

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

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

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

2010年4月1日

ルネサスエレクトロニクス株式会社

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

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

## ご注意書き

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

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

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

# SH7080 シリーズ用 E200F エミュレータ

## プロファイル、リアルタイムプロファイル測定方法

### 要旨

E200F エミュレータは、プロファイル機能とリアルタイムプロファイル機能を使って各関数のパフォーマンスを測定することができます。関数ごとの所要時間の統計情報を取得することによって、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。本書では、SH7080 シリーズ用 E200F エミュレータを用いてプロファイル機能とリアルタイムプロファイル機能の使い方を説明します。

本書の使用例では、E200F エミュレータを単体で使用して説明しています。本書の内容は、異なるターゲットマイコンであっても E200F エミュレータであれば共通で利用することが可能です。

### 目次

1.	仕様 .....	2
1.1	プロファイル機能について .....	2
1.2	リアルタイムプロファイル機能について .....	2
2.	使用機能説明 .....	3
3.	ソフトウェアの準備 .....	3
3.1	はじめに .....	3
3.2	E200Fエミュレータ付属品ソフトウェアのインストール .....	3
3.3	その他の必要なソフトウェアのインストール .....	3
4.	動作説明 .....	4
4.1	High-performance Embedded Workshopの起動 .....	4
4.2	ワークスペースを開く .....	5
4.3	スタック情報ファイルのロード .....	11
4.4	プロファイル測定方法 .....	14
4.5	オプションメニューの説明 .....	20
4.6	リアルタイムプロファイル測定方法 .....	24
5.	よくある質問 .....	32
5.1	プロファイル機能を使用する上で注意すべき事がありますか？ .....	32
5.2	プロファイル機能使用中に使用できなくなる機能はありますか？ .....	32
5.3	リアルタイムプロファイル機能を使用する上で注意すべき事がありますか？ .....	33
6.	関連ドキュメント .....	34

## 1. 仕様

### 1.1 プロファイル機能について

E200Fエミュレータに搭載されてい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。

プロファイル測定結果を表示するプロファイルウィンドウには、各関数/変数のプロファイルデータを表示する「List」シート、関数の各呼び出し位置におけるプロファイルデータを表示する「Tree」シートの2つがあります。また、プロファイル測定結果の「プロファイル-チャート」を表示させることも可能です。

但し、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するため、リアルタイム性がなくなります。また、オンチップパフォーマンス測定機能とプロファイル機能は、同時に使用できません。

#### 【注意事項】

本機能は、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度実行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ため、測定値は目安と考えてください。

### 1.2 リアルタイムプロファイル機能について

E200F エミュレータに搭載されているリアルタイムプロファイル機能は、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。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せん。

また、オプションとして拡張プロファイルユニットを E200F 本体に接続することで測定できるエリアを拡大できます。

- ・ 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)

なお、本書の説明では拡張プロファイルユニットを使用していません。接続方法などの詳細は、ルネサス Web サイトに掲載されている E200F エミュレータユーザズマニュアルを別途ご参照ください。

#### 【注意事項】

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。

## 2. 使用機能説明

本書では、E200F エミュレータ付属の CD-ROM、またはルネサス Web サイトからのダウンロードパッケージに含まれるサンプルプログラムを使用してプロファイル機能とリアルタイムプロファイル機能の使い方を説明します。

使用する各ツールのバージョンは次のとおりです。

エミュレータソフトウェア

E200F エミュレータデバッグ Verion 2.04.00

コンパイラパッケージ

無償評価版 SuperH ファミリ用 C/C++コンパイラパッケージ V.9.01

## 3. ソフトウェアの準備

### 3.1 はじめに

E200F エミュレータに付属する CD-ROM のソフトウェアをインストールします。

これにより本書で使用するサンプルプログラム(tutorial ワークスペース)がパソコン上に展開されます。

High-performance Embedded Workshop をインストール済みのパソコン上に E200F エミュレータに付属する CD-ROM のソフトウェアをインストールすることは可能です。この場合、インストール作業中に一部のダイアログ表示が省略されることがあります。

### 3.2 E200F エミュレータ付属品ソフトウェアのインストール

E200F エミュレータに付属する CD-ROM 内の HewInstMan.exe を実行してください。

インストールの詳細につきましては、ルネサス Web サイトに掲載の E200F エミュレータ用ユーザーズマニュアルを参照していただき、インストール作業中は画面の指示に従って操作をしてください。ここではインストール手順は省略します。

### 3.3 その他の必要なソフトウェアのインストール

(1) 本書では、サンプルプログラムを一部変更して動作の確認を行います。このため SuperHファミリ用 C/C++コンパイラパッケージを使用します。製品版コンパイラパッケージを購入済みの場合、製品版コンパイラパッケージをインストールしてください。

(2) 製品版コンパイラパッケージを購入前の場合、E200F エミュレータに付属する CD-ROM に同梱する無償評価版コンパイラパッケージを利用できます。

無償評価版コンパイラパッケージは、ルネサスのウェブサイトからダウンロードすることも可能です。無償評価版の SuperH ファミリ用 C/C++コンパイラパッケージは、[ルネサストップページ]→[サポート]→[ダウンロード]→[ダウンロード検索]でカテゴリー選択を開いて「無償評価版」を選択して検索することにより見つけることができます。ルネサスのウェブサイトへのリンク情報は、本書の最後の章にあります。無償評価版の制限事項およびインストール方法は、ダウンロードページから入手してください。

(3) ソフトウェアのインストールの際に、オートアップデートユーティリティを選択した場合、インターネット経由で各ツールの最新版の存在を確認することが可能になります。

## 4. 動作説明

本章では、High-performance Embedded Workshop (HEW) を起動してプロファイル機能とリアルタイムプロファイル機能の使い方について説明します。手順としては以下のようになります。

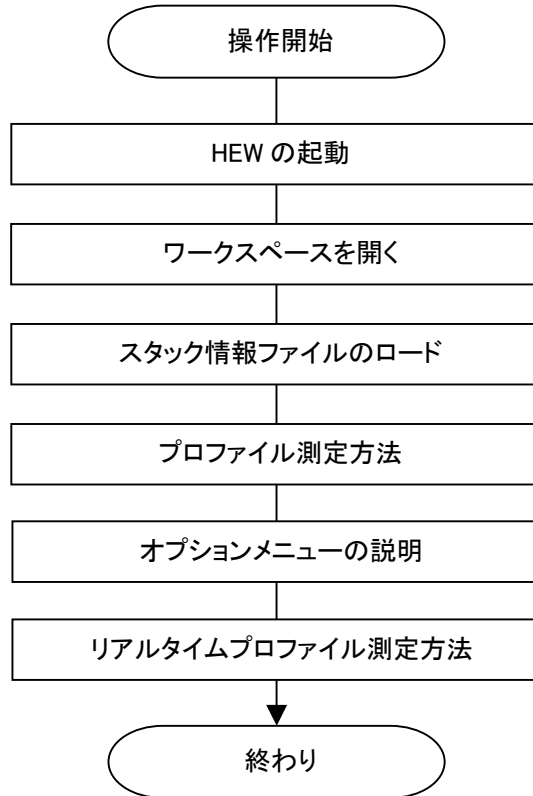


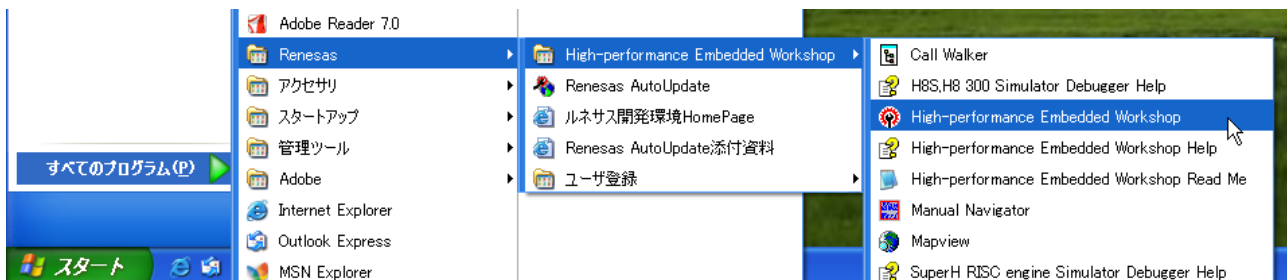
図 4.1 サンプルプログラムの実行手順

### 4.1 High-performance Embedded Workshop の起動

まず、始めにユーザシステムを接続したE200FエミュレータとホストコンピュータをUSBケーブルで接続し、デバッグ操作が可能であることを確認してください。

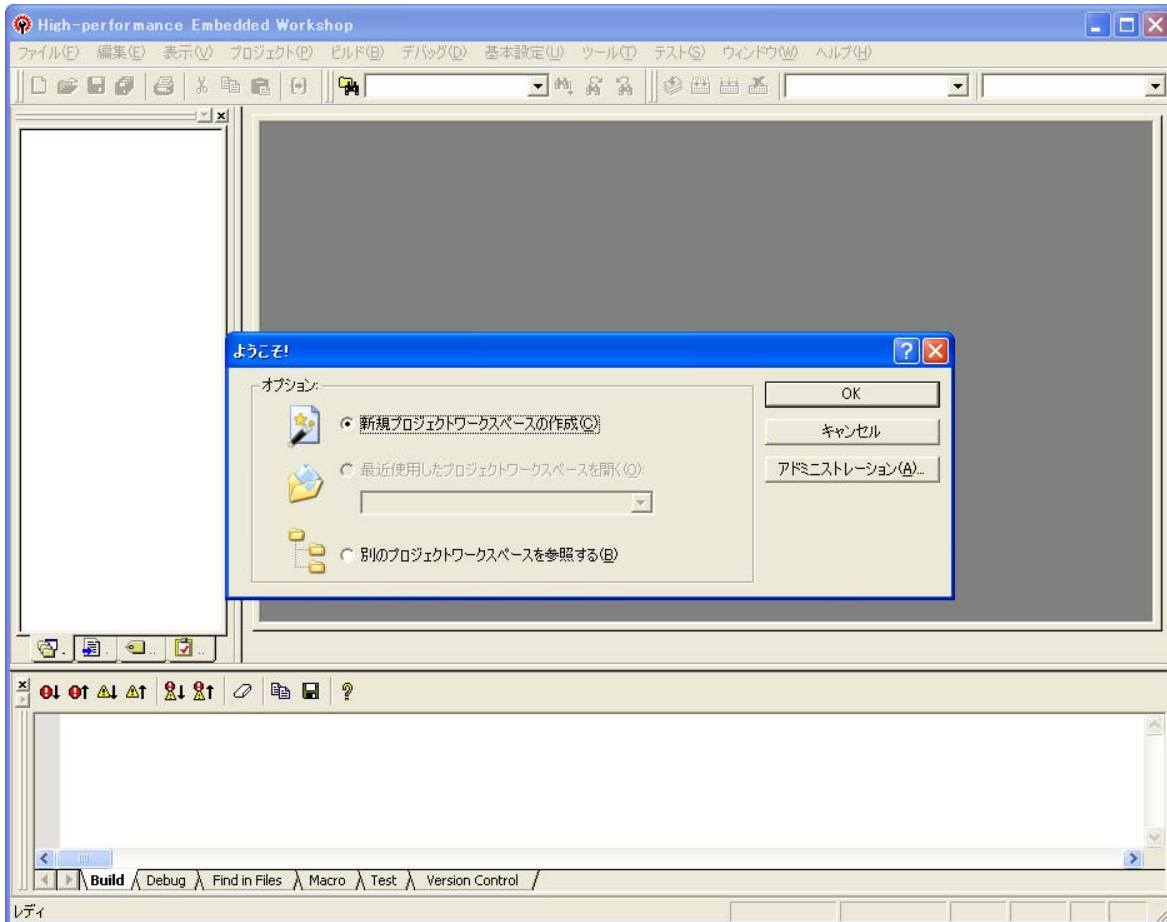
次にHigh-performance Embedded Workshopを起動します。

[スタート]メニューの[すべてのプログラム]から[Renesas]→[High-performance Embedded Workshop]→[High-performance Embedded Workshop]で起動できます。

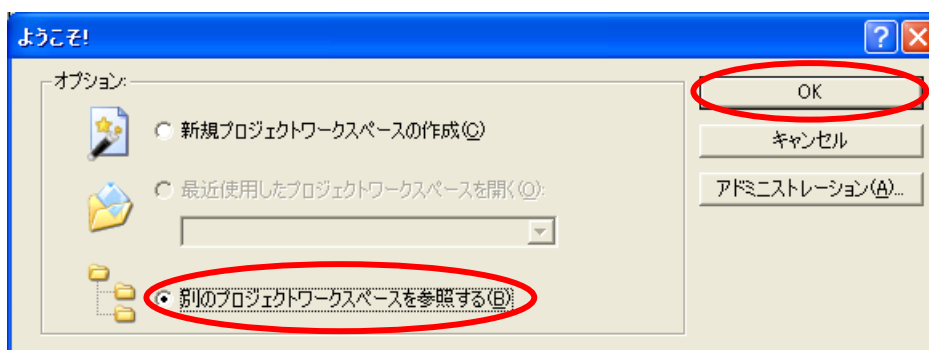


### 4.2 ワークスペースを開く

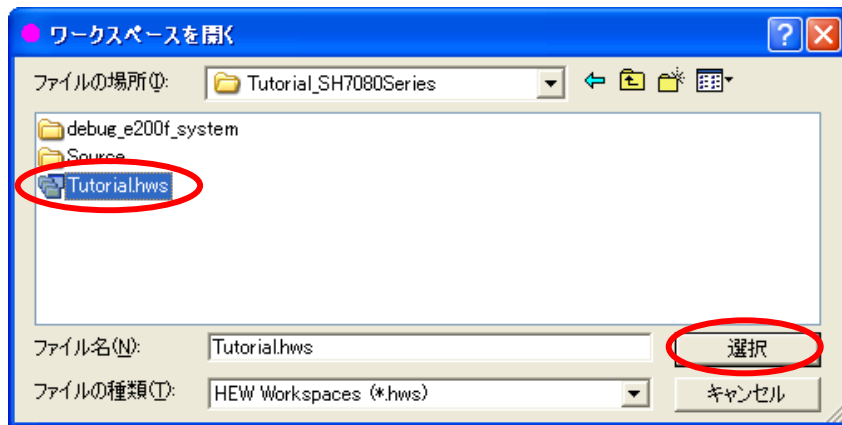
(1) High-performance Embedded Workshop 上に[ようこそ]ダイアログボックスが表示されます。



[ようこそ]ダイアログボックス内の[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て[OK]ボタンを押してください。



(2) [ワークスペースを開く]ダイアログボックスが表示されます。



本製品の CD-ROM のインストールが完了している場合、標準では次のフォルダ位置にワークスペース "Tutorial.hws" が格納されています。フォルダ位置を確認しながら順番に指定してください。ワークスペース "Tutorial.hws" が見つかりましたら指定し[選択]ボタンを押してください。

C:\¥WorkSpace¥Tutorial¥E200F¥SH7080Series¥Tutorial_SH7080Series¥Tutorial.hws
C:\¥WorkSpace └Tutorial └E200F └SH7080Series └Tutorial_SH7080Series └Tutorial.hws

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

<High-performance Embedded Workshop インストール先ディレクトリ>

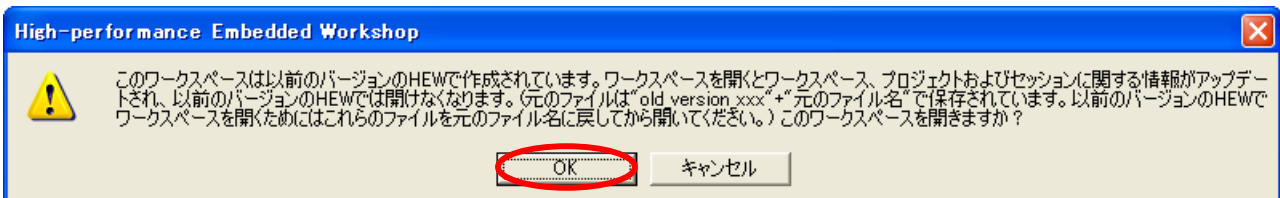
¥Tools¥Renesas¥DebugComp¥Platform¥E200F¥SH7080Series¥Tutorial\_SH7080Series

ディレクトリ例:

C:\¥hew3¥Tools¥Renesas¥DebugComp¥Platform¥E200F¥ SH7080Series¥Tutorial\_SH7080Series

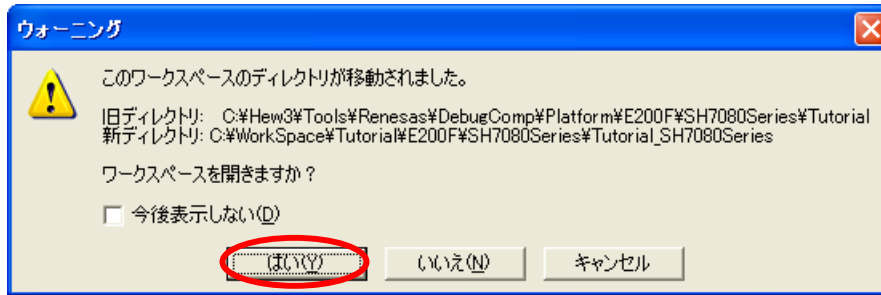
C:\¥hew2¥Tools¥Renesas¥DebugComp¥Platform¥E200F¥ SH7080Series¥Tutorial\_SH7080Series

(3) ワークスペースのバージョンが古い場合は次のダイアログボックスが表示されます。新しいバージョンにアップデートするために[OK]ボタンを押してください。

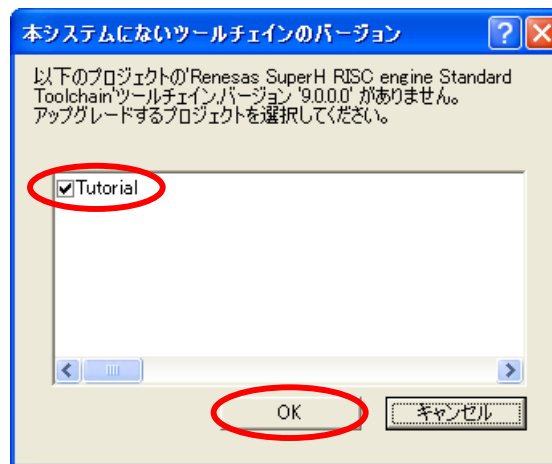




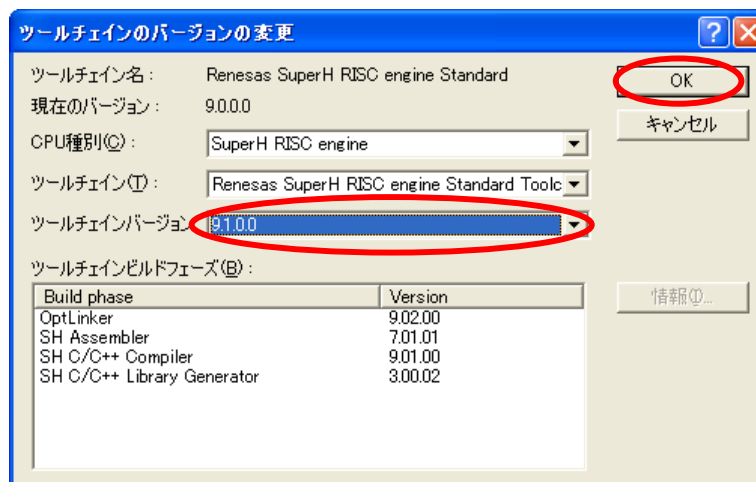
- (4) [ウォーニング]ダイアログボックスが表示されます。ディレクトリが移動されたワークスペースを開くため[はい]ボタンを押してください。



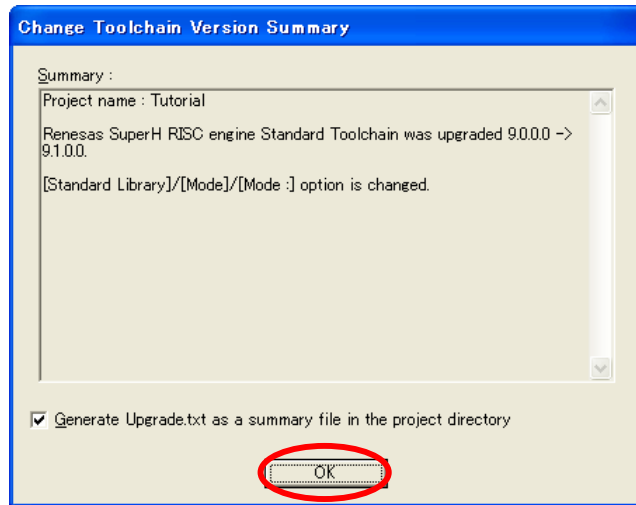
- (5) [本システムにないツールチェーンのバージョン]ダイアログボックスが表示される場合は、プロジェクト名を選択して[OK]ボタンを押してください。



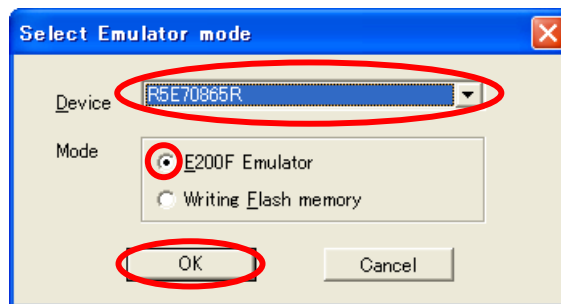
- (6) [ツールチェーンのバージョンの変更]ダイアログボックスが表示される場合は、利用するツールチェーンバージョンを選択して[OK]ボタンを押してください。



(7) [Summary]ダイアログボックスが表示される場合は、そのまま[OK]ボタンを押してください。

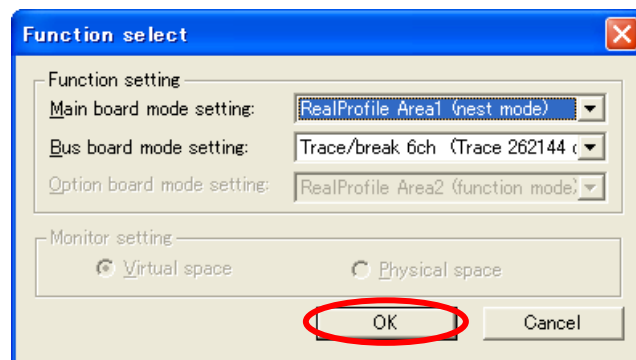


(8) ワークスペースが開かれると、[Select Emulator mode]ダイアログボックスが開きます。[Device]にデバッグ対象マイコン名称を指定し、[Mode]を[E200F Emulator]を選択して[OK]ボタンを押してください。

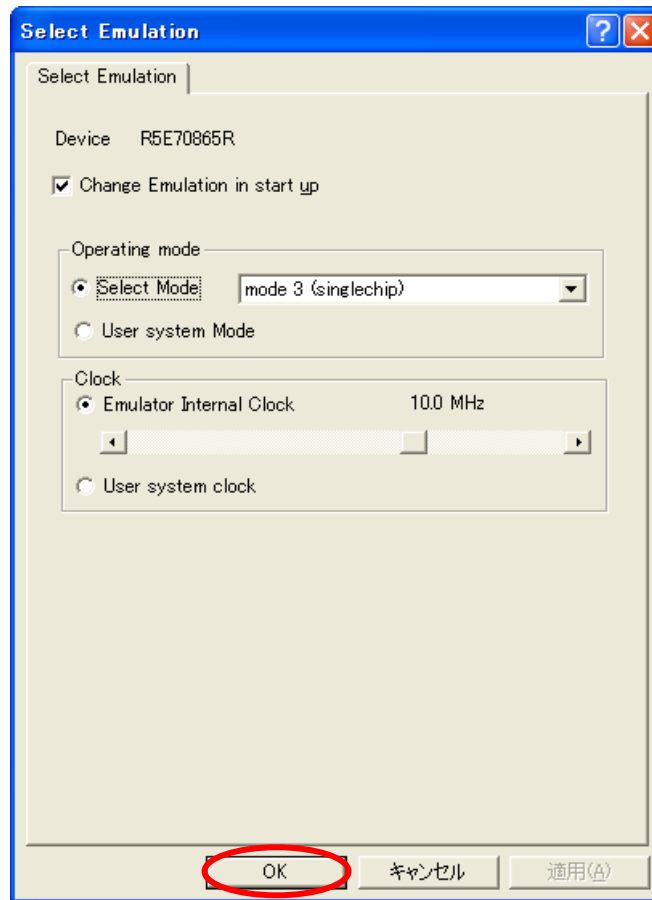


本書の例では、"R5E70865R" を選択しています。

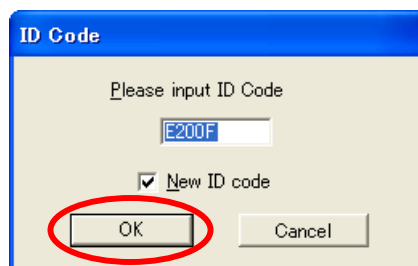
(9) [Function select]ダイアログボックスが開きます。デフォルトのまま[OK]ボタンを押してください。



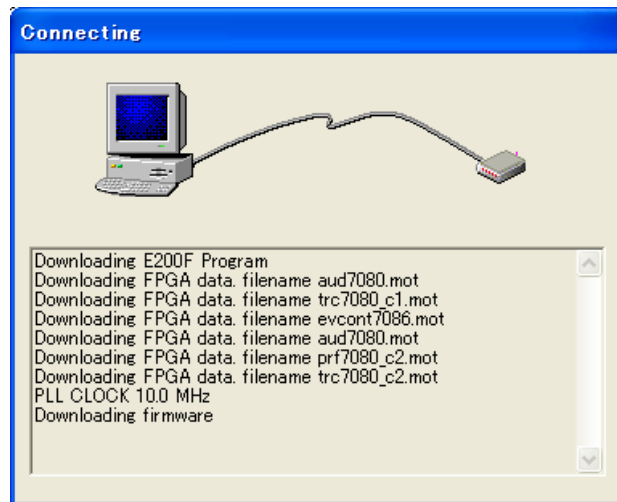
(10) [Select Emulator]ダイアログボックスが開きます。デフォルトのまま[OK]ボタンを押してください。



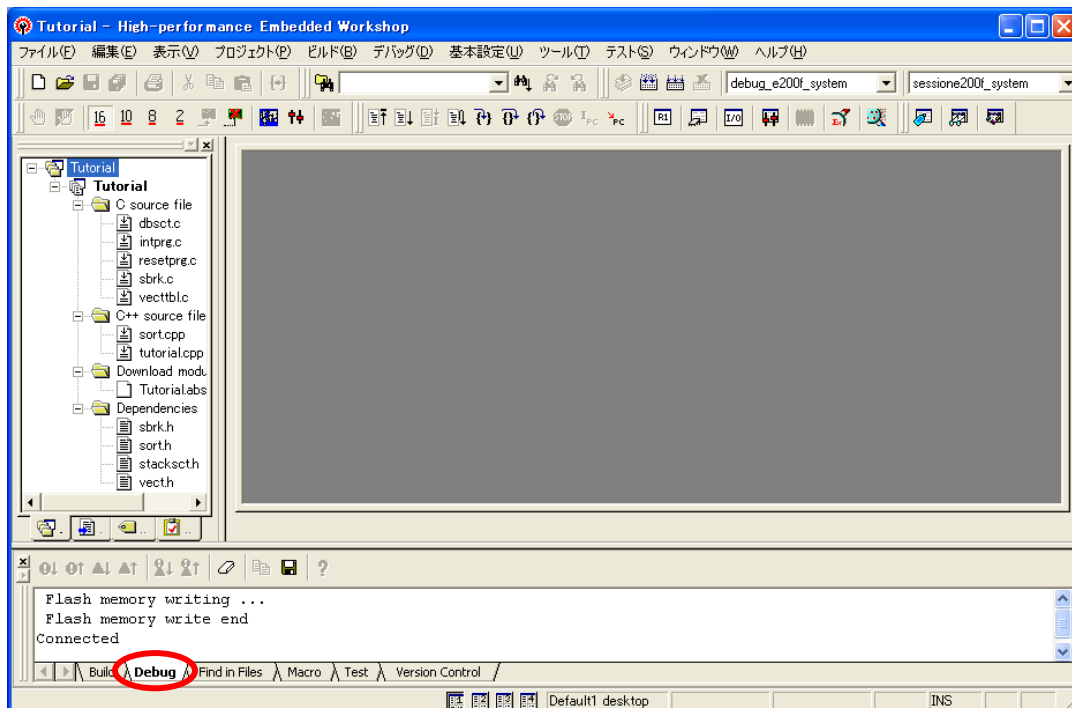
(11) [ID Code]ダイアログボックスが開きます。デフォルトの[E200F]のまま[OK]ボタンを押してください。



(12) E200F エミュレータ接続中は下図の[接続]ダイアログボックスが表示されます。



(13) E200F エミュレータの接続が完了して High-performance Embedded Workshop の画面が操作可能になります。

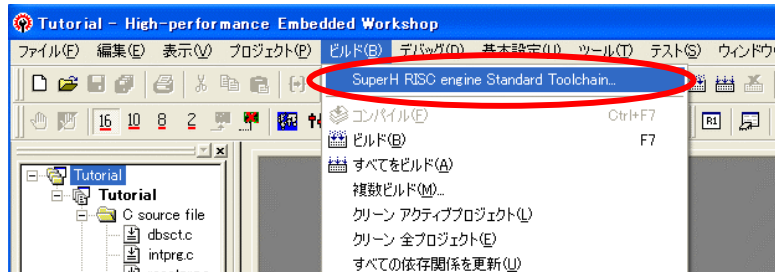


E200F エミュレータの接続が成功すると、アウトプットウィンドウの[Debug]タブには“Connected”が表示されます。

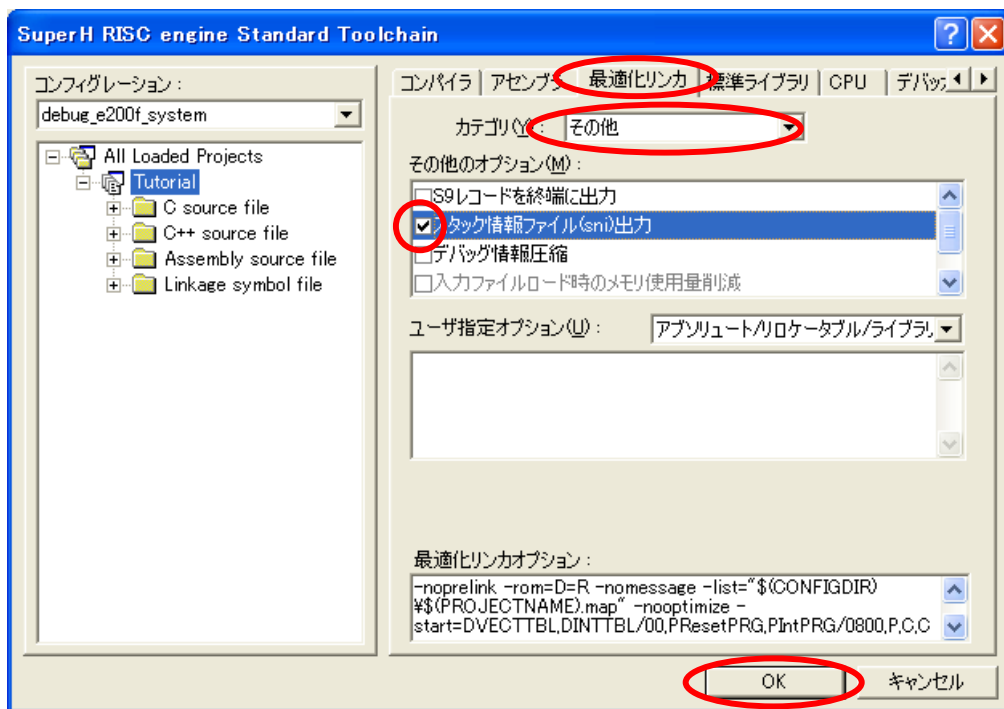
### 4.3 スタック情報ファイルのロード

プロファイル測定を行う場合には、各関数のスタックサイズ情報を得るためにスタック情報ファイルのロードを行う必要があります。スタック情報ファイルのロードを行わない場合、プロファイル測定結果にスタックサイズ情報は出力されません。

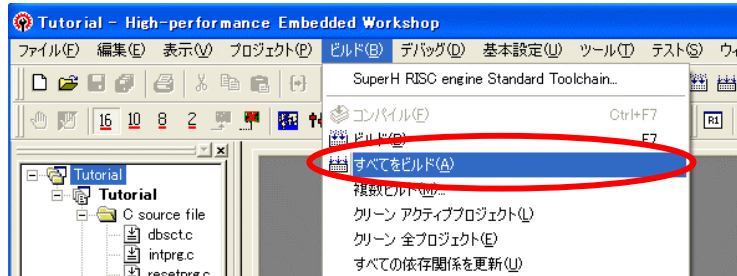
- (1) 最初に、スタック情報ファイルを作成します。[ビルド]メニューの[SuperH RISC engine Standard Toolchain...]を選択してください。



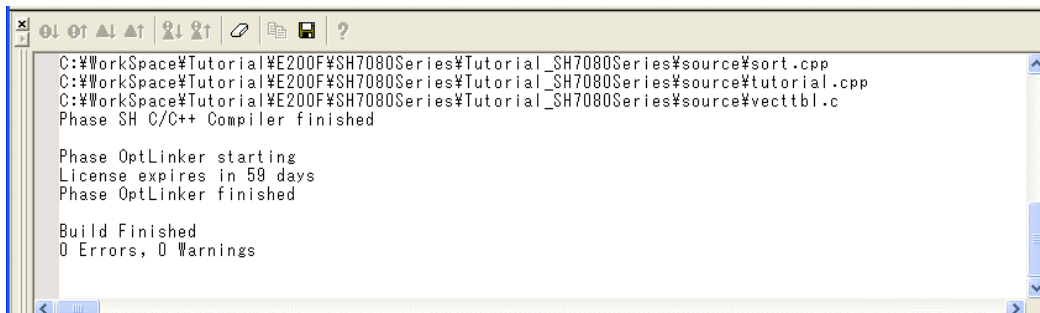
- (2) [SuperH RISC engine Standard Toolchain]ダイアログボックスが表示されるので、[最適化リンカ]タブを選んで[カテゴリ]を“その他”にして、[その他のオプション]の“スタック情報ファイル(sni)出力”にチェックを入れて[OK]を押してください。



(3) [ビルド]メニューの[すべてをビルド]を選択してビルド作業を行ってください。



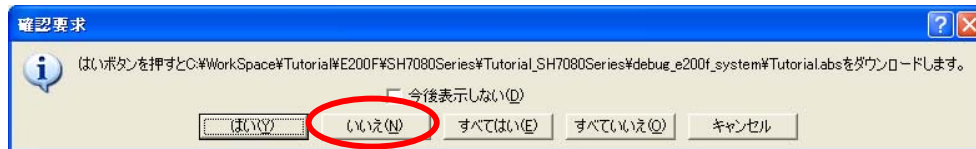
(4) ビルド作業が終了するとアウトプットウィンドウに下図のようなコメントが出力されます。



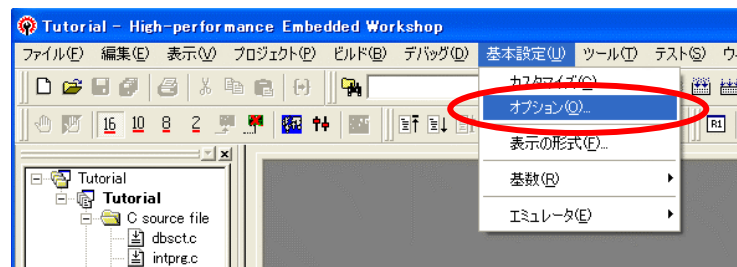
ビルド作業が終了すると下記ディレクトリにスタック情報ファイル“Tutorial.sni”が作成されます。

C:\Workspace\tutorial\%E200F%\SH7080Series\tutorial\_SH7080Series\debug\_e200f\_system

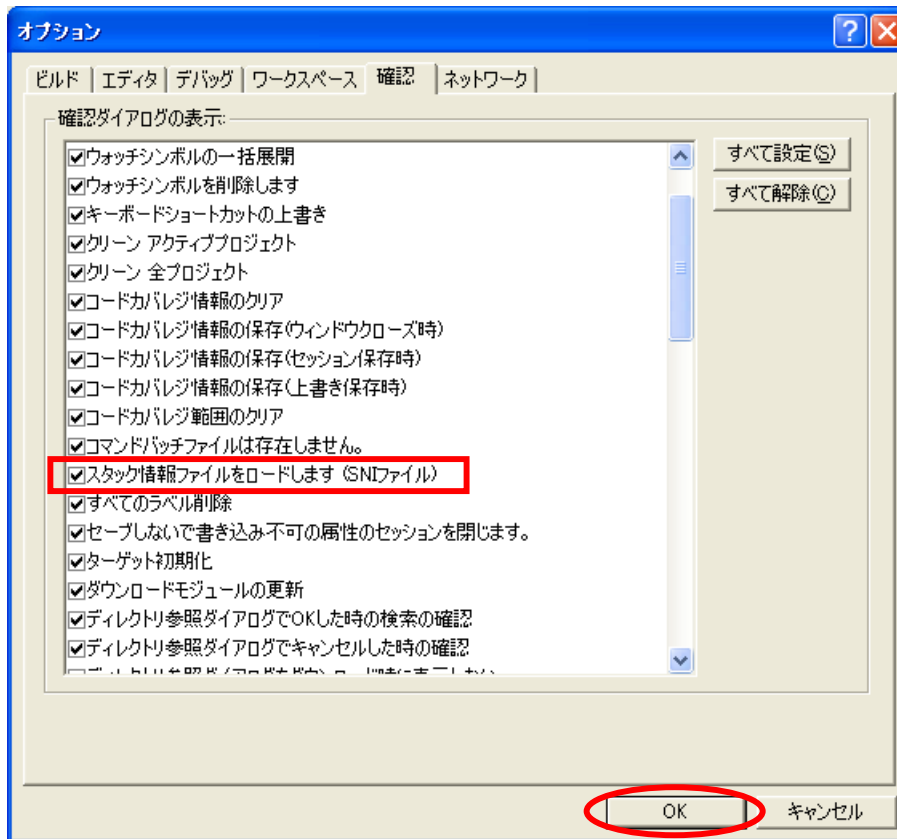
(5) 下図のような[確認要求]ダイアログボックスが表示されるので、[いいえ]を選択してください。



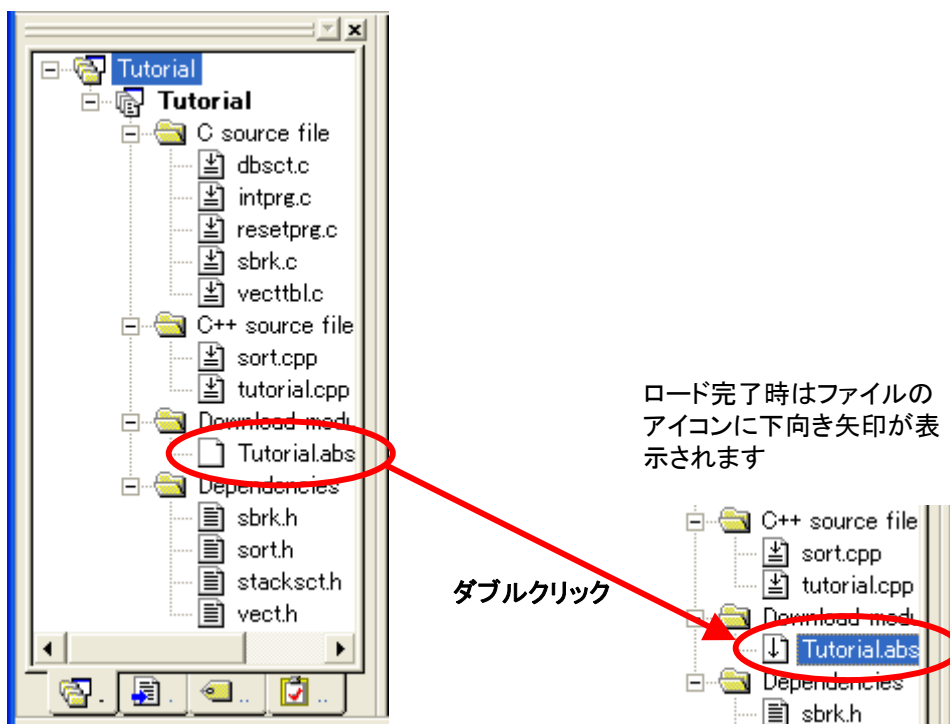
(6) 次に、スタック情報ファイルをロードします。[基本設定]メニューの[オプション]を選択してください。



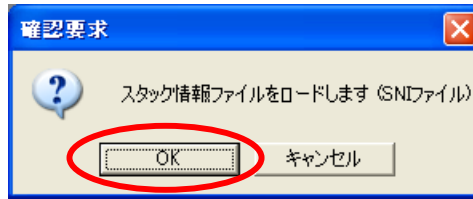
- (7) [オプション]ダイアログボックスが表示されるので、[確認]タブを選んで“スタック情報ファイルをロードします(SNIファイル)”にチェックを入れて[OK]を押してください(デフォルト状態で既にチェックは入っています)。



- (8) プログラムをダウンロードするためワークスペースのロードモジュールファイル(拡張子.abs)をダブルクリックしてください。



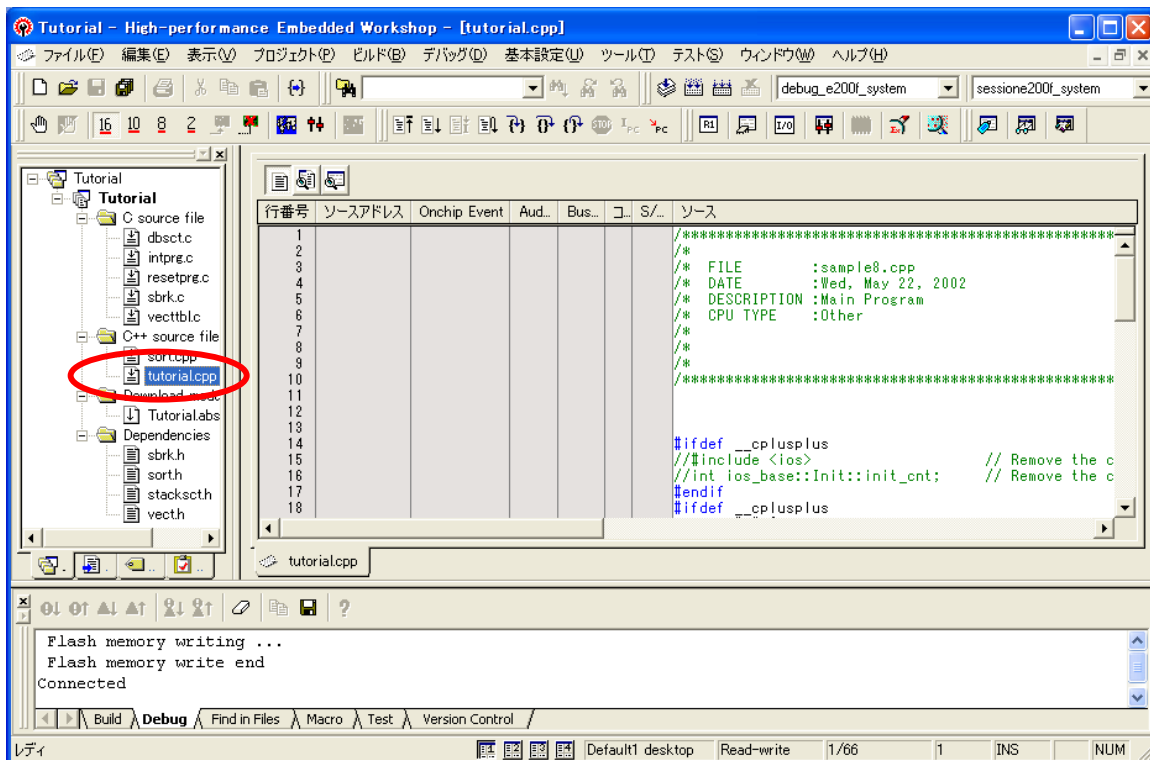
- (9) プログラムをダウンロードすると、[確認要求]ダイアログボックスが表示されるので[OK]を押してスタック情報ファイルをロードしてください。



### 4.4 プロファイル測定方法

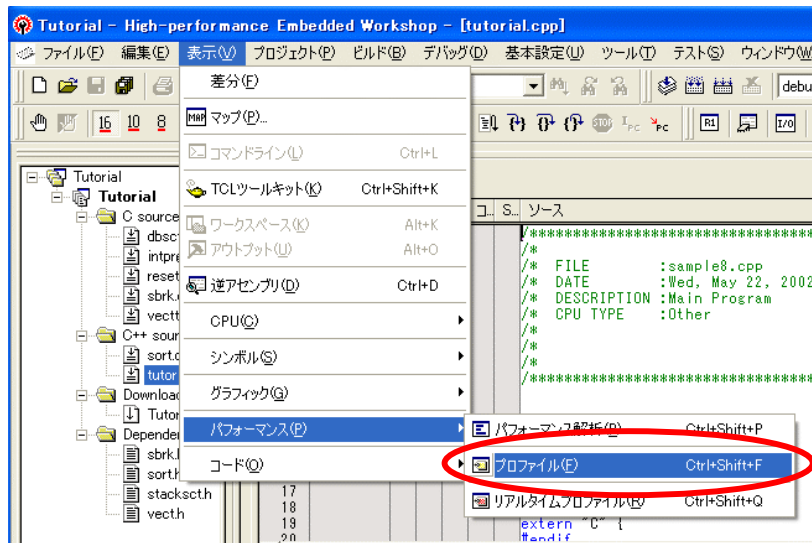
この章では、ソフトウェアブレイクを使ったプロファイル測定を行う方法を説明します。

- (1) ワークスペース内のソースファイル “tutorial.cpp” をダブルクリックしてソースウィンドウに表示させてください。

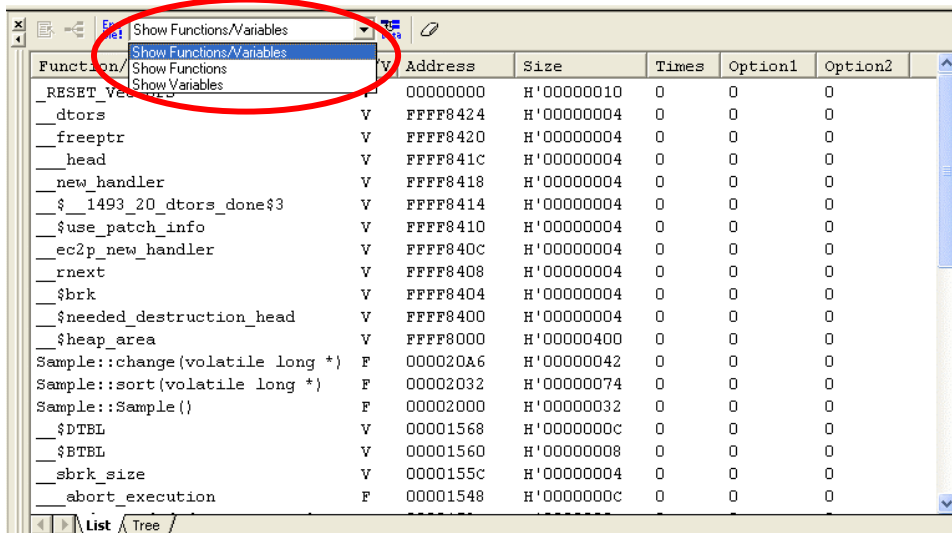




(2) 次に、[表示]メニューの[パフォーマンス]の中の[プロファイル]を選択してください。



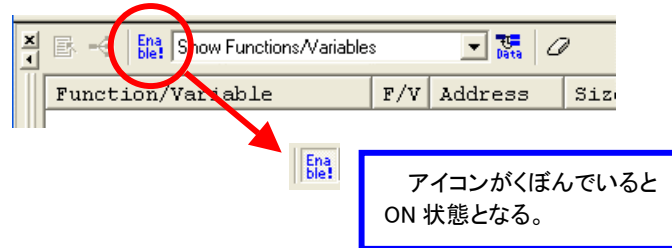
(3) プロファイル測定結果を表示するプロファイルウィンドウが開きます。まずは、プロファイルウィンドウの表示設定を選択します。表示設定は全部で3種類ありますが、今回は“Show Functions/Variables”を選択してください。



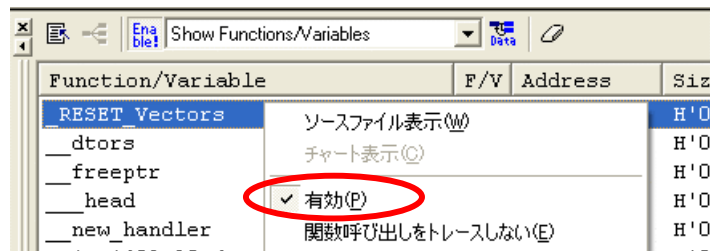
表示設定の内容は下記のとおりです。

- [Show Functions/Variables] → 関数およびグローバル変数の両方を表示します。
- [Show Functions] → 関数のみを表示します。
- [Show Variables] → グローバル変数のみを表示します。

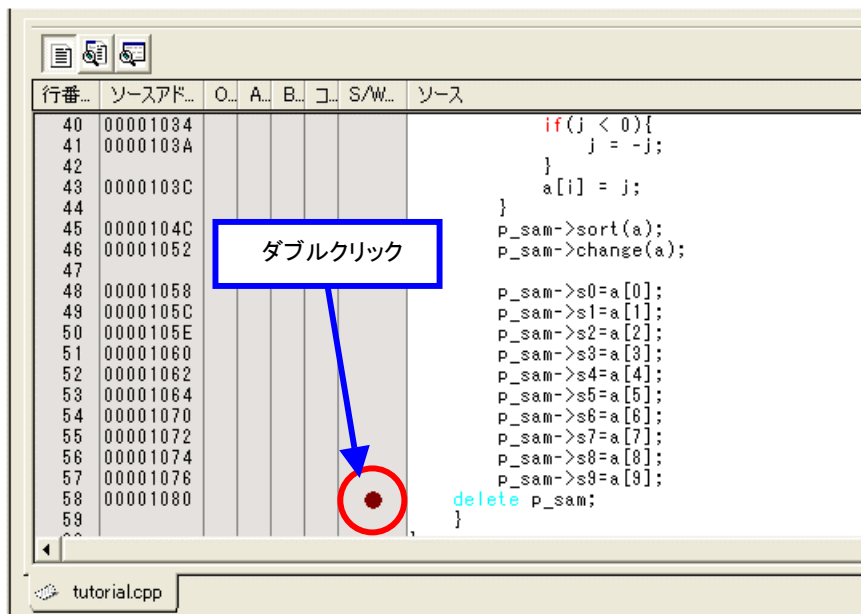
- (4) プロファイル測定を行う前に[有効]アイコンをクリックして、プロファイルデータ収集 ON の状態にしてください。OFF の状態になっているとプロファイルウィンドウに測定結果が表示されません。



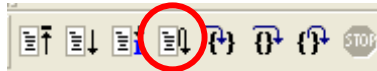
プロファイルウィンドウ上で右クリックして表示されるメニューからも行えます。[有効]にチェックが入っているとプロファイルデータ収集 ON の状態になります。



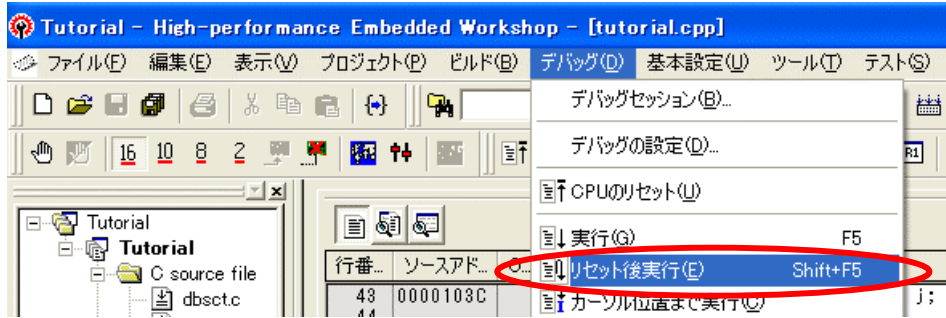
- (5) ソースファイル“tutorial.cpp”の行番号 58 の[S/W ブレークポイント]カラムをダブルクリックして、ソフトウェアブレークを設定してください。



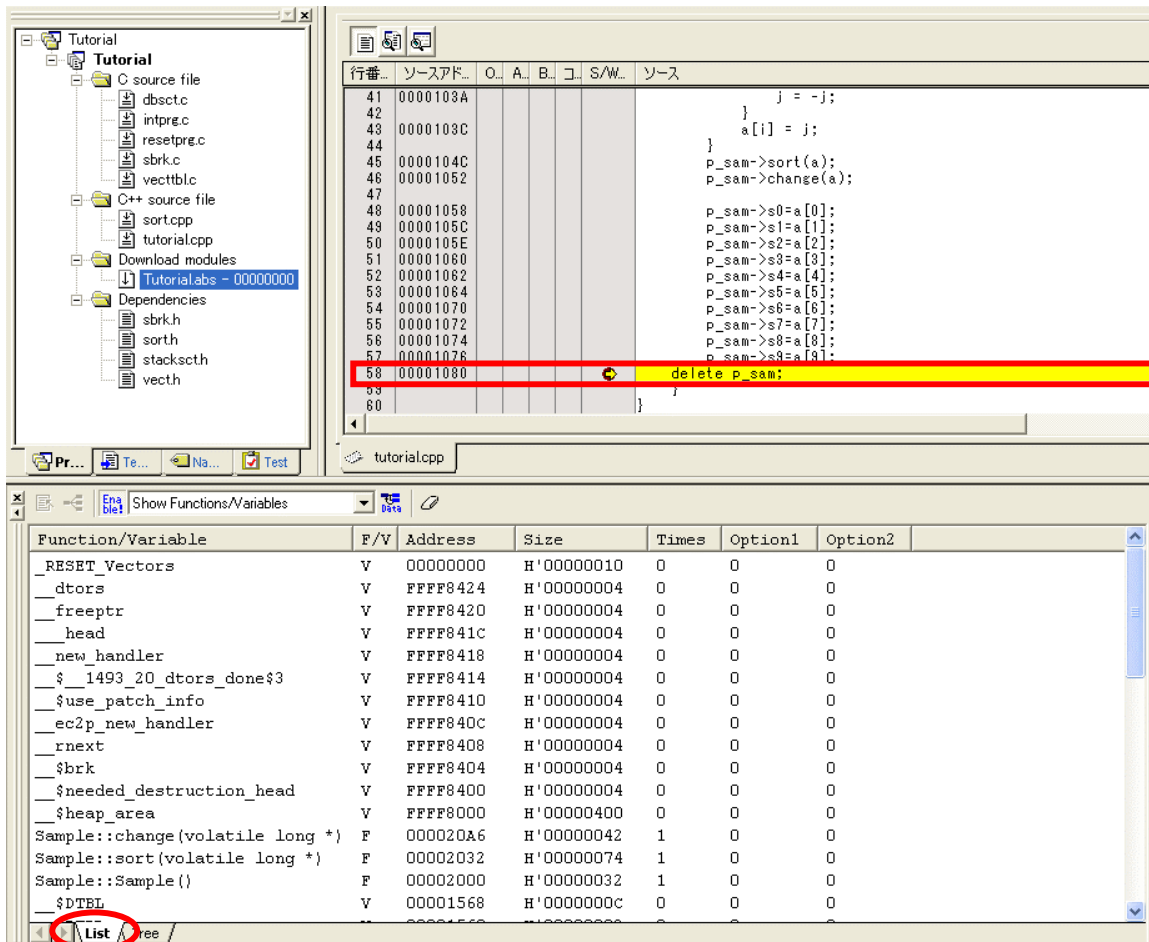
(6) [リセット後実行]アイコンをクリックして、プログラムを実行してください。



リセット後実行は、[デバッグ]メニューから行えます。



(7) ブレーク設定した行でプログラムが停止して、プロファイルウィンドウに測定結果が表示されます。[List]タブ上には、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。



[Tree]タブ上には、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。

Function	Address	Size	Stack Size	Times	Option1	Option2
C:\Workspace\Tutorial\E200F\SH7080S...						
_PowerON_Reset_PC	00000800	H'0000002E	H'00000000	1	0	0
main	0000101c	H'0000006c	H'00000028	1	0	0
Sample::Sample()	00002000	H'00000032	H'00000004	1	0	0
operator new(unsigned long)	00001158	H'00000064	H'0000001c	1	0	0
malloc	0000127c	H'00000000	H'00000000	1	0	0
\$morecor	0000130c	H'0000005c	H'00000010	1	0	0
__divlu	00001408	H'00000000	H'00000000	1	0	0
_sbrk	00001000	H'0000001c	H'00000000	1	0	0
free	000011E4	H'00000000	H'00000000	1	0	0
Sample::change(volat...)	000020A6	H'00000042	H'00000028	1	0	0
Sample::sort(volat...)	00002032	H'00000074	H'00000018	1	0	0
_rand	000011BC	H'00000028	H'00000004	10	0	0
operator delete(void *)	00001144	H'00000014	H'00000000	0	0	0

プロファイルウィンドウ上で右クリックして表示されるメニューの[関数呼び出しをトレースしない]にチェックが入っていると、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。この場合、上図とは“Times”の値が異なってきますのでご注意ください(関数が呼び出されてもカウントしない状態です)。

表示項目の内容は下記のとおりです。

- “F/V” → 関数(F)か、グローバル変数(V)かのどちらかを表示
- “Address” → 関数を配置しているメモリのアドレスを表示
- “Size” → 関数のサイズを表示
- “Times” → 関数の呼び出し(アクセス)回数を表示
- “Stack Size” → 関数のスタックサイズを表示
- “Option” → 追加した測定項目の結果を表示 (4.5章で説明)

- (8) プロファイルウィンドウの[List]タブ上の関数を選択して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。[ソース表示]アイコンからも行えます。

[ソース表示]アイコン

ダブルクリックすると

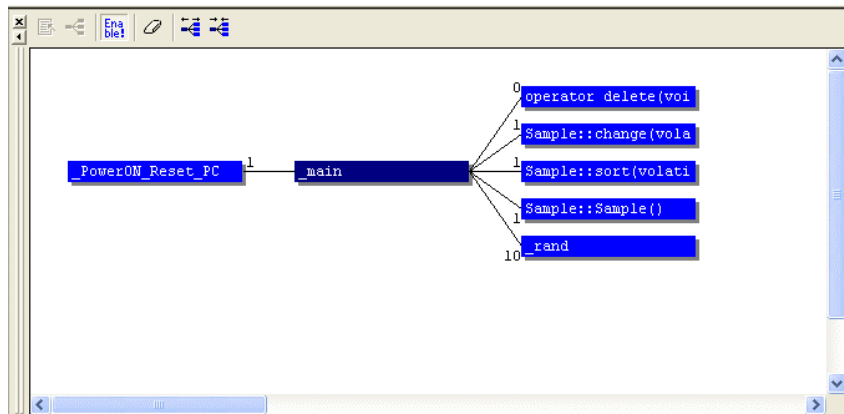
該当するソースプログラムを表示

Function/Variable	F/V	Address	Size	Times	Option1	Option2
_rand	F	000011BC	H'00000028	10	0	0
operator new(unsigned long)	F	00001158	H'00000064	1	0	0
operator delete(void *)	F	00001144	H'00000014	0	0	0
__CALL_END	F	00001130	H'00000014	0	0	0
__CALL_INIT	F	00001108	H'00000028	1	0	0
000010B0	F	000010B0	H'00000000	1	0	0
abort	F	00001088	H'00000018	0	0	0
main	F	0000101c	H'0000006c	1	0	0
_sbrk	F	00001000	H'0000001c	1	0	0
_Dummy	F	0000084c	H'00000004	0	0	0
_INT_Illegal_code	F	00000848	H'00000004	0	0	0
_Manual_Reset_PC	F	0000082E	H'0000001A	0	0	0
_PowerON_Reset_PC	F	00000800	H'0000002E	1	0	0
_INT_Vectors	V	00000010	H'00000008	0	0	0

- (9) 次に、プロファイルウィンドウの[List]タブ上の関数を選択して[チャート表示]アイコンをクリックしてください。ここでは“\_main”を選択します。

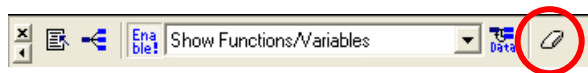
Func	Chart View	Table	F/V	Address	Size	Times	Option1	Option2
_rand			F	000011BC	H'00000028	10	0	0
operator new(unsigned long)			F	00001158	H'00000064	1	0	0
operator delete(void *)			F	00001144	H'00000014	0	0	0
__CALL_END			F	00001130	H'00000014	0	0	0
__CALL_INIT			F	00001108	H'00000028	1	0	0
000010B0			F	000010B0	H'00000000	1	0	0
abort			F	00001098	H'00000018	0	0	0
<b>_main</b>			<b>F</b>	<b>0000101C</b>	<b>H'0000006C</b>	<b>1</b>	<b>0</b>	<b>0</b>
_SOK			F	00001000	H'0000001C	1	0	0
_Dummy			F	0000084C	H'00000004	0	0	0
_INT_Illegal_code			F	00000848	H'00000004	0	0	0
_Manual_Reset_PC			F	0000082E	H'0000001A	0	0	0
_PowerON_Reset_PC			F	00000800	H'0000002E	1	0	0
_INT_Vectors			V	00000010	H'00000008	0	0	0

- (10) プロファイルチャートウィンドウが開きます。プロファイル測定結果をプロファイルチャートとして表示することができます。



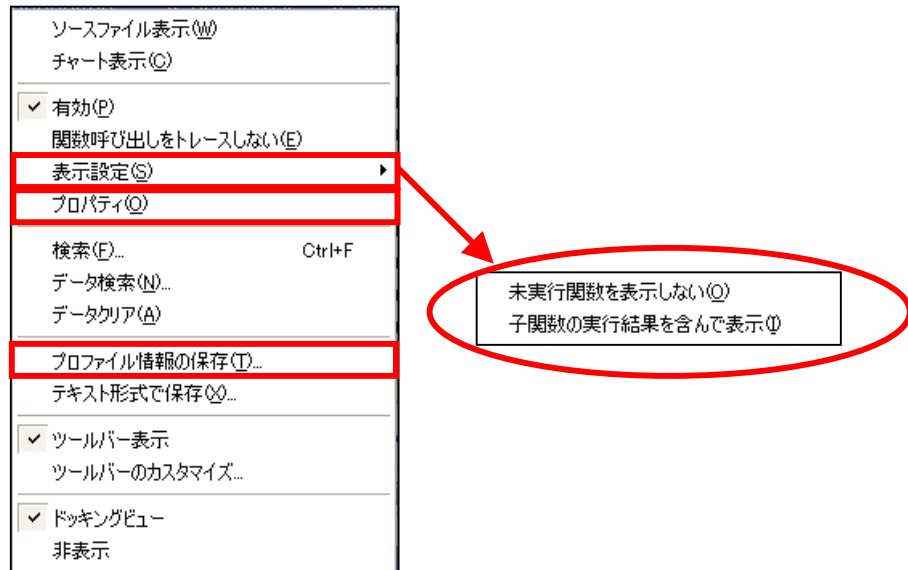
プロファイルチャートウィンドウは、着目する関数を中心に表示し、その左側には着目した関数を呼び出した関数、右側には着目している関数が呼び出した関数をそれぞれ表示します。また、呼び出しを行った回数も表示します。

- (11) 引き続きプロファイル測定を行う場合、[クリア]アイコンをクリックして測定値を初期化してください(前回の測定値が保持されるため)。初期化せずに測定を行いますと、前回の測定値を加算した結果が表示されます。



### 4.5 オプションメニューの説明

この章では、プロファイルウィンドウ上で右クリックすることで表示されるオプションメニューの一部について説明します。



(1) [表示設定]の[未実行関数を表示しない]を選択すると、実行した関数のみ表示することができます。

Function/Variable	F/V	Address	Size	Times	Option1	Option2
RESET_Vectors	V	00000000	H'00000010	0	0	0
__dtors	V	FFFF8424	H'00000004	0	0	0
__freeptr	V	FFFF8420	H'00000004	0	0	0
__head	V	FFFF841C	H'00000004	0	0	0
__new_handler	V	FFFF8418	H'00000004	0	0	0
\$_1493_20_dtors_done\$3	V	FFFF8414	H'00000004	0	0	0
\$_use_patch_info	V	FFFF8410	H'00000004	0	0	0
__ec2p_new_handler	V	FFFF840C	H'00000004	0	0	0
__rnext	V	FFFF8408	H'00000004	0	0	0
__\$brk	V	FFFF8404	H'00000004	0	0	0
__\$needed_destruction_head	V	FFFF8400	H'00000004	0	0	0
__\$heap_area	V	FFFF8000	H'00000400	0	0	0
Sample::change (volatile long *)	F	000020A6	H'00000042	1	0	0
Sample::sort (volatile long *)	F	00002032	H'00000074	1	0	0

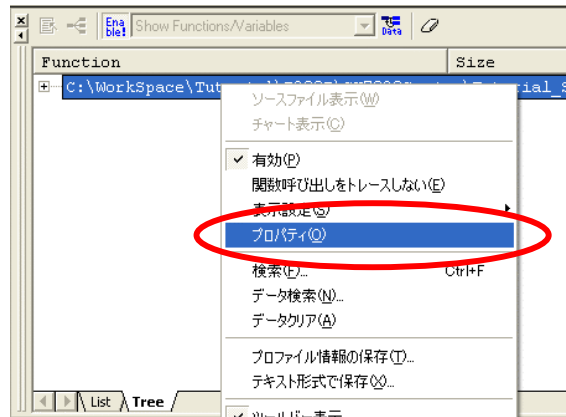
[未実行関数を表示しない]を選択すると

Function/Variable	F/V	Address	Size	Times	Option1	Option2
Sample::change (volatile long *)	F	000020A6	H'00000042	1	0	0
Sample::sort (volatile long *)	F	00002032	H'00000074	1	0	0
Sample::Sample ()	F	00002000	H'00000032	1	0	0
__divlu	F	00001408	H'00000000	1	0	0
__\$morecor	F	0000130C	H'0000005C	1	0	0
__malloc	F	0000127C	H'00000000	1	0	0
__free	F	000011E4	H'00000000	1	0	0
__rand	F	000011BC	H'00000028	10	0	0
operator new (unsigned long)	F	00001158	H'00000064	1	0	0
__CALL_INIT	F	00001108	H'00000028	1	0	0
000010B0	F	000010B0	H'00000000	1	0	0
__main	F	0000101C	H'0000006C	1	0	0
__\$brk	F	00001000	H'0000001C	1	0	0
PowerON_Reset_PC	F	00000800	H'0000002E	1	0	0

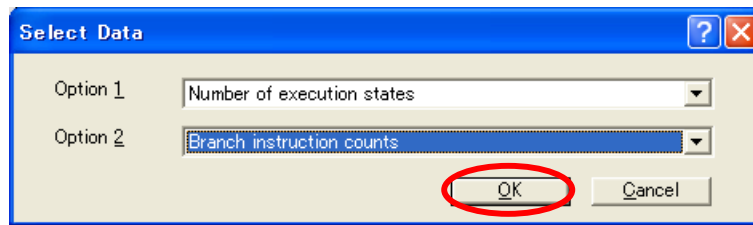
アクセス回数“0”の関数は省かれて表示される

最適化リンカージェネリタが出力するスタック情報ファイル(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。

- (2) [表示設定]の[子関数の実行結果を含んで表示]を選択すると、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んで表示することができます。オプションメニューから[プロパティ]を選択してください。



[Select Data]ダイアログボックスが表示され、測定項目を追加することができます。ここでは、[Option1]に“Number of execution states”、[Option2]に“Branch instruction counts”を設定して[OK]を押してください。



追加できる測定項目の内容は下記のとおりです。

“Elapsed time”	→ 実行サイクル数を測定
“Number of execution states”	→ 実行ステート数を測定
“Branch instruction counts”	→ 分岐命令回数を測定
“Number of execution instructions”	→ 実行命令数を測定
“Exception/interrupt counts”	→ 例外・割り込み回数を測定
“Interrupt counts”	→ 割り込み回数を測定
“URAM area access counts”	→ URAM エリア命令・データアクセス回数を測定
“URAM area instruction access counts”	→ URAM エリア命令アクセス回数を測定
“URAM area data access counts”	→ URAM エリアデータアクセス回数を測定

プログラムを実行させてプロファイルウィンドウに測定結果を表示させてください。このとき、測定結果を見やすくするため、オプションメニューの[未実行関数を表示しない]はチェックを入れてください。

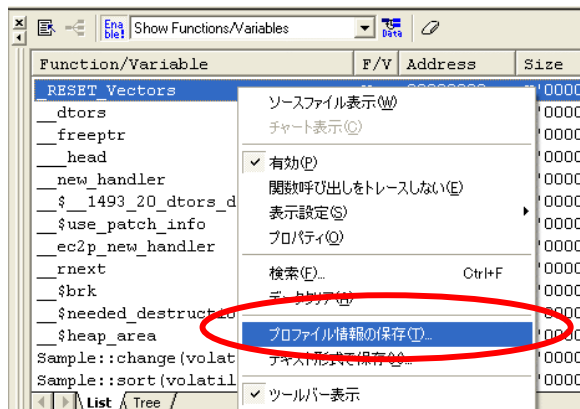
Function/Variable	F/V	Address	Size	Times	Option1	Option2
Sample::change (volatile long *)	F	000020A6	H'00000042	1	212	38
Sample::sort (volatile long *)	F	00002032	H'00000074	1	639	120
Sample::Sample ()	F	00002000	H'00000032	1	26	4
__divlu	F	00001408	H'00000000	1	20	2
__\$morecor	F	0000130C	H'0000005C	1	36	8
__malloc	F	0000127C	H'00000000	1	64	12
__free	F	000011E4	H'00000000	1	15	6
__rand	F	000011BC	H'00000028	10	177	40
operator new(unsigned long)	F	00001158	H'00000064	1	28	6
__CALL_INIT	F	00001108	H'00000028	1	17	6
000010B0	F	000010B0	H'00000000	1	1113	546
__main	F	0000101C	H'0000006C	1	118	42
__sbrk	F	00001000	H'0000001C	1	7	2
PowerON_Reset_PC	F	00000800	H'0000002E	1	14	4

[子関数の実行結果を含んで表示]を選択すると

Function/Variable	F/V	Address	Size	Times	Option1	Option2
Sample::change (volatile long *)	F	000020A6	H'00000042	1	212	38
Sample::sort (volatile long *)	F	00002032	H'00000074	1	639	120
Sample::Sample ()	F	00002000	H'00000032	1	196	40
__divlu	F	00001408	H'00000000	1	20	2
__\$morecor	F	0000130C	H'0000005C	1	63	12
__malloc	F	0000127C	H'00000000	1	142	30
__free	F	000011E4	H'00000000	1	15	6
__rand	F	000011BC	H'00000028	10	177	40
operator new(unsigned long)	F	00001158	H'00000064	1	170	36
__CALL_INIT	F	00001108	H'00000028	1	17	6
000010B0	F	000010B0	H'00000000	1	1113	546
__main	F	0000101C	H'0000006C	1	1342	280
__sbrk	F	00001000	H'0000001C	1	7	2
PowerON_Reset_PC	F	00000800	H'0000002E	1	1373	290

各関数内で呼び出された子関数の実行結果が含まれた測定結果が表示される。例えば、関数“\_main”は実行ステート数が 118 から 1342 に増えていることが確認できる。

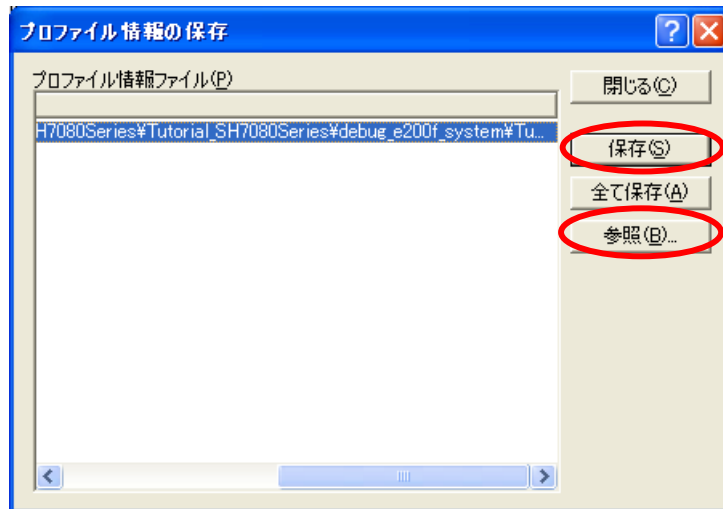
- (3) プロファイル測定結果は、プロファイル情報ファイル(拡張子 “.pro” )として保存することができます。オプションメニューから[プロファイル情報の保存]を選択してください。



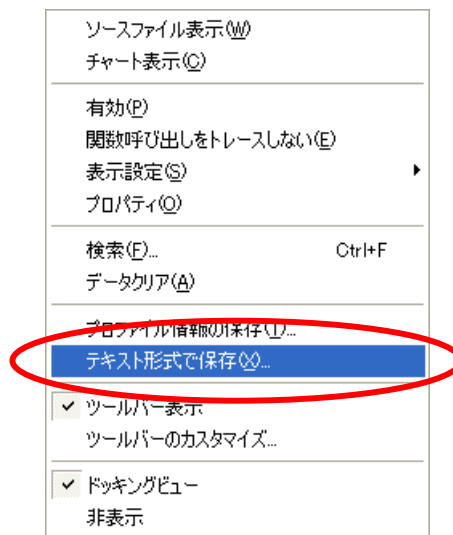
最適化リンケージエディタは、プロファイル情報を元に、ユーザプログラムの最適化を行うことができます。但し、オプションメニューの[関数呼び出しをトレースしない]にチェックして測定した結果のプロファイル情報では、最適化リンケージエディタによる最適化は行えません。詳細は、最適化リンケージエディタのマニュアルを参照ください。



[プロファイル情報の保存]ダイアログボックスが表示されるので、[参照]ボタンからディレクトリを選択して[保存]ボタンを押してください。指定したディレクトリにプロファイル情報ファイル(拡張子 “.pro”)が生成されます。



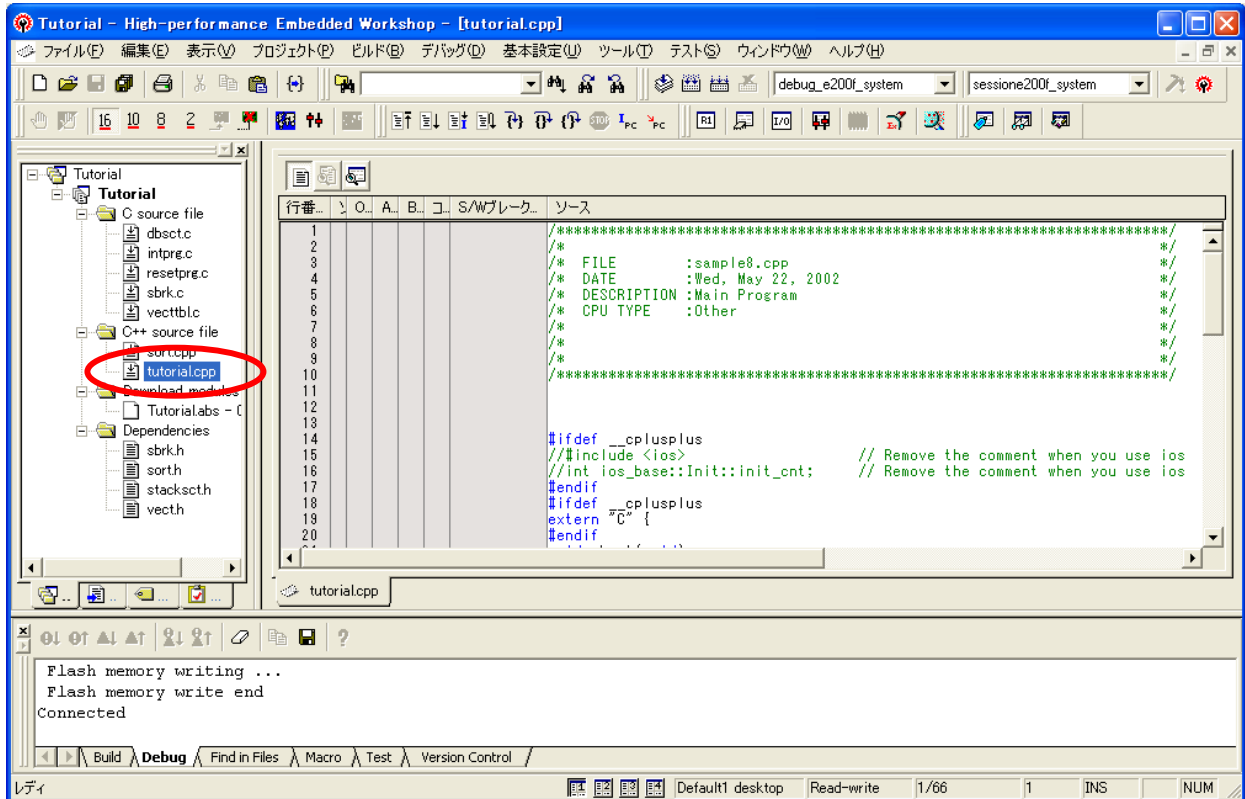
オプションメニューの[テキスト形式で保存]を選択すると、プロファイル測定結果の内容をテキスト形式で保存することもできます。



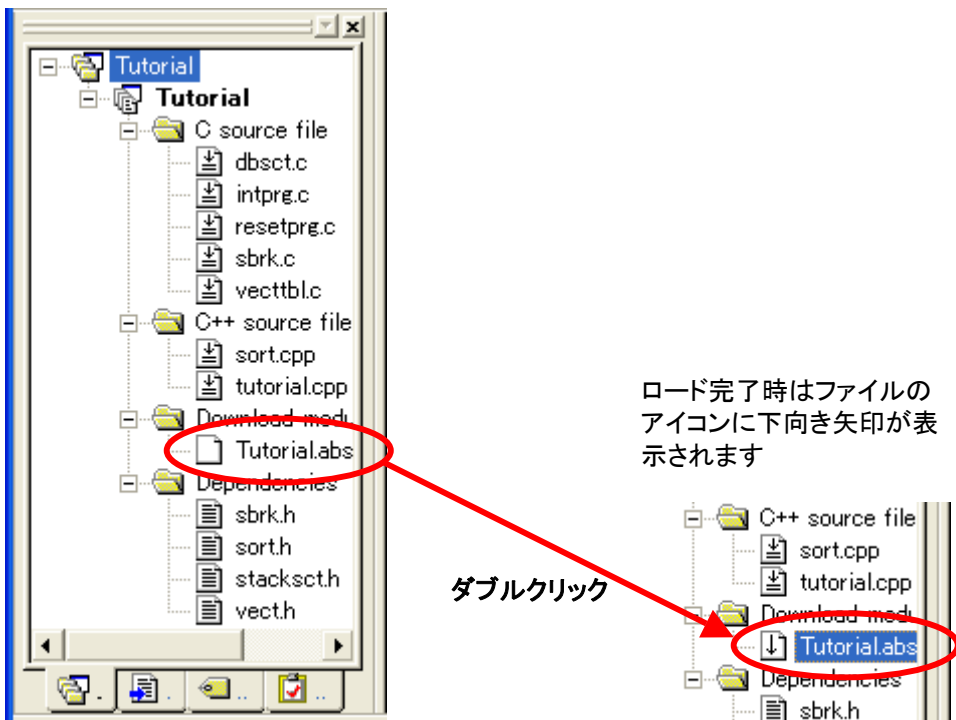
### 4.6 リアルタイムプロファイル測定方法

この章では、ソフトウェアブレークを使ったリアルタイムプロファイル測定を行う方法を説明します。

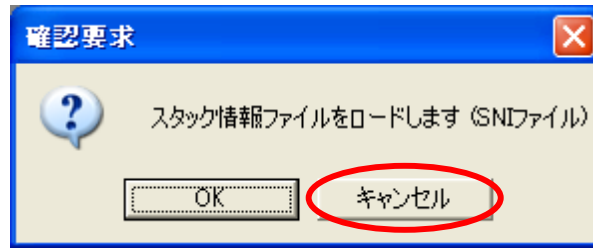
- (1) ワークスペース内のソースファイル “tutorial.cpp” をダブルクリックしてソースウィンドウに表示させてください。



- (2) プログラムをダウンロードするため、ワークスペースのロードモジュールファイル(拡張子.abs)をダブルクリックしてください。

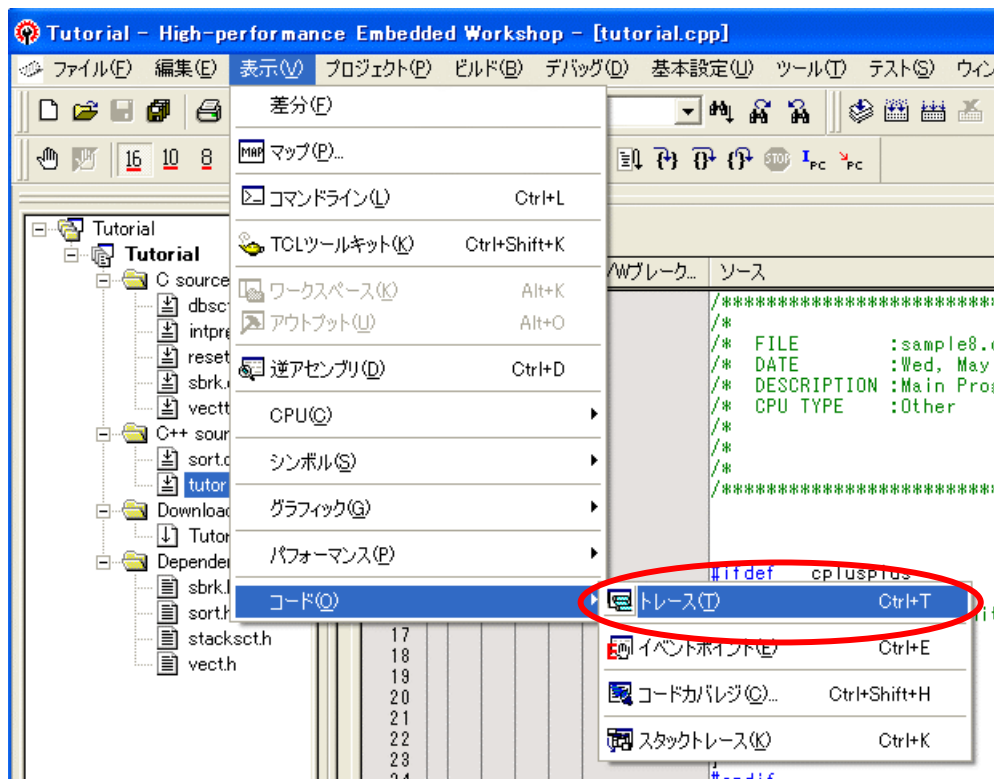


- (3) [確認要求]ダイアログボックスが表示された場合は[キャンセル]を押してください。



上記ダイアログボックスは、スタック情報ファイルの作成・オプション設定を行っていないと表示されません。詳細は「4.3 章 スタック情報ファイルのロード」をご参照ください。

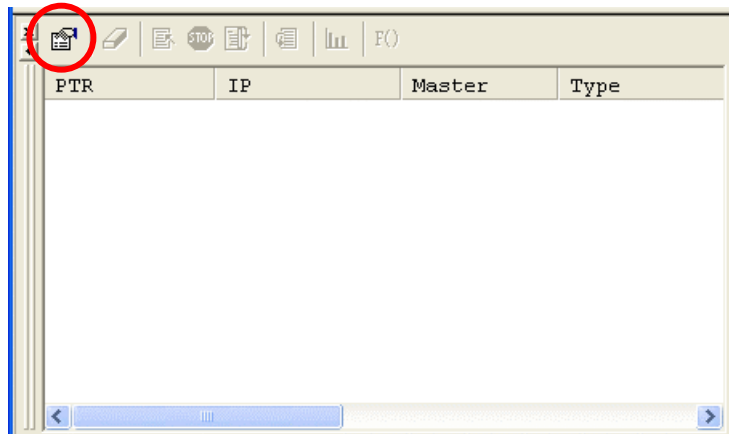
- (4) トレースモードを AUD トレースモードに設定します。[表示]メニューの[コード]の中の[トレース]を選択してください。



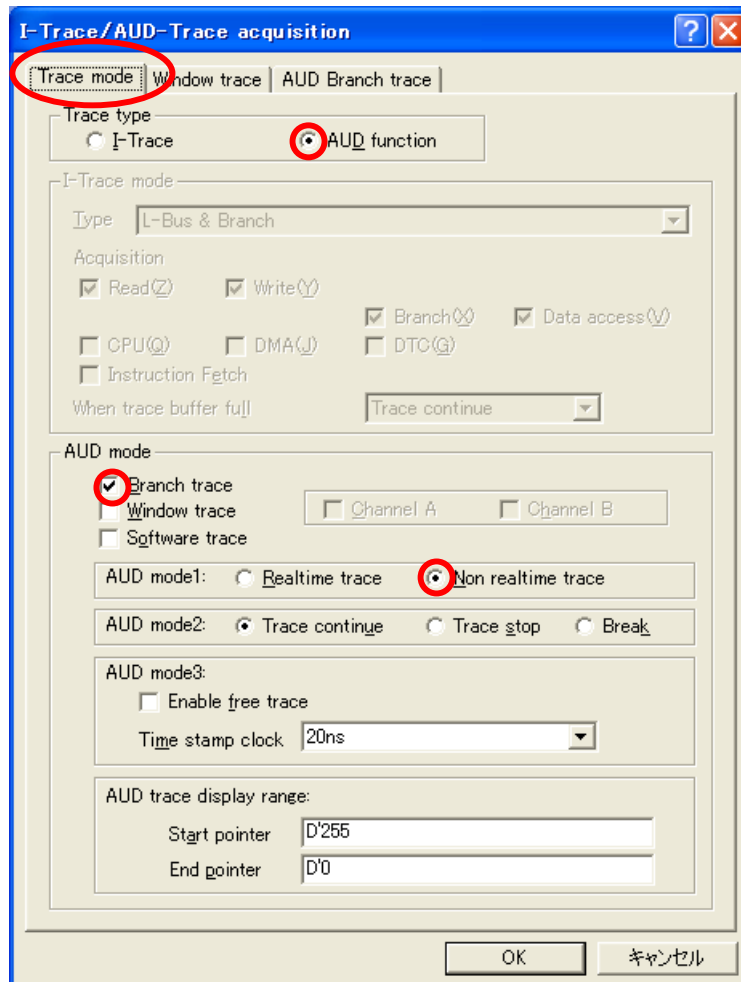
- (5) [トレースウィンドウの選択]ダイアログボックスが表示されるので、“Internal/AUD”を選択して[OK]を押してください。



(6) トレースウィンドウが開くので、左上にある[設定]アイコンをクリックしてください。

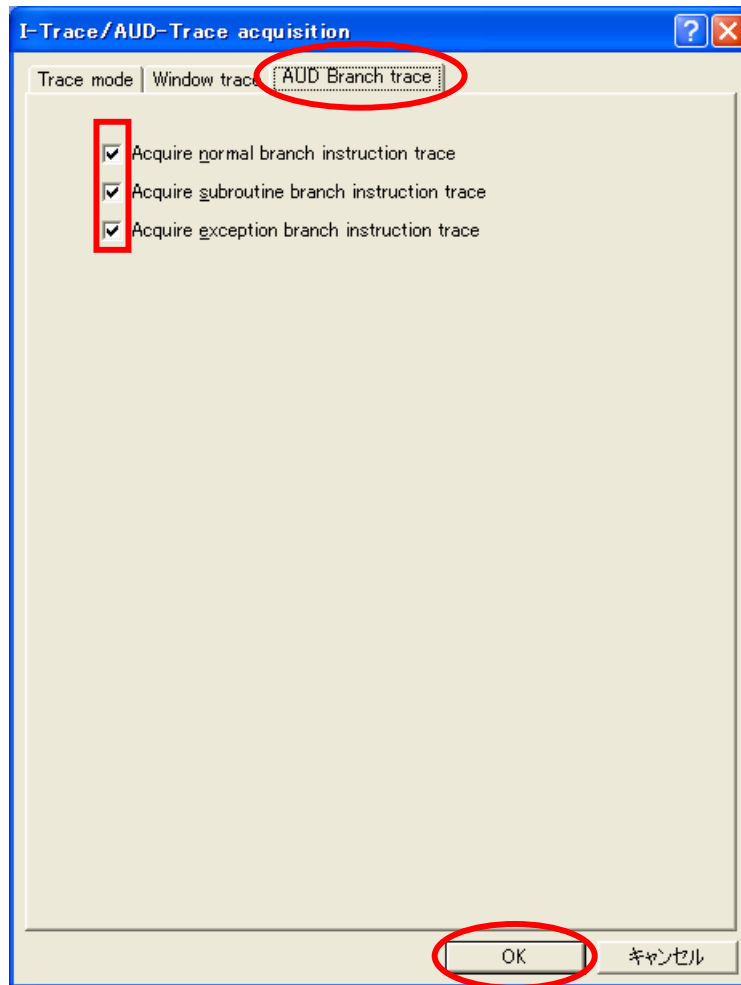


(7) [I-Trace/AUD-Trace acquisition]ダイアログボックスが表示されるので、[Trace mode]タブ内の[Trace type]に“AUD function”を選択して、[AUD mode]の“Branch trace”にチェックを入れてください。また、[AUD mode1]は“Non realtime trace”にチェックを入れてください。(※1)



※1. [AUD mode1]を“Realtime trace”に設定するとトレースデータを消失することがあり、正しくリアルタイムプロファイル測定が行えない場合があります。

[AUD Branch trace]タブを選択して、すべての項目にチェックを入れて[OK]を押してください。



【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[I-Trace/AUD-Trace acquisition]ダイアログボックス内で以下の設定を行ってください。

[Trace mode]タブ内

[Trace type] → “AUD function”を選択

[AUD mode] → “Branch trace”チェックボックスをチェック

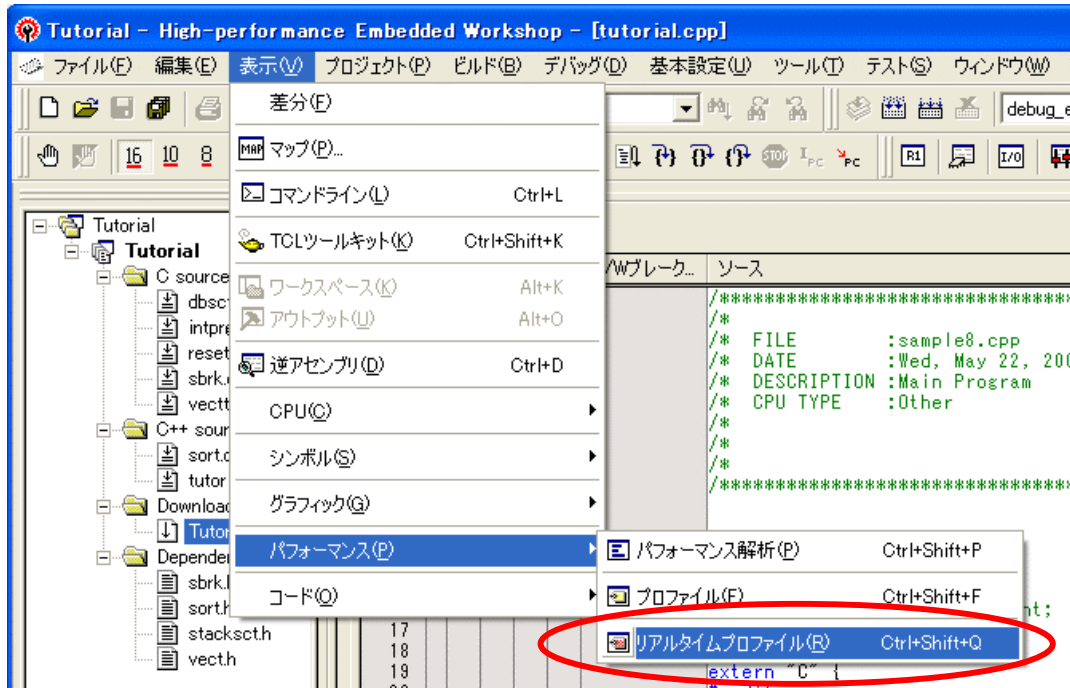
[AUD Branch trace]タブ内

[Acquire normal branch instruction trace] → チェックボックスをチェック

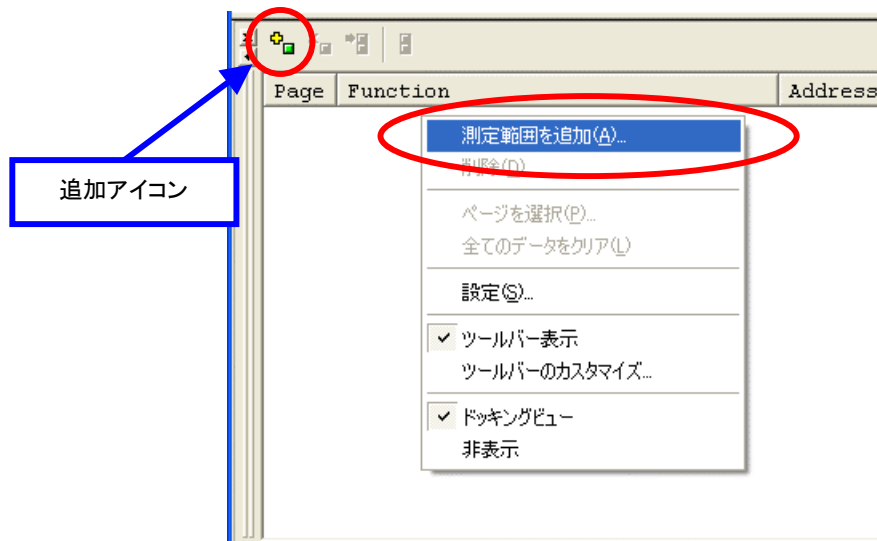
[Acquire subroutine branch instruction trace] → チェックボックスをチェック

[Acquire exception branch instruction trace] → チェックボックスをチェック

- (8) リアルタイムプロファイルウィンドウを開きます。[表示]メニューの[パフォーマンス]の中の[リアルタイムプロファイル]を選択してください。

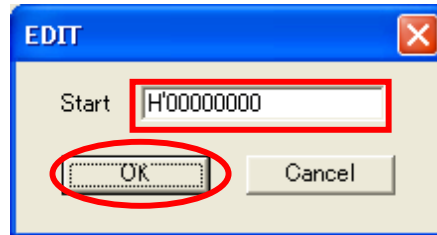


- (9) リアルタイムプロファイルウィンドウ上で右クリックして、[測定範囲を追加]を選択してください。



リアルタイムプロファイルウィンドウの左上にある[追加]アイコンをクリックしても同じ作業が行えます。

- (10) [EDIT]ダイアログボックスが表示されるので、リアルタイムプロファイルの測定範囲を指定します。今回は“H'00000000”を入力して[OK]を押してください。



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。(本書では、拡張プロファイルユニットを使用しておりません)

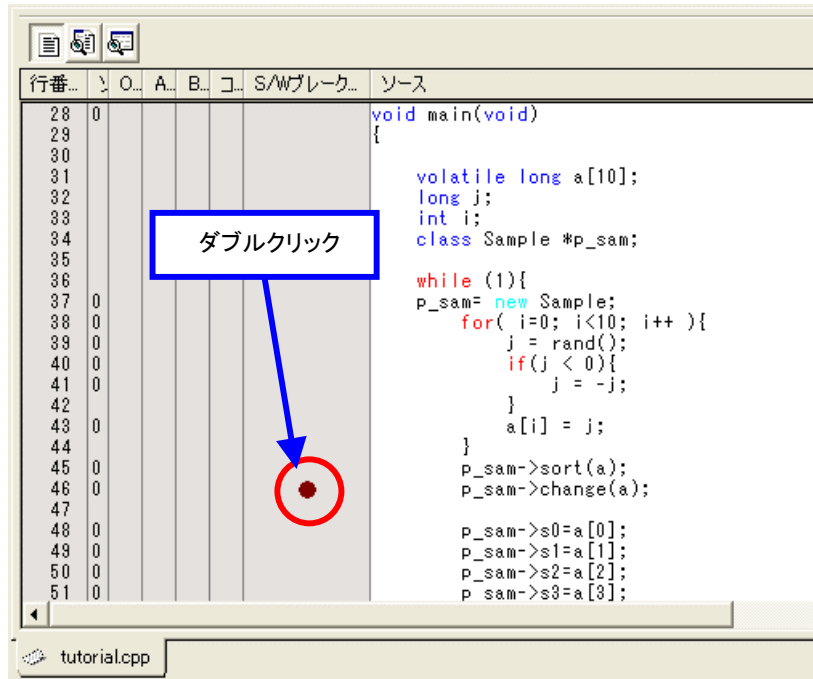
- (11)リアルタイムプロファイルウィンドウに各関数の情報が表示されます。まだ測定は行っていないので“Count”と“Time”の項目はゼロとなっています。

Page	Function	Address	Size	Count	Time
1	_PowerON_Reset_PC	00000800	H'2E	0	0000h000min000s000ms000us000ns
1	_Manual_Reset_PC	0000082E	H'1A	0	0000h000min000s000ms000us000ns
1	_INT_Illegal_code	00000848	H'4	0	0000h000min000s000ms000us000ns
1	_Dummy	0000084C	H'4	0	0000h000min000s000ms000us000ns
1	_sbrk	00001000	H'1C	0	0000h000min000s000ms000us000ns
1	_main	0000101C	H'6C	0	0000h000min000s000ms000us000ns
1	_abort	00001088	H'18	0	0000h000min000s000ms000us000ns
1	__CALL_INIT	00001108	H'28	0	0000h000min000s000ms000us000ns
1	__CALL_END	00001130	H'14	0	0000h000min000s000ms000us000ns
1	operator delete(void *)	00001144	H'14	0	0000h000min000s000ms000us000ns
1	operator new(unsigned long)	00001158	H'64	0	0000h000min000s000ms000us000ns
1	_rand	000011BC	H'28	0	0000h000min000s000ms000us000ns
1	__free	000011E4	H'98	0	0000h000min000s000ms000us000ns
1	__malloc	0000127C	H'90	0	0000h000min000s000ms000us000ns
1	__default_new_handler()	00001368	H'4	0	0000h000min000s000ms000us000ns
1	__call_dtors()	0000136C	H'7C	0	0000h000min000s000ms000us000ns
1	__process_needed_destructions()	000014DC	H'34	0	0000h000min000s000ms000us000ns
1	__already_marked_for_destruction	00001510	H'6	0	0000h000min000s000ms000us000ns
1	__record_needed_destruction	00001516	H'26	0	0000h000min000s000ms000us000ns
1	__std_needed_destruction_list	0000153C	H'C	0	0000h000min000s000ms000us000ns
1	__abort_execution	00001548	H'C	0	0000h000min000s000ms000us000ns
1	Sample::Sample()	00002000	H'32	0	0000h000min000s000ms000us000ns
1	Sample::sort(volatile long *)	00002032	H'74	0	0000h000min000s000ms000us000ns
1	Sample::change(volatile long *)	000020A6	H'42	0	0000h000min000s000ms000us000ns

表示内容は下記のとおりです。

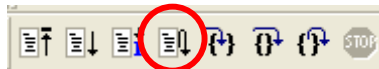
- “Address” → 関数を配置しているメモリ上のアドレス
- “Size” → 関数のサイズ
- “Count” → 関数をコールした回数
- “Time” → 実行時間の累積

- (12) プログラムを実行する前にソフトウェアブレークの設定を行います。今回は、ソースファイル“tutorial.cpp”の行番号 46 の[S/W ブレークポイント]カラムをダブルクリックして、ソフトウェアブレークを設定してください。

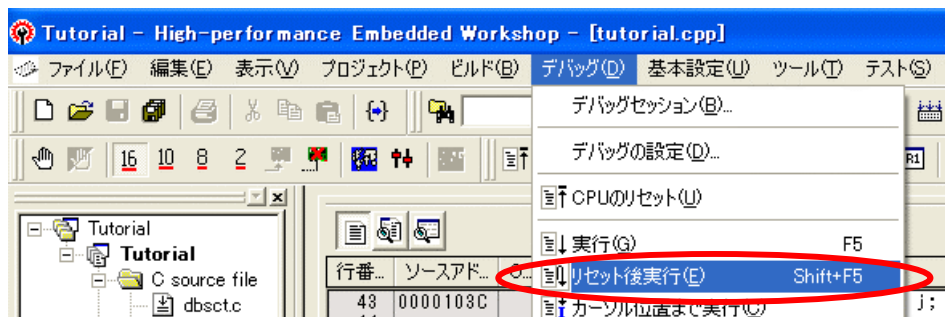


内容としては、プログラムを実行すると main 関数内の change 関数を実行する前にブレークをかけてプログラムを停止させます。

- (13) プログラムを実行してリアルタイムプロファイル測定を開始します。[リセット後実行]アイコンをクリックして、プログラムを実行してください。

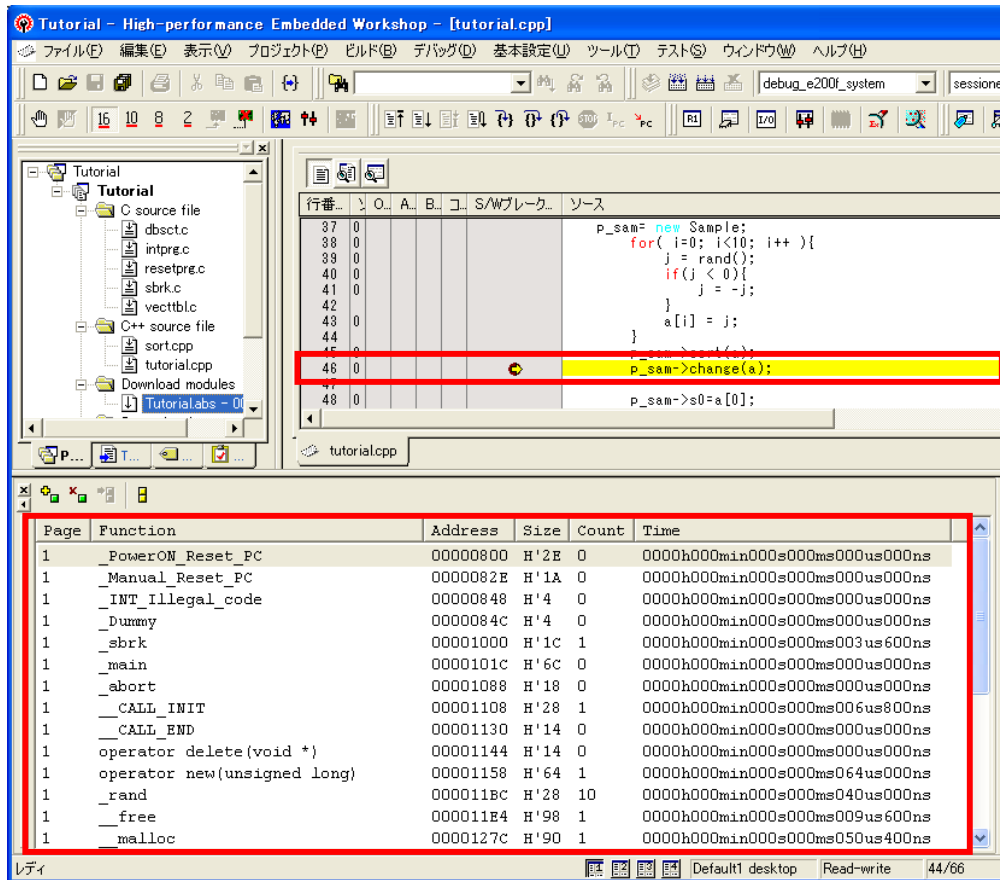


リセット後実行は、[デバッグ]メニューからも行えます。





(14) ブレーク設定した行でプログラムが停止して、リアルタイムプロファイルウィンドウに測定結果が表示されます。



測定結果を確認すると、rand 関数の“Count”が“10”、change 関数の“Count”が“0”になっていることが確認できます。プログラム上で rand 関数は 10 回コールされており、プログラムを実行すると change 関数が実行される前にブレークがかかり停止するので正しく測定できていることがわかります。

Page	Function	Address	Size	Count	Time
1	PowerON_Reset_PC	00000800	H'2E	0	0000h000min000s000ms000us000ns
1	Manual_Reset_PC	0000082E	H'1A	0	0000h000min000s000ms000us000ns
1	__INT_Illegal_code	00000848	H'4	0	0000h000min000s000ms000us000ns
1	__Dummy	0000084C	H'4	0	0000h000min000s000ms000us000ns
1	__sbrk	00001000	H'1C	1	0000h000min000s000ms003us600ns
1	__main	0000101C	H'6C	0	0000h000min000s000ms000us000ns
1	__abort	00001088	H'18	0	0000h000min000s000ms000us000ns
1	__CALL_INIT	00001108	H'28	1	0000h000min000s000ms006us800ns
1	__CALL_END	00001130	H'14	0	0000h000min000s000ms000us000ns
1	operator delete(void *)	00001144	H'14	0	0000h000min000s000ms000us000ns
1	operator new(unsigned long)	00001158	H'64	1	0000h000min000s000ms0064us000ns
1	rand	000011BC	H'28	10	0000h000min000s000ms040us000ns
1	__free	000011E4	H'98	1	0000h000min000s000ms009us600ns
1	__malloc	0000127C	H'90	1	0000h000min000s000ms050us400ns
1	default_new_handler()	00001368	H'4	0	0000h000min000s000ms000us000ns
1	__call_dtors()	0000136C	H'7C	0	0000h000min000s000ms000us000ns
1	__process_needed_destructions()	000014DC	H'34	0	0000h000min000s000ms000us000ns
1	__already_marked_for_destruction	00001510	H'6	0	0000h000min000s000ms000us000ns
1	__record_needed_destruction	00001516	H'26	0	0000h000min000s000ms000us000ns
1	__std_needed_destruction_list	0000153C	H'C	0	0000h000min000s000ms000us000ns
1	__abort_execution	00001548	H'C	0	0000h000min000s000ms000us000ns
1	Sample::Sample()	00002000	H'32	1	0000h000min000s000ms071us600ns
1	Sample::sort(volatible long *)	00002032	H'74	1	0000h000min000s000ms178us000ns
1	Sample::change(volatible long *)	000020A6	H'42	0	0000h000min000s000ms000us000ns

以上でプロファイル機能とリアルタイムプロファイル機能の使い方の説明を終わります。

## 5. よくある質問

### 5.1 プロファイル機能を使用する上で注意すべき事がありますか？

- (1) 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て、測定データを収集してから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るため、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。
- (4) 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限が戻るため、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時の分岐命令の実行がトレースウイ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\*\*” と表示されます。
- (5) 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。但し、オプションメニューの[関数呼び出しをトレースしない]に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。

### 5.2 プロファイル機能使用中に使用できなくなる機能はありますか？

- (1) 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) プロファイル機能イネーブル時には、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) 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) 内蔵トレース機能で利用できる継続トレース機能は、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。プロファイルデータを正常に測定できません。
- (5) 内蔵トレース、AUDトレース、外部バストレースの 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。
- (6) 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。

### 5.3 リアルタイムプロファイル機能を使用する上で注意すべき事がありますか？

- (1) R5F70834A/R5F70835A/R5F70844A/R5F70845A/R5F70854A/R5F70855A/R5F70865A/R5E71464R/R5F71464R/R5F71494R/R5F71491R/SH7124/SH7125 のエバチップユニット接続なしでのデバッグの場合、AUD トレース機能は使用できません。AUD トレースの出力がないため、AUD Eventpoint 機能の条件が成立しません。また、AUD パフォーマンス測定機能、コードカバレッジ機能、リアルタイムプロファイル機能も使用できません。  
R5F70834A/R5F70835A/R5F70844A/R5F70845A/R5F70854A/R5F70855A/R5F70865A/R5E71464R/R5F71464R/R5F71494R/R5F71491R/SH7124/SH7125 のエバチップユニット接続ありでのデバッグの場合はこれらの機能を使用することができます。
- (2) メインユニットは 512kbyte × 8 ブロックの測定用メモリ(Main)を 1 つ搭載しています。また、拡張プロファイルユニットは 512kbyte × 8 ブロックの測定用メモリ 2 つ(Option\_1、Option\_2)を搭載しています。次の点にご注意ください。
  - ・ Main のメモリに設定した測定対象領域は Option\_1、Option\_2 の領域からは、測定対象領域外になります。
  - ・ Option\_1 のメモリに設定した測定対象領域は Main、Option\_2 の領域からは、測定対象領域外になります。
  - ・ Option\_2 のメモリに設定した測定対象領域は Main、Option\_1 の領域からは、測定対象領域外になります。
- (3) Nest モードで使用した場合に、測定対象外の関数が測定対象内の関数を呼び出し(呼び出し元アドレス:N)、その後、測定対象内の関数が測定対象外の呼び出し元に戻る場合、呼び出し元の戻りアドレスが N+4 の場合のみ、測定対象内の関数を正しく測定できます。戻りアドレスが N+4 以外の場合は正しく測定できません。
- (4) 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。
- (5) 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。
- (6) リアルタイムプロファイル機能は、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。

## 6. 関連ドキュメント

E200F エミュレータおよび、HEW には本書で取り上げた機能以外にも便利な機能を豊富に備えています。各製品の仕様の詳細、技術情報、制限事項など有用な情報を記載していますので下記の関連ドキュメントも合わせて参照してください。

### 【E200F エミュレータ関連ドキュメント】

- SH-2A、SH-2 E200F エミュレータユーザーズマニュアル
- SH-2A SH-2 E200F エミュレータ ご使用上の注意事項
- SH-2A、SH-2 E200F エミュレータユーザーズマニュアル 別冊 SH7086 ご使用時の補足説明

### 【High-performance Embedded Workshop 関連ドキュメント】

- High-performance Embedded Workshop ユーザーズマニュアル
- High-performance Embedded Workshop リリースノート

### 【CPU 関連ドキュメント】

- SH7080 グループ ハードウェアマニュアル
- SH-2A、SH2A-FPU ソフトウェアマニュアル

### 【SH7080 シリーズ C コンパイラパッケージ関連ドキュメント】

- SuperH ファミリ用 C/C++コンパイラ、アセンブラ、最適化リンケージエディタ コンパイラパッケージユーザーズマニュアル
- SuperH ファミリ用 C/C++コンパイラパッケージ アプリケーションノート

本製品に関する情報は以下のルネサス・ウェブサイトをご覧ください:

日本サイト: <http://japan.renesas.com/e200f>

グローバルサイト: <http://www.renesas.com/e200f>

## ホームページとサポート窓口

ルネサステクノロジホームページ

<http://japan.renesas.com/>

お問い合わせ先

<http://japan.renesas.com/inquiry>[csc@renesas.com](mailto:csc@renesas.com)

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2007.10.25	—	初版発行

**本資料ご利用に際しての留意事項**

1. 本資料は、お客様に用途に応じた適切な弊社製品をご購入いただくための参考資料であり、本資料中に記載の技術情報について弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的、あるいはその他軍事用途の目的で使用しないでください。また、輸出に際しては、「外国為替及び外国貿易法」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、海底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会下さい。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、その他お気づ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。