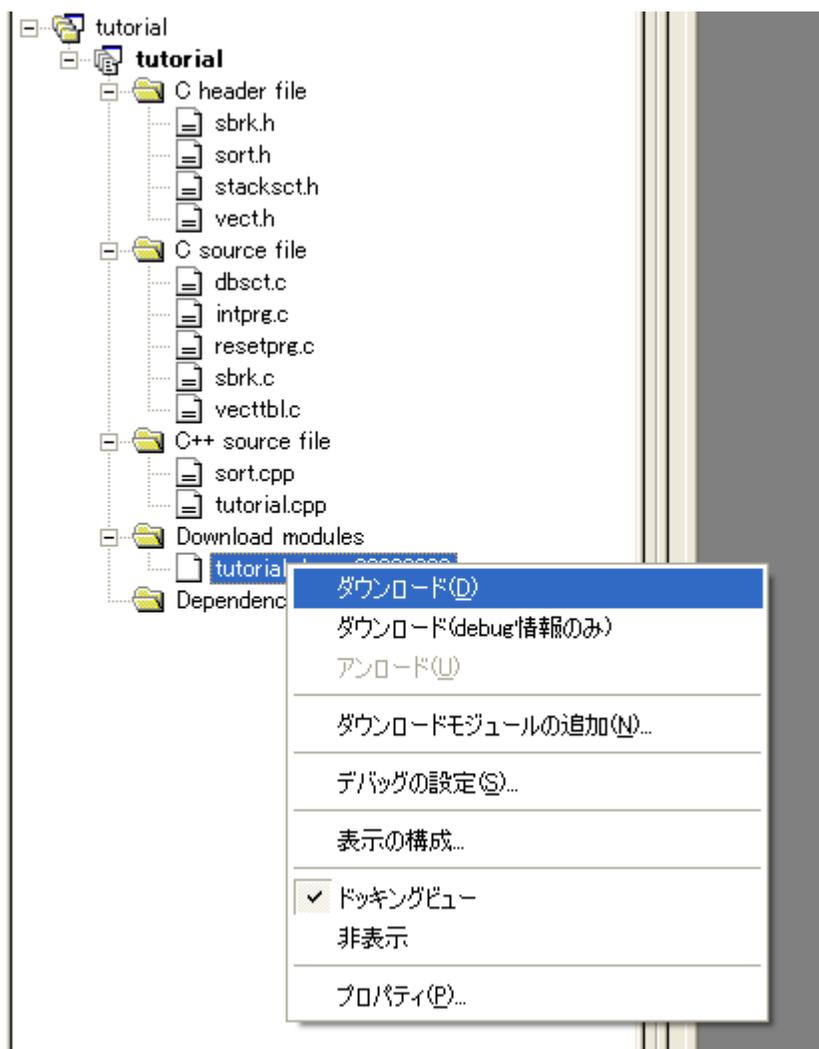


図 6.4 チュートリアルプログラムのダウンロード

## 6.6.2 ソースプログラムを表示する

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

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

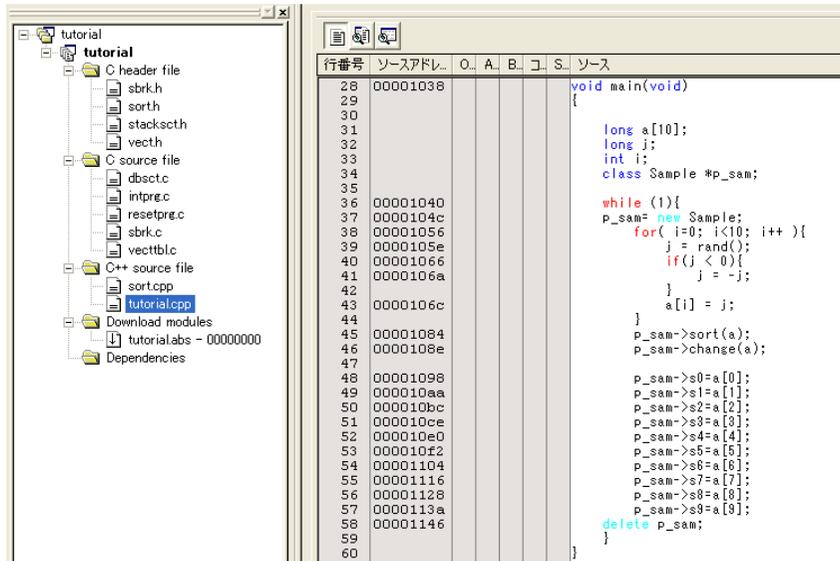


図 6.5 [エディタ]ウィンドウ (ソースプログラムの表示)

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。

## 6.7 PC ブレークポイントの設定

簡単なデバッグ機能の 1 つに PC ブレークポイントがあります。

[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。

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。

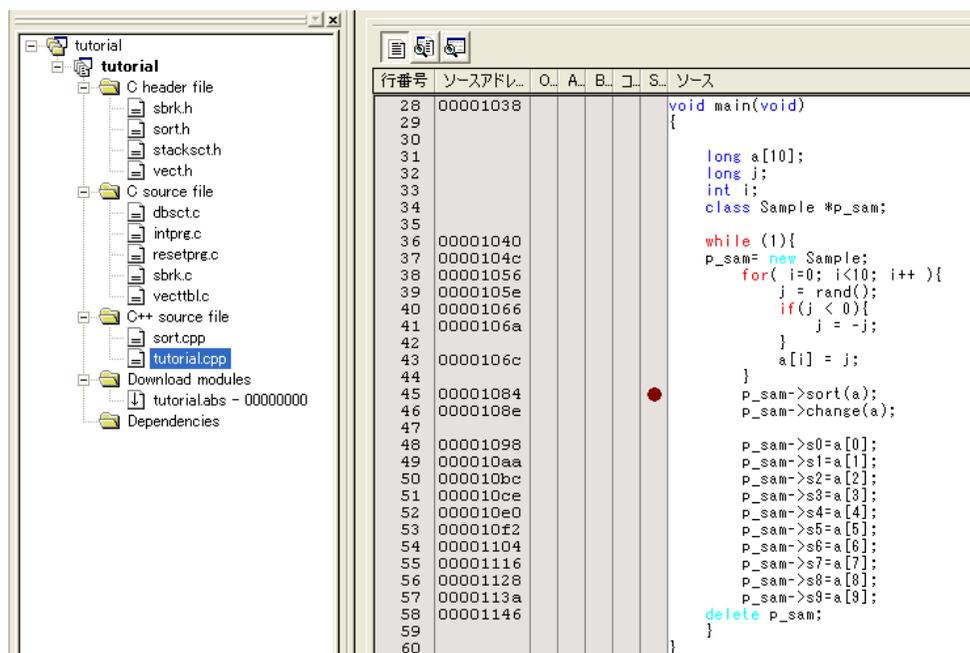


図 6.6 [エディタ]ウィンドウ (PC ブレークポイントの設定)

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。



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。

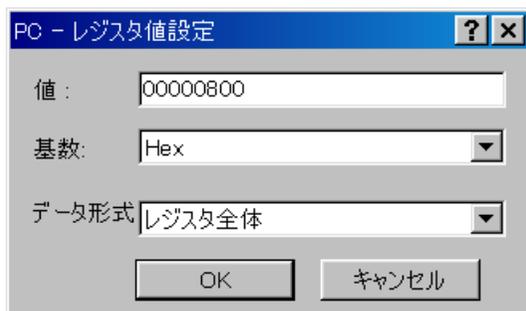


図 6.8 [レジスタ値設定]ダイアログボックス (PC)

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'00010000を設定してください。



図 6.9 [レジスタ値設定]ダイアログボックス (R15)

## 6.9 プログラムの実行

プログラムの実行方法について説明します。

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



図 6.10 [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、実行 PC アドレスの表示となります。プログラムはブレークポイントを設定したところまで実行されます。プログラムが停止した位置を示すために [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 インストールドライブ >  
`¥Workspace¥Tutorial¥E200F¥xxxx¥Tutorial¥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 6.8 よりやり直してください。

```
28 00001038 void main(void)
29
30
31 long a[10];
32 long j;
33 int i;
34 class Sample *p_sam;
35
36 00001040 while (1){
37 0000104c p_sam= new Sample;
38 00001056 for( i=0; i<10; i++ ){
39 0000105e     j = rand();
40 00001066     if(j < 0){
41 0000106a         j = -j;
42
43 0000106c     }
44         a[i] = j;
45 00001084     }
46 0000108e     p_sam->sort(a);
47         p_sam->change(a);
48
49 00001098     p_sam->s0=a[0];
50 000010aa     p_sam->s1=a[1];
51 000010bc     p_sam->s2=a[2];
52 000010ce     p_sam->s3=a[3];
53 000010e0     p_sam->s4=a[4];
54 000010f2     p_sam->s5=a[5];
55 00001104     p_sam->s6=a[6];
56 00001116     p_sam->s7=a[7];
57 00001128     p_sam->s8=a[8];
58 0000113a     p_sam->s9=a[9];
59 delete p_sam;
60 }
```

図 6.11 [エディタ]ウィンドウ (ブレーク状態)

[ステータス]ウィンドウで最後に発生したブレークの要因が確認できます。

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

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

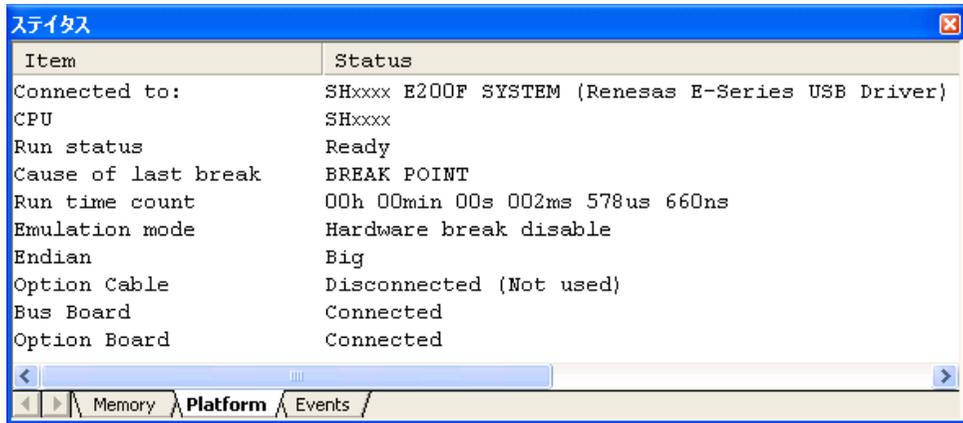


図 6.12 [ステータス]ウィンドウ

**【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。

## 6.10 ブレークポイントの確認

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

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

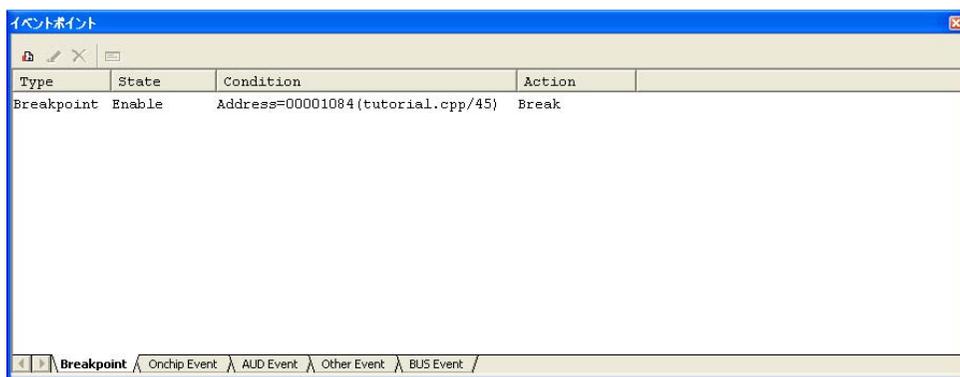


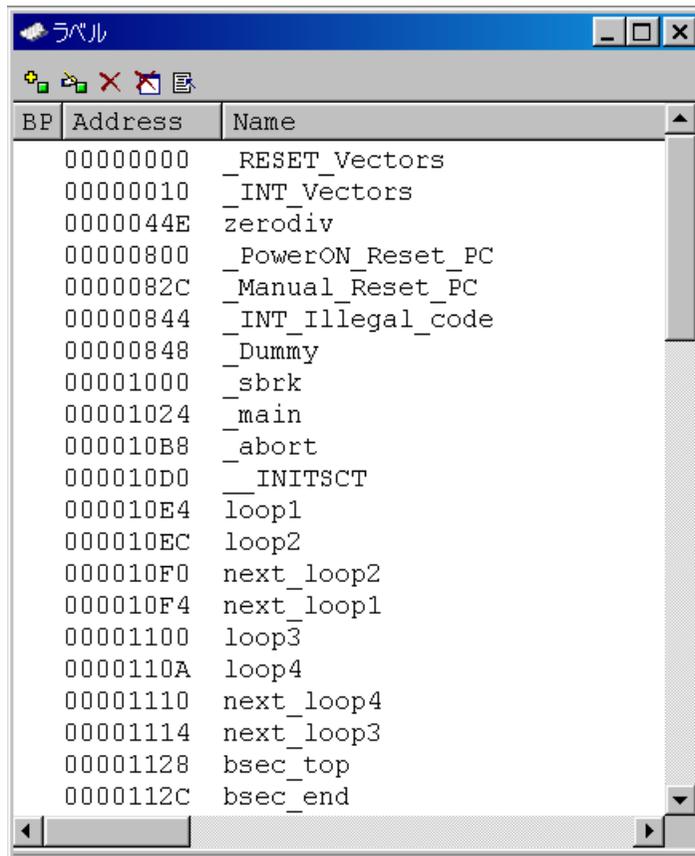
図 6.13 [Event]ウィンドウ

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

## 6.11 シンボルの参照

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

[表示]メニューの[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。



The screenshot shows a window titled 'ラベル' (Labels) with a table of symbols. The table has three columns: 'BP', 'Address', and 'Name'. The symbols listed include various system and user-defined labels such as \_RESET\_Vectors, \_INT\_Vectors, zerodiv, and several loop and abort symbols.

BP	Address	Name
	00000000	_RESET_Vectors
	00000010	_INT_Vectors
	0000044E	zerodiv
	00000800	_PowerON_Reset_PC
	0000082C	_Manual_Reset_PC
	00000844	_INT_Illegal_code
	00000848	_Dummy
	00001000	_sbrk
	00001024	_main
	000010B8	_abort
	000010D0	__INITSCT
	000010E4	loop1
	000010EC	loop2
	000010F0	next_loop2
	000010F4	next_loop1
	00001100	loop3
	0000110A	loop4
	00001110	next_loop4
	00001114	next_loop3
	00001128	bsec_top
	0000112C	bsec_end

図 6.14 [ラベル]ウィンドウ

## 6.12 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。

[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 " を入力してください。



図 6.15 [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。

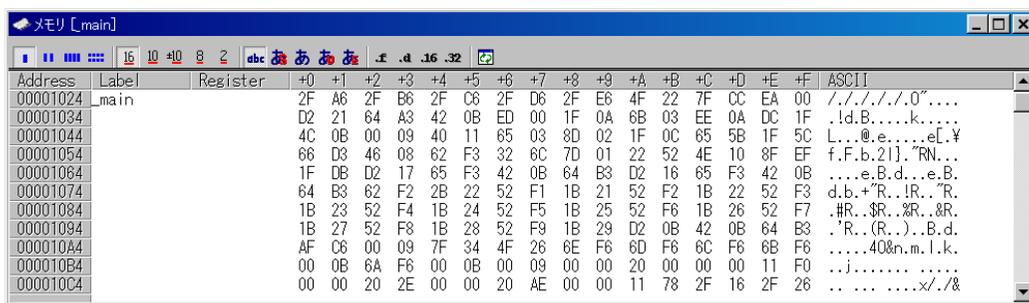


図 6.16 [メモリ]ウィンドウ

## 6.13 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。

[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。

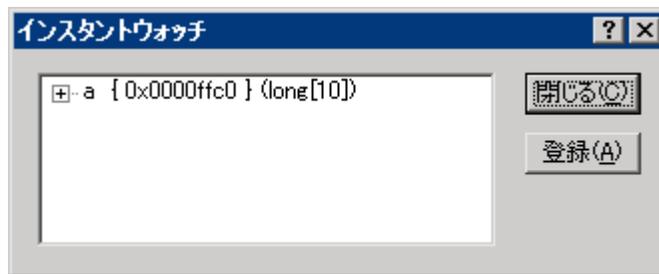


図 6.17 [インスタントウォッチ]ダイアログボックス

[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。

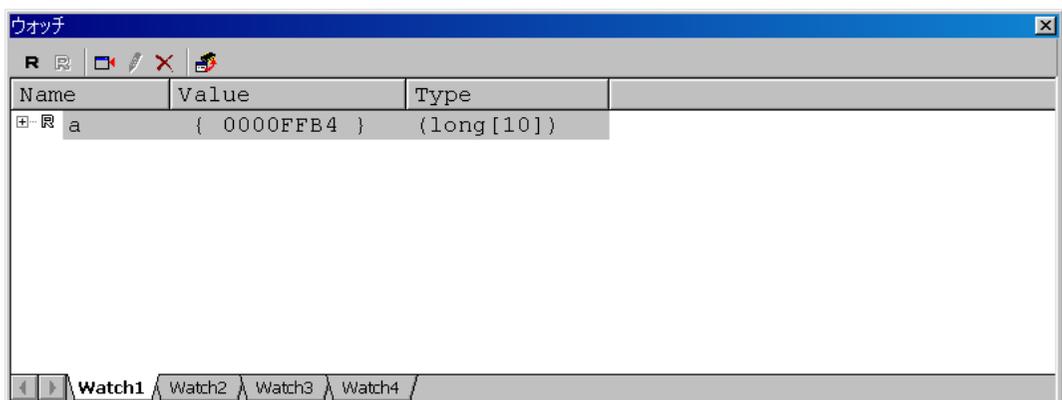


図 6.18 [ウォッチ]ウィンドウ (配列の表示)

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

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

以下のダイアログボックスが表示されますので、変数 `i` を入力してください。

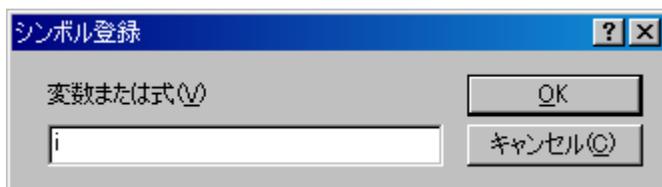


図 6.19 [シンボル登録]ダイアログボックス

[OK]ボタンをクリックします。

[ウォッチ]ウィンドウに、`int` 型の変数 `i` が表示されます。

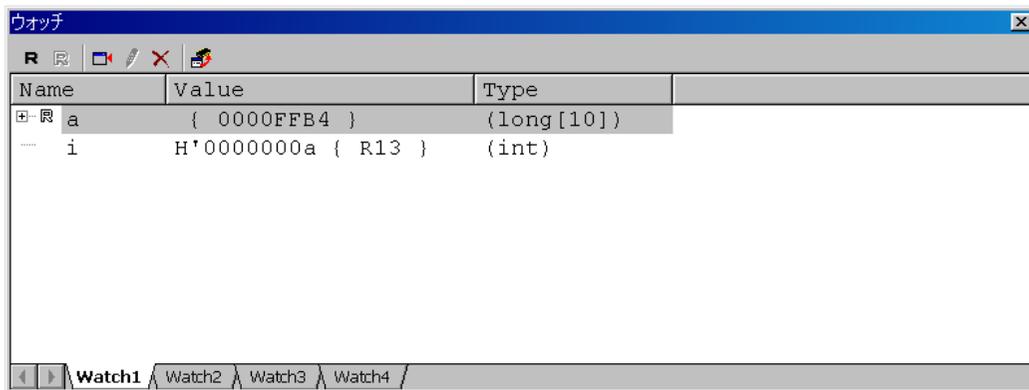


図 6.20 [ウォッチ]ウィンドウ (変数の表示)

[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。

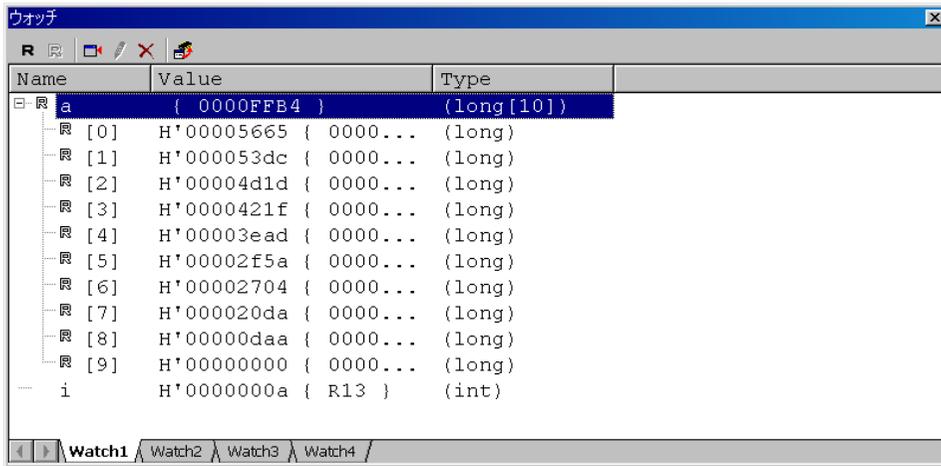


図 6.21 [ウォッチ]ウィンドウ (配列要素の表示)

## 6.14 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。

[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。

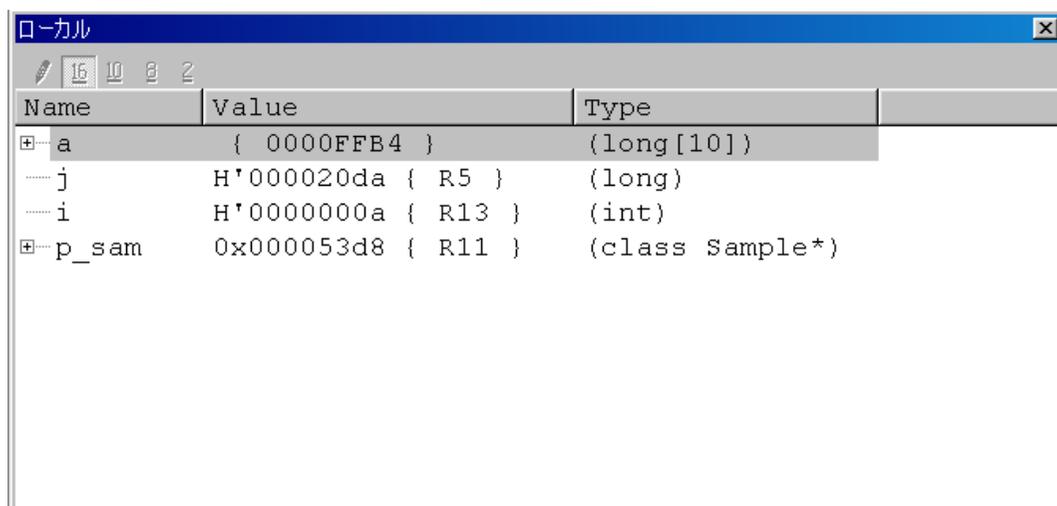


図 6.22 [ローカル]ウィンドウ

[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。

## 6.15 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。

### 6.15.1 ステップインの実行

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

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



図 6.23 [ステップイン]ボタン

```
12 00002000 Sample::Sample()
13 00002002 {
14 00002014     s0=0;
15 00002018     s1=0;
16 0000201a     s2=0;
17 0000201c     s3=0;
18 0000201e     s4=0;
19 00002020     s5=0;
20 00002022     s6=0;
21 00002024     s7=0;
22 00002026     s8=0;
23 00002028     s9=0;
24
25
26 00002032 ↪ void Sample::sort(long #a)
27 {
28     long t;
29     int i, j, k, gap;
30
31     gap = 5;
32     while( gap > 0 ){
33         for( k=0; k<gap; k++){
34             for( i=k+gap; i<10; i=i+gap ){
35                 for( j=i-gap; j>=k; j=j-gap){
36                     if(a[j]>a[j+gap]){
37                         t = a[j];
38                         a[j] = a[j+gap];
39                         a[j+gap] = t;
40                     }
41                     else
42                         break;
43                 }
44             }
45         }
46         gap = gap/2;
47     }
48 }
```

図 6.24 [エディタ]ウィンドウ (ステップイン)

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

## 6.15.2 ステップアウトの実行

ステップアウト機能はコール関数の中から抜け出し、コール元プログラムの次のステートメントで停止します。sort 関数の中から抜け出すために、[デバッグ]メニューから[ステップアウト]を選択するか、またはツールバーの[ステップアウト]ボタンをクリックしてください。

### 【留意事項】

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



図 6.25 [ステップアウト]ボタン

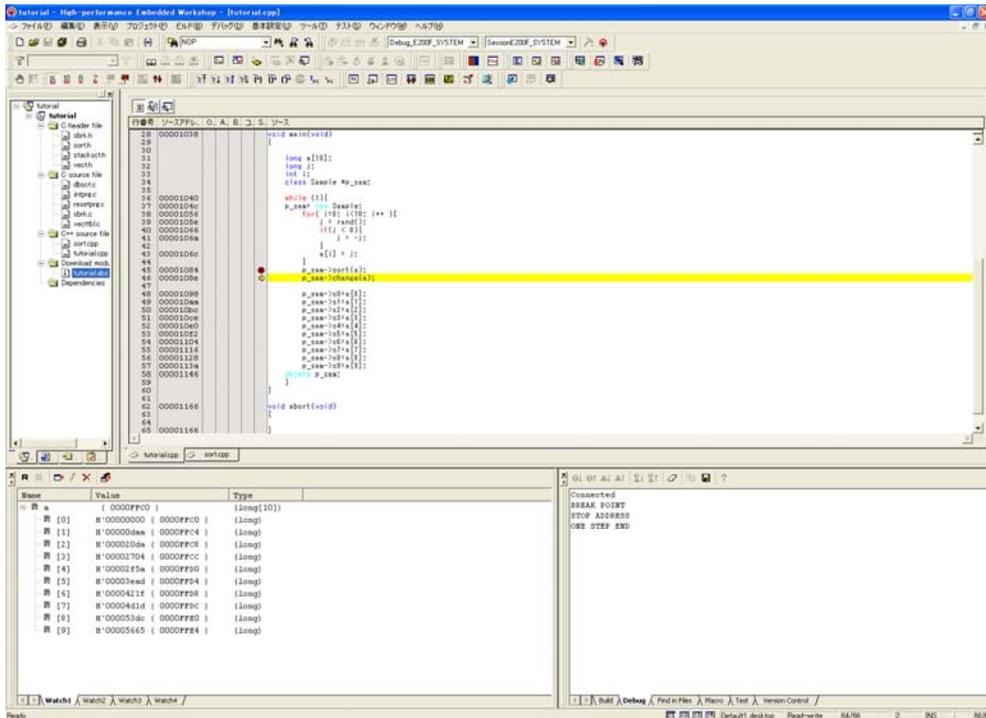


図 6.26 [High-performance Embedded Workshop]ウィンドウ (ステップアウト)

[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。

## 6.15.3 ステップオーバーの実行

ステップオーバー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。

「6.15.1 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、[デバッグ]メニューから[ステップオーバー]を選択するか、またはツールバーの[ステップオーバー]ボタンをクリックしてください。



図 6.27 [ステップオーバー]ボタン

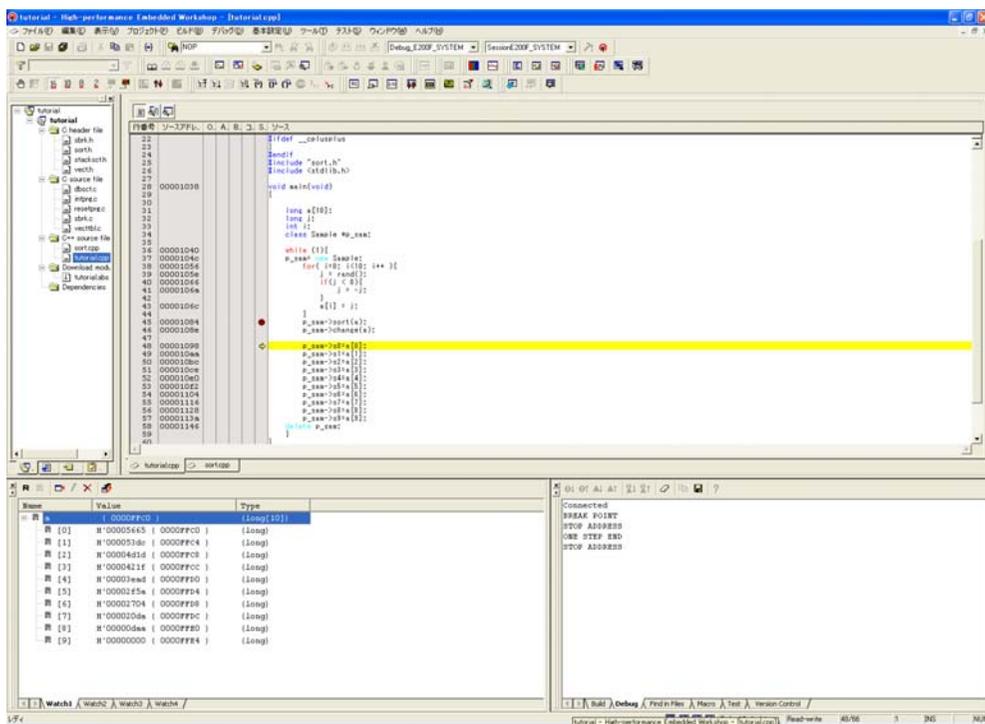


図 6.28 [High-performance Embedded Workshop]ウィンドウ (ステップオーバー)

## 6.16 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、[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 6.29 [実行]ボタン

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



図 6.30 [STOP]ボタン

## 6.17 ブレーク機能

E200F エミュレータは、PC ブレーク機能とイベントポイントによる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[Event]ウィンドウの[Breakpoint]シートで行います。また、イベントポイント条件の設定はイベント種類により、[Onchip Event], [AUD Event], [Other Event], [BUS Event]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。

### 6.17.1 PC ブレーク機能

E200F エミュレータは、1000 ポイントまで PC ブレークを設定することができます。

本章では、「6.7 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。

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

[Breakpoint]シートを開きます。



図 6.31 [Event]ウィンドウ (PC ブレーク設定前)

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

[Address]エディットボックスにアドレス H'00001098 を入力してください。

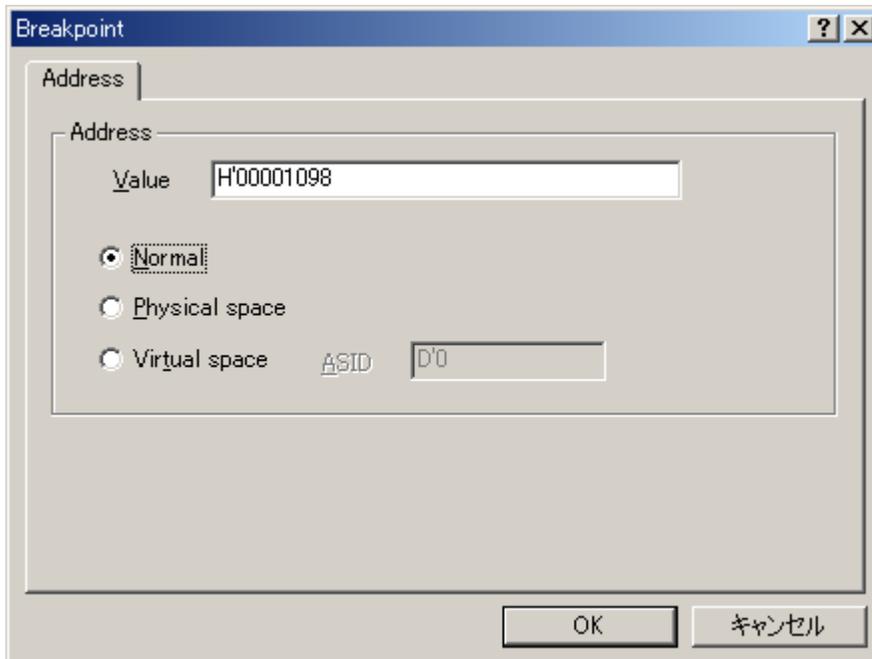


図 6.32 [Breakpoint]ダイアログボックス

**【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。

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

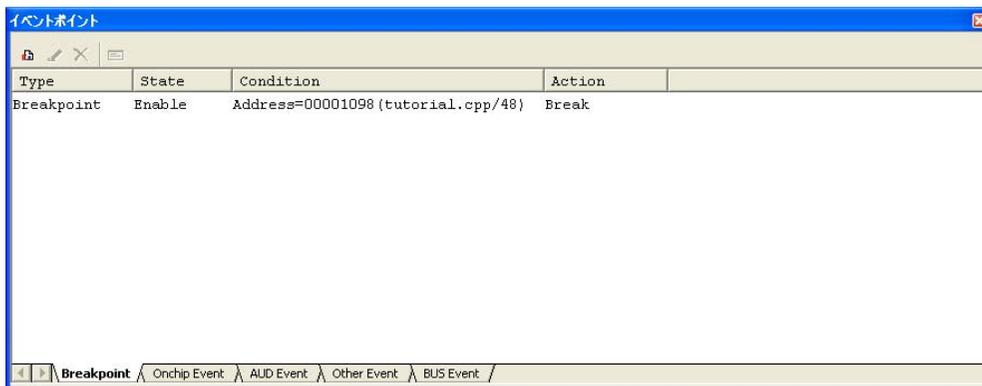


図 6.33 [Event]ウィンドウ (PC ブレーク設定時)

### 【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。

「6.8 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 ' 00010000) を[レジスタ]ウィンドウに設定して、[実行]ボタンをクリックしてください。

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 00001038 void main(void)
29
30
31     long a[10];
32     long j;
33     int i;
34     class Sample *p_sam;
35
36 00001040     while (1){
37 0000104c     p_sam= new Sample;
38 00001056         for( i=0; i<10; i++ ){
39 0000105e             j = rand();
40 00001066             if(j < 0){
41 0000106a                 j = -j;
42
43 0000106c                 a[i] = j;
44             }
45 00001084             p_sam->sort(a);
46 0000108e             p_sam->change(a);
47
48 00001098             p_sam->s0=a[0];
49 000010aa             p_sam->s1=a[1];
50 000010bc             p_sam->s2=a[2];
51 000010ce             p_sam->s3=a[3];
52 000010e0             p_sam->s4=a[4];
53 000010f2             p_sam->s5=a[5];
54 00001104             p_sam->s6=a[6];
55 00001116             p_sam->s7=a[7];
56 00001128             p_sam->s8=a[8];
57 0000113a             p_sam->s9=a[9];
58 00001146             delete p_sam;
59         }
60     }
61
62 00001166 void abort(void)
63     {

```

図 6.34 実行停止時の[エディタ]ウィンドウ (PC ブレーク)

[ステータス]ウィンドウの表示内容は、以下のようになります。

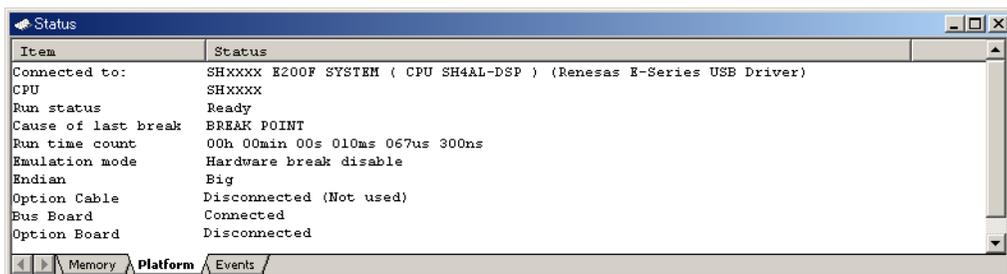


図 6.35 [ステータス]ウィンドウの表示内容 (PC ブレーク)

#### 【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。

## 6.18 イベントポイントによるブレーク機能

E200F エミュレータは、Onchip Eventpoint, AUD Eventpoint, Ohter Eventpoint, BUS Eventpoint を設定できます。ここでは、例として Onchip Eventpoint によるブレークの設定方法を説明します。他のイベントポイントの設定方法は、「5.6 イベントポイントを使用する」を参照してください。

### 6.18.1 Onchip Eventpoint によるブレーク

Onchip Eventpoint のイベントチャンネル 1 (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。

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

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

次は Onchip Eventpoint を設定します。

[Onchip Event]をクリックしてください。

Onchip Eventpoint は、13 箇所まで独立に条件を設定することができます。ここでは、イベントチャンネル 1 を設定します。

#### 【留意事項】

Onchip Eventpoint のイベントチャンネルの本数は、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。

[Event]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。

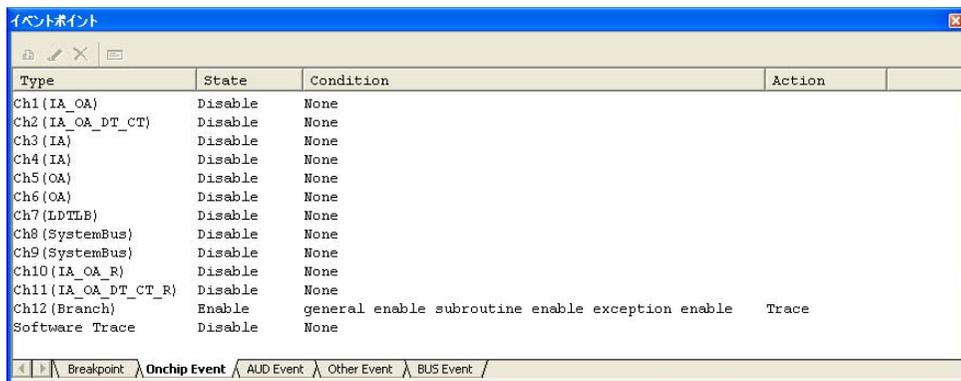


図 6.36 [Event]ウィンドウ ([Onchip Event]シート)

[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。

[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 ' 00001084 を入力してください。

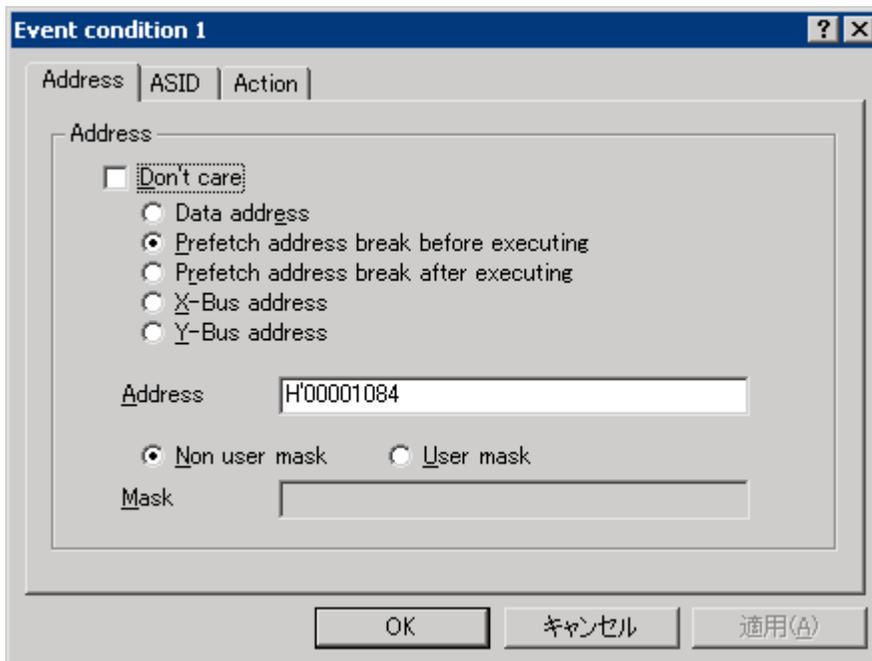


図 6.37 [Event condition 1]ダイアログボックス ([Address]ページ)

#### 【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。

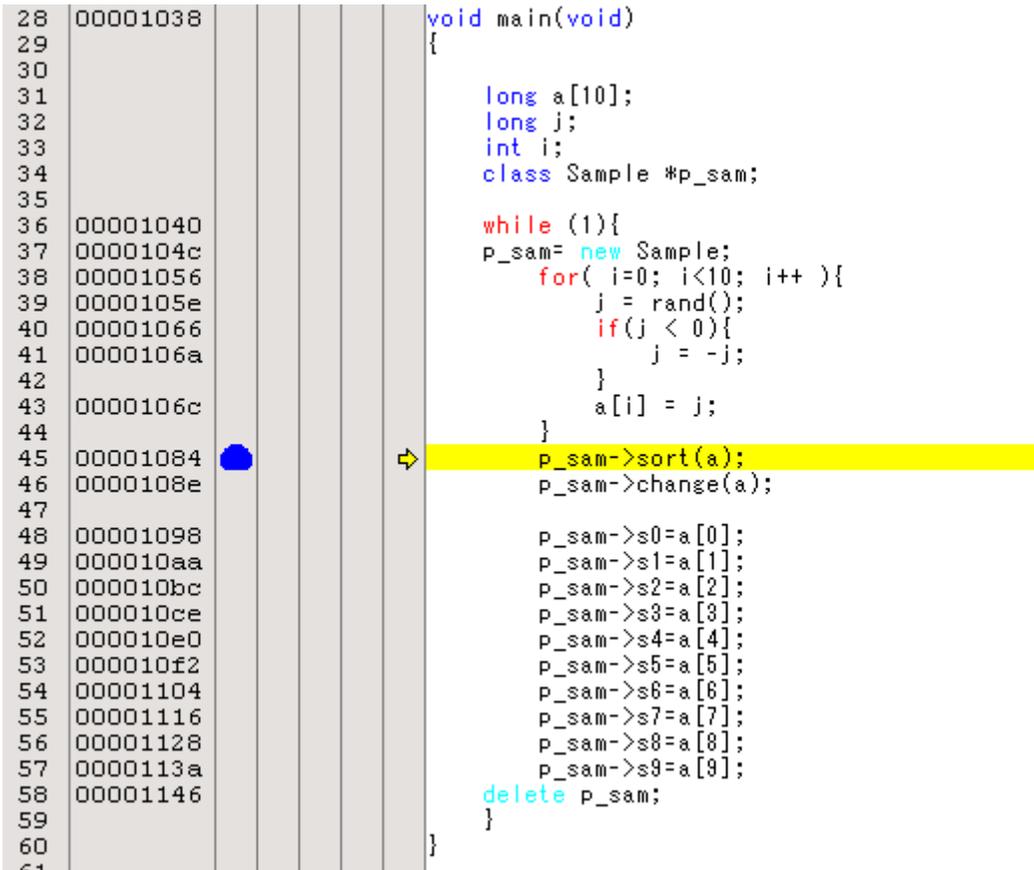
State 行の 1 ポイント目の表示が "Disable" から "Enable" に変わります。

Condition 行の 1 ポイント目の表示が "None" から "Address = H ' 00001084 pc" に変わります。

「6.8 レジスタ内容の変更」で設定したプログラムカウンタ、スタックポインタ (PC=H ' 00000800、R15=H ' 00010000) を[レジスタ]ウィンドウに設定して、[実行]ボタンをクリックしてください。

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 00001040 while (1){
37 0000104c p_sam= new Sample;
38 00001056     for( i=0; i<10; i++ ){
39 0000105e         j = rand();
40 00001066         if(j < 0){
41 0000106a             j = -j;
42
43 0000106c         }
44             a[i] = j;
45 00001084 p_sam->sort(a);
46 0000108e     p_sam->change(a);
47
48 00001098     p_sam->s0=a[0];
49 000010aa     p_sam->s1=a[1];
50 000010bc     p_sam->s2=a[2];
51 000010ce     p_sam->s3=a[3];
52 000010e0     p_sam->s4=a[4];
53 000010f2     p_sam->s5=a[5];
54 00001104     p_sam->s6=a[6];
55 00001116     p_sam->s7=a[7];
56 00001128     p_sam->s8=a[8];
57 0000113a     p_sam->s9=a[9];
58 00001146     delete p_sam;
59
60 }
61
```

図 6.38 実行停止時の[エディタ]ウィンドウ (Onchip Eventpoint チャンネル 1)

[ステータス]ウィンドウの表示内容は、以下のようになります。

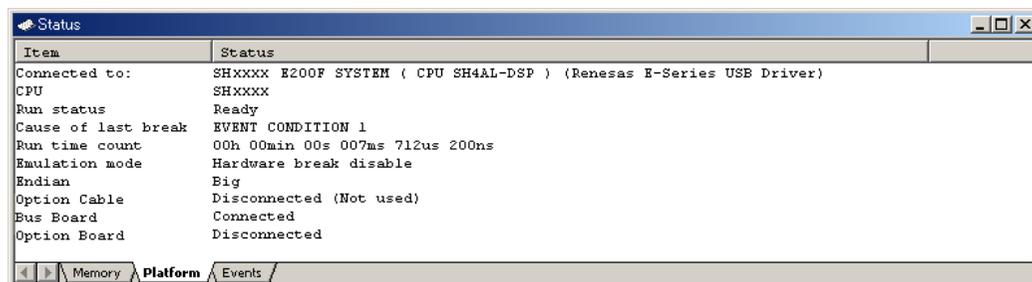


図 6.39 [ステータス]ウィンドウの表示内容 ([Onchip Event Ch1(IA\_OA)])

**【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。

## 6.18.2 Sequential Onchip Eventpoint によるブレーク

Onchip Eventpoint の組み合わせにより、シーケンシャルブレークができます。

Onchip Eventpoint の成立条件を次のように設定します。

1. Ch1 (IA\_OA)

アドレス H ' 00001084 の命令を実行する直前にブレーク条件が成立します。

2. Ch2 (IA\_OA\_DT\_CT)

アドレス H ' 0000106C の命令を実行する直前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、これらのイベントポイントをシーケンシャルとする設定を行います。

[Event]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。[Sequential setting]ダイアログボックスが開きます。

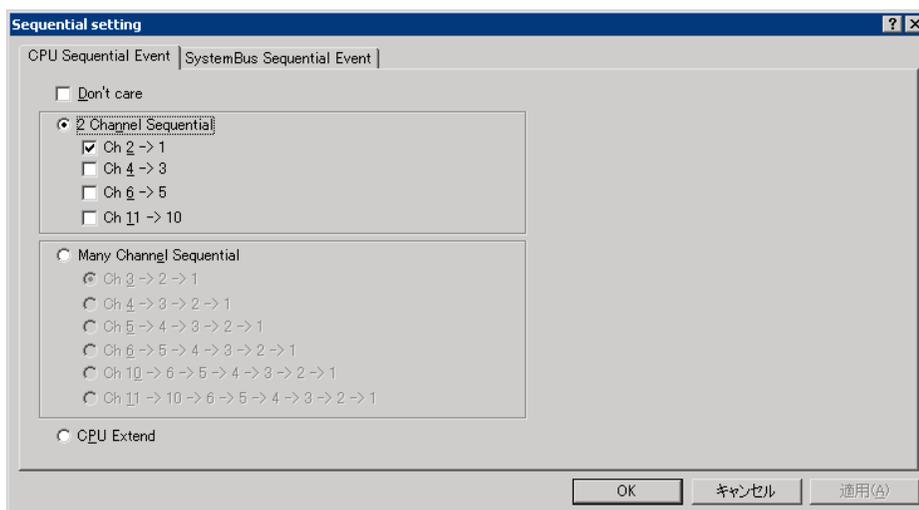


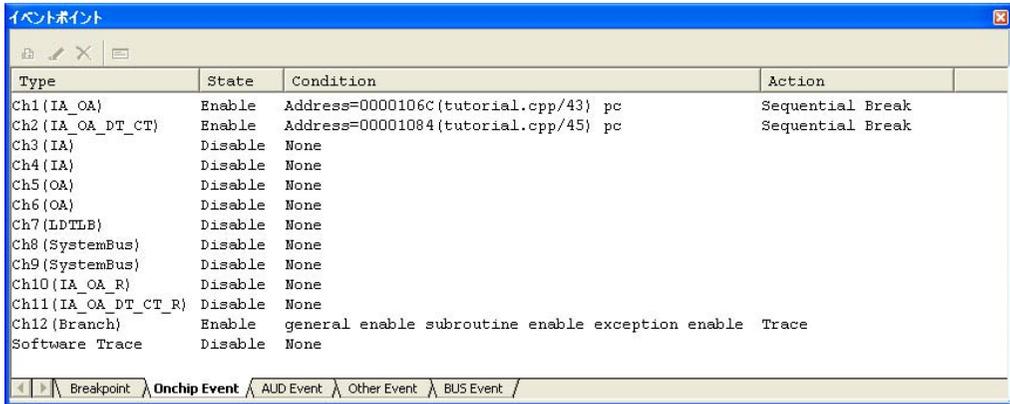
図 6.40 [Sequential setting]ダイアログボックス

### 【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。

[2 Channel Sequential]ラジオボタンを選択し、[Ch 2 -> 1]を選択した後[OK]ボタンをクリックしてください。

設定完了後、[Event]ウィンドウの状態は以下のようになっています。



Type	State	Condition	Action
Ch1 (IA_OA)	Enable	Address=0000106C(tutorial.cpp/43) pc	Sequential Break
Ch2 (IA_OA_DT_CT)	Enable	Address=00001084(tutorial.cpp/45) pc	Sequential Break
Ch3 (IA)	Disable	None	
Ch4 (IA)	Disable	None	
Ch5 (OA)	Disable	None	
Ch6 (OA)	Disable	None	
Ch7 (IDTLB)	Disable	None	
Ch8 (SystemBus)	Disable	None	
Ch9 (SystemBus)	Disable	None	
Ch10 (IA_OA_R)	Disable	None	
Ch11 (IA_OA_DT_CT_R)	Disable	None	
Ch12 (Branch)	Enable	general enable subroutine enable exception enable	Trace
Software Trace	Disable	None	

図 6.41 [Onchip Event]シート

**【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、「6.8 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'00010000) を[レジスタ]ウィンドウに設定して、[Go]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。

イベントチャンネル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 00001040 while (1){
37 0000104c p_sam= new Sample;
38 00001056 for( i=0; i<10; i++ ){
39 0000105e j = rand();
40 00001066 if(j < 0){
41 0000106a j = -j;
42
43 0000106c }
44 a[i] = j;
45 00001084 p_sam->sort(a);
46 0000108e p_sam->change(a);
47
48 p_sam->s0=a[0];
49 000010aa p_sam->s1=a[1];
50 000010bc p_sam->s2=a[2];
51 000010ce p_sam->s3=a[3];
52 000010e0 p_sam->s4=a[4];
53 000010f2 p_sam->s5=a[5];
54 00001104 p_sam->s6=a[6];
55 00001116 p_sam->s7=a[7];
56 00001128 p_sam->s8=a[8];
57 0000113a p_sam->s9=a[9];
58 00001146 delete p_sam;
59 }
60
61

```

図 6.42 実行停止時の[エディタ]ウィンドウ (シーケンシャルブレイク)

[ステータス]ウィンドウの表示内容は、以下のようになります。

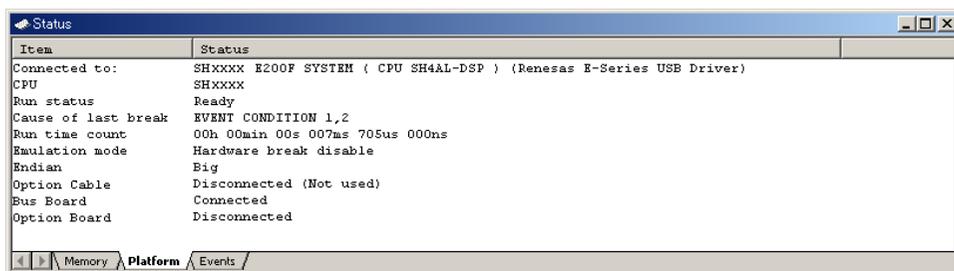


図 6.43 [ステータス]ウィンドウの表示内容 (シーケンシャルブレイク)

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

次に、[Event]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。[Sequential setting]ダイアログボックス (図 6.40) が開きます。

[Don't care]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。

### 6.19 トレース機能

E200F エミュレータには以下の5種類のトレース機能があります。

- 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ので、リアルタイムにトレースを取得することができます。

取得できる情報は以下です。

- (a) トレース情報の種類  
分岐情報、CPUからのメモリアクセス情報、Trace Rn 命令実行時のPC値とRn値
- (b) トレース取得アドレス値
- (c) データ値
- (d) ニモニク
- (e) オペランド
- (f) ソース行

**【留意事項】**

1. トレース取得できる分岐命令の数は、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。

- AUDトレース機能

デバイスのAUD端子をE200Fエミュレータに接続している場合に便利な、大容量のトレース機能です。トレース取得できる情報の数は、分岐元/分岐先の組を1個とすると最大262,144個です。

取得できる情報は以下です。

- (a) トレース情報の種類  
分岐情報、CPUからのメモリアクセス情報、Trace Rn命令実行時のPC値とRn値
- (b) トレース取得アドレス値
- (c) データ値
- (d) 外部プローブ端子状態
- (e) タイムスタンプ値
- (f) ニモニク
- (g) オペランド
- (h) ソース行

## 【留意事項】

1. 製品によっては、AUD 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。
2. 製品によっては、AUD トレース機能で取得できるトレース情報の種類が異なります。各製品の仕様、取得分岐数については、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、AUD トレース表示数削減のため、複数回ループする場合においては IP のみのカウントアップします。

## • 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。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。

デバイスのAUD端子をE200Fエミュレータに接続していない場合に便利な、大容量のトレース機能です。

## 【留意事項】

製品によっては、メモリ出力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。

## • 外部バストレース機能

デバイスの外部バス端子をE200Fエミュレータに接続している場合に便利な、大容量のトレース機能です。

外部バストレース機能では、バスサイクル毎に最大262,144サイクル分の情報を取得できます。

取得できる情報は以下です。

- (a) 外部バスアドレス値
- (b) 外部バスデータ値
- (c) 割り込み信号状態
- (d) タイムスタンプ値
- (e) ニモニック
- (f) オペランド

## 【留意事項】

トレース取得できる情報は、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。

## • I/Oアナライザ機能によるトレース機能

E200Fエミュレータでは、I/Oアナライザ機能の一つとしてトレース取得ができる場合があります。

### 6.19.1 [Internal/AUD/Usermemory trace]ウィンドウの表示方法

[表示]メニューの[コード]サブメニューから[トレース]を選択してください。[Trace Window Type]ダイアログボックスが表示されます。



図 6.44 [Trace Window Type]ダイアログボックス

[Internal/AUD/Usermemory trace]を選択し[OK]ボタンをクリックすると、[Internal/AUD/Usermemory trace]ウィンドウを表示します。

## (1) 内蔵トレース機能

内蔵トレースの取得方法を以下に説明します。

## (a) トレース取得モードを設定する

マウスの右ボタンで[Internal/AUD/Usermemory trace]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[Acquisition...]を選択してください。[Internal/AUD/Usermemory acquisition]ダイアログボックスが表示されます。

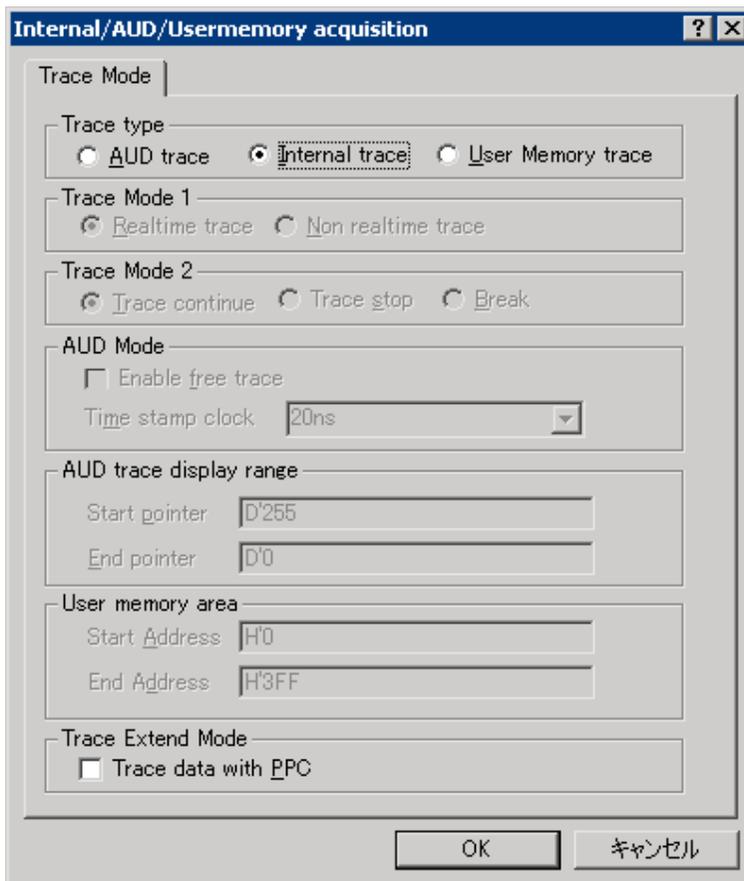


図 6.45 [Internal/AUD/Usermemory acquisition]ダイアログボックス

[Trace type]に[Internal trace]を選択し[OK]ボタンをクリックしてください。

(b) トレース取得条件を設定する

分岐元/分岐先情報の取得を例に説明します。

他のトレース情報を取得するための設定は、「5.6 イベントポイントを使用する」を参照してください。

[Event]ウィンドウの[Onchip Event]シートで[Ch12 (Branch)]を選択してダブルクリックしてください。[Event condition 12]ダイアログボックスが開きます。

取得する分岐情報のタイプを選択して[Action]ページで[Acquire trace]を選択した後に[OK]ボタンをクリックしてください。

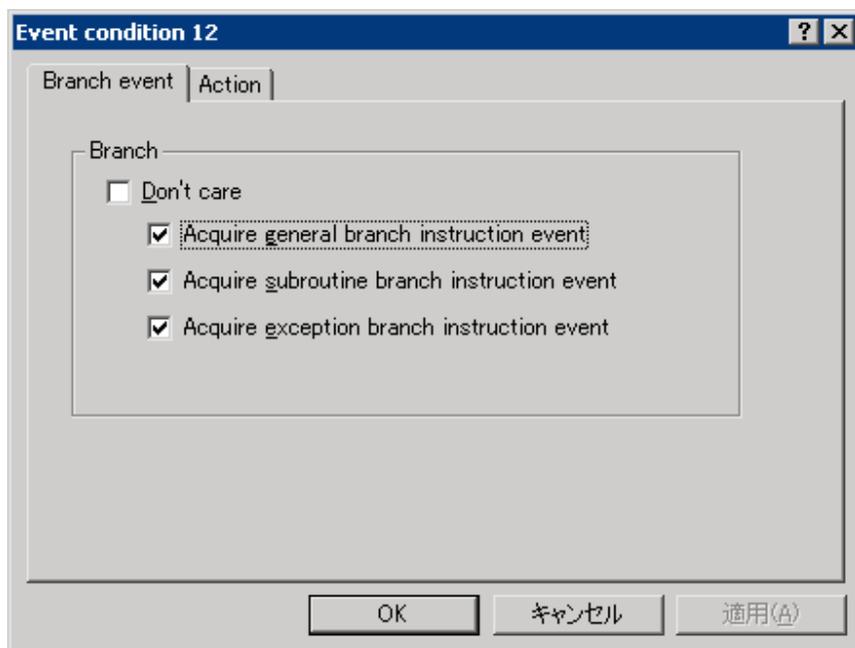


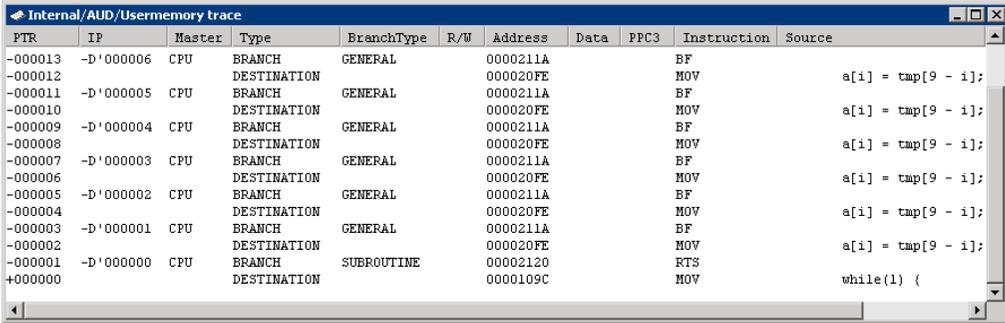
図 6.46 [Event condition 12]ダイアログボックス

【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。

## (c) トレース結果を表示する

「6.17.1 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[Internal/AUD/Usermemory trace]ウィンドウに内蔵トレース結果を表示します。



PTR	IP	Master	Type	BranchType	R/W	Address	Data	PPC3	Instruction	Source
-000013	-D'000006	CPU	BRANCH	GENERAL		0000211A			BF	
-000012			DESTINATION			000020FE			MOV	a[i] = tmp[9 - i];
-000011	-D'000005	CPU	BRANCH	GENERAL		0000211A			BF	
-000010			DESTINATION			000020FE			MOV	a[i] = tmp[9 - i];
-000009	-D'000004	CPU	BRANCH	GENERAL		0000211A			BF	
-000008			DESTINATION			000020FE			MOV	a[i] = tmp[9 - i];
-000007	-D'000003	CPU	BRANCH	GENERAL		0000211A			BF	
-000006			DESTINATION			000020FE			MOV	a[i] = tmp[9 - i];
-000005	-D'000002	CPU	BRANCH	GENERAL		0000211A			BF	
-000004			DESTINATION			000020FE			MOV	a[i] = tmp[9 - i];
-000003	-D'000001	CPU	BRANCH	GENERAL		0000211A			BF	
-000002			DESTINATION			000020FE			MOV	a[i] = tmp[9 - i];
-000001	-D'000000	CPU	BRANCH	SUBROUTINE		00002120			RTS	
+000000			DESTINATION			0000109C			MOV	while(1) {

図 6.47 [Internal/AUD/Usermemory trace]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】

トレース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。

### (2) AUD トレース機能

デバイスの AUD 端子を E200F 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 端子からチップ内の情報を出力するため、Onchip Eventpoint を設定する必要があります。

ここでは、アドレス H' FFC0 ~ H' FFE4 間 (配列 a) のメモリアクセス情報取得を例として説明します。

#### (a) Onchip Eventpoint を設定する

AUD トレース取得条件が AUD 端子から出力されるチップ実行情報に設定できるように、「5.6 イベントポイントを使用する」を参照して、Onchip Eventpoint を設定してください。

ここでは、オンチップイベントチャンネル Ch5 (OA) にウィンドウトレース H' 0 ~ H' 10000 を設定します。

[Event]ウィンドウの[Onchip Event]シートで[Ch5 (OA)]を選択してダブルクリックしてください。[Event condition 5]ダイアログボックスが開きます。[Window address]ページで、AUD 端子から出力するメモリアクセス情報の範囲を指定します。

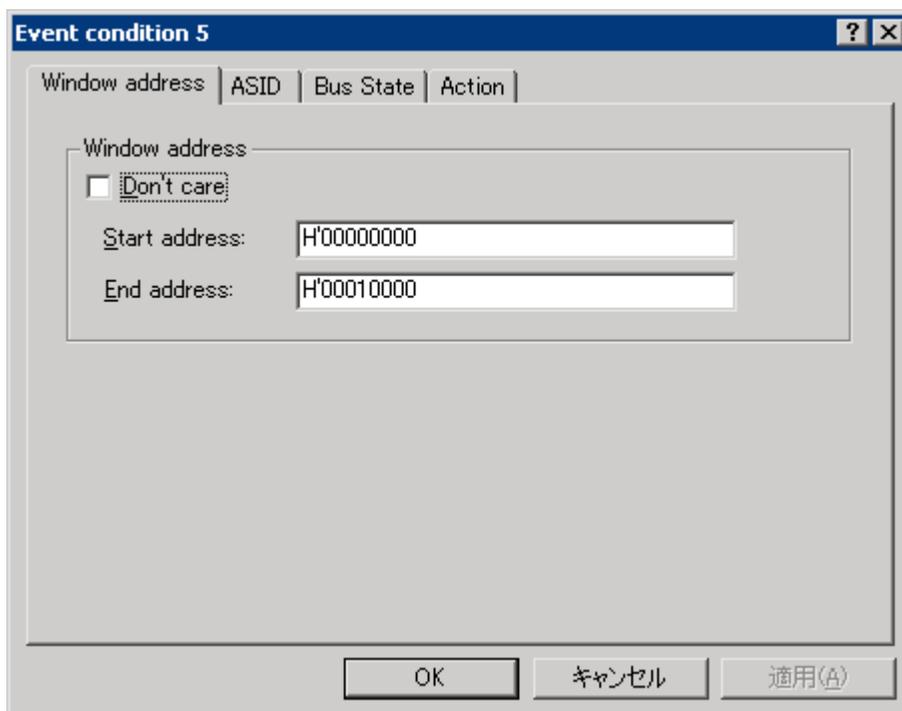


図 6.48 Onchip Eventpoint 設定 ([Ch5 (OA)チャンネル]

[Start address]に H' 0、[End address]に H' 10000 を入力します。

[Action]ページで[Acquire trace]を選択した後に[OK]ボタンをクリックしてください。

(b) トレース取得モードを設定する

[Internal/AUD/Usermemory trace]ウィンドウを表示してください。

マウスの右ボタンで[Internal/AUD/Usermemory trace]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[Acquisition...]を選択してください。[Internal/AUD/Usermemory acquisition]ダイアログボックスが表示されます。

AUD トレースの取得条件を設定します。

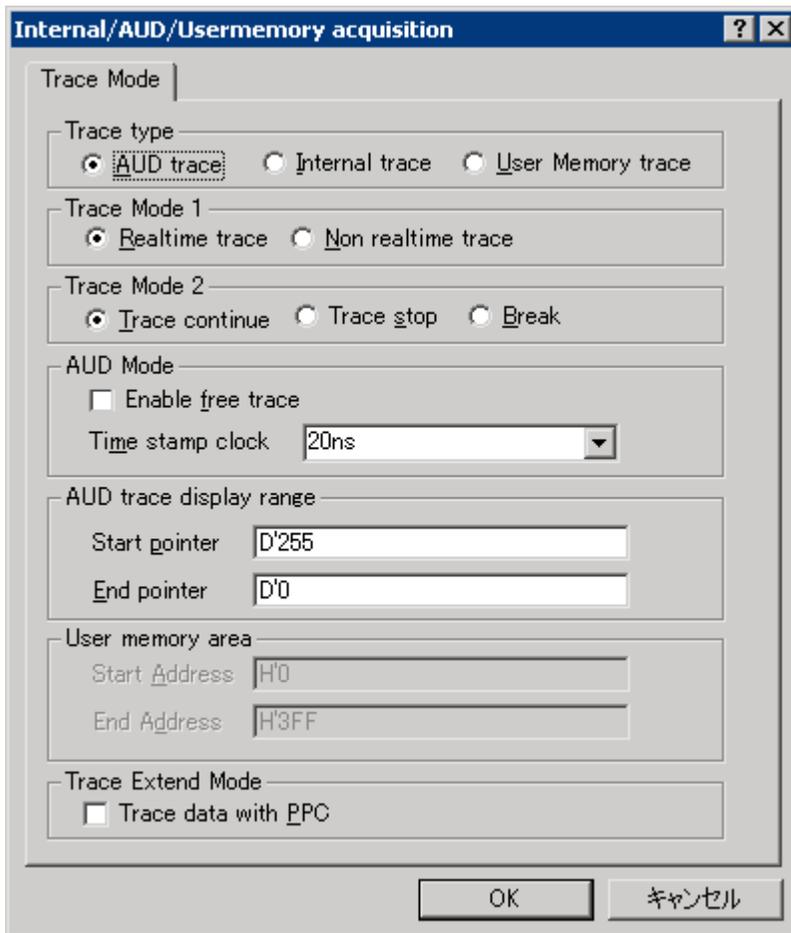


図 6.49 [Internal/AUD/Usermemory acquisition]ダイアログボックス



## (c) トレース取得条件を設定する

メモリアクセス情報の取得を例に説明します。

他のトレース情報を取得するための設定は、「5.6 イベントポイントを使用する」を参照してください。

[Event]ウィンドウの[AUD Event]シートで[Ch2 (Delay)]を選択してダブルクリックしてください。[Ch2 (Delay)]ダイアログボックスが開きます。

[General]ページで、取得するトレース情報のタイプを選択できます。

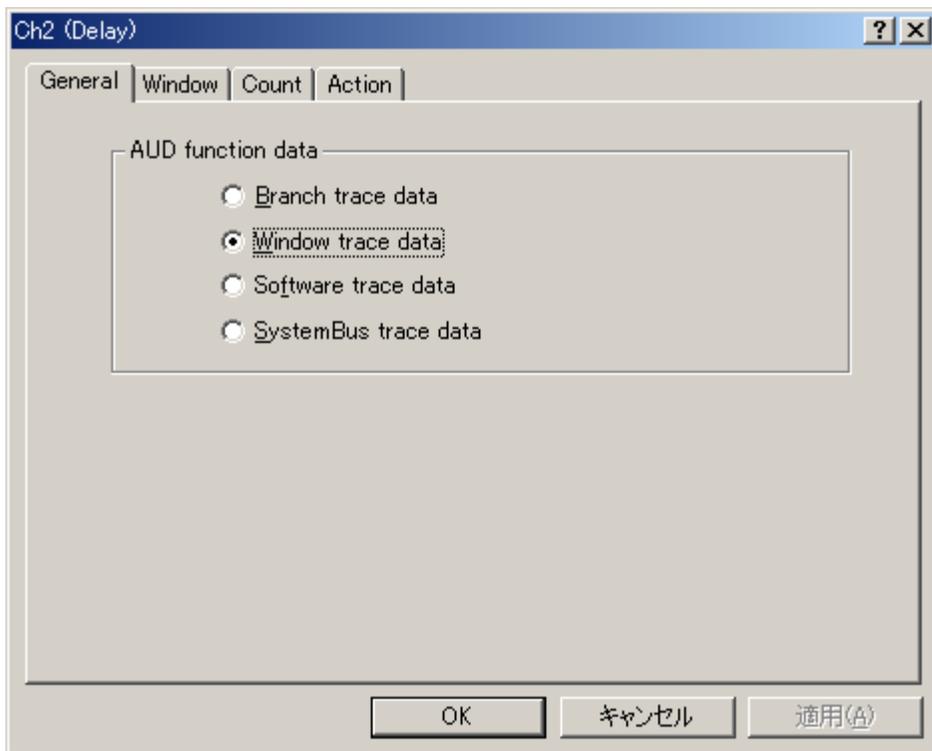


図 6.50 [Ch2 (Delay)]ダイアログボックス ([General]ページ)

[Window trace data]を選択し[Window]ページをクリックしてください。

[Window]ページでメモリアクセス範囲条件を設定できます。

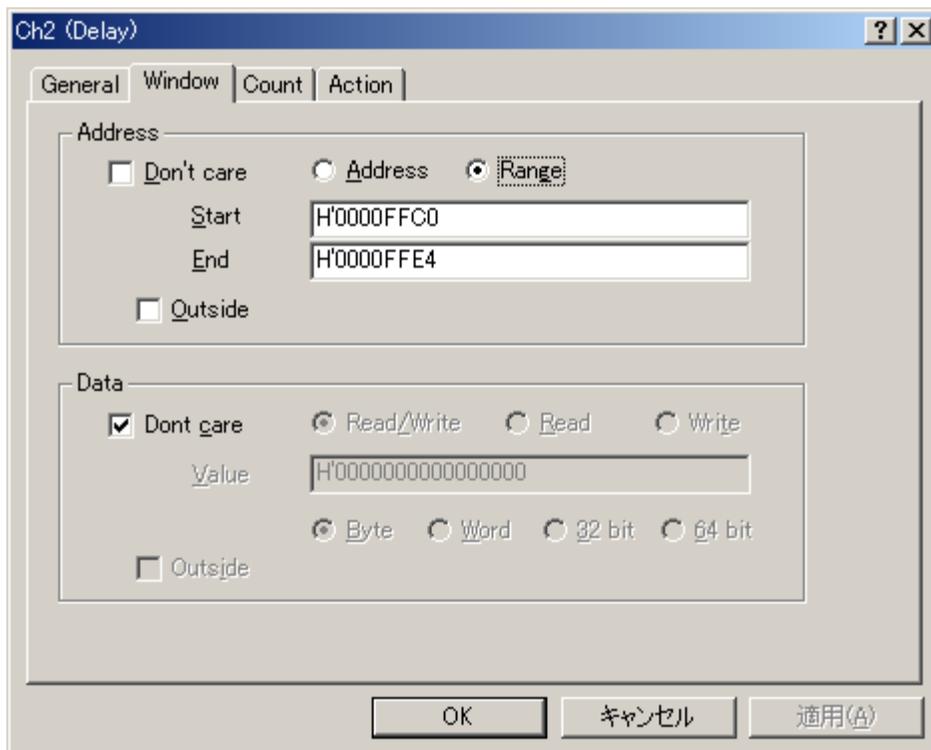


図 6.51 [Ch2 (Delay)]ダイアログボックス ([Window]ページ)

アドレス範囲間のトレース取得として、[Range]を選択します。

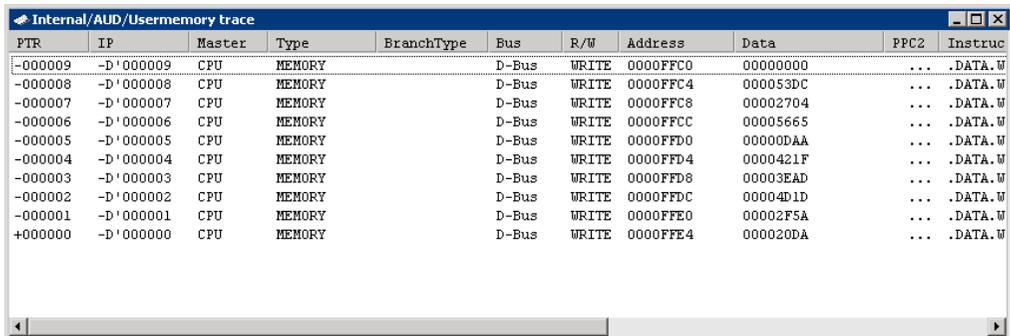
[Start]に H ' FFC0、[End]に H ' FFE4 を入力します。

アドレス H ' FFC0 ~ H ' FFE4 間 (配列 a) のメモリアクセス情報を取得します。

[Action]ページで[Trace get]を選択した後[OK]ボタンをクリックしてください。

(d) トレース結果を表示する

「6.17.1 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[Internal/AUD/Usermemory trace]ウィンドウにトレース結果を表示します。



PTR	IP	Master	Type	BranchType	Bus	R/W	Address	Data	PPC2	Instruc
-000009	-D'000009	CPU	MEMORY		D-Bus	WRITE	0000FFC0	00000000	...	.DATA.W
-000008	-D'000008	CPU	MEMORY		D-Bus	WRITE	0000FFC4	000053DC	...	.DATA.W
-000007	-D'000007	CPU	MEMORY		D-Bus	WRITE	0000FFC8	00002704	...	.DATA.W
-000006	-D'000006	CPU	MEMORY		D-Bus	WRITE	0000FFCC	00005665	...	.DATA.W
-000005	-D'000005	CPU	MEMORY		D-Bus	WRITE	0000FFD0	00000DAA	...	.DATA.W
-000004	-D'000004	CPU	MEMORY		D-Bus	WRITE	0000FFD4	0000421F	...	.DATA.W
-000003	-D'000003	CPU	MEMORY		D-Bus	WRITE	0000FFD8	00003EAD	...	.DATA.W
-000002	-D'000002	CPU	MEMORY		D-Bus	WRITE	0000FFDC	00004D1D	...	.DATA.W
-000001	-D'000001	CPU	MEMORY		D-Bus	WRITE	0000FFE0	00002F5A	...	.DATA.W
+000000	-D'000000	CPU	MEMORY		D-Bus	WRITE	0000FFE4	000020DA	...	.DATA.W

図 6.52 [Internal/AUD/Usermemory trace]ウィンドウ (表示例)

[Watch]ウィンドウを開いて配列 a の値を確認できます。(「6.13 変数の参照」を参照してください)

### (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取得方法を以下に説明します。

#### (a) トレース取得モードを設定する

[Internal/AUD/Usermemory trace]ウィンドウを表示してください。

マウスの右ボタンで[Internal/AUD/Usermemory trace]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[Acquisition...]を選択してください。[Internal/AUD/Usermemory acquisition]ダイアログボックスが表示されます。

メモリ出力トレースの取得モードを設定します。

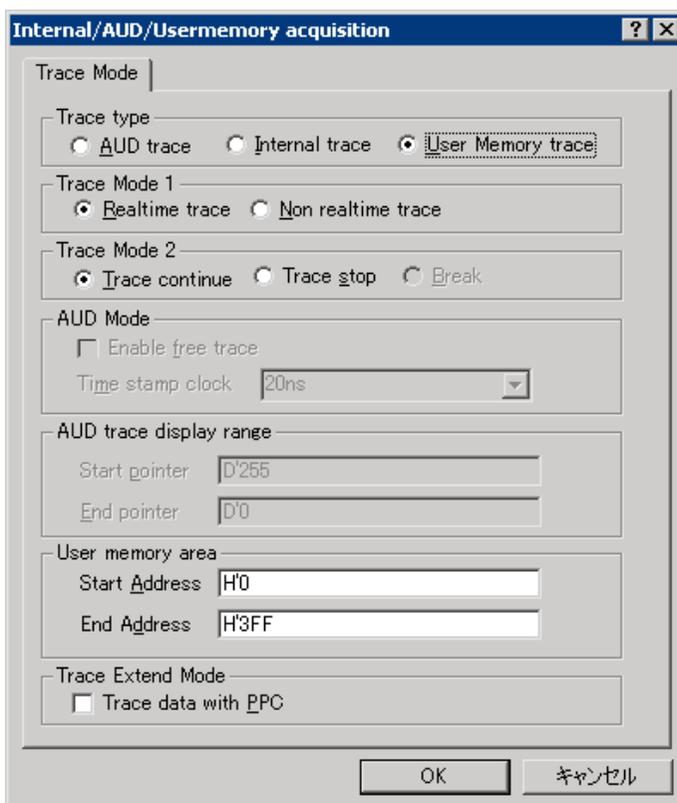


図 6.53 [Internal/AUD/Usermemory acquisition]ダイアログボックス

[Trace type]	トレース機能の種類を選択します。 AUD trace : AUD トレース機能を選択します。 Internal trace : 内蔵トレース機能を選択します。 User Memory trace : ユーザメモリ出力トレース機能を選択します。
[Trace Mode 1]	トレース情報が連続して発生した場合の動作モードを選択します。 Realtime trace : 一部のトレース情報を出力しません。 Non realtime trace : トレース情報が出力されるまで CPU が待たされます。
[Trace Mode 2]	トレースバッファがフルになったときの動作を設定します。 Trace continue : 古いトレース情報に上書きして、常に最新の情報を取得します。 Trace stop : 以後のトレース情報を取得しません。
[User memory area]	ユーザメモリ出力トレース機能を使用するときのトレース表示範囲を設定します。 Start Address : 表示範囲の先頭アドレスを設定します。 End Address : 表示範囲の終了アドレスを設定します。
[Trace Extend Mode]	トレース取得項目として、内蔵パフォーマンスカウンタ Ch3, Ch4 を含めるかを設定します。 Trace data with PPC : チェックすると、内蔵パフォーマンスカウンタ Ch3, Ch4 を含めます。

**【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。

[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。

## 6 チュートリアル

### (b) トレース取得条件を設定する

メモリ出力トレース機能の取得条件設定は、内蔵トレース機能の取得条件設定と同じです。「6.19.1 (1) 内蔵トレース機能」を参照してください。

### (c) トレース結果を表示する

「6.17.1 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[Internal/AUD/Usermemory trace]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。

P/P	IP	Rst/c	Type	BranchType	Bus	R/W	Address	Data	FPC3	FPC4	Instruction	Source	Label
-00001	-0'000015	CPF	BRANCH	GENERAL			000020B	...	...	...	BF/S	#0'200010	...
-00000			DESTINATION	GENERAL			000020B	...	...	...	MOV	#5, #0	...
-000029	-0'000014	CPF	BRANCH	GENERAL			000020C	...	...	...	BF/S	#0'200010	...
+000028			DESTINATION	GENERAL			000020B	...	...	...	MOV	#5, #0	...
-000017	-0'000013	CPF	BRANCH	GENERAL			000020C	...	...	...	BF/S	#0'200010	...
-000026			DESTINATION	GENERAL			000020B	...	...	...	MOV	#5, #0	...
-000025	-0'000012	CPF	BRANCH	GENERAL			000020C	...	...	...	BF/S	#0'200010	...
-000024			DESTINATION	GENERAL			000020B	...	...	...	MOV	#5, #0	...
+000023	-0'000011	CPF	BRANCH	GENERAL			000020C	...	...	...	BF/S	#0'200010	...
+000012			DESTINATION	GENERAL			000020B	...	...	...	MOV	#5, #0	...
+000021	-0'000010	CPF	BRANCH	GENERAL			000020C	...	...	...	BF/S	#0'200010	...
-000020			DESTINATION	GENERAL			000020B	...	...	...	MOV	#5, #0	...
-000019	-0'000009	CPF	BRANCH	GENERAL			000020E	...	...	...	BF/S	#0'200010	...
-000018			DESTINATION	GENERAL			000020D	...	...	...	MOV	#0'09, #2	...
-000017	-0'000008	CPF	BRANCH	GENERAL			000020E	...	...	...	BF/S	#0'200010	...
-000016			DESTINATION	GENERAL			000020D	...	...	...	MOV	#0'09, #2	...
+000015	-0'000007	CPF	BRANCH	GENERAL			000020E	...	...	...	BF/S	#0'200010	...
-000014			DESTINATION	GENERAL			000020D	...	...	...	MOV	#0'09, #2	...
-000013	-0'000006	CPF	BRANCH	GENERAL			000020E	...	...	...	BF/S	#0'200010	...
-000012			DESTINATION	GENERAL			000020D	...	...	...	MOV	#0'09, #2	...
-000011	-0'000005	CPF	BRANCH	GENERAL			000020E	...	...	...	BF/S	#0'200010	...
-000010			DESTINATION	GENERAL			000020D	...	...	...	MOV	#0'09, #2	...
+000009	-0'000004	CPF	BRANCH	GENERAL			000020E	...	...	...	BF/S	#0'200010	...
-000008			DESTINATION	GENERAL			000020D	...	...	...	MOV	#0'09, #2	...
-000007	-0'000003	CPF	BRANCH	GENERAL			000020E	...	...	...	BF/S	#0'200010	...
-000006			DESTINATION	GENERAL			000020D	...	...	...	MOV	#0'09, #2	...
-000005	-0'000002	CPF	BRANCH	GENERAL			000020E	...	...	...	BF/S	#0'200010	...
+000004			DESTINATION	GENERAL			000020D	...	...	...	MOV	#0'09, #2	...
-000003	-0'000001	CPF	BRANCH	GENERAL			000020E	...	...	...	BF/S	#0'200010	...
-000002			DESTINATION	GENERAL			000020D	...	...	...	MOV	#0'09, #2	...
+000001	-0'000000	CPF	BRANCH	SUBROUTINE			000020E	...	...	...	DTS		...
+000000			DESTINATION				00001076	...	...	...	MOV.L	#0'15, #2	...

図 6.54 [Internal/AUD/Usermemory trace]ウィンドウ (表示例)

### 6.19.2 [BUS/MFI trace]ウィンドウの表示方法

[表示]メニューの[コード]サブメニューから[トレース]を選択してください。[Trace Window Type]ダイアログボックスが表示されます。



図 6.55 [トレースウィンドウの選択]ダイアログボックス

[BUS/MFI trace]を選択し[OK]ボタンをクリックすると、[BUS/MFI trace]ウィンドウを表示します。

### (1) 外部バストレース機能

デバイスの外部バス端子を E200F 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。

外部バストレースの取得方法を以下に説明します。

#### (a) バストレースに関する基本設定を行う

「2.7.2 外部バストレース/ブレーク機能と外部バスパフォーマンス機能」を参考に、バス用端子のマルチプレクス状態や接続されているメモリについて設定してください。

#### (b) トレース取得モードを設定する

マウスの右ボタンで[BUS trace]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[Acquisition...]を選択してください。[BUS acquisition]ダイアログボックスが表示されます。

外部バストレースの取得モードを以下のように設定します。

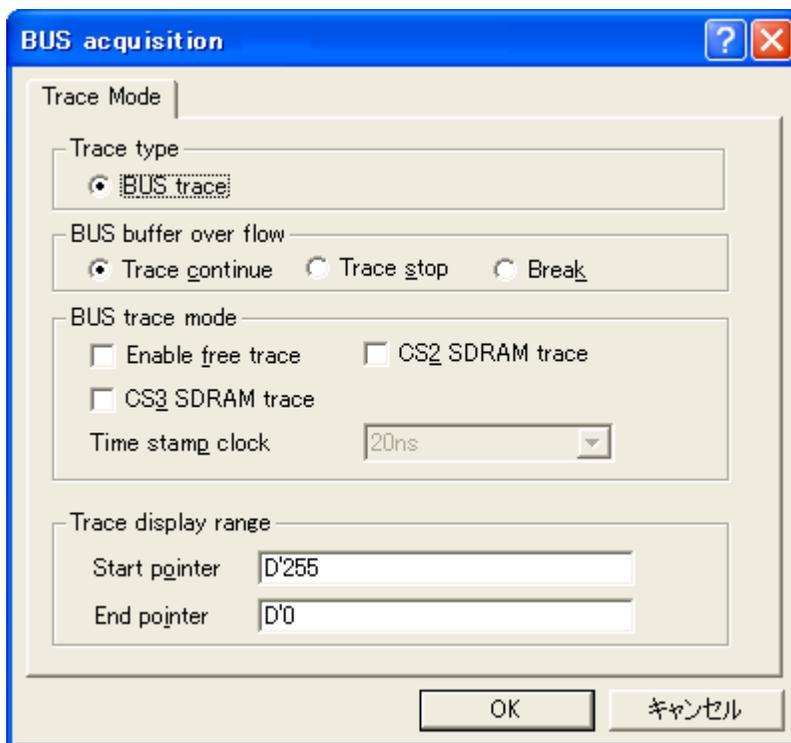


図 6.56 [BUS acquisition]ダイアログボックス

## (c) トレース取得条件を設定する

アドレス条件の設定を例に説明します。

他のトレース条件設定は、「5.6 イベントポイントを使用する」を参照してください。

[Event]ウィンドウの[Bus Event]シートで[Ch3 (Normal)]を選択してダブルクリックしてください。[Ch3 (Normal)]ダイアログボックスが開きます。

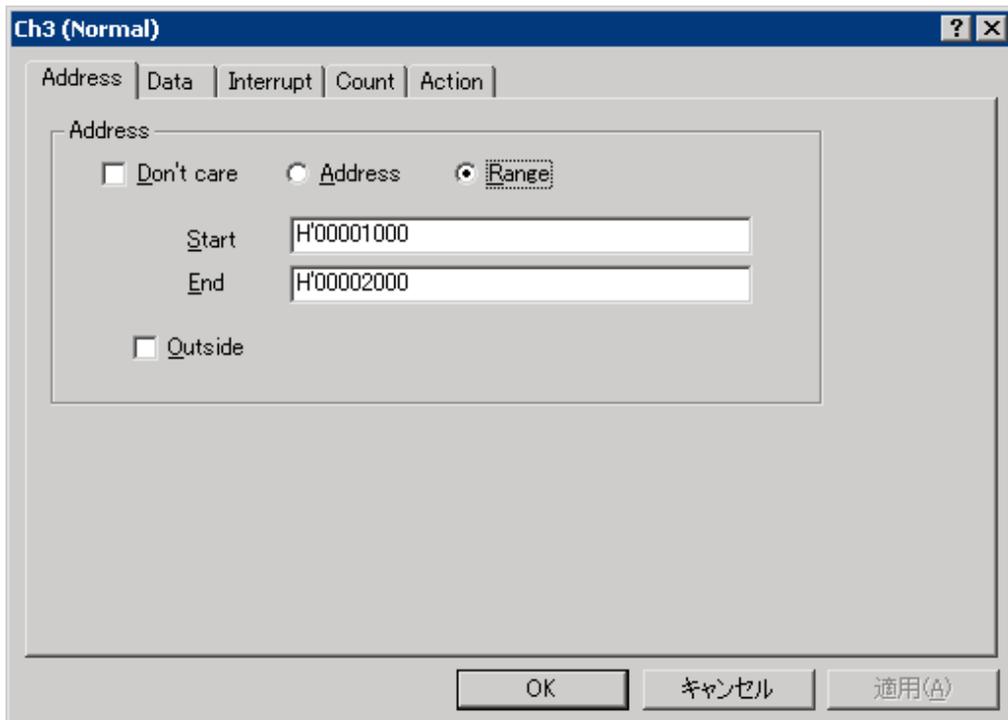


図 6.57 [Ch3 (Normal)]ダイアログボックス ([Address]ページ)

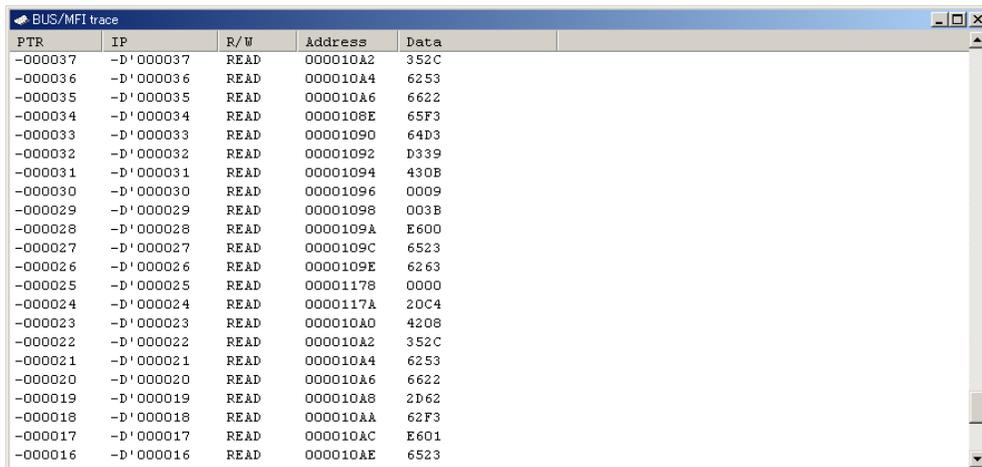
[Range]を選択した後、[Start]に H ' 1000 を入力し、[End]に H ' 2000 を入力してください。

[Action]ページで[Trace get]を選択した後[OK]ボタンをクリックしてください。

外部メモリ H ' 1000 ~ H ' 2000 にアクセスしたときのトレース情報を取得します。

## (d) トレース結果を表示する

「6.17.1 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 [BUS/MFI trace] ウィンドウにトレース結果を表示します。



PTR	IP	R/W	Address	Data
-000037	-D'000037	READ	000010A2	352C
-000036	-D'000036	READ	000010A4	6253
-000035	-D'000035	READ	000010A6	6622
-000034	-D'000034	READ	0000108E	65F3
-000033	-D'000033	READ	00001090	64D3
-000032	-D'000032	READ	00001092	D339
-000031	-D'000031	READ	00001094	430B
-000030	-D'000030	READ	00001096	0009
-000029	-D'000029	READ	00001098	003B
-000028	-D'000028	READ	0000109A	E600
-000027	-D'000027	READ	0000109C	6523
-000026	-D'000026	READ	0000109E	6263
-000025	-D'000025	READ	00001178	0000
-000024	-D'000024	READ	0000117A	20C4
-000023	-D'000023	READ	000010A0	4208
-000022	-D'000022	READ	000010A2	352C
-000021	-D'000021	READ	000010A4	6253
-000020	-D'000020	READ	000010A6	6622
-000019	-D'000019	READ	000010A8	2D62
-000018	-D'000018	READ	000010AA	62F3
-000017	-D'000017	READ	000010AC	E601
-000016	-D'000016	READ	000010AE	6523

図 6.58 [BUS/MFI trace] ウィンドウ

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

## 【注意事項】

トレース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 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

E200F 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 E200F 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、E200F 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 E200F 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。

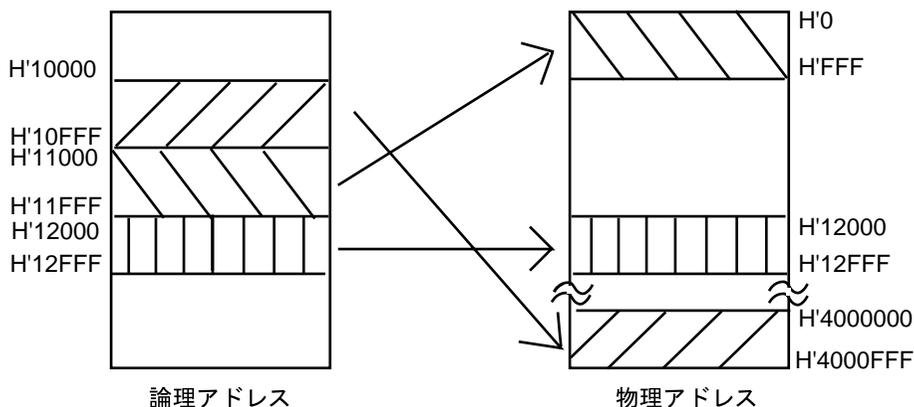


図 6.59 VP\_MAP テーブルによるアドレス変換

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、E200F 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 範囲外の時、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 テーブル範囲外るとき TLB エラーになります。

表 6.2 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。

## 6.21 スタックトレース機能

E200Fエミュレータでは、スタック情報を用いて、現在のPCがある関数がどの関数からコールされているかを表示します。

### 【留意事項】

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

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。

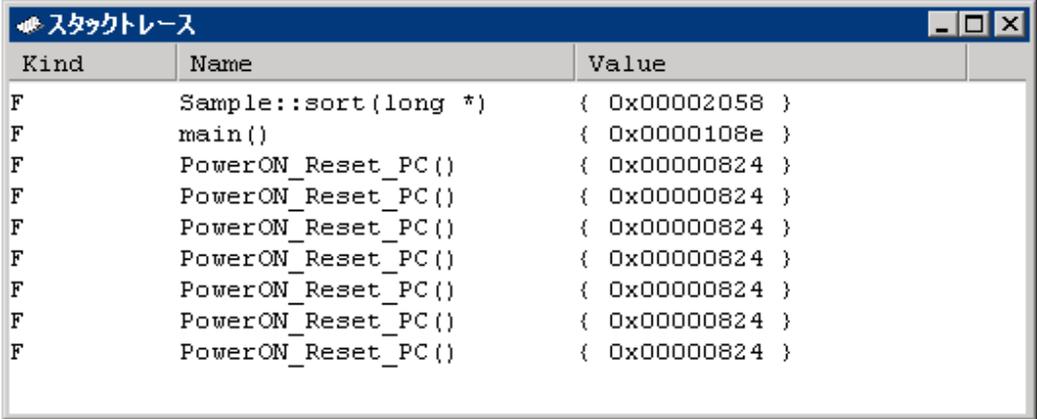
sort関数内の行の[S/W ブレークポイント]カラムをダブルクリックして、PCブレークポイントを設定してください。

26	00002032					void Sample::sort(long *a)
27						{
28						long t;
29						int i, j, k, gap;
30						
31	00002038					gap = 5;
32	0000203a					while( gap > 0 ){
33	0000203e					for( k=0; k<gap; k++){
34	00002044					for( i=k+gap; i<10; i=i+gap ){
35	00002050					for(j=i-gap; j>=k; j=j-gap){
36	00002058					if(a[j]>a[j+gap]){
37	00002070					t = a[j];
38	0000207a					a[j] = a[j+gap];
39	0000208e					a[j+gap] = t;
40						}
41						else
42						break;
43						}
44						}
45						}
46	000020ae					gap = gap/2;
47						}
48						
49						void Sample::change(long *a)
50	000020c4					{
51						long tmp[10];
52						

図 6.60 [エディタ]ウィンドウ (PCブレークポイントの設定)

「6.8 レジスタ内容の変更」で設定したプログラムカウンタ、スタックポインタ (PC=H'00000800、R15=H'00010000) を[レジスタ]ウィンドウに設定して、[実行]ボタンをクリックしてください。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。



Kind	Name	Value
F	Sample::sort(long *)	{ 0x00002058 }
F	main()	{ 0x0000108e }
F	PowerON_Reset_PC()	{ 0x00000824 }
F	PowerON_Reset_PC()	{ 0x00000824 }
F	PowerON_Reset_PC()	{ 0x00000824 }
F	PowerON_Reset_PC()	{ 0x00000824 }
F	PowerON_Reset_PC()	{ 0x00000824 }
F	PowerON_Reset_PC()	{ 0x00000824 }
F	PowerON_Reset_PC()	{ 0x00000824 }
F	PowerON_Reset_PC()	{ 0x00000824 }

図 6.61 [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[Edit]カラムを再度ダブルクリックして、PC ブレークポイントを解除します。

【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 フラッシュメモリへのダウンロード機能

E200F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】

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。

#### (1) ライト/消去モジュールと E200F 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、E200F エミュレータファームウェアから分岐します。

E200F 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 E200F 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 6.3 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 R7(L):ベリファイオプション 0=ベリファイなし 1=ベリファイあり 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 ベリファイエラー=' BT '
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。

## (2) 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、[オプション]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。

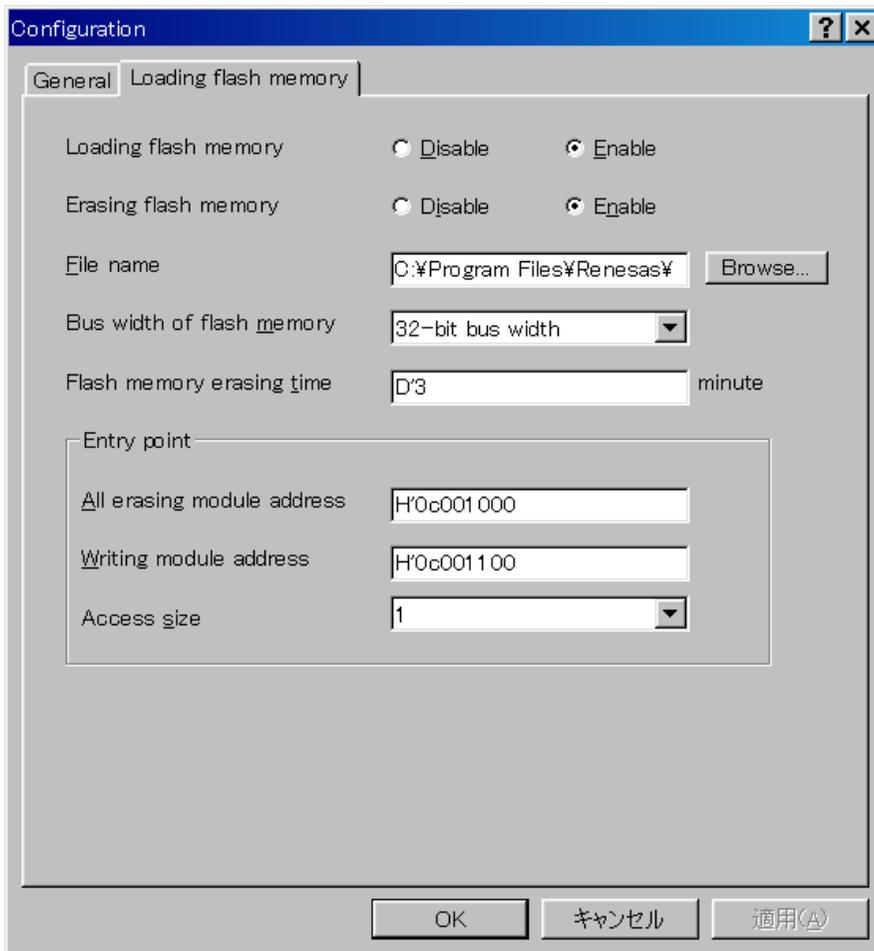


図 6.62 [Loading flash memory]ページ

## 6 チュートリアル

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

表 6.4 [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。

【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。

### (3) 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。

## (4) フラッシュメモリダウンロード例

以下に示すような結線で接続されている、(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 6.5 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

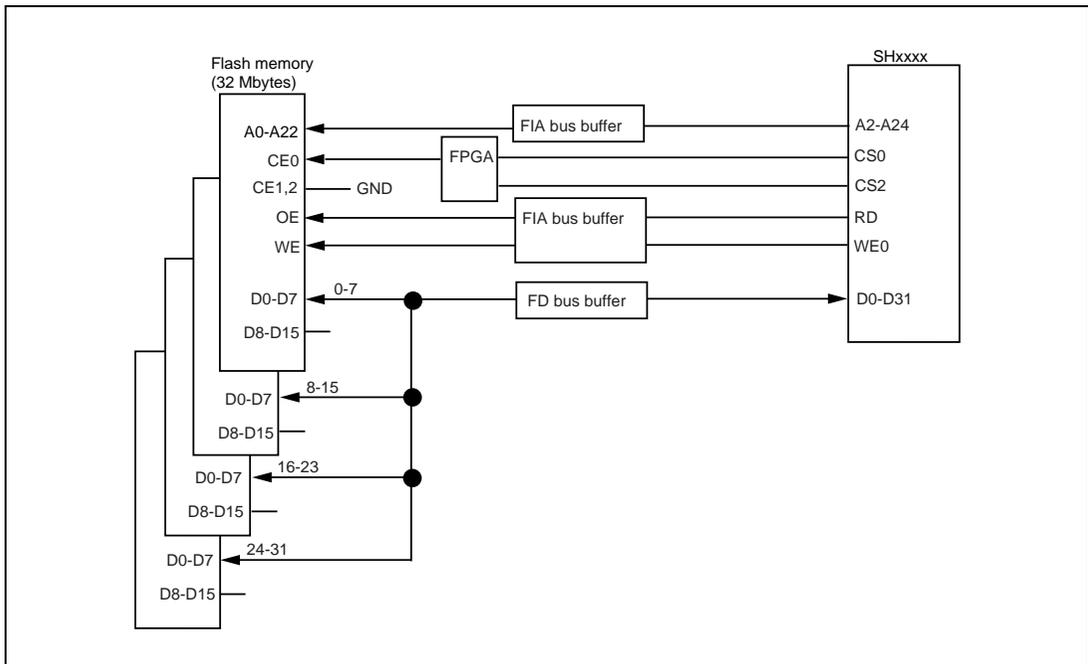


図 6.63 フラッシュメモリ結線図

表 6.6 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

SDRAM を使用するため、バスコントローラを設定します。

[Configuration]ダイアログボックスの[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。

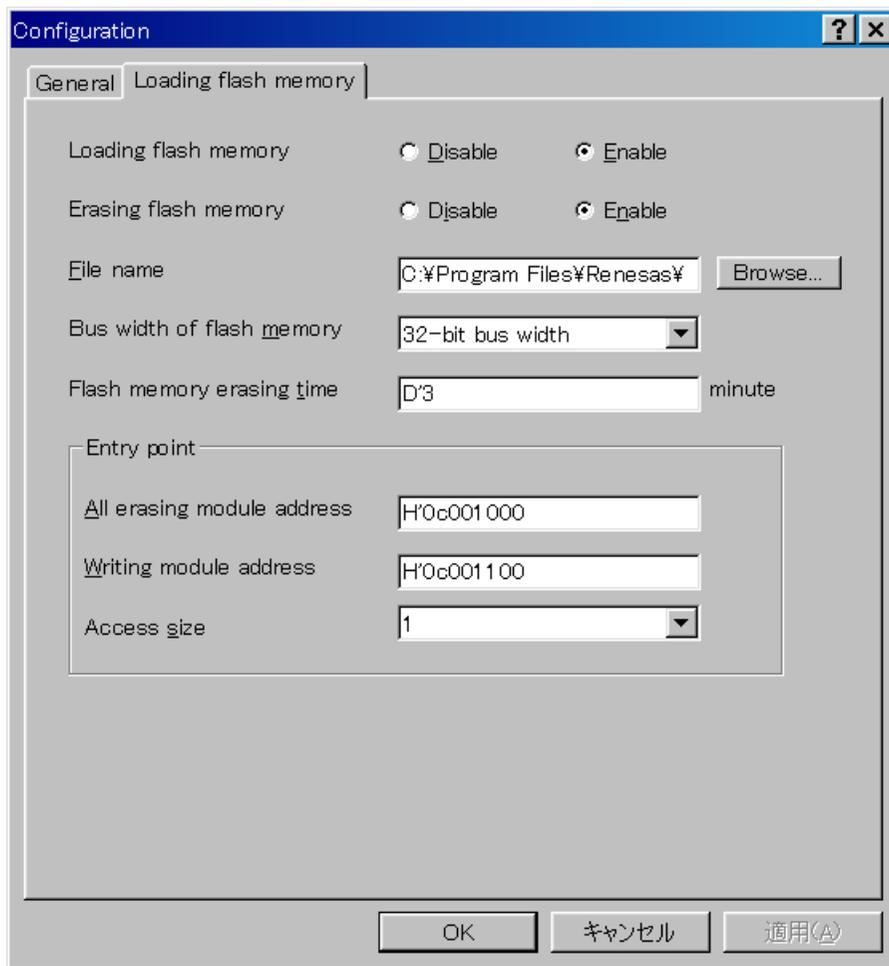


図 6.64 [Loading flash memory]ページ

- 【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、ペ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 分かかります（サンプル例の場合）。

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。

## 6.23 さてつぎは？

このチュートリアルでは、E200F エミュレータのいくつかの主な特徴と、High-performance Embedded Workshop の使い方を紹介しました。

E200F エミュレータで提供されるエミュレーション機能を使用することによって、高度なデバッグを行うことができます。それによって、ハードウェアとソフトウェアの問題が発生する条件を正確に分離し、識別すると、それらの問題点を効果的に調査することができます。



---

## 7. 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。

詳細は、High-performance Embedded Workshop ユーザーズマニュアルを参照してください。

- ツールの変更をしたいが、[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® 2000, Windows® XP上で日本語入力ができない、または日本語の文字が正しく表示されない

[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。

- 自分のホストマシンでワークスペースを開いた。同時に、他の人が他のホストマシン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。

## 7 トラブルシューティング

---

この他にもルネサスの WEB([www.renesas.com](http://www.renesas.com))に掲載されている E200F エミュレータ、High-performance Embedded Workshop に関する FAQ を参照してください。

---

## 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」にしたが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**

添付の「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：

修理のために本エミュレータを輸送される場合、本エ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。エ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。またエミュレータを梱包する場合、必ずエミュレータ添付の導電性エ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エ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 メニュー一覧

---

GUI メニューの一覧を表 A.1 に示します。

表 A.1 GUI メニュー一覧

メニュー	メニューオプション	ショートカットキー	ツールバーボタン	備考
表示	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			

表 A.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	CPU	レジスタ	Ctrl+R		[レジスタ]ウィンドウを表示します
		メモリ...	Ctrl+M		[メモリ]ウィンドウを表示します
		IO	Ctrl+I		[IO]ウィンドウを表示します
		ステータス	Ctrl+U		[ステータス]ウィンドウを表示します
		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
	モニタ (M)...	モニタ設定 (E)...	Shift+Ctrl+E		[Monitor Setting]ダイアログボックスを表示します
		ウィンドウの選択 (S)...			[ウィンドウの選択]ダイアログボックスを表示します
	拡張モニタ				[拡張モニタ]ウィンドウを表示します
	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	

表 A.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	コード	イベントポイント	Ctrl+E		[Event]ウィンドウを表示します
		トレース	Ctrl+T		[Trace]ウィンドウを表示します
		スタックトレース	Ctrl+K		[スタックトレース]ウィンドウを表示します
	グラフィック	画像...	Shift+Ctrl+G		[画像]ウィンドウを表示します
		波形...	Shift+Ctrl+V		[波形]ウィンドウを表示します
	パフォーマンス	パフォーマンス解析	Shift+Ctrl+P		[パフォーマンス解析]ウィンドウを表示します
		プロファイル	Shift+Ctrl+F		[Profile]ウィンドウを表示します
リアルタイムプロファイル		Shift+Ctrl+R		[リアルタイムプロファイル]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

表 A.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	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
	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

表 A.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	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 B コマンドライン機能

---

E200F 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 C High-performance Embedded Workshop の 注意事項

---

### (1) ロードモジュール作成後のソースファイル位置移動に関する注意事項

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

### (2) ソースレベル実行機能

- ソースファイル

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

- Step

標準 C ライブラリ等にも移行します。上位関数に戻るには Step Out を使用してください。

また、for および while 文では、1 回のステップでは次の行に進みません。進める場合はもう一度ステップしてください。

### (3) ファイルアクセス中の操作について

ロードモジュールのダウンロード中、[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、[コードカバレジ]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¥E200F¥Tutorial  
¥Tutorial¥Debug\_SHxxxx\_E200F\_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

- BREAKPOINT解除

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

BREAKPOINT IS DELETED A=xxxxxxx

上記メッセージが表示された場合は、[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 1000 個です。したがって BREAKPOINT を 1000 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 1000 個以下で使用してください。

(12) Timeout error 表示時の注意事項

Timeout error が表示された場合、E200F 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 E200F 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 E200F エミュレータを再接続してください。

(13) [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ください。

(14) 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。

(15) [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

(16) 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。

(17) 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いません。

---

## 付録 D 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、E200F エミュレータ ( R0E0200F0EMU00、R0E0200F2EMU00 ) をご購入頂き、厚く御礼申し上げます。

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

故障症状調査書

ご購入営業担当 行

お客様ご芳名 会社名 \_\_\_\_\_

担当者名 \_\_\_\_\_

様

TEL \_\_\_\_\_

調査項目	症 状
1 故障発生 年月日、時期	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 <ul style="list-style-type: none"> <li>• E200F エミュレータ(R0E0200F0EMU00, R0E0200F2EMU00) シリアル No. _____ レビジョン _____ ( ケース側面に表示しています : シリアル No.は数字 4 桁、 レビジョンはそれに続くアルファベットです )</li> <li>• トレースユニット ( R0E0200F0ETU00 ) シリアル No. _____ レビジョン _____ ( 基板上に捺印表示しています )</li> <li>• 拡張プロファイラユニット ( R0E0200F0EPU00 ) シリアル No. _____ レビジョン _____ ( 基板上に捺印表示しています )</li> <li>• 付属 CD-R(R0E0200F0EMU00S) バージョン V ( CD-R に V.x.xx release と表示しています(x : 数字) )</li> <li>• ご使用になっている PC メーカー名 _____ 型式 _____ 使用 OS{ Windows®2000, Windows®XP }</li> </ul>
4 エラー発生時の 設定内容	(1) 動作モード : モード _____ (2) ターゲットシステム電圧 : _____ V (3) 使用クロック : ( Xtal 発振子、外部クロック入力 いずれかに ) (4) 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。

--



---

ルネサスマイクロコンピュータ開発環境システム  
ユーザズマニュアル  
SH-4A、SH4AL-DSP用 E200F エミュレータ

発行年月日 2004年6月4日 Rev.1.00

2007年7月25日 Rev.6.00

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

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

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

営業お問合せ窓口  
株式会社ルネサス販売

# RENESAS

<http://www.renesas.com>

本			社	〒100-0004	千代田区大手町2-6-2 (日本ビル)	(03) 5201-5350
京	浜	支	社	〒212-0058	川崎市幸区鹿島田890-12 (新川崎三井ビル)	(044) 549-1662
西	東	京	支	〒190-0023	立川市柴崎町2-2-23 (第二高島ビル2F)	(042) 524-8701
東	北	支	社	〒980-0013	仙台市青葉区花京院1-1-20 (花京院スクエア13F)	(022) 221-1351
い	わ	き	支	〒970-8026	いわき市平小太郎町4-9 (平小太郎ビル)	(0246) 22-3222
茨	城	支	店	〒312-0034	ひたちなか市堀口832-2 (日立システムプラザ勝田1F)	(029) 271-9411
新	潟	支	店	〒950-0087	新潟市東大通1-4-2 (新潟三井物産ビル3F)	(025) 241-4361
松	本	支	社	〒390-0815	松本市深志1-2-11 (昭和ビル7F)	(0263) 33-6622
中	部	支	社	〒460-0008	名古屋市中区栄4-2-29 (名古屋広小路プレイス)	(052) 249-3330
関	西	支	社	〒541-0044	大阪市中央区伏見町4-1-1 (明治安田生命大阪御堂筋ビル)	(06) 6233-9500
北	陸	支	社	〒920-0031	金沢市広岡3-1-1 (金沢パークビル8F)	(076) 233-5980
広	島	支	店	〒730-0036	広島市中区袋町5-25 (広島袋町ビルディング8F)	(082) 244-2570
島	取	支	店	〒680-0822	鳥取市今町2-251 (日本生命鳥取駅前ビル)	(0857) 21-1915
九	州	支	社	〒812-0011	福岡市博多区博多駅前2-17-1 (博多プレステージ5F)	(092) 481-7695

■技術的なお問合せおよび資料のご請求は下記へどうぞ。

総合お問合せ窓口：コンタクトセンター E-Mail: [csc@renesas.com](mailto:csc@renesas.com)



# SH-4A、SH4AL-DSP 用 E200F エミュレータ ユーザズマニュアル



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

RJJ10B0141-0600