

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情報の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。  
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）  
特定水準： 航空機器、航空宇宙機器、海中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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

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

# H8S、H8/300 シリーズ シミュレータ／デバッガ

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



## 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。



---

# はじめに

---

## このマニュアルについて

このマニュアルは、High-performance Embedded Workshop (HEW)に付属しているシミュレータ・デバッガのデバッグ機能について説明します。

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。

HEW 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、HEW アプリケーションの日本語版について説明します。

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

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

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

その他、記載されている製品名は各社の商標または登録商標です。

## このマニュアルの記号

このマニュアルで使われている記号の意味を説明します。

表 1: 記号一覧

記号	意味
[Menu->Menu Option]	太字と '->' はメニューオプションを示します (例 [File->Save As...])
FILENAME.C	大文字の名前はファイル名を示します
“ <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



---

# 目次

---

1.	はじめに	1
2.	シミュレータ・デバッガの機能	3
2.1	特長	3
2.2	デバッグ対象プログラム	3
2.3	シミュレーション範囲	4
2.4	メモリ管理	5
2.5	命令実行リセット処理	5
2.6	例外処理	6
2.7	H8S/2000CPU 特殊機能	6
2.8	H8S/2600CPU 特殊機能	6
2.9	H8SX シリーズ CPU 特殊機能	7
2.10	制御レジスタ	7
2.11	タイマ (H8S シリーズ、H8SX シリーズ)	8
2.11.1	サポート範囲	8
2.11.2	制御レジスタ	8
2.11.3	クロック	8
2.11.4	タイマを使用する	8
2.11.5	タイマ使用時の注意	9
2.12	トレース	9
2.13	標準入出力およびファイル入出力処理	9
2.14	命令実行サイクル数の計算	10
2.15	ブレイク条件	11
2.16	浮動小数点データ	13
2.17	関数呼び出し履歴の表示	14
2.18	パフォーマンス測定	14
2.18.1	プロファイラ	14
2.18.2	パフォーマンス解析	14
2.19	擬似割込み	14
2.20	カバレッジ	15
3.	デバッグ	17
3.1	ワークスペースの作成	17
3.1.1	ターゲットの選択	17
3.1.2	シミュレータ用ワークスペースの設定	19
3.2	シミュレータ・デバッガの設定を変更する	20

3.2.1	シミュレータシステムの設定を変更する.....	20
3.2.2	メモリマップおよびメモリリソースの設定を変更する.....	22
3.2.3	メモリマップ設定ダイアログボックス.....	23
3.2.4	メモリリソース設定ダイアログボックス.....	24
3.3	周辺機能シミュレーションを設定する.....	25
3.3.1	周辺機能シミュレーションモジュールを登録する.....	25
3.3.2	周辺機能のアドレスを変更する.....	26
3.3.3	周辺機能の割り込み要因情報を変更する.....	27
3.3.4	割り込み優先順位レジスタのメモリリソースを確保する.....	28
3.3.5	接続されている周辺機能を確認する.....	28
3.4	シミュレータ・デバッガのブレークポイントを使用する.....	28
3.4.1	ブレークポイントを一覧表示する.....	29
3.4.2	ブレークポイントを設定する.....	31
3.4.3	ブレークポイントの設定内容を変更する.....	38
3.4.4	ブレークポイントを有効にする.....	38
3.4.5	ブレークポイントを無効にする.....	39
3.4.6	ブレークポイントを削除する.....	39
3.4.7	ブレークポイントをすべて削除する.....	39
3.4.8	ブレークポイントのソース行を表示する.....	39
3.4.9	入出力ファイルを閉じる.....	39
3.4.10	入出力ファイルをすべて閉じる.....	39
3.5	トレース情報を見る.....	39
3.5.1	トレースウィンドウを開く.....	39
3.5.2	トレース情報取得条件を設定する.....	39
3.5.3	トレース情報を取得する.....	40
3.5.4	トレース情報を検索する.....	41
3.5.5	トレース情報をクリアする.....	42
3.5.6	トレース情報をファイルに保存する.....	42
3.5.7	ソースファイルを表示する.....	42
3.5.8	ソース表示を整形する.....	42
3.5.9	統計情報を解析する.....	43
3.6	プロファイル情報を見る.....	44
3.6.1	スタック情報ファイル.....	44
3.6.2	プロファイル情報ファイル.....	45
3.6.3	スタック情報ファイルのロード.....	46
3.6.4	プロファイルを有効にする.....	47
3.6.5	測定方法を指定する.....	47
3.6.6	ユーザプログラムを実行し結果を確認する.....	47
3.6.7	List シート.....	47
3.6.8	Tree シート.....	48
3.6.9	プロファイルチャートウィンドウ.....	51
3.6.10	表示データの種類および用途.....	52
3.6.11	プロファイル情報ファイルを作成する.....	53
3.6.12	注意事項.....	54
3.7	パフォーマンスを解析する.....	54
3.7.1	パフォーマンス解析ウィンドウを開く.....	54

3.7.2	評価関数を設定する .....	55
3.7.3	データ収集を開始する .....	55
3.7.4	データをリセットする .....	55
3.7.5	評価関数を削除する .....	55
3.7.6	すべての評価関数を削除する .....	55
3.7.7	現在の表示内容をセーブする .....	55
3.8	コードカバレッジを測定する .....	56
3.8.1	カバレッジウィンドウを開く .....	56
3.8.2	カバレッジ情報をすべて取得する .....	58
3.8.3	カバレッジ情報をすべてクリアする .....	58
3.8.4	ソースファイルを表示する .....	58
3.8.5	新しいカバレッジ測定範囲を設定する .....	58
3.8.6	カバレッジ測定範囲を変更する .....	58
3.8.7	カバレッジ測定範囲を削除する .....	60
3.8.8	カバレッジ情報を取得する .....	60
3.8.9	カバレッジ情報をクリアする .....	60
3.8.10	カバレッジ情報をファイルに保存する .....	60
3.8.11	カバレッジ情報をファイルからロードする .....	61
3.8.12	最新の情報に更新する .....	61
3.8.13	Confirmation Request ダイアログボックス .....	61
3.8.14	カバレッジ情報を保存ダイアログボックス .....	62
3.8.15	エディタウィンドウへのカバレッジ結果表示 .....	63
3.8.16	逆アセンブリウィンドウへのカバレッジ結果表示 .....	64
3.9	手動で擬似割込みを発生させる .....	65
3.9.1	トリガウィンドウ .....	65
3.9.2	GUI I/O ウィンドウ .....	67
3.10	標準入出力およびファイル入出力を行う .....	68
3.10.1	I/O シミュレーションウィンドウを開く .....	68
3.10.2	入出力機能 .....	69
3.11	仮想入出力パネルを作成する .....	71
3.11.1	GUI 入出力ウィンドウを開く .....	71
3.11.2	ボタンを作成する .....	72
3.11.3	ラベルを作成する .....	73
3.11.4	LED を作成する .....	75
3.11.5	テキスト文字列を作成する .....	78
3.11.6	アイテムのサイズ/配置を変更する .....	78
3.11.7	アイテムをコピーする .....	79
3.11.8	アイテムを削除する .....	79
3.11.9	グリッド線を表示する .....	79
3.11.10	入出力パネル情報を保存する .....	80
3.11.11	入出力パネル情報を読み込む .....	80
4.	ウィンドウ .....	81
5.	コマンドライン .....	83
5.1	コマンド一覧 (機能別) .....	83
5.1.1	実行関連 .....	83

5.1.2	ダウンロード関連 .....	83
5.1.3	レジスタ操作関連 .....	84
5.1.4	メモリ操作関連 .....	84
5.1.5	アセンブル/逆アセンブル関係 .....	84
5.1.6	ブレーク設定関連 .....	84
5.1.7	トレース関連 .....	85
5.1.8	カバレッジ計測関連 .....	85
5.1.9	パフォーマンス測定関連 .....	85
5.1.10	ウォッチ関連 .....	85
5.1.11	スクリプト/ログファイル関連 .....	86
5.1.12	マップ関連 .....	86
5.1.13	シミュレーションの設定関連 .....	86
5.1.14	標準入出力およびファイル入出力関連 .....	86
5.1.15	ユーティリティ関連 .....	87
5.1.16	プロジェクト/ワークスペース関連 .....	87
5.1.17	テスト支援機能関連 .....	87
5.2	コマンド一覧 (アルファベット順) .....	88
6.	メッセージ一覧 .....	91
6.1	インフォメーションメッセージ .....	91
6.2	エラーメッセージ .....	92
7.	チュートリアル .....	95
7.1	デバッグの準備 .....	95
7.1.1	サンプルプログラム .....	95
7.1.2	サンプルプログラムの作成 .....	95
7.2	デバッグのための設定 .....	96
7.2.1	メモリリソースの確保 .....	96
7.2.2	サンプルプログラムのダウンロード .....	97
7.2.3	ソースプログラムの表示 .....	98
7.2.4	PC ブレークポイントの設定 .....	99
7.2.5	プロファイラの設定 .....	100
7.2.6	I/O シミュレーションの設定 .....	101
7.2.7	トレース情報取得条件の設定 .....	102
7.2.8	スタックポインタ、プログラムカウンタの設定 .....	103
7.3	デバッグ開始 .....	103
7.3.1	プログラムの実行 .....	103
7.3.2	トレースバッファの使い方 .....	106
7.3.3	トレース検索の実行 .....	107
7.3.4	I/O シミュレーションの確認 .....	108
7.3.5	ブレークポイントの確認 .....	108
7.3.6	変数の参照 .....	109
7.3.7	プログラムのステップ実行 .....	110
7.3.8	プロファイラ情報の確認 .....	113

---

# 1. はじめに

---

シミュレータ・デバッガは、ルネサステクノロジのマイクロコンピュータのCPUシミュレーション機能およびデバッグ機能を持っています。シミュレータ・デバッガを利用することによって、C/C++言語やアセンブリ言語で作成されたプログラムを効率よくデバッグすることができます。

シミュレータ・デバッガは、High-performance Embedded Workshop (HEW)とともに動作します。HEWは、ルネサステクノロジ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かれ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シミュレータ・デバッガのアクセス、計測、および変更に関して、HEWは高機能でしかも直観的な手段を提供することを目的としています。

シミュレータ・デバッガを使用するにあたっては、シミュレータ・デバッガ、およびHEWのヘルプも参照ください。

## 1. はじめに

---

---

## 2. シミュレータ・デバッガの機能

---

本章では、H8S,H8/300シリーズ シミュレータ・デバッガの機能について説明します。

### 2.1 特長

本シミュレータ・デバッガには、次のような特長があります。

- (1) ホスト計算機上で動作するので、実機がなくてもプログラムのデバッグを開始することができます。システム全体の開発期間を短縮できます。
- (2) シミュレーション時にプログラムの命令実行サイクル数を計算します。これにより実機がなくても性能評価が行えます。
- (3) 擬似割込み機能、およびI/Oシミュレーション機能を持ち、簡易的なシステムレベルシミュレーションを行えます。
- (4) 下記のような機能を持ち、プログラムのテスト、およびデバッグを効率よく進めることができます。
  - H8S、H8/300 シリーズの各 CPU に対応
  - ユーザプログラムの実行中に異常が発生した場合、異常を無視して続行するか、または停止するかを制御する機能
  - プロファイルデータ取得、および関数単位のパフォーマンス測定
  - 豊富なブレーク機能
  - メモリマップの設定・編集
  - 関数呼び出し履歴の表示
  - C/C++およびアセンブラソースレベルのカバレジ表示
  - イメージ表示、波形表示による視覚的デバッグ機能
- (5) Windows®上で動作し、ブレークポイント・メモリマップ・パフォーマンス・トレースをダイアログボックス上で設定することができます。H8S、H8/300マイコンの各々のメモリマップに対応した環境設定もダイアログボックス上で行うことができます。

また、下記のような特徴を持ちます。

- 直観的なユーザインタフェース
- オンラインヘルプ
- 共通した表示と操作性

### 2.2 デバッグ対象プログラム

シミュレータ・デバッガでは、ELF/DWARF2 フォーマットのロードモジュールがシンボリックデバッグ可能です。その他のフォーマットのロードモジュールについては、ダウンロードと命令実行はできますが、シンボリックデバッグはできません。詳しくは、「High-performance Embedded Workshop V.4.03 ユーザーズマニュアル」の「17.18.2 Elf/Dwarf2 のサポート」を参照してください。

### 2.3 シミュレーション範囲

- (1) H8/300、H8/300L、H8/300H、H8S/2600、H8S/2000およびH8SXシリーズのシミュレーションをサポートします。
- (2) シミュレータ・デバッガは、H8SX、H8S、H8/300シリーズマイコンの下記機能をサポートしています。
- 全実行命令
  - 例外処理
  - レジスタ
  - 全アドレス空間
  - 表 2-1に示す各 CPU モード

表2-1 プラットフォームと CPU モードの対応

デバッグ プラットフォーム名	対応 CPU
H8/300 Simulator	H8/300
H8/300L Simulator	H8/300L
H8/300HA Simulator	H8/300H アドバンスモード
H8/300HN Simulator	H8/300H ノーマルモード
H8S/2600A Simulator	H8S/2600 アドバンスモード
H8S/2600N Simulator	H8S/2600 ノーマルモード
H8S/2000A Simulator	H8S/2000 アドバンスモード
H8S/2000N Simulator	H8S/2000 ノーマルモード
H8SX Normal Simulator	H8SX ノーマルモード
H8SX Advanced Simulator	H8SX アドバンスモード
H8SX Maximum Simulator	H8SX マキシマムモード

- (3) シミュレータ・デバッガは、H8SX、H8S、H8/300シリーズマイコンの下記機能をサポートしていません。下記機能を使用したプログラムは、H8SX、H8S、H8/300シリーズ用エミュレータを使用してデバッグしてください。
- デュアルポート RAM
  - タイマ (H8/300 シリーズのみ未サポート)
  - パルス幅変換器 (PWM)
  - シリアルコミュニケーションインタフェース (SCI)
  - A/D 変換器
  - I/O ポート
  - 割込みコントローラ



## 2.4 メモリ管理

### (1) メモリマップの設定

メモリマップは、シミュレーション時のメモリアクセスサイクル数の計算に使用します。設定できる項目は次の通りです。

- メモリ種別
- メモリ領域の先頭位置、終了位置
- メモリアクセスのサイクル数
- メモリのデータバス幅

詳細は、「3.2.2 メモリマップおよびメモリリソースの設定を変更する」を参照してください。  
なお、ユーザプログラムは、内蔵 I/O 空間を除くすべてのメモリで実行可能です。

### (2) メモリリソースの設定

ユーザプログラムをロードして実行させるためにメモリリソースを設定する必要があります。設定できる項目は以下の通りです。

- 開始アドレス
- 終了アドレス
- アクセス種別

アクセス種別は、読み書き可能、読み出しのみ可能、書き込みのみ可能があります。

ユーザプログラムで、読み出しのみ可能メモリへ書き込みを行う等の不正なアクセスを行ったときはエラーとなるので、誤ったメモリアクセスを検出することができます。

EEPROM については、他のメモリと異なりアクセス種別が読み出しのみ可能な場合でも、EEPROM 命令により書き込みできます。反対に書き込み可能でも、EEPROM 命令以外では書き込みできません。

メモリリソース設定の詳細は、「3.2.2 メモリマップおよびメモリリソースの設定を変更する」を参照してください。

## 2.5 命令実行リセット処理

シミュレータ・デバッガでは、以下の場合に命令実行数および命令実行サイクル数をリセットします。

命令シミュレーション停止後再実行までにPCを変更した  
実行開始アドレスを指定したRunコマンドを実行した  
イニシャライズまたはプログラムをロードした

### 2.6 例外処理

シミュレータ・デバッガでは、TRAPA 命令 (H8/300H、H8S、および H8SX シリーズのみ)、トレース例外 (H8S、H8SX シリーズのみ) の発生を検出し、例外処理をシミュレーションします。これにより、例外発生時のシミュレーションも行うことができます。

例外処理のシミュレーションは、次の手順で行います。

- (a) 命令の実行中に例外の発生を検出します。
- (b) スタック領域に PC と CCR を退避します。EXR の有効ビットが ON のときには EXR も退避します。退避処理でエラーが発生した場合は、例外処理を中止し、例外処理エラーが発生したことを表示後、シミュレータ・デバッガのコマンド待ちに戻ります。
- (c) CCR の I ビットを 1 にセットします。
- (d) ベクタ番号に対応するベクタアドレスから、スタートアドレスを読み出します。読み出し処理でエラーが発生した場合は、例外処理を中止し、例外処理エラーが発生したことを表示後、シミュレータ・デバッガのコマンド待ちに戻ります。
- (e) スタートアドレスから命令実行を行います。スタートアドレスが 0 の場合は、例外処理を中止し、例外処理エラーが発生したことを表示後、シミュレータ・デバッガのコマンド待ち状態に戻ります。

### 2.7 H8S/2000CPU 特殊機能

H8S/2000 シリーズ シミュレータ・デバッガでは割り込みモードをサポートします。

割り込みモードは、SYSCR の INTM1、INTM0 ビットで設定します。

SYSCR は、INTM1、INTM0 ビットのみサポートします。

SYSCR のアドレスの初期値は H'FF39 ですが、[シミュレータシステム] ダイアログボックスの [SYSCR アドレス] で変更できます。変更したアドレスは、セッションファイルに保存します。

設定可能な割り込みモードは CPU により異なりますが、シミュレータ・デバッガでは、モード 0、モード 1、モード 2 をサポートします。

イベント機能により割り込みを発生する場合のコントロールレベルは下記ようになります。

- ・ [ブレイク種別の選択] ダイアログボックスの [動作種別] フィールドで、[割り込み] を選択し、[優先順位] に 0 を設定  
コントロールレベルは 0
- ・ [ブレイク種別の選択] ダイアログボックスの [動作種別] フィールドで、[割り込み] を選択し、[優先順位] に 0 以外を設定  
コントロールレベルは 1

### 2.8 H8S/2600CPU 特殊機能

#### (1) MAC 命令

H8S/2600CPU では、積和演算 (MAC 命令) が行えます。この命令では飽和演算、非飽和演算が選択できます。本シミュレータ・デバッガでは内蔵 I/O の SYSCR レジスタのビット 7 (以下 MACS ビットとする) の値により判定します。

MACS ビット	0 :	非飽和演算
	1 :	飽和演算

## (2) EXR レジスタ

H8S/2600CPU では EXR レジスタを使用できます。また、このレジスタの有効/無効を設定することもできます。本シミュレータ・デバッガでは内蔵 I/O の SYSCR レジスタのビット 5(以下 EXR ビットとする)の値により判定します。

EXR ビット            0 : EXR 無効  
                         1 : EXR 有効

SYSCR アドレスは、[シミュレータシステム]ダイアログボックスの [SYSCR アドレス]で設定します。

**【注】** SYSCR アドレスは内蔵 I/O に設定してください。内蔵 I/O 以外に SYSCR アドレスを設定している場合は MACS ビットを 0 (非飽和)、EXR ビットを 0 (EXR 無効)と判断しますのでご注意ください。

詳しくは、「3.2.1 シミュレータシステムの設定を変更する」を参照してください。

## 2.9 H8SX シリーズ CPU 特殊機能

### (1) 乗算器、除算器の有効 / 無効切替

乗算器および除算器の有効 / 無効を切り替えられます。これにより、乗算器、除算器有無によるプログラムパフォーマンスの違いを確認できます。乗算器および除算器の有効 / 無効は、[シミュレータシステム]ダイアログボックスの [乗算器]および[除算器]で設定します。

### (2) フェッチモード

16ビットフェッチおよび32ビットフェッチを指定できます。これにより、フェッチサイズによるプログラムパフォーマンスの違いを確認できます。SYSCRレジスタのFETCHMDビットで指定してください。

### (3) エンディアン

MCUモードがアドバンストまたはマキシマムモードでアドレス空間サイズが16Mまたは256Mの場合のみENDIANCRレジスタによりエンディアンを設定できます。

メモリ上のデータ格納形式としてビッグエンディアンの他にリトルエンディアンのバイト順をサポートしています。これにより、リトルエンディアンで作成したユーザプログラムのシミュレーション、デバッグが可能になります。

### (4) 割り込み制御選択モード

INTCR(割り込みコントロールレジスタ)のINTM1ビットのみサポートします。本ビットにより割り込み制御モードを選択できます。

## 2.10 制御レジスタ

H8S/2600 シリーズおよび H8SX シリーズでは、メモリにマッピングした制御レジスタとして、SYSCR (システムコントロールレジスタ) をサポートしています。これにより、積和演算、EXR アクセスを行っているユーザプログラムのシミュレーション、デバッグを行うことができます。

SYSCR アドレスは、[シミュレータシステム]ダイアログボックスの [SYSCR アドレス]で設定します。制御レジスタの変更や表示は [IO]ウィンドウをご利用ください。

詳しくは、「3.2.1 シミュレータシステムの設定を変更する」、「High-performance Embedded

Workshop V.4.03 ユーザーズマニュアル」の「17.6 I/Oレジスタを見る」を参照してください。

### 2.11 タイマ (H8S シリーズ、H8SX シリーズ)

#### 2.11.1 サポート範囲

H8S シリーズおよび H8SX シリーズシミュレータ・デバッガでは 16 ビットタイマパルスユニット 0 のチャンネル 0 をサポートしています。

また、オーバフロー/コンペアマッチによる割り込みだけをサポートしており、インプットキャプチャ等の端子入出力をともなう機能はサポートしていません。

#### 2.11.2 制御レジスタ

シミュレータ・デバッガではタイマの制御レジスタをサポートしています。シミュレータ・デバッガでサポートしている制御レジスタを表 2-2 に示します。

表中サポート状況の はサポート、 は「2.11.1 サポート範囲」で説明した機能に関するビットのみサポートしています。

タイマ制御レジスタのアドレス、割り込みベクタ番号、および割り込み優先順位レジスタ位置は[周辺モジュールの構成]ダイアログボックスで参照・変更することができます。[周辺モジュールの構成]ダイアログボックスの詳細については、「3.3 周辺機能シミュレーションを設定する」を参照してください。

表2-2 シミュレータ・デバッガでサポートするタイマ制御レジスタ

タイマ名	サポートする制御レジスタ	サポート状況
TPU0	TSTR	
	TCR	
	TIER	
	TSR	
	TCNT	
	TGRA	
	TGRB	
	TGRC	
	TGRD	

#### 2.11.3 クロック

シミュレータ・デバッガでは、メモリアクセスにかかわる内部クロック、周辺機能用クロック、タイマを動かすクロックをサポートします。

メモリマップで指定するサイクル数は内部クロックになります。内部クロックと周辺機能用クロックの比は、[周辺機能シミュレーションの設定]ダイアログボックスで設定してください。

タイマを動かすクロックの分周率は、タイマ制御レジスタで指定してください。

#### 2.11.4 タイマを使用する

タイマを使用するにはシミュレータ・デバッガ起動時に表示される[周辺機能シミュレーションの設定]ダイアログボックスでタイマモジュールを登録する必要があります。

タイマモジュール登録の詳細については、「3.3 周辺機能シミュレーションを設定する」を参照してください。

### 2.11.5 タイマ使用時の注意

- (1) 0 クリアだけ可能なビットも、1 をライト可能となっています。
- (2) タイマ割り込み発生時にブレークするか、しないかを選択できます。

[シミュレータシステム]ダイアログボックスまたは EXEC\_STOP\_SET コマンドで設定してください。

## 2.12 トレース

シミュレータ・デバッガは、実行結果をトレースバッファに書き込みます。トレース情報の取得条件は、[トレース取得]ダイアログボックスで指定します。[トレース取得]ダイアログボックスは、[トレース]ウィンドウ上で右クリックしてポップアップメニューを表示し、[設定...]を選択することによって表示できます。取得したトレース情報は、[トレース]ウィンドウに表示します。

トレース情報は検索することができます。検索条件は、[トレース検索]ダイアログボックスで設定します。[トレース検索]ダイアログボックスは、[トレース]ウィンドウ上で右クリックしてポップアップメニューを表示し、[検索...]を選択することによって表示できます。

詳しくは、「3.5 トレース情報を見る」を参照してください。

## 2.13 標準入出力およびファイル入出力処理

シミュレータ・デバッガでは、ユーザプログラムから標準入出力およびファイル入出力を行うことができます。入出力機能を利用する場合は、必ず [I/O シミュレーション]ウィンドウをオープンしておいてください。

サポートしている入出力処理は以下の通りです。機能コードには、16 ビットアドレス版、24 ビットアドレス版、32 ビットアドレス版があります。使用する CPU に合わせて選択してください。

表2-3 入出力機能一覧

番号	機能コード	機能名	内容
1	H'01 (16 ビットアドレス) H'11 (24 ビットアドレス) H'21 (32 ビットアドレス)	GETC	標準入力からの 1 バイト入力
2	H'02 (16 ビットアドレス) H'12 (24 ビットアドレス) H'22 (32 ビットアドレス)	PUTC	標準出力への 1 バイト出力
3	H'03 (16 ビットアドレス) H'13 (24 ビットアドレス) H'23 (32 ビットアドレス)	GETS	標準入力からの 1 行入力
4	H'04 (16 ビットアドレス) H'14 (24 ビットアドレス) H'24 (32 ビットアドレス)	PUTS	標準出力への 1 行出力
5	H'05 (16 ビットアドレス) H'15 (24 ビットアドレス) H'25 (32 ビットアドレス)	FOPEN	ファイルのオープン

## 2. シミュレータ・デバッガの機能

番号	機能コード	機能名	内 容
6	H'06	FCLOSE	ファイルのクローズ
7	H'07 (16 ビットアドレス) H'17 (24 ビットアドレス) H'27 (32 ビットアドレス)	FGETC	ファイルからの1バイト入力
8	H'08 (16 ビットアドレス) H'18 (24 ビットアドレス) H'28 (32 ビットアドレス)	FPUTC	ファイルへの1バイト出力
9	H'09 (16 ビットアドレス) H'19 (24 ビットアドレス) H'29 (32 ビットアドレス)	FGETS	ファイルからの1行入力
10	H'0A (16 ビットアドレス) H'1A (24 ビットアドレス) H'2A (32 ビットアドレス)	FPUTS	ファイルへの1行出力
11	H'0B	FEOF	エンドオブファイルのチェック
12	H'0C	FSEEK	ファイルポインタの移動
13	H'0D	FTELL	ファイルポインタの現在位置を得る

入出力機能の詳細は、「3.10 標準入出力およびファイル入出力を行う」を参照してください。

### 2.14 命令実行サイクル数の計算

シミュレータ・デバッガでの命令実行サイクル数計算は、H8S、H8/300 シリーズ プログラミングマニュアルに記載している計算式とメモリマップのデータバス幅およびアクセスステート数を使用して行います。ただし、以下の理由によりシミュレータ・デバッガで計算した命令実行サイクル数と、実機でプログラムを実行した場合の命令実行サイクル数が異なることがあります。

#### (1) MOVFPE、MOVTPE 命令

E クロック同期命令のデータ転送サイクル数は、9～16 と幅を持った値となっています。シミュレータ・デバッガでは「11+オペランドアクセスサイクル数」で計算します。オペランドアクセスサイクル数は、メモリのデータバス幅およびアクセスステート数より求めます。

#### (2) EEPMOV 命令

EEPROM 書き込み専用命令のサイクル数は、命令読み出しに要するサイクル数と、データを転送するのに要するサイクル数の合計となります。

#### (3) SLEEP 命令

シミュレータ・デバッガでは、SLEEP 命令がプログラム停止用命令として使用している場合を考慮して、サイクル数を加算しません。

#### (4) 標準入出力およびファイル入出力処理

標準入出力およびファイル入出力処理は、シミュレータ・デバッガ固有の機能であるため、サイクル数に加算しません。なお、標準入出力およびファイル入出力処理とは、BSR、JSR 命令で I/O シミュレーションアドレスで指定した位置への分岐が完了してから入出力処理を行いコール元に戻るまでです。

## 2.15 ブレーク条件

ユーザプログラムのシミュレーションを中断する条件として以下のものがあります。

- ブレーク系コマンドの条件成立によるブレーク
- ユーザプログラムの実行時エラー検出によるブレーク
- トレースバッファ満杯によるブレーク
- SLEEP 命令実行によるブレーク
- [停止]ボタンによるブレーク

### (1) ブレーク系コマンドの条件成立によるブレーク

ブレーク条件を設定するコマンドには次の5種類があります。

- BREAKPOINT : 命令実行位置によるブレーク
- BREAK\_ACCESS : メモリ範囲のアクセスによるブレーク
- BREAK\_CYCLE : 実行サイクル数によるブレーク
- BREAK\_DATA : メモリ書き込みデータ値によるブレーク
- BREAK\_REGISTER : レジスタ書き込みデータ値によるブレーク
- BREAK\_SEQUENCE : 実行順序を指定したブレーク

ブレーク条件成立時の動作を[Stop]と指定した場合、そのブレーク条件が成立するとプログラムを中断します。詳しくは、「3.4 シミュレータ・デバッガのブレークポイントを使用する」を参照してください。

ユーザプログラム実行中にブレーク条件が成立しプログラムが中断した場合、ブレークポイントの命令を実行しないで停止するか、実行してから停止するかを表 2-4 に示します。

表2-4 ブレーク条件成立時の処理

コマンド名	ブレーク条件成立命令	
	実行する	実行しない
BREAKPOINT		
BREAK_ACCESS		
BREAK_CYCLE		
BREAK_DATA		
BREAK_REGISTER		
BREAK_SEQUENCE		

BREAKPOINT、BREAK\_SEQUENCE の場合、実行命令の先頭位置以外にブレークポイントを設定するとブレークを検出できません。

ユーザプログラム実行中にブレーク条件が成立すると、ブレーク条件成立のメッセージをステータスバーに表示して、命令実行を中断します。

### (2) ユーザプログラムの実行時エラー検出によるブレーク

シミュレータ・デバッガでは、CPU の例外発生機能では検出できないプログラムの誤りを検出するためにシミュレーションエラーを設けています。これらのエラーが発生した場合に、シミュレーションを停止するか、続行するかを [シミュレータシステム] ダイアログボックスにより選択できます。エラーの種類、エラーメッセージ、エラー発生要因、および続行時のシミュレータ・デバッガの動作を表 2-5 に示します。

## 2. シミュレータ・デバッガの機能

表2-5 シミュレーションエラー一覧

エラーの種類/メッセージ	エラー発生要因	続行モード時処理
アドレスエラー/Address Error	<ul style="list-style-type: none"> <li>PC 値が奇数</li> <li>内蔵 I/O 空間からの命令フェッチ</li> <li>奇数アドレスからのワードアクセス</li> <li>奇数アドレスからのロングワードアクセス</li> </ul>	デバイスと同一動作をする
メモリアクセスエラー/ Memory Access Error (Address:H'nnnnnn)	<ul style="list-style-type: none"> <li>確保していないメモリ領域をアクセスしようとした</li> <li>書き込み不可属性を持つメモリへ書き込みを行おうとした</li> <li>読み出し不可属性を持つメモリから読み出しを行おうとした</li> <li>メモリが存在しない領域をアクセスしようとした</li> <li>EEPROM 命令以外の命令での EEPROM 書き込み</li> </ul>	メモリへの書き込み時、何も書き込まない メモリ読み出し時、全ビット"1"を読み出す
不当命令/ Illegal Instruction	<ul style="list-style-type: none"> <li>命令ではないコードの実行</li> <li>MOV.B Rn,@-SP または MOV.B @SP+,Rn の実行</li> </ul>	常に停止する そのまま実行するが結果は保証しない
命令実行不正/ Illegal Operation	<ul style="list-style-type: none"> <li>DAA 命令、DAS 命令で CCR の C フラグ、H フラグと補正前の値の関係不正</li> <li>DIVXU 命令、DIVXS 命令のゼロ除算またはオーバフロー</li> </ul>	そのまま実行するが結果は保証しない

停止モードの場合、シミュレーションエラーが発生するとシミュレータ・デバッガは、命令実行を中止してエラーメッセージを表示後、コマンド待ち状態に戻ります。シミュレーションエラー停止後の PC の状態を表 2-6 に示します。なお、シミュレーションエラー停止後 CCR の内容は変化しません。

表2-6 シミュレーションエラー停止時のレジスタ

エラーの種類	PC の内容
アドレスエラー、 メモリアクセスエラー	命令読込時： エラーが発生した命令の先頭アドレス 命令実行時： エラーが発生した命令の次命令のアドレス
不当命令	エラーが発生した命令の先頭アドレス
命令実行不正	エラーが発生した命令の次命令のアドレス

シミュレーションエラーが発生する命令を組み込んだプログラムのデバッグは、次の手順で行ってください。

- 最初は停止モードで実行させて、意図している箇所以外にエラーがないかどうかを確認してください。
- 確認が完了したら、続行モードで実行してください。

**【注】** 停止モードでエラーが発生して停止した状態から、モードを続行モードに変更してシミュ



レーションを再開すると、正しくシミュレーションできない場合があります。シミュレーションを再開する場合は、レジスタ内容、メモリの内容をエラー発生前の状態に戻してから再実行するようにしてください。

### (3) トレースバッファ満杯によるブレーク

[トレース取得]ダイアログボックスの [トレースバッファ満杯時の動作]で [停止]モードを指定し、命令実行中にトレースバッファが満杯になると、シミュレータ・デバッガは、実行を中断します。中断時には以下のメッセージを[アウトプット]ウィンドウに表示します。

Trace Buffer Full

### (4) SLEEP 命令実行によるブレーク

命令実行時に、SLEEP 命令を実行すると、シミュレータ・デバッガは実行を中断します。中断時には、以下のメッセージを[アウトプット]ウィンドウに表示します。

Sleep

**【注】** 実行を再開する場合は、PC の値を再開位置の命令アドレスに変更してください。

### (5) [停止]ボタンによるブレーク

命令実行中にユーザにより強制的に実行を中断することができます。中断時には以下のメッセージをステータスバーに表示します。

Stop

Go、Step コマンドにより実行を再開できます。

## 2.16 浮動小数点データ

実数データとして浮動小数点数を指定することができます。これにより、データ値等で浮動小数点を扱う場合の操作が容易になります。浮動小数点を指定できる項目は次の通りです。

- ・[ブレーク種別の選択]ダイアログボックスにおいて、ブレーク種別を [ブレークデータ] や [ブレークレジスタ] と指定したときのデータ
- ・[メモリ]ウィンドウにおけるデータ
- ・[メモリフィル]ダイアログボックスにおけるデータ
- ・[メモリ検索]ダイアログボックスにおけるデータ
- ・レジスタ値編集ダイアログボックスでの入力値

浮動小数点データフォーマットは、ANSI C の浮動小数点フォーマットに準拠しています。

シミュレータ・デバッガでは、浮動小数点数の 10 進→2 進変換で発生する丸めのモードに RN (最近値丸め) を使用します。

なお、10 進→2 進変換および 2 進→10 進変換で非正規化数を指定した場合、非正規化数のまま処理します。また、10 進→2 進変換時にオーバフローが発生した場合、無限大を設定します。

### 2.17 関数呼び出し履歴の表示

シミュレーションの中断時に、関数の呼び出し履歴を [スタックトレース] ウィンドウに表示します。これにより、プログラムの動作の流れを確認することができます。また、[スタックトレース] ウィンドウ上で関数名を選択することにより、該当するソースプログラムを [エディタ] ウィンドウ上に表示します。これにより、中断している関数の他に、その関数を呼び出した元の関数をチェックすることができます。

関数呼び出し履歴を更新するのは、以下のような場合です。

- 「2.15 ブレーク条件」に示す条件によりシミュレーションが中断した時
- 上記中断した状態で、レジスタの値を変更した時
- シミュレーションをステップ実行している時

詳しくは、「High-performance Embedded Workshop V.4.03 ユーザーズマニュアル」の「17.15 関数呼び出し履歴を見る」を参照してください。

### 2.18 パフォーマンス測定

シミュレータ・デバッガはユーザプログラムのパフォーマンスを測定するためにプロファイラ機能およびパフォーマンス解析機能を提供します。

#### 2.18.1 プロファイラ

プロファイラは、ユーザプログラムの全体について関数とグローバル変数のアドレス、サイズ、関数の呼び出し回数、およびプロファイルデータを表示します。プロファイルデータは CPU により異なります。

プロファイル情報はリスト形式、ツリー形式、チャート形式で表示します。

プロファイル情報を用いることにより、サイズが小さく、呼び出し回数が多い関数をインライン関数にするなどの最適化を検討することが出来ます。

また、ファイルに出力したプロファイル情報を用いて、最適化リンケージエディタで動的情報に基づいた最適化を実行することができます。

詳しくは、「3.6 プロファイル情報を見る」および「最適化リンケージエディタマニュアル 4.2.4 最適化オプション PROfile」を参照してください。

#### 2.18.2 パフォーマンス解析

パフォーマンス解析はユーザプログラム内の指定関数について実行サイクル数、呼び出し回数を表示します。指定関数のみについてパフォーマンスデータを取得するため、プロファイラよりも高速なシミュレーションが可能です。詳しくは、「3.7 パフォーマンスを解析する」を参照してください。

### 2.19 擬似割り込み

シミュレータ・デバッガでは、シミュレーション中に擬似割り込みを発生させることができます。擬似割り込みを発生させるには、以下の 2 通りの方法があります。

- (1) ブレーク条件成立時の動作による擬似割り込み発生

ブレーク系コマンドで、ブレーク条件成立時の動作に [Interrupt] を指定することにより、擬似割込

みを発生させることができます。

詳しくは、「3.4 シミュレータ・デバッガのブレークポイントを使用する」を参照してください。

(2) ウィンドウによる擬似割り込み発生

[トリガ]ウィンドウ、または[GUI I/O]ウィンドウのボタンをクリックすることにより、擬似割り込みを発生させることができます。

詳しくは、「3.9 手動で擬似割り込みを発生させる」を参照してください。

なお、擬似割り込みが発生してからその割り込みを受け付けるまでの間に、次の擬似割り込みが発生した場合は、優先順位の高い割り込みだけを受け付けます。

(3) 擬似割り込み発生によるブレーク

擬似割り込み発生時にブレークするか、しないかを選択できます。

[シミュレータシステム]ダイアログボックスまたはEXEC\_STOP\_SETコマンドで設定してください。

**【注】** 擬似割り込みでは、割り込みを受け付けるかどうかの判定には、ベクタ番号ではなく割り込み情報の[優先順位]を使用します。優先順位にH'8以上を指定した場合は、常に割り込みを受け付けます。また、割り込みコントローラはシミュレーションしていません。

## 2.20 カバレッジ

シミュレータ・デバッガでは、ユーザが指定した測定範囲について命令実行中に命令カバレッジ情報を収集できます。

測定範囲は直接アドレスを指定して設定するほかに、ソースファイル名を指定してそのファイルに含まれる全関数を設定することができます。

命令カバレッジ情報を利用することで各命令の実行状態を観察できます。さらにプログラムのどの部分が未実行であるかを容易に特定できます。

収集した命令カバレッジ情報は[カバレッジ]ウィンドウに表示します。

命令カバレッジ情報は命令実行済のソース行に対応するカラムを [エディタ]ウィンドウ上に強調表示します。

また、測定対象のアドレス範囲または関数について、カバレッジ統計情報をパーセント形式で表示します。これによりプログラムがどのくらい実行されているかを定量的に把握できます。

命令カバレッジ情報はファイルへの保存およびファイルからのロードが行えます。ロードできるのは".COV"ファイル形式のみです。

詳しくは、「3.8 コードカバレッジを測定する」を参照してください。



---

## 3. デバッグ

---

この章では、シミュレータ・デバッガ特有の操作と関連するウィンドウおよびダイアログボックスについて説明します。

HEWで共通な下記機能については、HEWのヘルプを参照ください。

- ・ デバッグの準備
- ・ プログラムを表示する
- ・ メモリ内容を参照 / 設定する
- ・ メモリ内容を波形形式で表示する
- ・ メモリ内容を画像形式で表示する
- ・ 任意のアドレスのメモリ内容を参照 / 設定する
- ・ I/Oレジスタを参照 / 設定する
- ・ レジスタを参照 / 設定する
- ・ プログラムを実行 / 停止、リセットする
- ・ 現在の状態の参照
- ・ 複数デバッグプロットフォームの同期
- ・ コマンドラインインタフェースのデバッグ
- ・ Elf / Dwarf2のサポート
- ・ ラベルを参照 / 設定する

### 3.1 ワークスペースの作成

シミュレータ・デバッガを使用するためには、まずHEWでワークスペースを作成する必要があります。ここでは、シミュレータ・デバッガ特有の説明のみをします。ワークスペース作成の詳細は、HEWのマニュアルを参照ください。

#### 3.1.1 ターゲットの選択

HEWを起動し、新規プロジェクトワークスペースの作成を行う場合、下記ダイアログボックスを表示します。

ここで、シミュレータ・デバッガのターゲットを選択してください。

### 3. デバッグ



図 3-1 デバッガターゲット設定画面 (7/9)

[ターゲット] デバッガターゲットを設定します。デバッガターゲットを選択(チェック)してください。デバッガターゲットは未選択でも複数選択してもかまいません。

[ターゲットタイプ] [ターゲット]に表示するターゲットの種類を指定します。

【注】 2/9 で選択した[オペレーティングモード] と違う動作モードのデバッガターゲットを選択した場合、デバッグが出来ない場合があります。

### 3.1.2 シミュレータ用ワークスペースの設定

新規プロジェクト-8/9では、シミュレータ用ワークスペースの設定を行うことができます。

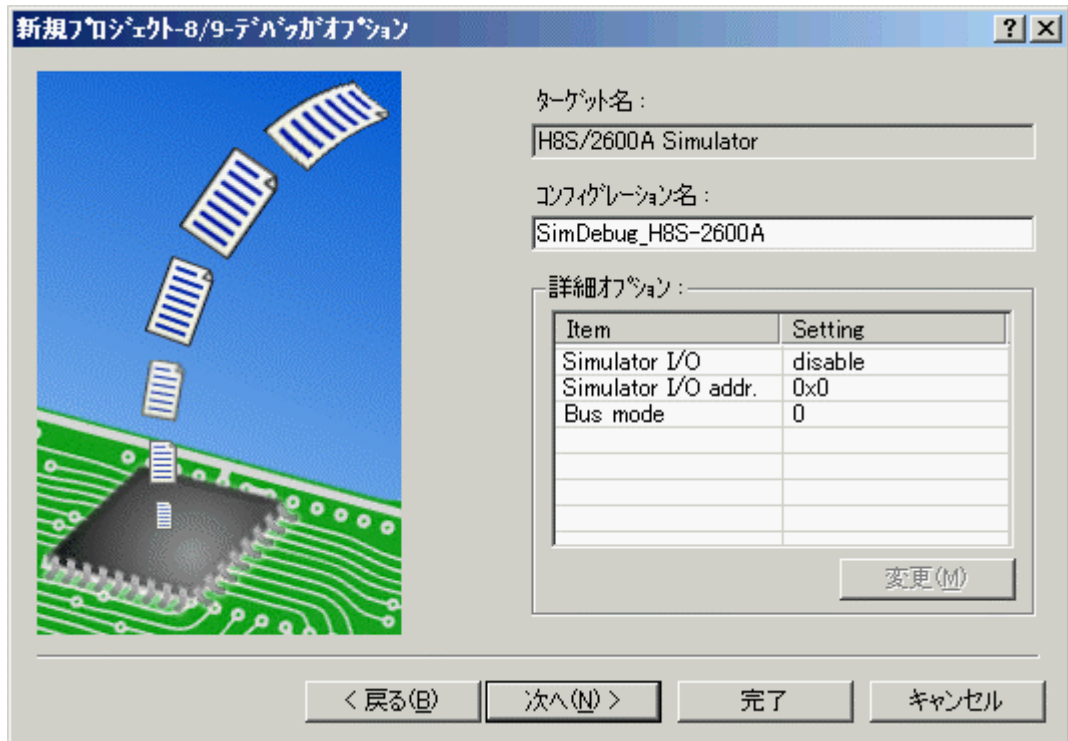


図 3-2 デバッガオプション設定画面 (8/9)


- [コンフィギュレーション名] HEWはデフォルトで[Release]と[Debug]の二つのコンフィグレーションを作成しますが、デバッガターゲットを選択すると選択したターゲット用のコンフィグレーションも作成します(ターゲット名を含んだ略称となります)。このコンフィグレーション名は[コンフィギュレーション名]で変更できます。
- [詳細オプション] デバッガターゲットのオプションを設定します。変更する場合は、[Item]を選択して[変更]をクリックしてください。なお、変更できない項目の場合、[Item]を選択しても[変更]はグレーのままです。
- [Simulator I/O] ユーザプログラムから標準入出力またはファイル入出力を行うI/Oシミュレーションは有効([Enable])または無効([Disable])
- [Simulator I/O addr.] 上記I/Oシミュレーション開始アドレス
- [Bus mode] 現状未使用

## 3.2 シミュレータ・デバッガの設定を変更する

本節ではシミュレータ・デバッガ起動後にシミュレータシステム、メモリマップ、およびメモリリソースを変更する方法について説明します。

### 3.2.1 シミュレータシステムの設定を変更する

I/O シミュレーションの開始位置、実行モード等の設定変更は、[シミュレータシステム]ダイアログボックスの[システム]タブで行います。

[シミュレータシステム]ダイアログボックス[システム]タブを開くには、[基本設定->シミュレータ->システム...]を選択するか、[シミュレータシステム]ツールバーボタンをクリックします。

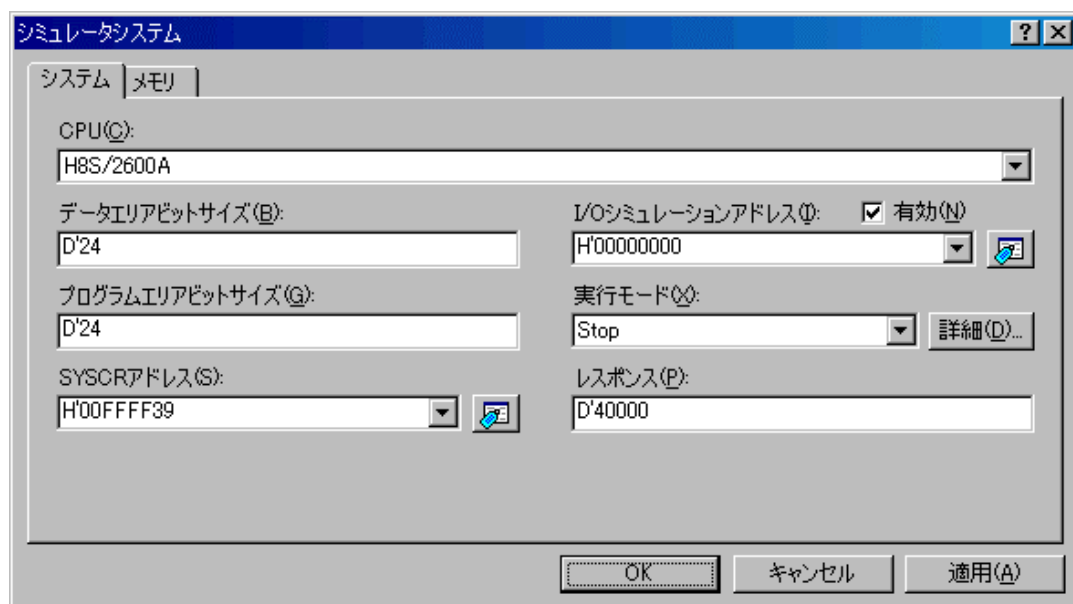


図 3-3 シミュレータシステムダイアログボックス(システムタブ)



本ダイアログボックスでは下記項目を設定または表示します。

[CPU]	現在設定している CPU。 (CPU は、[デバッグの設定]ダイアログボックスにより設定します。)						
[データエリアビット サイズ]	アドレス空間のビット数を指定します。各 CPU で指定できる値は以下の通りです。  <table> <tr> <td>H8/300、H8/300L、H8/300HN、H8S/2600N、H8S/2000N :</td> <td>16</td> </tr> <tr> <td>H8/300HA :</td> <td>17 ~ 24</td> </tr> <tr> <td>H8S/2600A、H8S/2000A :</td> <td>17 ~ 32</td> </tr> </table>	H8/300、H8/300L、H8/300HN、H8S/2600N、H8S/2000N :	16	H8/300HA :	17 ~ 24	H8S/2600A、H8S/2000A :	17 ~ 32
H8/300、H8/300L、H8/300HN、H8S/2600N、H8S/2000N :	16						
H8/300HA :	17 ~ 24						
H8S/2600A、H8S/2000A :	17 ~ 32						
[プログラムエリア ビットサイズ]	プログラム領域のビット数を指定します。各 CPU で指定できる値は以下の通りです。  <table> <tr> <td>H8/300、H8/300L、H8/300HN、H8S/2600N、H8S/2000N :</td> <td>16</td> </tr> <tr> <td>H8/300HA :</td> <td>アドレス空間ビット数 と同一</td> </tr> <tr> <td>H8S/2600A、H8S/2000A :</td> <td>17 ~ 24</td> </tr> </table>	H8/300、H8/300L、H8/300HN、H8S/2600N、H8S/2000N :	16	H8/300HA :	アドレス空間ビット数 と同一	H8S/2600A、H8S/2000A :	17 ~ 24
H8/300、H8/300L、H8/300HN、H8S/2600N、H8S/2000N :	16						
H8/300HA :	アドレス空間ビット数 と同一						
H8S/2600A、H8S/2000A :	17 ~ 24						
[SYSCR アドレス]	SYSCR レジスタのアドレスを指定します。						
[I/O シミュレーション アドレス]	ユーザプログラムから標準入出力またはファイル入出力を行うための I/O シミュレーションの開始位置を指定します。 [有効]      チェックすると I/O シミュレーションが有効となります。						
[レスポンス]	何命令ごとにウィンドウをリフレッシュするかを指定します。 (1~D'2,147,483,647、デフォルトは D'40000)						
[実行モード]	シミュレーションエラーが発生した場合のシミュレータ・デバッガ動作を規定します。 割り込みはシミュレーションエラーに含み、本設定に従います。 また、[詳細]ボタンで、割り込み発生時の動作を規定することもできます。 [停止]      シミュレーションを停止します。 [続行]      シミュレーションを続行します。						

変更内容は、[OK]ボタンまたは[適用]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。

### 3. デバッグ

#### 3.2.2 メモリマップおよびメモリリソースの設定を変更する

メモリマップの設定および変更とメモリリソースの設定および変更は、[シミュレータシステム]ダイアログボックスの[メモリ]タブで行います。

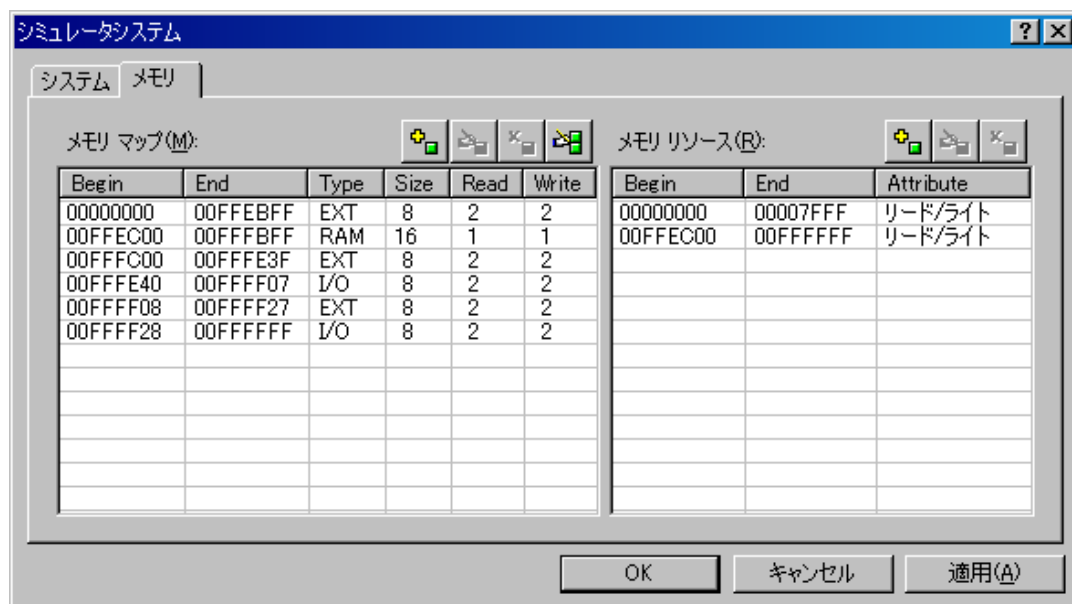


図 3-4 シミュレータシステムダイアログボックス(メモリタブ)

本ダイアログボックスでは以下の項目を設定します。

- [メモリマップ]                   メモリ情報として、メモリ種別とその先頭アドレス・終了アドレス、データバス幅、アクセスステート数を表示します。
- [メモリリソース]                現在設定しているメモリリソースのアクセス種別とその先頭アドレス・終了アドレスを表示します。

[メモリリソース]は次の各ボタンにより追加・変更・削除することができます。



[メモリリソース]の項目を追加します。クリックすることにより、[メモリリソース設定]ダイアログボックスが開き、追加することができます。



[メモリリソース]の項目を変更します。変更したい項目をリストボックス上で選択後、ボタンをクリックします。クリックすることにより、[メモリリソース設定]ダイアログボックスが開き、変更することができます。



[メモリリソース]の項目を削除します。削除したい項目をリストボックス上で選択後、ボタンをクリックします。

[メモリリソース]は、[H8S,H8/300 Standard Toolchain]ダイアログボックスの[デバッガ]シートの[メモリリソース]と同一の設定情報です。おたがいの変更が反映されます。

[メモリマップ]は、以下の各ボタンにより追加・変更・削除ができます。



[メモリマップ]の項目を追加します。クリックすると、[メモリマップ設定]ダイアログボックス(図 3-5 参照)が開き、追加することができます。



[メモリマップ]の項目を変更します。変更したい項目をリストボックス上で選択後、ボタンをクリックします。クリックすると、[メモリマップ設定]ダイアログボックス(図 3-5 参照)が開き、変更することができます。



[メモリマップ]の項目を削除します。削除したい項目をリストボックス上で選択後、ボタンをクリックします。

なお、[メモリマップ]・[メモリリソース]は、ボタンによりデフォルト値にリセットすることができます。

変更内容は、[OK]ボタンまたは[適用]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。

最適化リンケージエディタが出力するリンケージリストファイル(.map)がある場合、メモリマップおよびリンケージマップ情報に基づきメモリリソースを自動的に確保することができます。

詳しくは「High-performance Embedded Workshop V.4.03 ユーザーズマニュアル」の「13.1.9 メモリリソースを自動的に確保する」を参照してください。

### 3.2.3 メモリマップ設定ダイアログボックス

[メモリマップ設定]ダイアログボックスでは、対象 CPU のメモリマップを設定します。

各項目に表示する内容は、対象 CPU によって異なります。シミュレータ・デバッガはこれらの値をメモリアクセスサイクル数の算出に使用します。

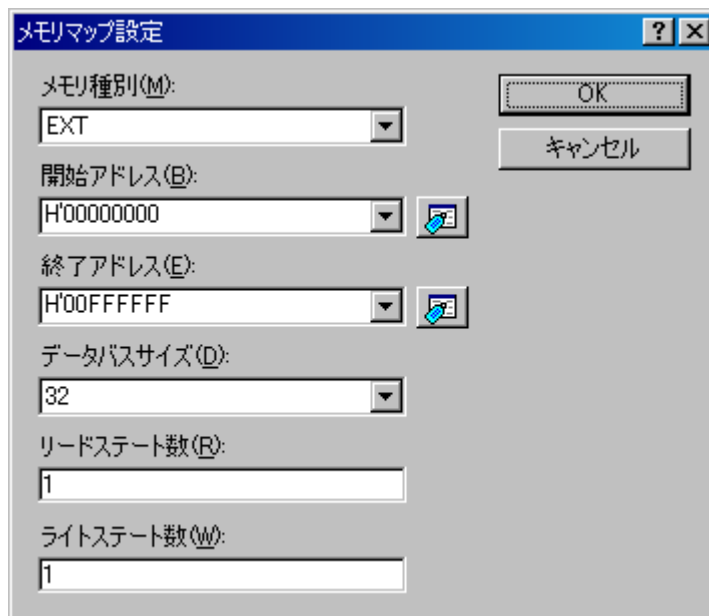


図 3-5 メモリマップ設定ダイアログボックス

### 3. デバッグ

本ダイアログボックスでは以下の項目を設定します。

[メモリ種別]	メモリ種別
	[ROM] 内蔵 ROM
	[RAM] 内蔵 RAM
	[EXT] 外部メモリ
	[IO] 内蔵 I/O
	[EEPROM] EEPROM
[開始アドレス]	メモリ種別に対応するメモリの先頭アドレス
[終了アドレス]	メモリ種別に対応するメモリの終了アドレス
[データバスサイズ]	メモリのデータバス幅
[リードステート数]	メモリのリードアクセスステート数
[ライトステート数]	メモリのライトアクセスステート数

変更内容は、[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。

**【注】** メモリリソースを確保している領域のメモリマップは、削除・変更することができません。あらかじめ、[シミュレータシステム]ダイアログボックスの[メモリ]タブによりメモリリソースを削除してから、メモリマップを削除・変更してください。

#### 3.2.4 メモリリソース設定ダイアログボックス

[メモリリソース設定]ダイアログボックスでは、メモリリソースの設定・変更を行います。

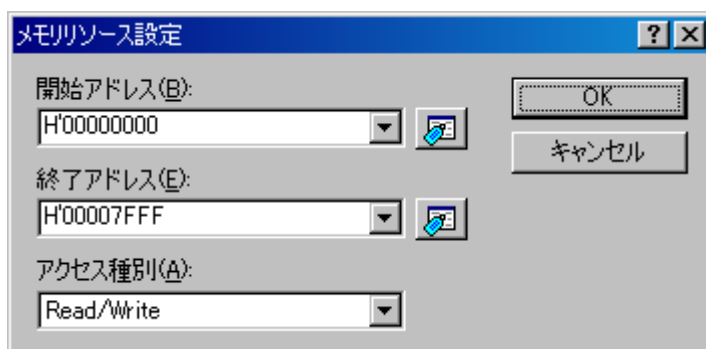


図 3-6 メモリリソース設定ダイアログボックス

本ダイアログボックスでは以下の項目を設定します。

[開始アドレス]	確保するメモリ領域の先頭アドレス
[終了アドレス]	確保するメモリ領域の終了アドレス
[アクセス種別]	アクセス種別
	[Read] 読み出しのみ可能
	[Write] 書き込みのみ可能
	[Read/Write] 読み書き可能

各項目を指定後、[OK]ボタンをクリックすることによりメモリリソースの設定・変更を行います。[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。

【注】メモリリソースを設定すると、PCのメモリを使用します。したがって、メモリリソースを大きく取りすぎると、PCの動作が極端に遅くなる場合があります。

### 3.3 周辺機能シミュレーションを設定する

シミュレータ・デバッガは周辺機能シミュレーションを DLL 形式のモジュールで実現しています。ここでは周辺機能シミュレーションを有効にするための周辺機能シミュレーションモジュールの登録方法および周辺機能シミュレーション構成の設定方法を説明します。

#### 3.3.1 周辺機能シミュレーションモジュールを登録する

周辺機能シミュレーションモジュールの登録は[周辺機能シミュレーションの設定]ダイアログボックスで行います。[周辺機能シミュレーションの設定]ダイアログボックスはシミュレータ・デバッガ起動時に表示されます。

本ダイアログボックスで周辺機能シミュレーションモジュールを登録すると、当該周辺機能シミュレーションモジュールが提供する周辺機能シミュレーションを利用可能となります。シミュレータ・デバッガ起動後は、周辺機能シミュレーションモジュール登録内容を変更することはできません。利用する周辺機能シミュレーションモジュールを変更する場合はシミュレータ・デバッガを再起動し、本ダイアログボックスを表示させてください。

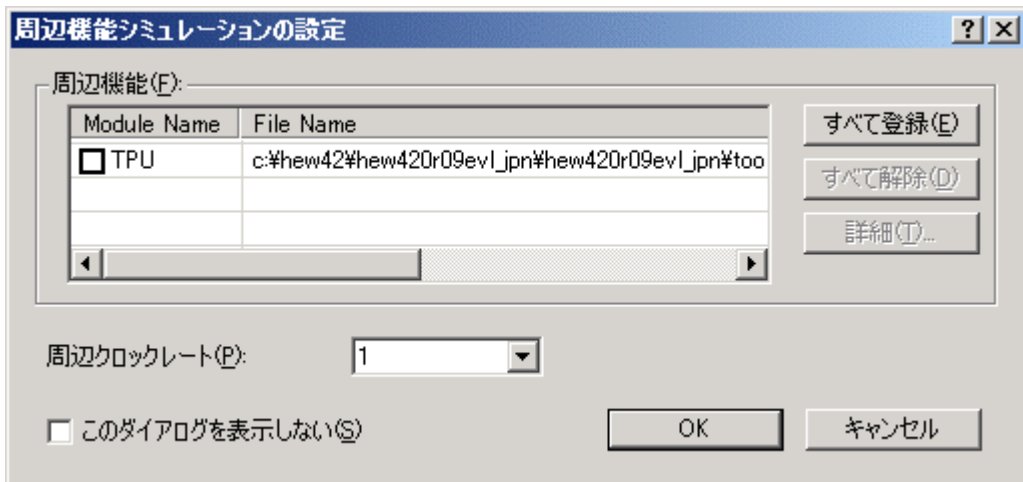


図 3-7 周辺機能シミュレーションの設定ダイアログボックス

### 3. デバッグ

本ダイアログボックスでは以下の項目を設定します。

- [周辺機能] 周辺機能シミュレーションモジュールの情報を表示します。
- [Module Name] シミュレーションする周辺機能名
- [File Name] 周辺機能シミュレーションモジュールファイル名
- [Module name]欄のチェックボックスをチェックした周辺機能シミュレーションモジュールが登録されて、利用可能となります。
- [すべて登録] すべての周辺機能シミュレーションモジュールを有効にします。
- [すべて解除] すべての周辺機能シミュレーションモジュールを無効にします。
- [詳細...] 周辺機能情報の表示、周辺機能の開始アドレス、および割り込み要因情報の変更を行うための[周辺モジュールの構成]ダイアログボックスを表示します。
- [周辺クロックレート] 周辺クロックと内部クロックの比(周辺1クロックが内部クロックいくつに相当するか)を指定します。  
(1,2,3,4,6,8,12,16,24,32から選択)

各項目を指定後、[OK]ボタンをクリックすることにより周辺機能シミュレーションの設定、変更を行います。[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。

[このダイアログを表示しない]チェックボタンをチェックすると、次回以降シミュレータ・デバッグ起動時に本ダイアログボックスを表示しなくなります。

#### 3.3.2 周辺機能のアドレスを変更する

周辺機能のアドレス変更は[周辺モジュールの構成]ダイアログボックス[アドレス]タブで行います。

[周辺モジュールの構成]ダイアログボックスを開くには、[周辺機能シミュレーションの設定]ダイアログボックスの[周辺機能]欄で周辺機能を選択して、[詳細...]ボタンをクリックします。



図 3-8 周辺モジュールの構成ダイアログボックス (アドレスタブ)



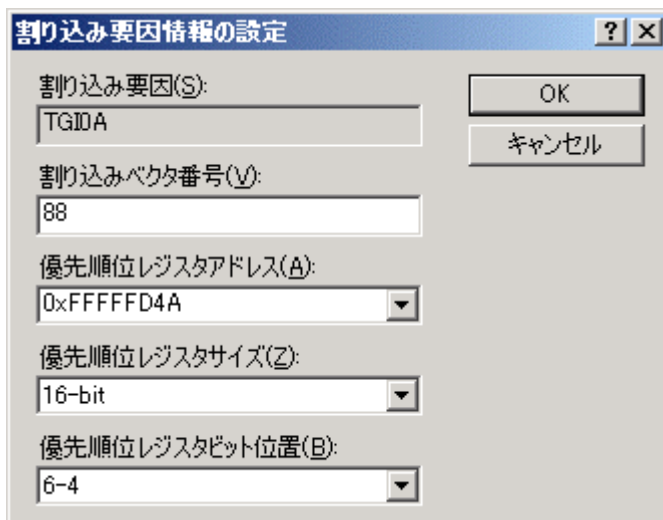


図 3-10 割り込み要因情報の設定ダイアログボックス

本ダイアログボックスでは以下の項目を表示、設定します。

割り込み要因	割り込み要因名
割り込みベクタ番号	割り込みベクタ番号 (接頭辞省略時は 10 進入力、10 進表示)
優先順位レジスタアドレス	割り込み優先順位レジスタのアドレス
優先順位レジスタサイズ	割り込み優先順位レジスタのサイズ
優先順位レジスタビット位置	割り込み優先順位レジスタ内ビット位置

各項目を指定後、[OK]ボタンをクリックすることにより割り込み要因情報を設定します。[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。

#### 3.3.4 割り込み優先順位レジスタのメモリリソースを確保する

周辺機能シミュレーションモジュールは割り込み優先順位レジスタ(IPR)メモリ領域を確保しません。割り込み優先順位レジスタ(IPR)領域は、ユーザ操作でメモリリソースを確保してください。メモリリソース設定の詳細は、「3.2.2 メモリマップおよびメモリリソースの設定を変更する」を参照してください。

#### 3.3.5 接続されている周辺機能を確認する

シミュレータ・デバッガ起動後は、[ステータス]ウィンドウの[Platform]シートの[Peripheral Modules]項目に接続されている周辺機能名を表示します。

### 3.4 シミュレータ・デバッガのブレークポイントを使用する


シミュレータ・デバッガでは HEW 標準の PC ブレークポイントとは別により高度なブレークポイント機能を持っています。

これらブレークポイントについて、ブレーク条件の設定、ブレーク条件成立時の動作、および設定されているブレークポイントの表示が行えます。



### 3.4.1 ブレークポイントを一覧表示する

現在設定されているブレークポイントを一覧表示するには[イベントポイント]ウィンドウを開きます。

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

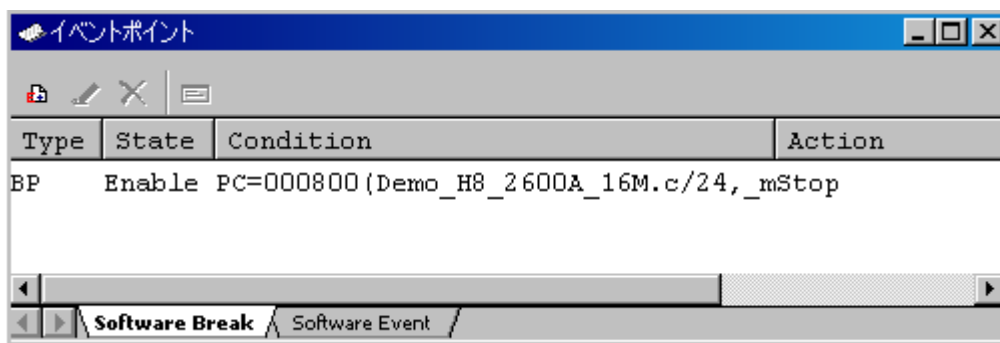


図 3-11 イベントポイントウィンドウ



### 3.4.2 ブレークポイントを設定する

[イベントポイント]ウィンドウのポップアップメニューで[設定...]を選択すると、[ブレーク種別の選択]ダイアログボックスが開きブレークポイントを設定できます。

[ブレーク種別の選択]ダイアログボックスからブレーク条件を設定する[条件の設定]ダイアログボックスとブレーク成立時の動作を設定する[動作の設定]ダイアログボックスが開きます。[動作種別]として、[停止]を指定する場合は[Software Break]タブで、[停止]以外を指定する場合は[Software Event]タブでポップアップメニューを選択します。

#### (1) ブレーク種別を選択する

[イベントポイント]ウィンドウのポップアップメニューで[設定]を選択すると、[ブレーク種別の選択]ダイアログボックスが開きます。

ブレーク種別は[ブレーク種別の選択]ダイアログボックスの[ブレーク種別]フィールドで選択します。



図 3-12 ブレーク種別の選択ダイアログボックス

選択内容を以下に示します。

ブレーク種別	内容
[PCブレークポイント]	実行命令位置によるブレークポイント
[ブレークアクセス]	メモリ範囲のアクセスによるブレーク
[ブレークデータ]	メモリのデータ値によるブレーク
[ブレークレジスタ]	レジスタのデータ値によるブレーク
[ブレークシーケンス]	実行順序を指定したブレークポイント
[ブレークサイクル]	サイクル数によるブレーク

#### (2) ブレーク条件を設定する

[ブレーク種別の選択]ダイアログボックスでブレーク種別を選択後[詳細]をクリックすると、各ブレーク種別の条件を設定するダイアログボックスを表示します。

##### (a) PC ブレークポイント



図 3-13 PC ブレークポイント条件の設定ダイアログボックス

### 3. デバッグ

PCブレイクポイント条件を設定します。

- [PCブレイクポイント] 1,024個まで指定可能
- [アドレス] ブレイクする命令の位置
- [回数] 指定位置の命令をフェッチする回数  
(接頭辞省略時は10進入力、10進表示)  
(1~16,383、省略すると1となります)

#### (b) ブレイクアクセス

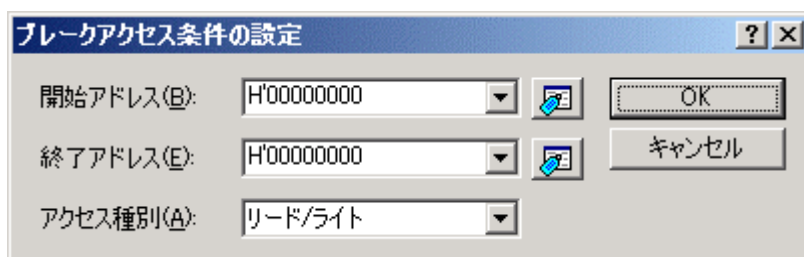


図 3-14 ブレイクアクセス条件の設定ダイアログボックス

ブレイクアクセス条件を設定します。

- [ブレイクアクセス] 1,024 個まで指定可能
- [開始アドレス] アクセスするとブレイクするメモリの開始位置
- [終了アドレス] アクセスするとブレイクするメモリの終了位置  
(省略すると開始位置のみが範囲となります)
- [アクセス種別] アクセス種別

#### (c) ブレイクレジスタ

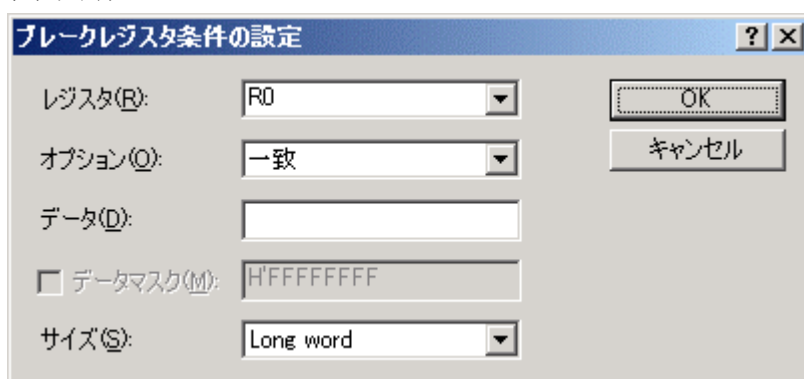


図 3-15 ブレイクレジスタ条件の設定ダイアログボックス

ブレイクレジスタ条件を設定します。

- [ブレイクレジスタ] 1,024 個まで指定可能
- [レジスタ] ブレイク条件を設定するレジスタ名
- [オプション] データの一致/不一致
- [データ] ブレイク条件となるデータ値  
(省略するとレジスタへ書き込むたびにブレイクします)
- [データマスク] マスク条件(0 を指定したビットがマスクされます)
- [サイズ] データのサイズ

## (d) ブレークシーケンス

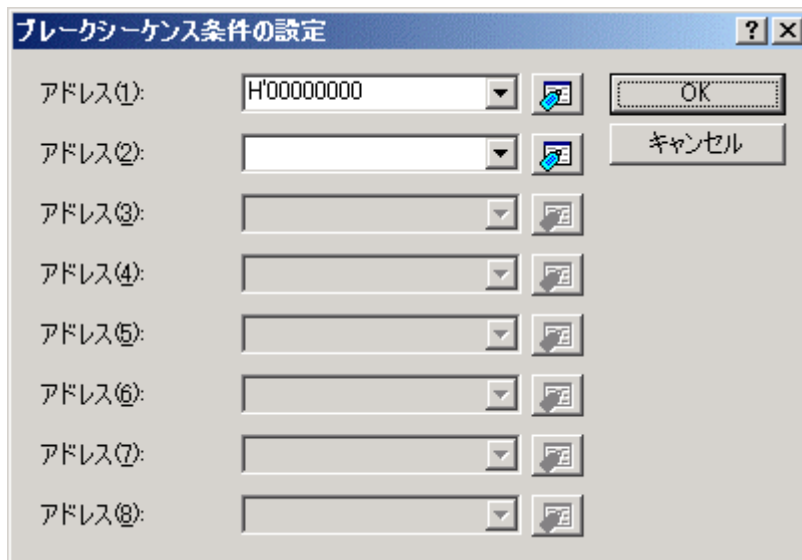


図 3-16 ブレークシーケンス条件の設定ダイアログボックス

ブレークシーケンス条件を設定します。

[ブレークシーケンス] 1組のみ指定可能

[アドレス(1)]~[アドレス(8)] ブレークの発生条件となる通過アドレス  
(8ポイントすべてを設定する必要はありません)

### 3. デバッグ

---

#### (e) ブレークサイクル

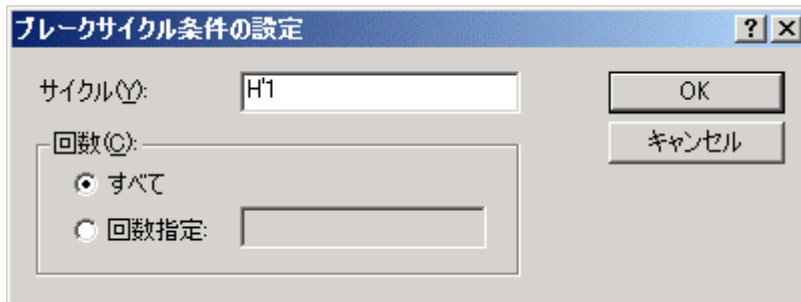


図 3-17 ブレークサイクル条件の設定ダイアログボックス

ブレークサイクル条件を設定します。

- |            |  |
|------------|--|
| [ブレークサイクル] | 1,024 個まで指定可能  |
| [サイクル]     | ブレーク判定を行なうサイクル数(1~H'FFFFFFFF)<br>[サイクル]×n のサイクルで条件が一致します<br>ただし、指定したサイクルと実際に条件が一致するサイクルは<br>ずれることがあります |
| [回数]       | ブレークが成立する回数  |
| [すべて]      | 条件が一致するごとにブレークが成立します   |
| [回数指定]     | 条件が一致した回数が[回数指定]以下の時だけ<br>ブレークが成立します<br>(接頭辞省略時は 16 進入力、16 進表示)(1~65,535)                              |

## (f) ブレークデータ

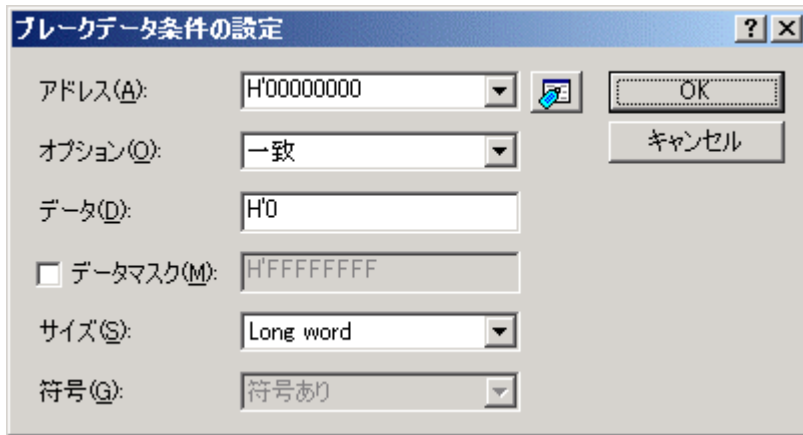


図 3-18 ブレークデータ条件の設定ダイアログボックス

ブレークデータ条件を設定します。

[ブレークデータ]	1,024 個まで指定可能	
[アドレス]	ブレーク判定を行うメモリの位置	
[オプション]	判定方法	
	一致	メモリのデータと指定値([データ])が一致で成立
	不一致	メモリのデータと指定値([データ])が不一致で成立
	符号反転 <sup>*1</sup>	前回メモリ書き込み値と今回メモリ書き込み値で符号が反転した場合に成立
	差分 <sup>*1</sup>	前回メモリ書き込み値と今回メモリ書き込み値の差分が指定値([データ])を超えた場合に成立
[データ]	ブレーク条件となるデータ値	
[データマスク]	マスク条件(0 を指定したビットがマスクされます)	
	データの一致/不一致オプション時のみ有効です	
[サイズ]	データのサイズ	
[符号]	データの符号	
	データの差分オプション時のみ有効です	

\*1: [符号反転]、および[差分]は前回書き込み値と比較するため、リセット後、およびブレーク成立後 1 回目の判定は常に不成立となります。

### 3. デバッグ

---

#### (3) 動作種別を選択する

各ブレーク条件設定ダイアログボックスで条件設定後[OK]をクリックすると、再度[ブレーク種別の選択]ダイアログボックスが開きます。

[ブレーク種別の選択]ダイアログボックスの[動作種別]フィールドで動作種別を選択します。

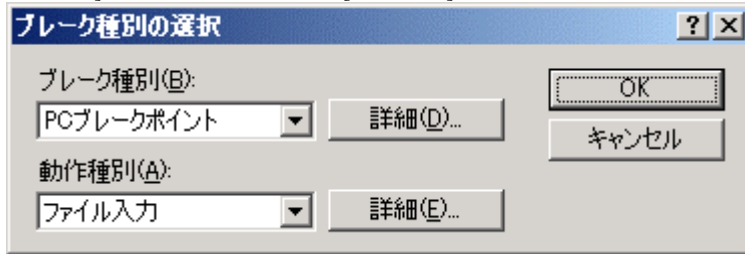


図 3-19 ブレーク種別の選択ダイアログボックス

選択内容を以下に示します。

動作種別	内容
[停止]	条件成立時にユーザプログラムの実行を停止します
[ファイル入力]	条件成立時に指定ファイルから読み込んだデータを指定メモリへ書き込みます
[ファイル出力]	条件成立時に指定メモリの内容を指定ファイルへ書き込みます
[割り込み]	条件成立時に割り込み処理を行います



## (4) 動作内容を設定する

[ブレーク種別の選択]ダイアログボックスで動作種別を選択後[詳細]をクリックすると、[停止]を除く各動作種別の内容を設定するダイアログボックスを表示します。

## (a) ファイル入力

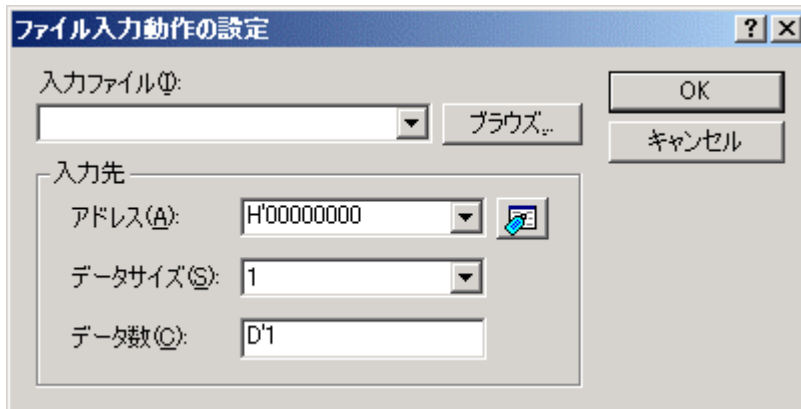


図 3-20 ファイル入力動作の設定ダイアログボックス

ファイル入力動作内容を設定します。

- |          |  |
|----------|--|
| [ファイル入力] | 条件成立時に指定ファイルから読み込んだデータを指定メモリへ書き込みます                      |
| [入力ファイル] | 読み込むデータファイルを指定します<br>ファイルの終端まで読み込んだら先頭から繰り返して読み込みます      |
| [アドレス]   | データを書き込むメモリのアドレスを指定します                                   |
| [データサイズ] | 読み込むデータ 1 個のサイズ(バイト数)を指定します(1/2/4/8)                     |
| [データ数]   | 読み込むデータの個数を指定します<br>(接頭辞省略時は 10 進入力、10 進表示)(1~H'FFFFFFF) |

## (b) ファイル出力

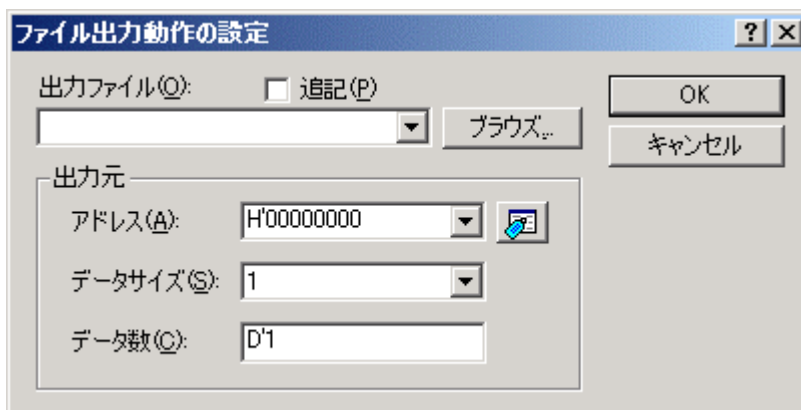


図 3-21 ファイル出力動作の設定ダイアログボックス

ファイル出力動作内容を設定します。

### 3. デバッグ

[ファイル出力]	条件成立時に指定メモリの内容を指定ファイルへ書き込みます
[出力ファイル]	書き込むデータファイルを指定します
[追記]	既存のファイルを[出力ファイル]で指定した場合にファイルの最後に追加出力するかを指定します
[アドレス]	データを読み出すメモリのアドレスを指定します
[データサイズ]	書き込むデータ 1 個のサイズ(バイト数)を指定します(1/2/4/8)
[データ数]	書き込むデータの個数を指定します (接頭辞省略時は 10 進入力、10 進表示)(1~H'FFFFFFF)

#### (c) 割り込み

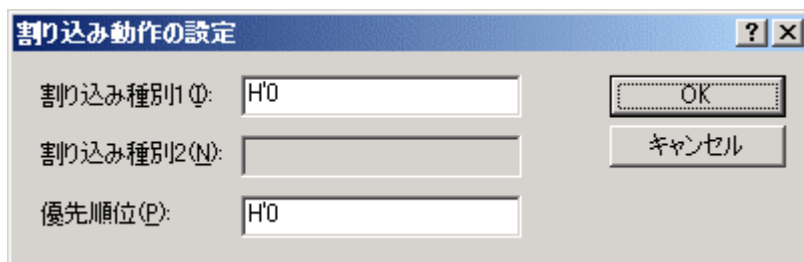


図 3-22 割り込み動作の設定ダイアログボックス

割り込み動作内容を設定します。

[割り込み]	条件成立時に割り込み処理を行います。詳細は「2.19 擬似割り込み」を参照してください。
[割り込み種別 1]	割り込みベクタ番号を指定します。(接頭辞省略時は 16 進入力、16 進表示)
[優先順位]	割り込み優先順位を指定します。(接頭辞省略時は 16 進入力、16 進表示)(0~H'11) 割り込みを受け付けるかどうかの判定は、選択されたデバッグプラットフォームの CPU 仕様に従います。ただし、優先順位に H'8 以上を指定した場合は、常に割り込みを受け付けます。

#### (d) 留意事項

複数の[ファイル入力]で同一ファイルを指定した場合、ブレイク成立順にファイルからデータを読み込みます。複数の[ファイル出力]で同一ファイルを指定した場合、ブレイク成立順にファイルヘッダを書き込みます。ただし、[ファイル入力]と[ファイル出力]で同一ファイルを指定した場合は、最初に成立した動作のみが有効となります。

### 3.4.3 ブレイクポイントの設定内容を変更する

変更したいブレイク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、[ブレイク種別の選択]ダイアログボックスが開き、ブレイク条件を変更することができます。[編集...]メニューはブレイクポイントを 1 個選択しているときのみ有効となります。

### 3.4.4 ブレイクポイントを有効にする

ブレイク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、選択しているブレイクポイントを有効にします。

### 3.4.5 ブレークポイントを無効にする

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

### 3.4.6 ブレークポイントを削除する

ブレーク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を削除します。ブレークポイントを削除しないで、詳細情報は保持したまま、条件が一致してもブレークを成立させないようにするには、[無効]オプションを使用します(「3.4.5 ブレークポイントを無効にする」参照)。

### 3.4.7 ブレーク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を削除します。

### 3.4.8 ブレークポイントのソース行を表示する

ブレークポイントを選択後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、ブレークポイントのある[ソース]または[逆アセンブリ]ウィンドウをオープンします。[ソースファイル表示]メニューはブレークポイントを1個選択しているときのみ有効となります。

### 3.4.9 入出力ファイルを閉じる

ブレークポイントを選択後ポップアップメニューから[ファイルを閉じる]を選択すると、選択した[ファイル入力] または[ファイル出力]のデータファイルを閉じ、ファイル読み出し位置をリセットします。

### 3.4.10 入出力ファイルをすべて閉じる


ポップアップメニューから[すべてのファイルを閉じる]を選択すると、すべての[ファイル入力] および[ファイル出力]のデータファイルを閉じ、ファイル読み出し位置をリセットします。

## 3.5 トレース情報を見る

シミュレータ・デバッグでは命令の実行結果をトレース情報として取得および表示することができます。

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

### 3.5.1 トレースウィンドウを開く

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

### 3.5.2 トレース情報取得条件を設定する

[トレース]ウィンドウを開いたら、トレース情報取得条件を設定します。

### 3. デバッグ

トレース情報取得条件は、[トレース取得]ダイアログボックスで設定します。  
[トレース取得]ダイアログボックスを開くには、ポップアップメニューから[設定...]を選択します。

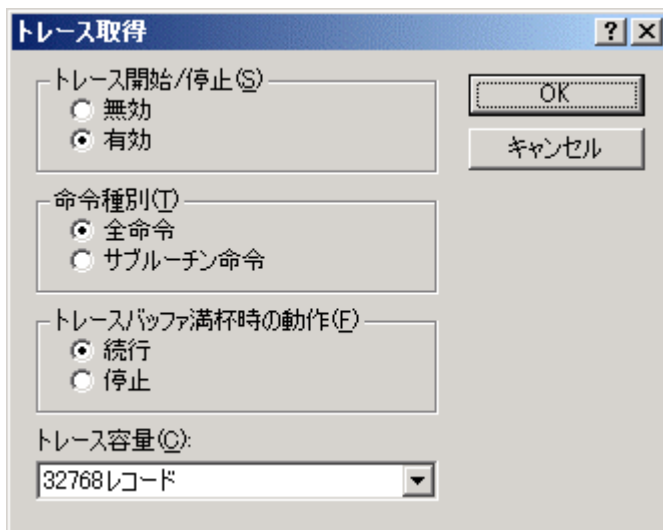


図 3-23 トレース取得ダイアログボックス

本ダイアログボックスでは、トレース情報の取得条件を設定します。

[トレース開始/停止]

- |      |             |
|------|-------------|
| [無効] | トレース情報の取得停止 |
| [有効] | トレース情報の取得開始 |

[命令種別]

- |            |                     |
|------------|---------------------|
| [全命令]      | 全命令のトレース情報を取得       |
| [サブルーチン命令] | サブルーチン命令のみトレース情報を取得 |

[トレースバッファ満杯時の動作]

- |      |                            |
|------|----------------------------|
| [続行] | トレース情報取得バッファが満杯になっても取得を続行  |
| [停止] | トレース情報取得バッファが満杯になった場合、実行停止 |

また、トレースバッファの大きさは、[トレース容量]により 1K(1,024)、4K(4,096)、16K(16,384)、32K(32,768)、64K(65,536)、128K(131,072)、または 256K(262,144)レコードの中から選択します。

指定した内容は、[OK]ボタンをクリックすることにより設定します。[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。

#### 3.5.3 トレース情報を取得する

トレース情報の取得を開始した状態で命令を実行すると、トレースを取得できます。  
取得したトレース情報は[トレース]ウィンドウに表示します。

PTR	Cycle	Address	CCR	Mult	Instruction	Access_Data	Source	Label
-00402	0000608107	000A9A	--H---C	---	MOV.L ER6,@-ER7	00FFFA6...	void change(long *a)	_chang
-00401	0000608111	000A9E	--H---C	---	MOV.L ER7,ER6	ER6<-00F...		
-00400	0000608122	000AA0	--H--Z-C	---	MOV.L ER2,@-ER7	00FFFA2...		
-00399	0000608133	000AA4	--H--Z-C	---	MOV.L ER5,@-ER7	00FFFB9E...		
-00398	0000608141	000AA8	----N---	---	SUB.W #H'002...	R7<-FB74		
-00397	0000608145	000AAC	-----	---	MOV.L ER0,ER5	ER5<-00F...		
-00396	0000608149	000AAE	----Z--	---	SUB.W R0,R0	R0<-0000	for(i=0; i<10; i++){	
-00395	0000608158	000AB0	----Z--	---	MOV.W R0,@(H...	00FFFB74...		
-00394	0000608166	000AB4	----Z--	---	BRA @(H'0AE2:8	PC<-0000...		
-00393	0000608175	000AE2	----Z--	---	MOV.W @(H'FF...	R0<-0000		
-00392	0000608183	000AE6	--H-N--C	---	CMP.W #H'000...			
-00391	0000608191	000AEA	--H-N--C	---	BLT @(H'0AB6:8	PC<-0000...		
-00390	0000608200	000AB6	--H--Z-C	---	MOV.W @(H'FF...	R0<-0000	tmp[i] = a[i];	
-00389	0000608204	000ABA	--H--Z-C	---	EXTS.L ER0	ER0<-000...		
-00388	0000608208	000ABC	--H--Z--	---	SHLL.L #2,ER0	ER0<-000...		
-00387	0000608212	000ABE	-----	---	ADD.L ER5,ER0	ER0<-00F...		

図 3-24 トレースウィンドウ

表示する項目は以下の通りです。

[PTR]	トレースバッファ内ポインタ (最後に実行した命令が0となります)
[Cycle]	累計命令実行サイクル数 (命令実行リセットによりクリアします)
[Address]	命令アドレス
[CCR]	コンディションコードレジスタ (CCR) の内容をニモニクで表示します。
[Mult]	乗算器内部のフラグをニモニクで表示します。(H8S/2600 シリーズのみ)
[Instruction]	命令ニモニク
[Access Data]	データアクセス (転送先<転送データの形式で表示)
[Source]	C/C++またはアセンブラソース
[Label]	ラベル

### 3.5.4 トレース情報を検索する

トレース情報を検索するには[トレース検索]ダイアログボックスを使用します。

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

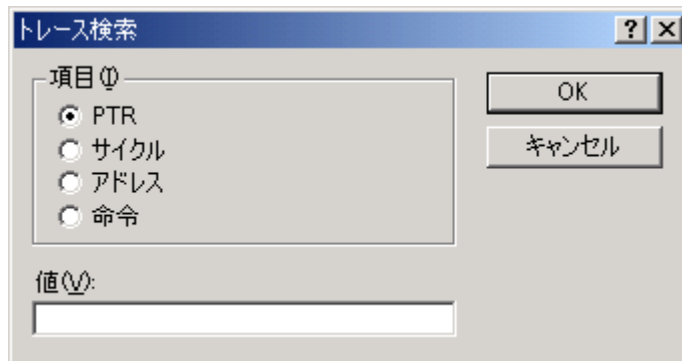


図 3-25 トレース検索ダイアログボックス

### 3. デバッグ

---

本ダイアログボックスは、トレース情報の検索条件を設定します。[項目]で検索対象項目を指定し、[値]で指定した内容を検索します。

[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。 -nmn の形式で指定してください。
[サイクル]	累計命令実行サイクル数
[アドレス]	命令アドレス
[命令]	命令ニモニック

[OK]ボタンをクリックすることにより、検索条件を設定し、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青色表示します。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。

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

#### 3.5.5 トレース情報をクリアする

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

#### 3.5.6 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]ファイルダイアログボックスを表示します。トレースバッファの内容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。このファイルはトレースバッファに再ロードできないことに注意してください。

#### 3.5.7 ソースファイルを表示する

トレースレコードに対応するソースファイルを[エディタ]ウィンドウに表示するには二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、選択した行をカーソルで示します。

#### 3.5.8 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

[ソースストリム]メニューを選択すると取り除いた空白を元に戻します。

### 3.5.9 統計情報を解析する

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

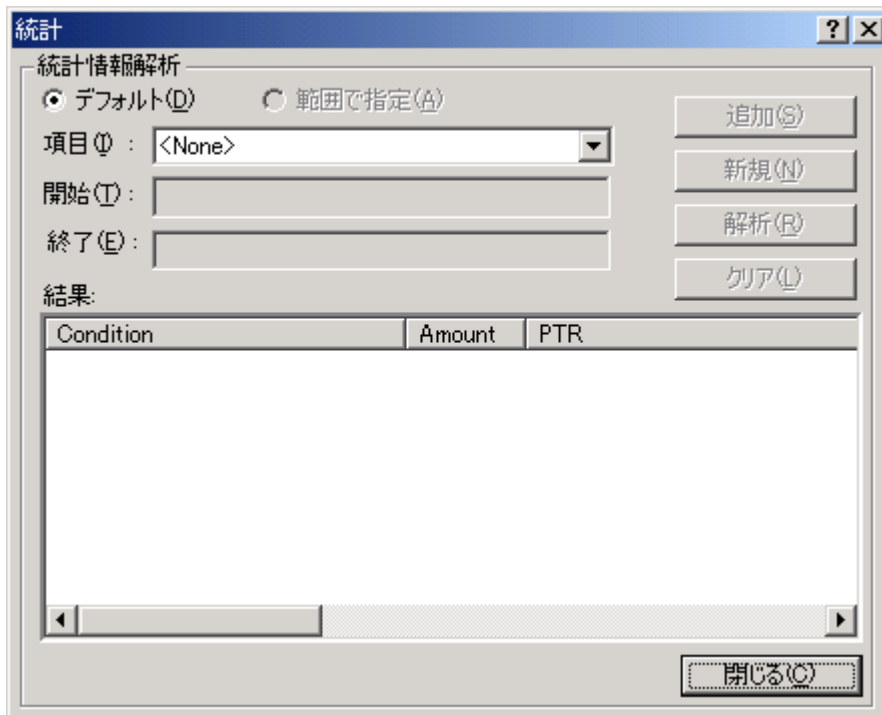


図 3-26 統計ダイアログボックス

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]で解析対象項目を指定し、[開始]および[終了]で入力値または文字列を指定します。

[デフォルト]を選択すると入力値または文字列を範囲で指定することはできません。範囲で指定する場合は[範囲で指定]を選択してください。

- [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]         すべての条件と統計情報解析結果を削除します

[閉じる]ボタンをクリックすると、ダイアログボックスを閉じます。

## 3.6 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。

HEW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。

### 3.6.1 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。HEW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。(但し、[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。)

HEW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、[Standard Toolchain]ダイアログボックスの[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。

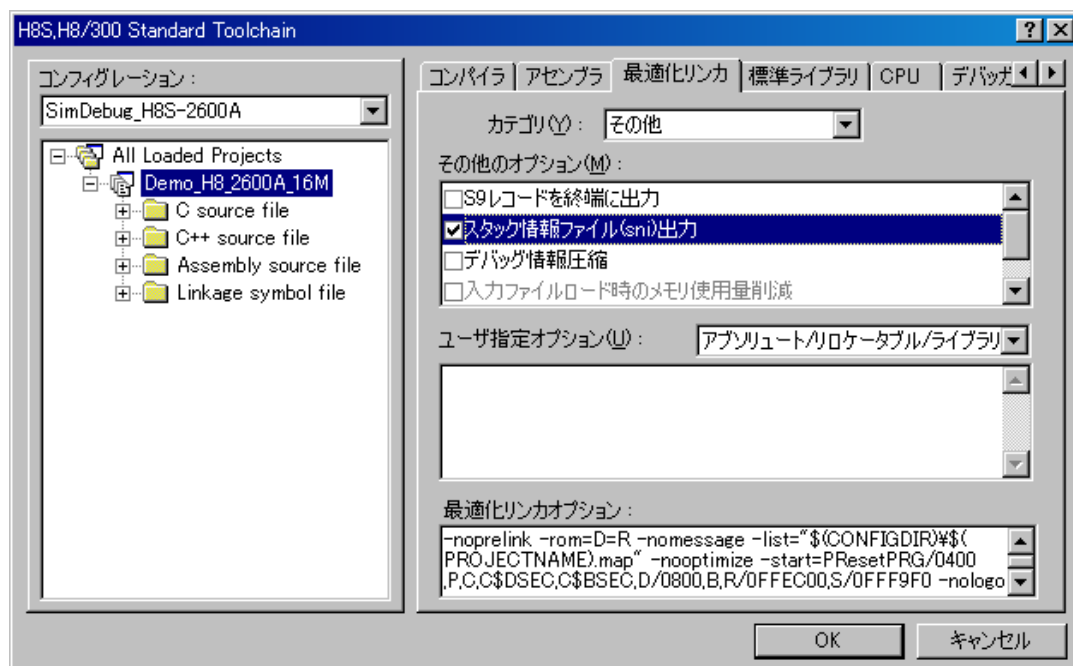


図 3-27 Standard Toolchain ダイアログボックス(1)



### 3.6.2 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]ウィンドウのポップアップメニューで[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グラム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]ダイアログボックスの[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。

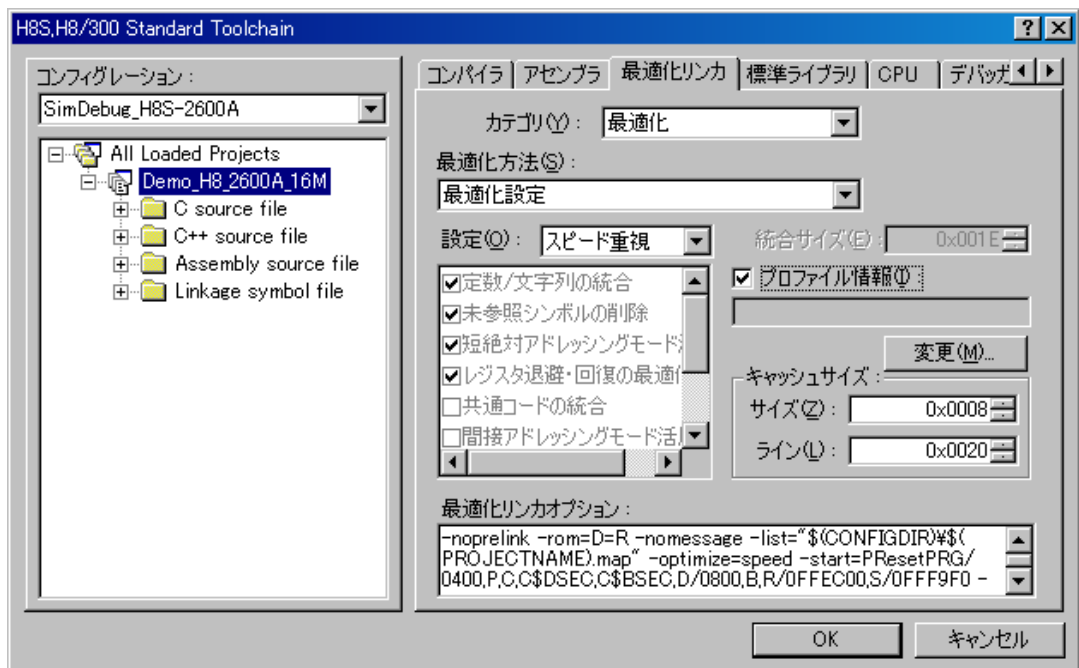


図 3-28 Standard Toolchain ダイアログボックス(2)

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、[最適化]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。

#### 3.6.3 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時
- [オプション]ダイアログボックス(メインメニューの[基本設定->オプション...])を選択すると開きます)の[確認]タブ(図 3-29)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

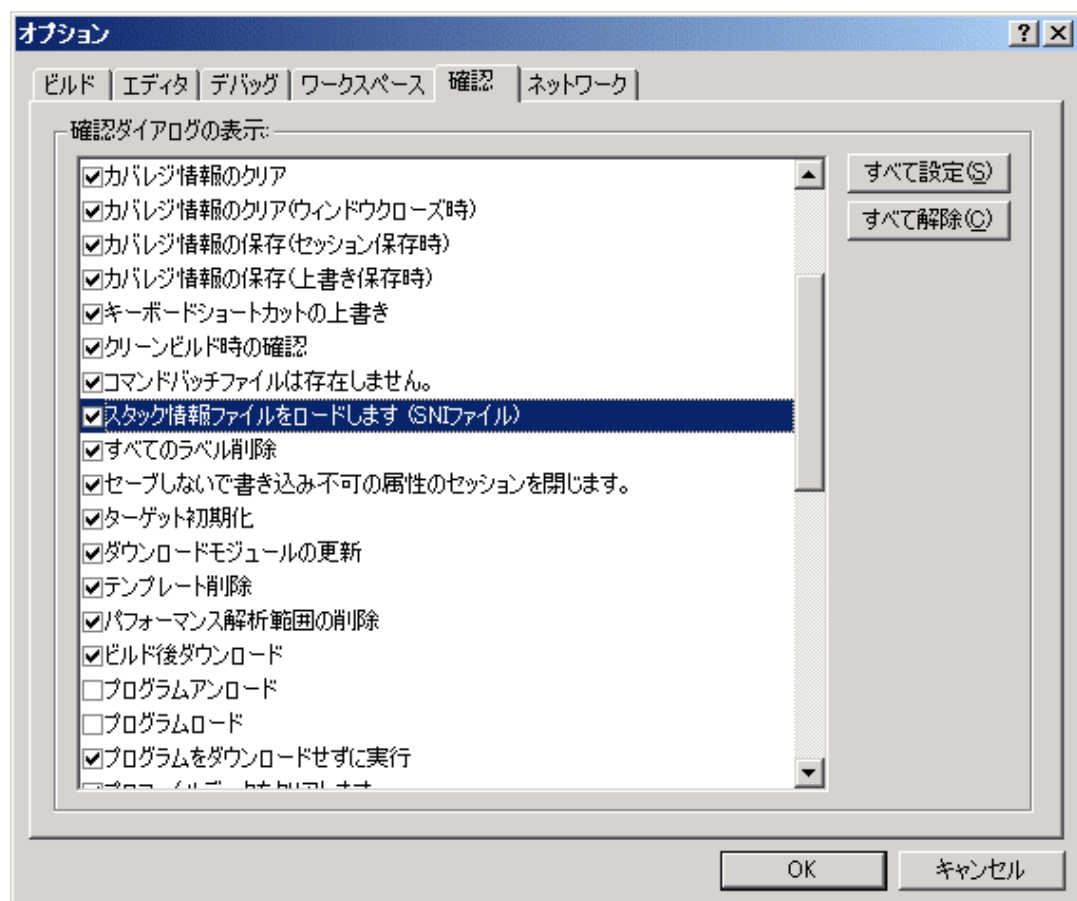


図 3-29 オプションダイアログボックス

### 3.6.4 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）。

### 3.6.5 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つ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。

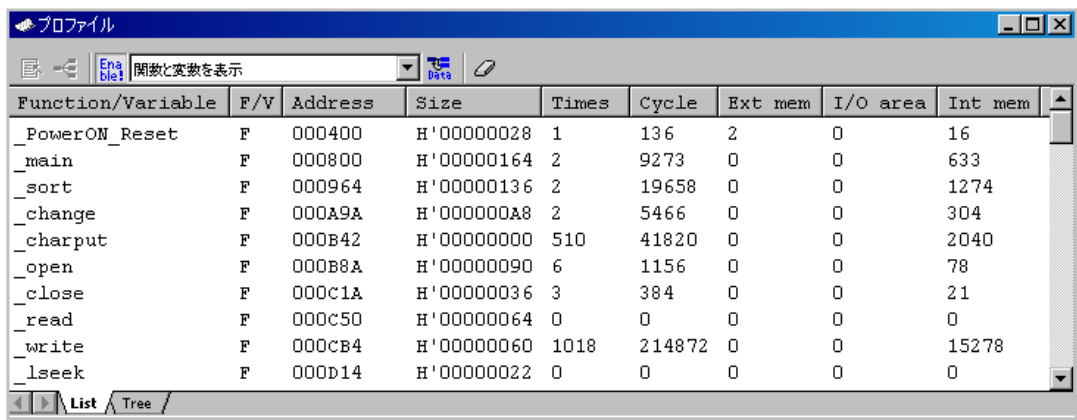
### 3.6.6 ユーザプログラムを実行し結果を確認する

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。

### 3.6.7 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	Cycle	Ext mem	I/O area	Int mem
_PowerON_Reset	F	000400	H'00000028	1	136	2	0	16
_main	F	000800	H'00000164	2	9273	0	0	633
_sort	F	000964	H'00000136	2	19658	0	0	1274
_change	F	000A9A	H'000000A8	2	5466	0	0	304
_charput	F	000B42	H'00000000	510	41820	0	0	2040
_open	F	000B8A	H'00000090	6	1156	0	0	78
_close	F	000C1A	H'00000036	3	384	0	0	21
_read	F	000C50	H'00000064	0	0	0	0	0
_write	F	000CB4	H'00000060	1018	214872	0	0	15278
_lseek	F	000D14	H'00000022	0	0	0	0	0

図 3-30 List シート

カ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。

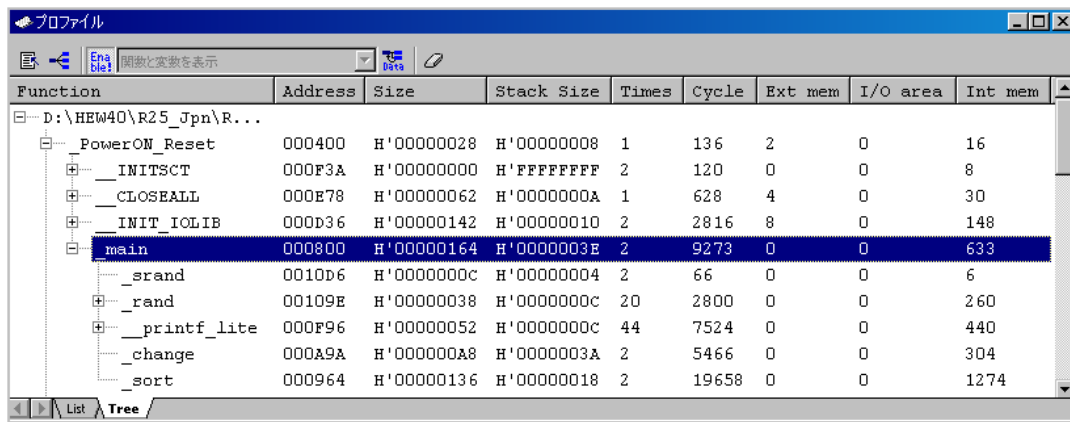
### 3. デバッグ

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については「3.6.8 Tree シート」を参照してください。

#### 3.6.8 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時のみ有効です。



Function	Address	Size	Stack Size	Times	Cycle	Ext mem	I/O area	Int mem
PowerON_Reset	000400	H'00000028	H'00000008	1	136	2	0	16
__INIT\$CT	000F3A	H'00000000	H'FFFFFFF	2	120	0	0	8
__CLOSEALL	000E78	H'00000062	H'0000000A	1	628	4	0	30
__INIT_IOLIB	000D36	H'00000142	H'00000010	2	2816	8	0	148
main	000800	H'00000164	H'0000003E	2	9273	0	0	633
__srand	0010D6	H'0000000C	H'00000004	2	66	0	0	6
__rand	00109E	H'00000038	H'0000000C	20	2800	0	0	260
__printf_lite	000F96	H'00000052	H'0000000C	44	7524	0	0	440
__change	000A9A	H'000000A8	H'0000003A	2	5466	0	0	304
__sort	000964	H'00000136	H'00000018	2	19658	0	0	1274

図 3-31 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測定のオン・オフを切り替えます。プロファイルデータ測定がオンのとき、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、最適化リンケージエディタによる最適化を行う場合も、本メニューをチェックしないでください。

## (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 3-32 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。

### 3. デバッグ

---

#### (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出来ます。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...

本シミュレータ・デバッガでは使用できません。

### 3.6.9 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。

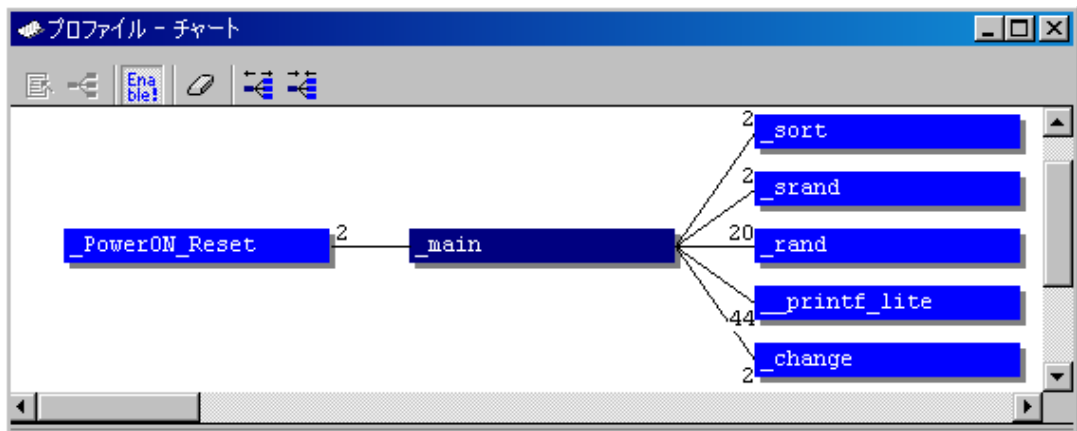


図 3-33 プロファイルチャートウィンドウ

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

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

(2) チャート表示

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

(3) 有効

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

(4) データクリア

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

### 3. デバッグ

---

#### (5) チャートウィンドウを複数開く

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

#### (6) プロファイル情報の保存...

[プロファイル情報の保存]ダイアログボックスを表示します。プロファイル結果をプロファイル情報ファイル(拡張子は".pro")に保存します。最適化リンケージエディタは、プロファイル情報を元に、ユーザプログラムの最適化を行うことが出来ます。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。

#### (7) 拡大

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

#### (8) 縮小

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

### 3.6.10 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。

#### (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

- [Cycle] (実行サイクル数)
- [Ext\_mem] (外部メモリアクセス回数)
- [I/O\_area] (内蔵 I/O アクセス回数)
- [Int\_mem] (内部メモリアクセス回数)

実行サイクル数は、当該関数コール命令実行時の累計実行サイクル数と当該関数からのリターン命令実行時の累計実行サイクル数の差から求めています。

## 3.6.11 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、Pop-up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こみます。[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こみます。

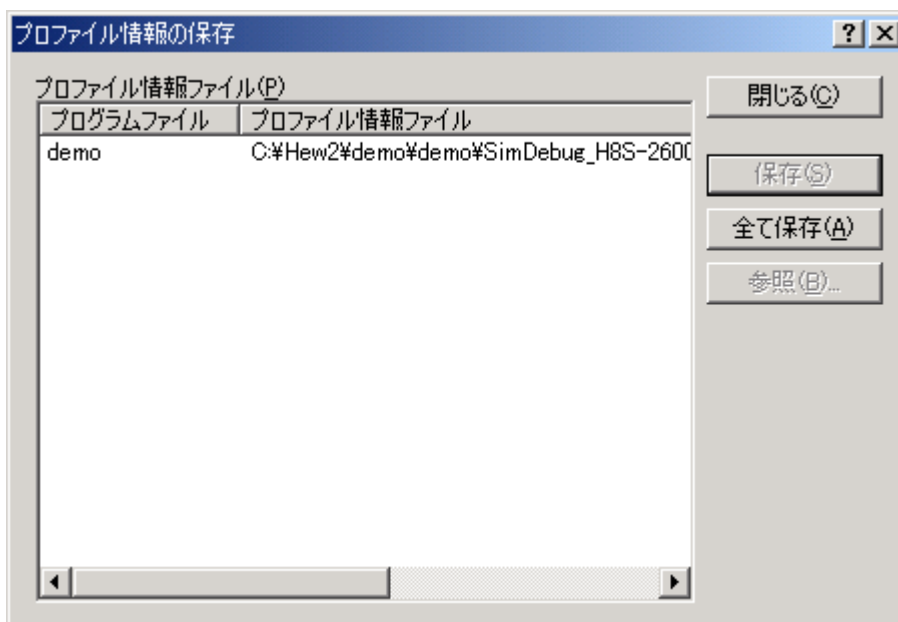


図 3-34 プロファイル情報の保存ダイアログボックス


### 3.6.12 注意事項

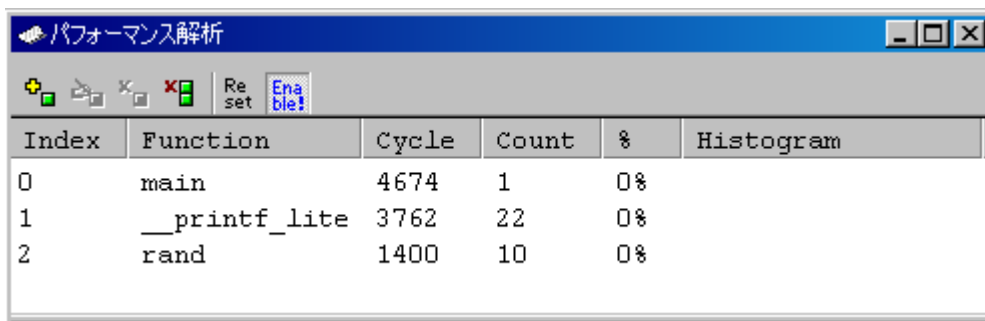
- (1) アプリケーションプログラムの実行サイクル数をプロファイル機能で測定した場合のデータには誤差があります。プロファイル機能では、アプリケーションプログラム全体の中で各関数が占める実行時間の比率を調べることができますが、より厳密に関数の実行サイクル数を測定したい場合には、パフォーマンス解析機能を使用してください。
- (2) デバッグ情報が無いロードモジュールでプロファイル情報の測定を行った場合、関数名を表示しない場合があります。
- (3) スタック情報ファイル(拡張子".SNI")はロードモジュールファイル(拡張子".ABS")と同一のディレクトリに置いてある必要があります。
- (4) 測定結果の蓄積はできません。
- (5) 測定結果の編集はできません。

## 3.7 パフォーマンスを解析する

関数名を指定してパフォーマンス解析する場合は、[パフォーマンス解析]ウィンドウを使用します。

### 3.7.1 パフォーマンス解析ウィンドウを開く

[パフォーマンス解析]ウィンドウを開くには、[表示->パフォーマンス->パフォーマンス解析]を選択するか、[パフォーマンス解析]ツールバーボタン  をクリックします。



Index	Function	Cycle	Count	%	Histogram
0	main	4674	1	0%	
1	__printf_lite	3762	22	0%	
2	rand	1400	10	0%	

図 3-35 パフォーマンス解析ウィンドウ

本ウィンドウは、指定関数ごとの実行サイクル数を表示します。  
実行サイクル数は、以下の計算で求めています。

実行サイクル数 = 指定関数からのリターン命令実行時累計実行サイクル数 - 指定関数コール命令実行時累計実行サイクル数

表示する項目は以下の通りです。

[Index]	設定条件のインデックス番号
[Function]	測定対象の関数名（または関数の開始アドレス）
[Cycle]	当該関数の累計実行サイクル数
[Count]	当該関数の累計呼び出し回数
[%]	プログラム全体の実行サイクル数に占める当該関数の実行サイクル数の割合
[Histogram]	上記割合のヒストグラム表示

### 3.7.2 評価関数を設定する

[パフォーマンス解析]ウィンドウを開いたら、評価する関数を設定します。ポップアップメニューから[範囲の追加...]を選択すると、[パフォーマンスオプション]ダイアログボックスが開きます。また、'Insert'キーでも[パフォーマンスオプション]ダイアログボックスを開くことができます。

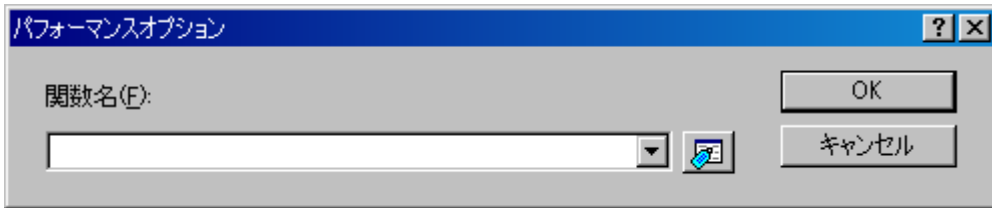


図 3-36 パフォーマンスオプションダイアログボックス

本ダイアログボックスでは、性能評価する関数（ラベルも可）を設定します。評価関数は255個まで設定可能です。

[OK]ボタンをクリックすることにより、評価関数を設定します。[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。

設定した評価関数を選択後、ポップアップメニューの[範囲の編集]を選択すると[パフォーマンスオプション]ダイアログボックスを表示して、評価関数を変更できます。また、'Enter'キーでも[パフォーマンスオプション]ダイアログボックスを開くことができます。

### 3.7.3 データ収集を開始する

ポップアップメニューの[有効]をチェックして、プログラムを実行するとパフォーマンスデータを収集できます。

### 3.7.4 データをリセットする

ポップアップメニューから[リセット]を選択すると、現在のプログラムのパフォーマンスデータをクリアします。

### 3.7.5 評価関数を削除する

評価関数を選択した状態で、ポップアップメニューから[範囲の削除]を選択すると、選択された評価関数を削除し、他の範囲のデータを再計算します。また、'Delete'キーでも評価関数を削除することができます。

### 3.7.6 すべての評価関数を削除する

ポップアップメニューから[全ての範囲を削除]を選択すると、現在の評価関数をすべて削除し、パフォーマンスデータをクリアします。

### 3.7.7 現在の表示内容をセーブする


現在ウィンドウに表示している内容をテキストファイルにセーブすることが出来ます。ポップアッ

メニューから[ファイルに保存...]を選択してください。

## 3.8 コードカバレッジを測定する

[カバレッジ]ウィンドウは、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 カバレッジおよび C1 カバレッジ)を収集し、結果を表示します。

### 3.8.1 カバレッジウィンドウを開く

[表示->コード->カバレッジ...]を選択するか[カバレッジ]ツールバーボタンをクリックすると、[カバレッジの設定]ダイアログボックスが開きます。

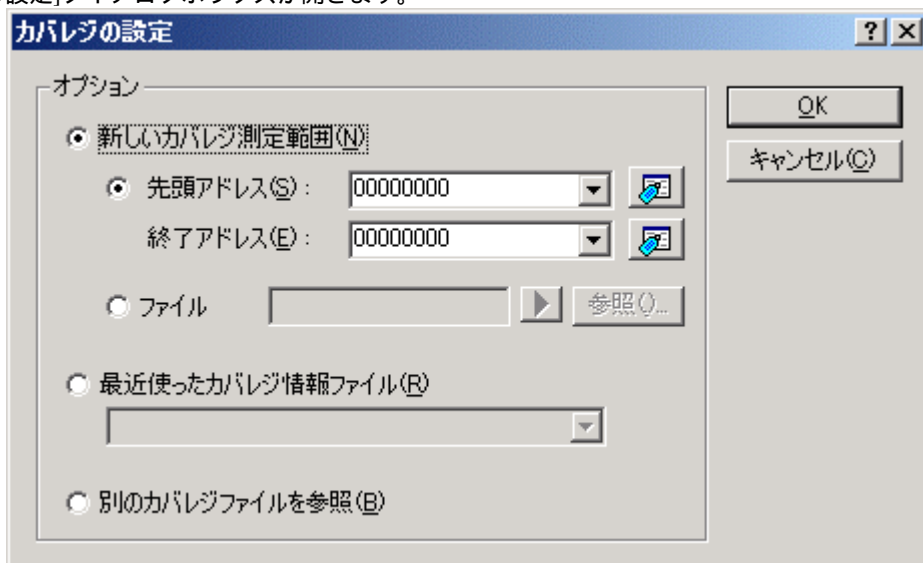


図 3-37 カバレッジの設定ダイアログボックス

[カバレッジの設定]ダイアログボックスでは、カバレッジ測定範囲を指定します。新たなカバレッジ範囲を設定するには、以下に示す 2 通りの方法があります。

- [新しいカバレッジ測定範囲]で開始アドレスおよび終了アドレスを指定する  
指定項目は以下のとおりです。

[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）

- [新しいカバレッジ測定範囲]でファイルを指定する  
指定項目は以下のとおりです。

[ファイル]      現在のプロジェクト中の“.C”または“.CPP”を拡張子にも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。拡張子を省略した場合は、“.C”を補います。“.C”または“.CPP”以外の拡張子を持つファイルは指定できません。プレースホルダまたは[参照...]ボタンが利用できます。

カバレッジ情報ファイルに保存した内容を再設定する場合は、[最近使ったカバレッジ情報ファイル]から選択するか、[別のカバレッジファイルを参照]でファイルオープンダイアログボックスを表示して選択します。

[最近使ったカバレッジ情報ファイル]では最近保存されたファイルを4個まで表示します。

[OK]ボタンをクリックすると[カバレッジ]ウィンドウを表示します。

アドレス指定で[カバレッジ]ウィンドウが既に表示されている場合は、そのウィンドウに設定が追加されます。

(1) カバレッジウィンドウ (アドレス指定)

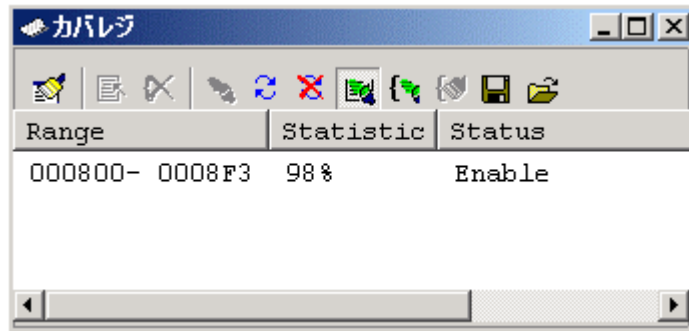


図 3-38 カバレッジウィンドウ(アドレス指定)

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)
[Status]	各カバレッジ範囲の有効/無効状態

[カバレッジ]ウィンドウをクローズすると取得したカバレッジ情報と取得条件の設定をクリアします。

(2) カバレッジウィンドウ (ソースファイル指定)

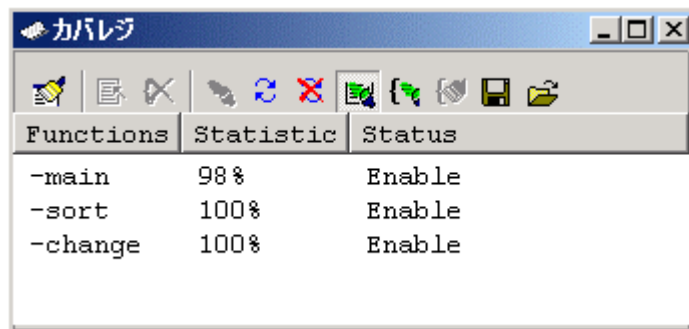


図 3-39 カバレッジウィンドウ(ソースファイル指定)

カバレッジ範囲とカバレッジ統計情報を表示します。

### 3. デバッグ

表示する項目は以下の通りです。

[Functions]	カバレジ対象の関数
[Statistic]	C0 カバレジ値(パーセント表示)
[Status]	各関数の有効/無効状態

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

[カバレジ]ウィンドウをクローズすると取得したカバレジ情報と取得条件の設定をクリアします。

#### 3.8.2 カバレジ情報をすべて取得する

ポップアップメニューで[すべて有効]をチェック後、命令を実行するとすべてのカバレジ情報を取得します。デフォルトで[すべて有効]はチェックされています。

#### 3.8.3 カバレジ情報をすべてクリアする

ポップアップメニューから[すべてクリア]を選択すると、取得したすべてのカバレジ情報をクリアします。

#### 3.8.4 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、[カバレジ]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。

#### 3.8.5 新しいカバレジ測定範囲を設定する

ポップアップメニューで[測定範囲追加...]を選択すると、[カバレジの設定]ダイアログボックス(図 3-37)を表示します。[カバレジの設定]ダイアログボックスについては、「3.8.1 カバレジウィンドウを開く」を参照してください。

#### 3.8.6 カバレジ測定範囲を変更する

##### (1) カバレジ測定範囲をアドレスで指定した場合

カバレジ測定範囲を選択し、ポップアップメニューで[測定範囲編集...]を選択すると、[カバレジ測定範囲]ダイアログボックスを表示します。

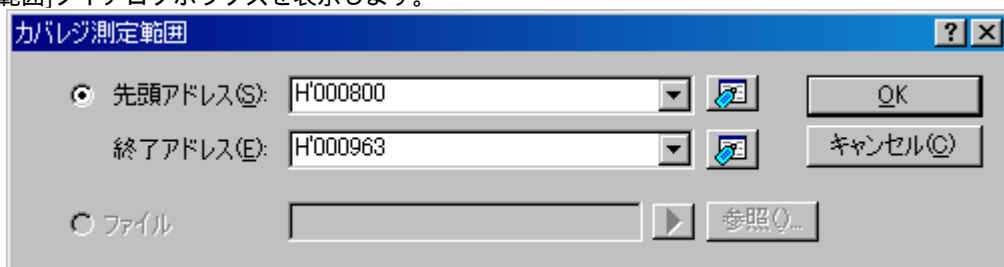


図 3-40 カバレジ測定範囲ダイアログボックス(アドレス指定)

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。

[先頭アドレス]	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測定範囲を変更します。

## (2) カバレッジ測定範囲をソースファイルで指定した場合

ポップアップメニューで[測定範囲編集...]を選択すると、[カバレッジ測定範囲]ダイアログボックスを表示します。

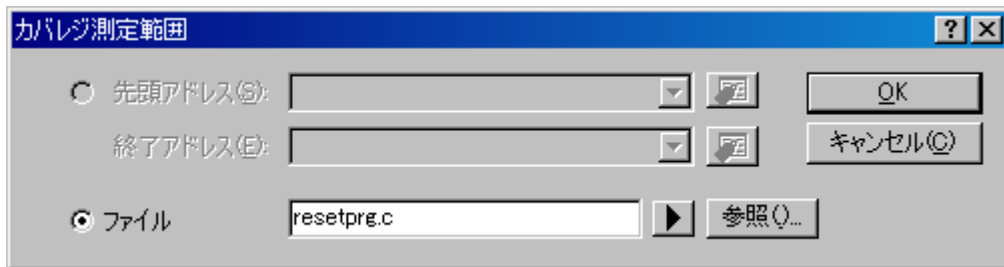


図 3-41 カバレッジ測定範囲ダイアログボックス(ソースファイル指定)

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"を型名にも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。ブレースホルダまたは[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。

### 3.8.7 カバレッジ測定範囲を削除する

カバレッジ測定範囲を選択し、ポップアップメニューから[カバレッジ測定範囲削除]を選択すると、選択したカバレッジ測定範囲を削除します。

### 3.8.8 カバレッジ情報を取得する

カバレッジ測定範囲を選択し、ポップアップメニューから[有効]をチェック後、命令を実行するとカバレッジ情報を取得します。デフォルトで[有効]はチェックされています。

### 3.8.9 カバレッジ情報をクリアする

カバレッジ測定範囲を選択し、ポップアップメニューから[クリア]を選択すると、取得したカバレッジ情報をクリアします。

### 3.8.10 カバレッ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、カバレッジ情報をファイルに保存するための[カバレッジ情報を保存]ダイアログボックスを表示します。

### 3. デバッグ

---



図 3-42 カバレッジ情報を保存ダイアログボックス

保存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。プレースホルダまたは[参照...]ボタンが使用できます。

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”または“.TXT”以外を入力するとエラーメッセージを出力します。

#### 3.8.11 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、カバレッジ情報をファイルからロードするための[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。

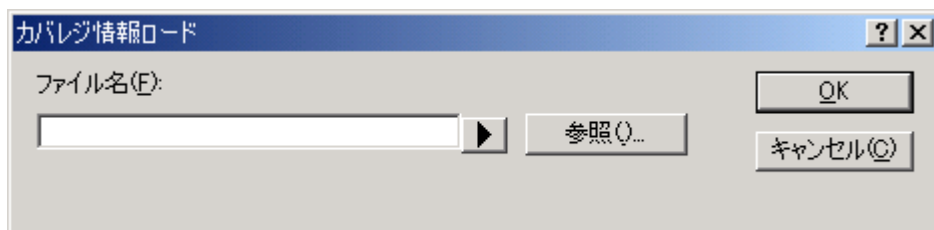


図 3-43 カバレッジ情報ロードダイアログボックス

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。プレースホルダまたは[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。

#### 3.8.12 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、[カバレッジ]ウィンドウ内容を最新に更新します。

#### 3.8.13 Confirmation Request ダイアログボックス

[すべてクリア]、[クリア]、[測定範囲編集...]、[測定範囲削除]をクリックするか、[カバレッジ]ウィンドウを閉じようとする時、確認のダイアログボックスを表示します。



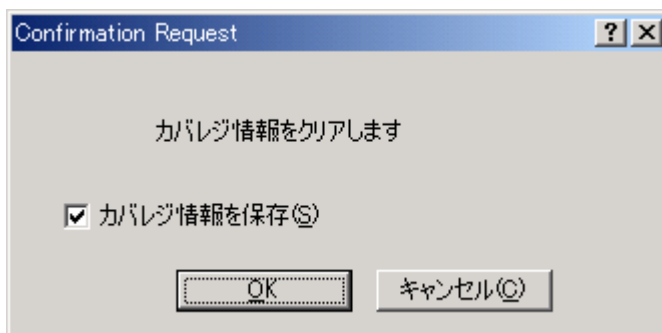


図 3-44 Confirmation Request ダイアログボックス

[OK]ボタンをクリックすると、カバレジデータをクリアします。

[カバレジ情報を保存]をチェックすると、「図 3-42」に示す[カバレジ情報を保存]ダイアログボックスを表示し、クリアする前にカバレジデータをファイルに保存できます。

### 3.8.14 カバレジ情報を保存ダイアログボックス

[ファイル->セッションの保存]メニューオプションをクリックすると、[カバレジ情報を保存]ダイアログボックスを表示します。[カバレジ]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。

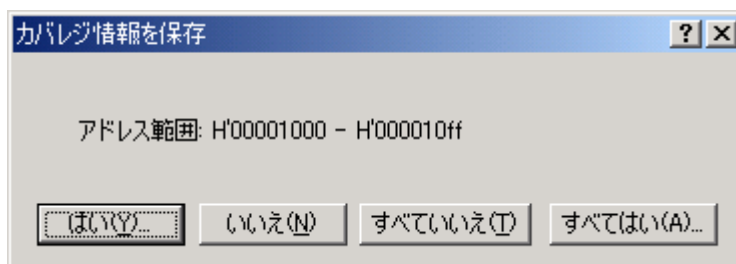


図 3-45 カバレジ情報を保存ダイアログボックス

- [カバレジ]ウィンドウが複数開いているとウィンドウ数分の[カバレジ情報を保存]ダイアログボックスが開きます。
- [すべていいえ]ボタンをクリックすると、すべてのカバレ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[カバレジ]ウィンドウデータを1個のファイルに保存します。

## 3.8.15 エディタウィンドウへのカバレッジ結果表示

命令実行済のソース行に対応するカバレッジ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。[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、対応するカバレッジカラムの表示も更新します。

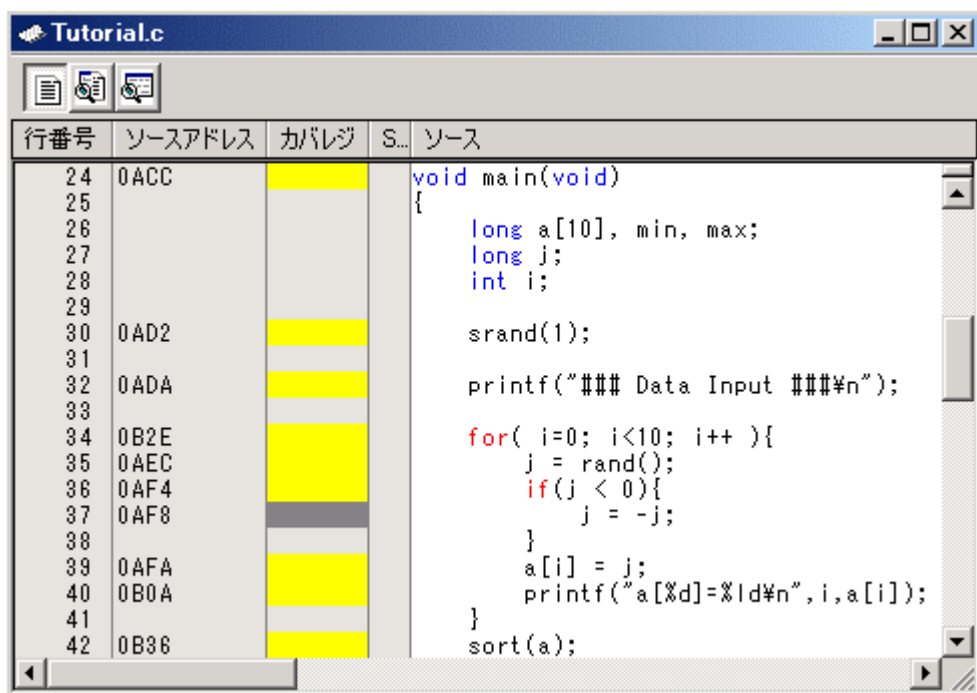
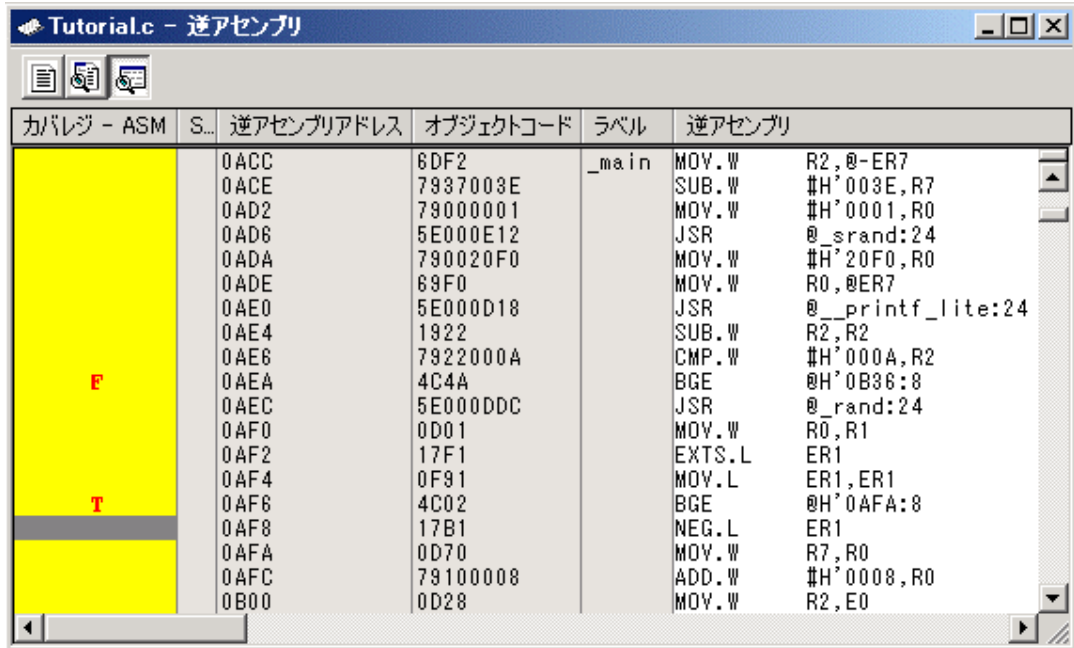


図 3-46 カバレッジカラム (ソース)

### 3.8.16 逆アセンブリウィンドウへのカバレジ結果表示

命令実行済の逆アセンブリ行に対応するカバレジ-ASM カラムを強調表示することで[逆アセンブリ]ウィンドウにもカバレジ結果を表示します。[カバレジ]ウィンドウでカバレジに関する設定を変更すると、対応するカバレジ-ASM カラムの表示も更新します。



カバレジ - ASM	S...	逆アセンブリアドレス	オブジェクトコード	ラベル	逆アセンブリ
		0ACC	6DF2	_main	MOV.W R2,@-ER7
		0ACE	7937003E		SUB.W #H'003E,R7
		0AD2	79000001		MOV.W #H'0001,R0
		0AD6	5E000E12		JSR @_srand:24
		0ADA	790020F0		MOV.W #H'20F0,R0
		0ADE	69F0		MOV.W R0,@ER7
		0AE0	5E000D18		JSR @_printf_lite:24
		0AE4	1922		SUB.W R2,R2
		0AE6	7922000A		CMP.W #H'000A,R2
F		0AEA	4C4A		BGE @H'0B36:8
		0AEC	5E000DDC		JSR @_rand:24
		0AF0	0D01		MOV.W R0,R1
		0AF2	17F1	ER1	EXTS.L ER1
		0AF4	0F91		MOV.L ER1,ER1
T		0AF6	4C02		BGE @H'0AFA:8
		0AF8	17B1		NEG.L ER1
		0AFA	0D70		MOV.W R7,R0
		0AFC	79100008		ADD.W #H'0008,R0
		0B00	0D28		MOV.W R2,E0

図 3-47 カバレジカラム (逆アセンブリ)

## 3.9 手動で擬似割込みを発生させる

ウィンドウ上のボタンを押下して手動で擬似割込みを発生させるには、[トリガ]ウィンドウ、または[GUI I/O]ウィンドウを使用します。

### 3.9.1 トリガウィンドウ


[トリガ]ウィンドウを開くには、[トリガ->CPU->トリガ]を選択するか、[トリガ]ツールバーボタン  をクリックします。



図 3-48 トリガウィンドウ

本ウィンドウは、手動で擬似割込みを発生させるためのトリガボタンを表示します。トリガボタン押下時に発生する擬似割込みの内容は、[トリガ設定]ダイアログボックスで設定します。

トリガボタンは最大 256 個まで設定できます。

シミュレータ・デバッガの擬似割込み処理については、「2.19 擬似割込み」を参照してください。

## (1) トリガボタンを設定する

各トリガボタン押下時に発生する擬似割り込みの内容を設定するには[トリガ設定]ダイアログボックスを使用します。

[トリガ設定]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[設定...]を選択します。

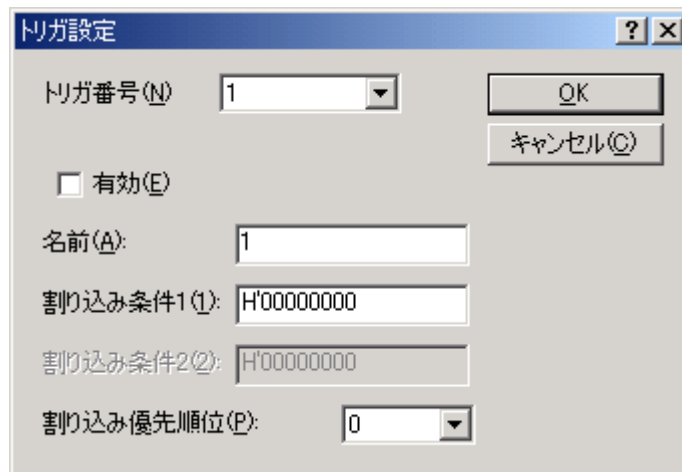


図 3-49 トリガ設定ダイアログボックス

本ダイアログボックスでは、トリガボタン押下時に発生する割り込みの内容を設定します。

- [トリガ番号] 設定するトリガボタンを選択します。
- [名前] [トリガ]ウィンドウに表示するトリガボタンの名前を指定します。
- [有効] チェックするとトリガボタンが有効になります。
- [割り込み条件 1] 割り込みベクタ番号を指定します。
- [割り込み優先順位] 割り込み優先順位を指定します。(接頭辞省略時は 16 進入力、16 進表示)(0~H'11)  
割り込みを受付けるかどうかの判定は、選択されたデバッグプラットフォームの CPU の仕様に従います。ただし、優先順位に H'8 以上を指定した場合は、常に割り込みを受付けます。

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。[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。

**【注】** 複数のトリガボタンの設定を変更した後に[キャンセル]ボタンをクリックすると、すべてのトリガボタンの設定変更が無効になります。

## (2) トリガボタンの数を変える


[トリガ]ウィンドウに表示するトリガボタンの数を変えるには、ポップアップメニューから[ボタンの数]のサブメニューで[4]、[16]、[64]、[256]を選択します。

## (3) トリガボタンのサイズを変える

[トリガ]ウィンドウに表示するトリガボタンのサイズを変えるには、ポップアップメニューから[サイズ]のサブメニューで[大]、[中]、[小]を選択します。

### 3. デバッグ

#### 3.9.2 GUI I/O ウィンドウ

[GUI I/O]ウィンドウを開くには、[表示->グラフィック->GUI I/O]を選択するか、[GUI I/O]ツールバーボタンをクリックします。

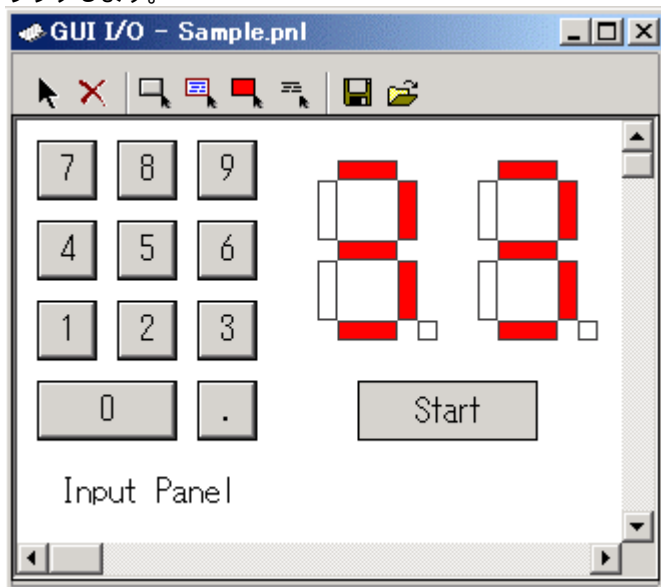



図 3-50 GUI I/O ウィンドウ

本ウィンドウは、手動で擬似割込みを発生させるためのボタンを設定します。ボタン押下時に発生する擬似割込みの内容は、[ボタンの設定]ダイアログボックスで設定します。

シミュレータ・デバッガの擬似割込み処理については、「2.19 擬似割込み」を参照してください。

##### (1) ボタンを設定する

擬似割込みを発生させるボタンを設定するには[ボタンの設定]ダイアログボックスを使用します。まずポップアップメニューの[ボタンの作成]を選択するか、ツールバーの[ボタンの作成]ボタンをクリックしてください。マウスポインターの形状が "+" になります。

この状態で作成するボタンの左上位置から右下位置までドラッグしてください。

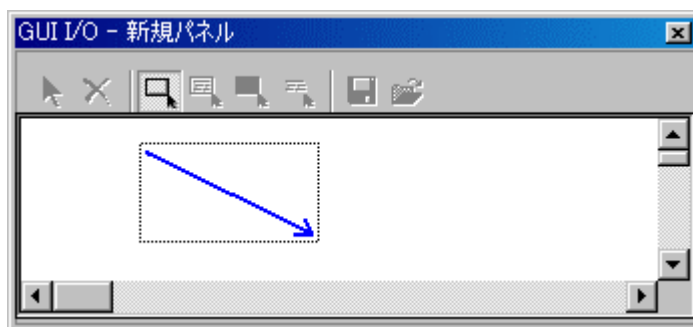


図 3-51 GUI I/O ウィンドウ (ボタンの作成)

次にこのボタンをダブルクリックして[ボタンの設定]ダイアログボックスを開きます。

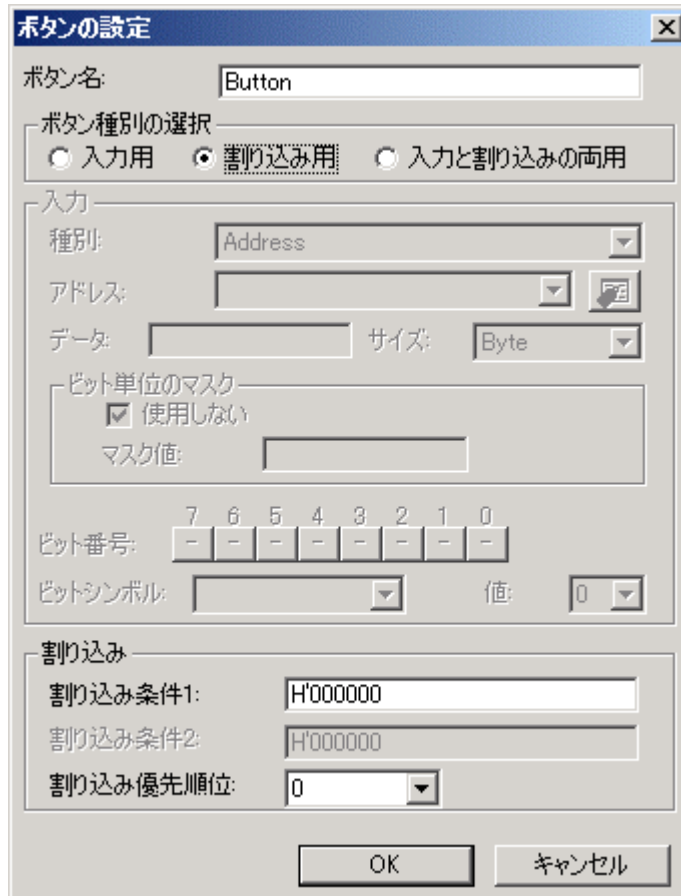


図 3-52 ボタンの設定ダイアログボックス


本ダイアログボックスでは、ボタン押下時に発生する割り込みの内容を設定します。

[ボタン名]	[GUI I/O]ウィンドウに表示するボタンの名前を指定します。
[ボタン種別の選択]	[割り込み用]、または[入力と割り込みの両用]を選択します。
[割り込み] [割り込み条件 1]	割り込みベクタ番号を指定します。
[割り込み優先順位]	割り込み優先順位を指定します。(接頭辞省略時は 16 進入力、16 進表示) (0 ~ H'11)
	割り込みを受けられるかどうかの判定は、選択されたデバッグプラットフォームの CPU の仕様に従います。ただし、優先順位に H'8 以上を指定した場合は、常に割り込みを受けれます。

## 3.10 標準入出力およびファイル入出力を行う

ユーザプログラムから標準入出力およびファイル入出力の I/O シミュレーションを行うには、[I/O シミュレーション]ウィンドウを利用します。

### 3.10.1 I/O シミュレーションウィンドウを開く

[I/O シミュレーション]ウィンドウを開くには、[表示->CPU->I/O シミュレーション]を選択するか、[I/O シミュレーション]ツールバーボタン  をクリックします。

### 3. デバッグ

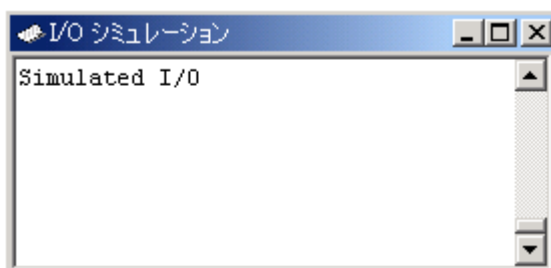


図 3-53 I/O シミュレーションウィンドウ

ユーザプログラムからの標準出力は本ウィンドウに出力されます。本ウィンドウ上でのキー入力がユーザプログラムへの標準入力となります。

#### 3.10.2 入出力機能

サポートしている入出力処理は以下の通りです。機能コードには、16 ビットアドレス版、24 ビットアドレス版、32 ビットアドレス版があります。使用する CPU に合わせて選択してください。

表 3-1 入出力機能一覧

番号	機能コード	機能名	内容
1	H'01 (16 ビットアドレス) H'11 (24 ビットアドレス) H'21 (32 ビットアドレス)	GETC	標準入力からの 1 バイト入力
2	H'02 (16 ビットアドレス) H'12 (24 ビットアドレス) H'22 (32 ビットアドレス)	PUTC	標準出力への 1 バイト出力
3	H'03 (16 ビットアドレス) H'13 (24 ビットアドレス) H'23 (32 ビットアドレス)	GETS	標準入力からの 1 行入力
4	H'04 (16 ビットアドレス) H'14 (24 ビットアドレス) H'24 (32 ビットアドレス)	PUTS	標準出力への 1 行出力
5	H'05 (16 ビットアドレス) H'15 (24 ビットアドレス) H'25 (32 ビットアドレス)	FOPEN	ファイルのオープン
6	H'06	FCLOSE	ファイルのクローズ
7	H'07 (16 ビットアドレス) H'17 (24 ビットアドレス) H'27 (32 ビットアドレス)	FGETC	ファイルからの 1 バイト入力
8	H'08 (16 ビットアドレス) H'18 (24 ビットアドレス) H'28 (32 ビットアドレス)	FPUTC	ファイルへの 1 バイト出力
9	H'09 (16 ビットアドレス) H'19 (24 ビットアドレス) H'29 (32 ビットアドレス)	FGETS	ファイルからの 1 行入力

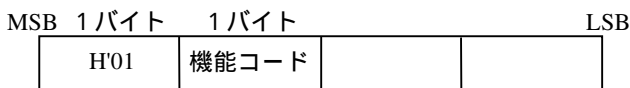


番号	機能コード	機能名	内 容
10	H'0A (16 ビットアドレス) H'1A (24 ビットアドレス) H'2A (32 ビットアドレス)	FPUTS	ファイルへの 1 行出力
11	H'0B	FEOF	エンドオブファイルのチェック
12	H'0C	FSEEK	ファイルポインタの移動
13	H'0D	FTELL	ファイルポインタの現在位置を得る

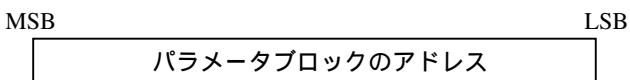
この機能を実現するためには、まず入出力用の特定の位置を [シミュレータシステム]ダイアログボックスの [I/O シミュレーションアドレス]で指定し、[有効]をチェック後、ユーザプログラムを実行します。シミュレータ・デバッガでは、ユーザプログラムの命令を実行中に、指定した位置へのサブルーチン分岐命令 (BSR, JSR) すなわち I/O シミュレーション命令を検出すると、R0, R1 ( H8/300, H8/300L シリーズ) または ER1 ( H8/300H, H8S, および H8SX シリーズ) の内容をパラメータとして入出力処理を行います。

したがって、I/O シミュレーションを行う前にユーザプログラムの中で次の設定をしておきます。

- ・ R0 レジスタ：機能コード



- ・ R1, または ER1 レジスタ：パラメータブロックのアドレス  
(パラメータブロックの内容は各機能の説明を参照してください。)



- ・ パラメータブロックおよび入出力バッファ領域の確保

なお、パラメータブロックの各パラメータにアクセスする場合は、該当するパラメータのサイズでアクセスしてください。

入出力処理が終了すると、I/O シミュレーション命令の次の命令からシミュレーションを再開します。

各入出力機能の詳細は、シミュレータヘルプを参照ください。

### 3. デバッグ

以下に標準入力（キーボード）から 1 文字入力する例を示します。I/O シミュレーションアドレスとしてラベル SYS\_CALL を指定します。

```
MOV.W    #H'0101, R0
MOV.W    #PARM, R1
JSR      @SYS_CALL
STOP     NOP
SYS_CALL NOP
PARM     .DATA.W  INBUF
INBUF    .RES.B   2
.END
```

#### 3.11 仮想入出力パネルを作成する

シミュレータ・デバッガにはユーザーターゲットシステムの簡単なキー入力パネルおよび出力パネルをウィンドウ上で模擬する GUI 入出力機能があります。この仮想的な入出力パネルを作成するには GUI 入出力ウィンドウを使用します。GUI 入出力ウィンドウでは、ウィンドウ上に仮想のボタンを配置してデータを入力することや仮想の LED を配置してデータを出力することができます。

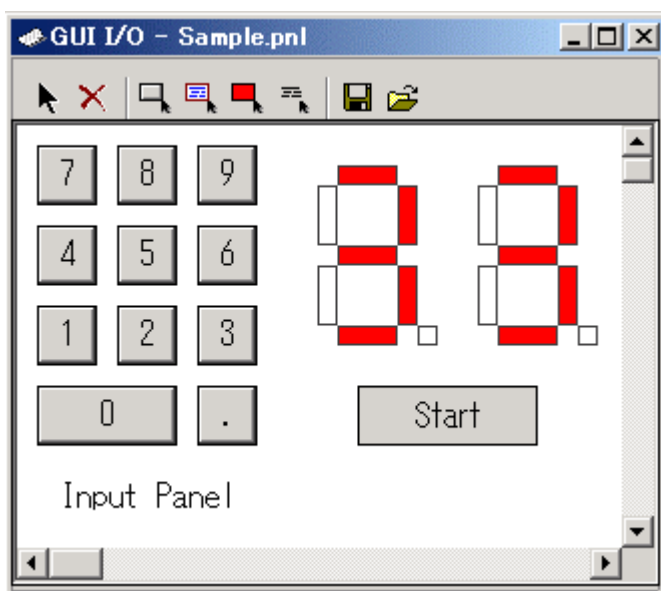



図 3-54 GUI I/O ウィンドウ例

##### 3.11.1 GUI 入出力ウィンドウを開く

[表示->グラフィック->GUI I/O]を選択するか、[GUI I/O]ツールバーボタンをクリックすると、GUI I/O ウィンドウが開きます。

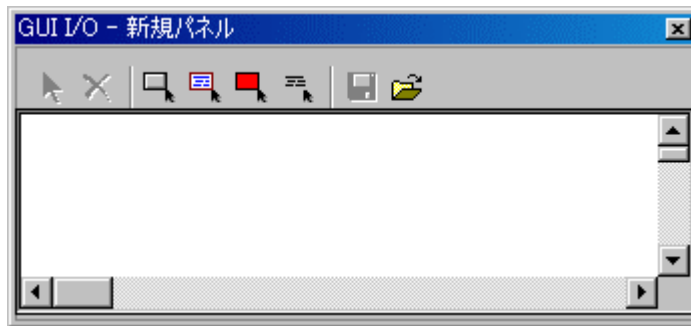



図 3-55 GUI I/O ウィンドウ

GUI 入出力ウィンドウでは、以下のアイテムが配置できます。

- ボタン  
ボタンを押下することにより、仮想ポートへのデータ入力や、仮想割り込みを発生させます。
- ラベル  
指定したアドレス(もしくはビット)に指定した値が書き込まれた際に、文字列を表示/消去します。
- LED  
指定したアドレス(もしくはビット)に指定した値が書き込まれた際に、指定した色で表示します(LED点灯の代用)。
- テキスト  
テキスト文字列を表示します。

### 3.11.2 ボタンを作成する

ツールバーの  ボタンをクリックするかポップアップメニューの[ボタンの作成]を選択してください。マウスカーソルの形状が "+" になります。

この状態で作成するボタンの左上位置から右下位置までドラッグしてください。

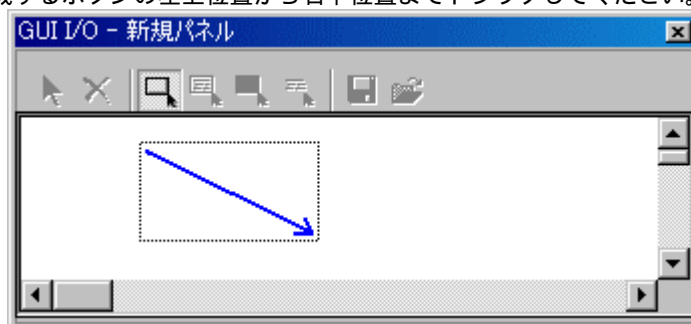


図 3-56 GUI I/O ウィンドウ (ボタン作成)

#### (1) ボタンクリック時のイベントを指定する



ツールバーの  ボタンをクリックした状態で、作成したボタンをダブルクリックしてください。以下のダイアログボックスが開きます。



図 3-57 ボタンの設定ダイアログボックス

ボタン名、入力ポートアドレスおよび入力データを指定してください。  
ボタン名に空白文字を使用することはできません。

#### 3.11.3 ラベルを作成する

ツールバーの  ボタンをクリックするか、ポップアップメニューの[ラベルの作成]を選択してください。マウスカーソルの形状が "+" に変化します。

この状態で作成する位置の左上位置から右下位置までドラッグしてください。ラベル枠を表示します。

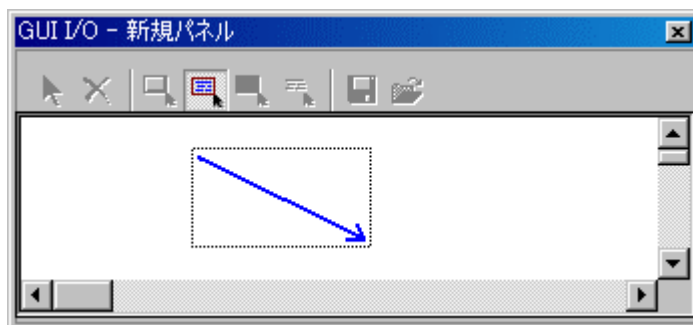



図 3-58 GUI I/O ウィンドウ (ラベル作成)

ツールバーの  ボタンをクリックするか、ポップアップメニューの[アイテムの選択]を選択後、ラベル枠をダブルクリックしてください。ラベル内容を設定するダイアログボックスが開きます。イベント発生時の動作を指定してください。ラベル名に空白文字を使用することはできません。

(1) 指定ビットへの任意の値書き込みの場合

アドレス 0x3E0 のビット 3 が 0 であれば"印刷中"、1 であれば"印刷可"と表示する設定を以下に示します。



図 3-59 ラベルの設定ダイアログボックス (ビット指定)

### 3. デバッグ

(2) 指定アドレスへの任意の値書き込みの場合

アドレス 0x3E0 にデータ 0x10 が書き込まれた時に"印刷中"、データ 0x20 が書き込まれた時に"印刷可"と表示する設定を以下に示します。

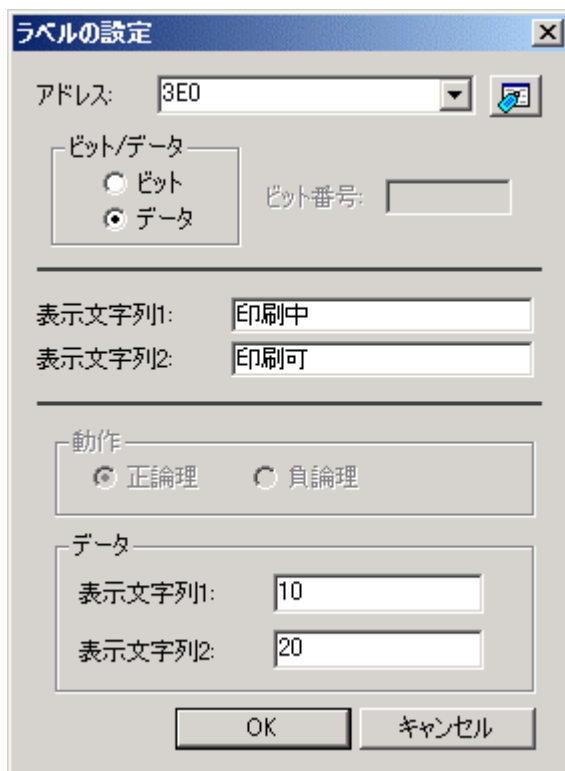



図 3-60 ラベルの設定ダイアログボックス (データ指定)

#### 3.11.4 LED を作成する

ツールバーの  ボタンをクリックするか、ポップアップメニューの[LED の作成]を選択してください。マウスカーソルの形状が "+" に変化します。

この状態で作成する位置の左上位置から右下位置までドラッグしてください。LED 出力枠を表示します。

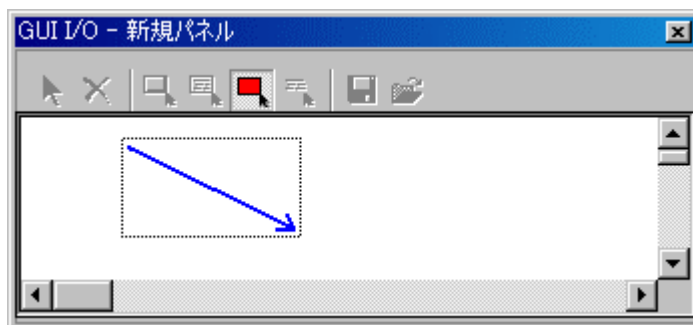



図 3-61 GUI I/O ウィンドウ (LED 作成)

ツールバーの  ボタンをクリックするか、ポップアップメニューの[アイテムの選択]を選択後、LED 出力枠をダブルクリックしてください。LED 出力を設定するダイアログボックスが開きます。イベント発生時の動作を指定してください。

(1) 指定ビットへの任意の値書き込みの場合

アドレス 0x3E0 のビット 2 が 0 であれば緑色、1 であれば赤色表示する設定を以下に示します。



図 3-62 LED の設定ダイアログボックス (ビット指定)

(2) 指定アドレスへの任意の値書き込みの場合

アドレス 0x3E0 にデータ 0x10 が書き込まれた時に緑色表示、データ 0x20 が書き込まれた時に赤色表示する設定を以下に示します。




図 3-63 LED の設定ダイアログボックス (データ指定)

表示色 1、表示色 2 ボタンをクリックするとカラー選択ダイアログボックスが開きます。



### 3.11.5 テキスト文字列を作成する

ツールバーの  ボタンをクリックするか、ポップアップメニューの[テキストの作成]を選択してください。マウスカーソルの形状が"+"に変化します。

この状態で作成するテキスト文字列の左上位置から右下位置までドラッグしてください。

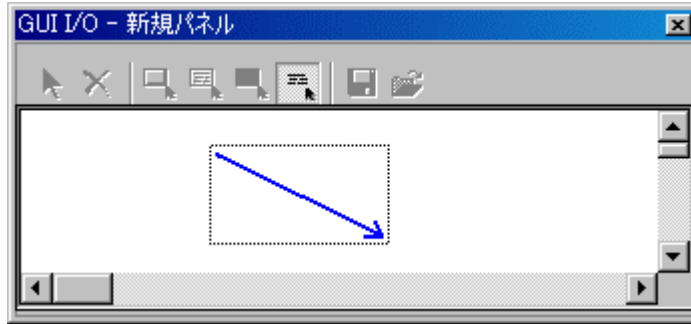



図 3-64 GUI I/O ウィンドウ (文字列作成)

#### (1) テキスト文字列の表示形式を指定する

ツールバーの  ボタンをクリックした状態で、作成したテキスト文字列をダブルクリックしてください。

以下のテキスト文字列の表示形式を設定するためのダイアログボックスが開きます。




図 3-65 テキストの設定ダイアログボックス

[フォント]ボタンを押して、表示するテキスト文字列のフォントとサイズを設定してください。

[テキスト]ボタンを押して、表示するテキスト文字列の文字色を設定してください。

[背景]ボタンを押して、表示するテキスト文字列の背景色を設定してください。

### 3.11.6 アイテムのサイズ/配置を変更する

ツールバーの  ボタンをクリックした状態で、変更するアイテムをクリックしてください。以下のようにアイテムが選択された状態となります。

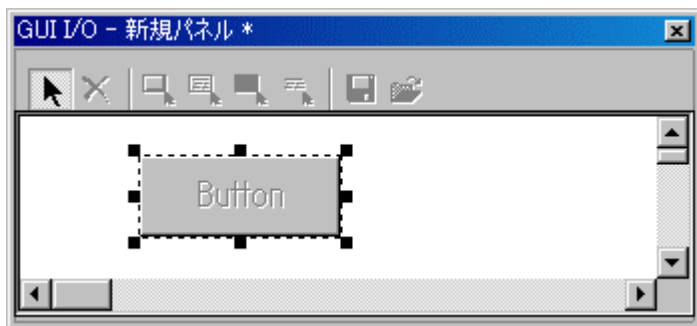




図 3-66 GUI I/O ウィンドウ (アイテムの選択)


この状態でドラッグすることにより、アイテムサイズの変更や移動ができます。

#### 3.11.7 アイテムをコピーする

ツールバーのボタンをクリックするか、ポップアップメニューの[コピー]を選択してください。マウскарソルの形状が"+"に変化します。


この状態でコピー元のアイテムをクリックしてください。ツールバーのボタンをクリックするか、ポップアップメニューの[貼り付け]を選択してください。コピー元と同サイズのアイテムが作成されます。

#### 3.11.8 アイテムを削除する

ツールバーのボタンをクリックするか、ポップアップメニューの[削除]を選択してください。マウскарソルの形状が"+"に変化します。

この状態で削除するアイテムをクリックしてください。

#### 3.11.9 グリッド線を表示する

ツールバーのボタンをクリックするか、ポップアップメニューの[グリッドの表示]を選択してください。背景にグリッドを表示します。

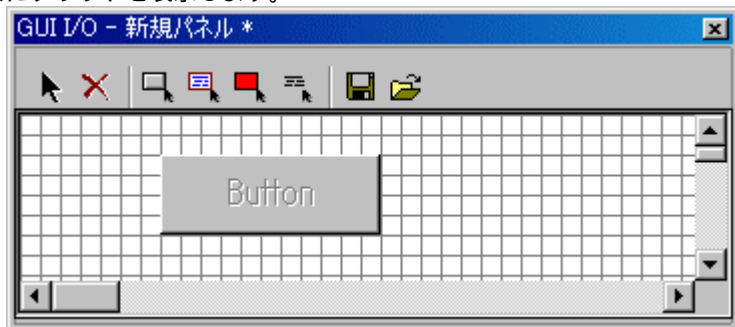





図 3-67 GUI I/O ウィンドウ (グリッドの表示)

再度ボタンをクリックすることにより、グリッドを非表示にします。

### 3.11.10 入出力パネル情報を保存する

作成した入出力パネルは、その情報をファイルに保存することにより、再利用することができます。ツールバーの  ボタンをクリックするか、ポップアップメニューの[保存]を選択してください。[GUI I/O パネルファイルの保存]ダイアログボックスが開きます。保存先ディレクトリとファイル名を指定してください。

### 3.11.11 入出力パネル情報を読み込む

ツールバーの  ボタンをクリックするか、ポップアップメニューの[読み込み]を選択してください。[GUI I/O パネルファイルの読み込み]ダイアログボックスが開きます。読み込むファイルを指定してください。読み込む前のパネル情報は削除されません。



---

## 4. ウィンドウ

---

ウィンドウ一覧を表 4-1に示します。

ツールバーボタンについては、シミュレータ・デバッガヘルプを参照ください。

表 4-1 ウィンドウ一覧

ウィンドウ名	機能
IO	I/O レジスタを見る
I/O シミュレーション	標準入出力およびファイル入出力を行う
イベントポイント	シミュレータ・デバッガのブレークポイントを使用する
ウォッチ	変数を表示する（任意の変数）
エディタ	ソースコードを表示する
画像	メモリ内容を画像形式で表示する
カバレッジ	コードカバレッジを測定する
逆アセンブリ	アセンブリ言語コードを表示する
コマンドライン	コマンドラインインタフェースでデバッグ
スタックトレース	関数呼び出し履歴を見る
ステータス	現在の状態の参照
トリガ	手動で擬似割り込みを発生させる
トレース	トレース情報を見る
波形	メモリ内容を波形形式で表示する
パフォーマンス解析	パフォーマンスを解析する
プロファイル/プロファイル チャート	プロファイル情報を見る
メモリ	メモリ領域を見る
ラベル	ラベルを参照/設定する
レジスタ	レジスタ内容を見る
ローカル	変数を表示する（ローカル変数）
GUI I/O	仮想入出力パネルを作成する

#### 4. ウィンドウ

---

---

## 5. コマンドライン

---

### 5.1 コマンド一覧 (機能別)

機能別のコマンド一覧を示します。  
各コマンドのシンタックスは、ヘルプを参照ください。

#### 5.1.1 実行関連

コマンド名	短縮形	説明
GO	GO	ユーザプログラムの実行
GO_RESET	GR	リセットベクタからのユーザプログラムの実行
GO_TILL	GT	テンポラリブレークポイントまでのユーザプログラムの実行
HALT	HA	ユーザプログラムの停止
RESET	RE	CPUのリセット
STEP	ST	ステップ実行(命令単位またはソース行単位)
STEP_MODE	SM	ステップモードの設定
STEP_OUT	SP	PC位置の関数を終了するまでのステップ実行
STEP_OVER	SO	ステップオーバ実行
STEP_RATE	SR	ステップ実行速度の設定、表示

#### 5.1.2 ダウンロード関連

コマンド名	短縮形	説明
BUILD	BU	カレントプロジェクトのビルド
BUILD_ALL	BL	カレントプロジェクトのすべてをビルド
BUILD_FILE	BF	ファイルのコンパイル
BUILD_MULTIPLE	BM	複数プロジェクトのビルド
CLEAN	CL	ビルドの中間ファイルおよび出力ファイルの削除
DEFAULT_OBJECT_FORMAT	DO	デフォルトオブジェクト(プログラム)フォーマットの設定
FILE_LOAD	FL	オブジェクト(プログラム)ファイルのロード
FILE_LOAD_ALL	LA	すべてのオブジェクト(プログラム)ファイルのロード
FILE_SAVE	FS	メモリ内容のファイルセーブ
FILE_UNLOAD	FU	オブジェクト(プログラム)ファイルのアンロード
FILE_UNLOAD_ALL	UA	すべてのオブジェクト(プログラム)ファイルのアンロード
FILE_VERIFY	FV	ファイル内容とメモリ内容の比較
GENERATE_MAKE_FILE	GM	カレントワークスペースの make ファイル作成

## 5. コマンドライン

### 5.1.3 レジスタ操作関連

コマンド名	短縮形	説明
REGISTER_DISPLAY	RD	レジスタ値の表示
REGISTER_SET	RS	レジスタ値の設定

### 5.1.4 メモリ操作関連

コマンド名	短縮形	説明
CACHE	-	メモリアクセスの有効化/無効化
MEMORY_COMPARE	MC	メモリ内容の比較
MEMORY_DISPLAY	MD	メモリ内容の表示
MEMORY_EDIT	ME	メモリ内容の変更
MEMORY_FILL	MF	指定データによるメモリ内容の一括変更
MEMORY_FIND	MI	メモリ内容の検索
MEMORY_MOVE	MV	メモリブロックの移動
MEMORY_TEST	MT	メモリブロックのテスト

### 5.1.5 アセンブル/逆アセンブル関係

コマンド名	短縮形	説明
ASSEMBLE	AS	アセンブルの実行
DISASSEMBLE	DA	逆アセンブル表示
SYMBOL_ADD	SA	シンボルの設定
SYMBOL_CLEAR	SC	シンボルの削除
SYMBOL_LOAD	SL	シンボル情報ファイルのロード
SYMBOL_SAVE	SS	シンボル情報のファイルセーブ
SYMBOL_VIEW	SV	シンボルの表示

### 5.1.6 ブレーク設定関連

コマンド名	短縮形	説明
BREAKPOINT	BP	実行命令位置によるブレークポイントの設定
BREAK_ACCESS	BA	メモリ範囲のアクセスによるブレーク条件の設定
BREAK_CLEAR	BC	ブレークポイントの削除
BREAK_CYCLE	BCY	サイクルによるブレーク条件の設定
BREAK_DATA	BD	メモリのデータ値によるブレーク条件の設定
BREAK_DATA_DIFFERENCE	BDD	メモリのデータ値の変化量(差分)によるブレーク条件の設定
BREAK_DATA_INVERSE	BDI	メモリのデータ値の符号反転によるブレーク条件の設定
BREAK_DISPLAY	BI	ブレークポイント一覧の表示
BREAK_ENABLE	BE	ブレークポイントの有効/無効の切換え
BREAK_REGISTER	BR	レジスタのデータ値によるブレーク条件の設定
BREAK_SEQUENCE	BS	実行順序を指定したブレークポイントの設定
SET_DISASSEMBLY_SOFT_BRE	SDB	逆アセンブリレベルのソフトウェアブレークポイントの設定



コマンド名	短縮形	説明
AK		または解除
SET_SOURCE_SOFT_BREAK	SSB	ソースレベルのソフトウェアブレークポイントの設定または解除
STATE_DISASSEMBLY_SOFT_BREAK	TDB	逆アセンブリレベルのソフトウェアブレークポイントの有効化または無効化
STATE_SOURCE_SOFT_BREAK	TSB	ソースレベルのソフトウェアブレークポイントの有効化または無効化

### 5.1.7 トレース関連

コマンド名	短縮形	説明
TRACE	TR	トレース情報の表示
TRACE_ACQUISITION	TA	トレース情報取得の有効/無効の切換え
TRACE_SAVE	TV	トレース情報をファイルへ出力
TRACE_STATISTIC	TST	統計情報解析の実行

### 5.1.8 カバレッジ計測関連

コマンド名	短縮形	説明
COVERAGE	CV	カバレッジ測定の有効化/無効化
COVERAGE_DISPLAY	CVD	カバレッジ情報の表示
COVERAGE_LOAD	CVL	カバレッジ情報のロード
COVERAGE_RANGE	CVR	カバレッジ範囲の設定
COVERAGE_SAVE	CVS	カバレッジ情報のセーブ

### 5.1.9 パフォーマンス測定関連

コマンド名	短縮形	説明
ANALYSIS	AN	性能分析機能の有効化/無効化
ANALYSIS_RANGE	AR	性能評価関数の設定、表示
ANALYSIS_RANGE_DELETE	AD	性能分析範囲の削除
PROFILE	PR	プロファイルの有効化/無効化
PROFILE_DISPLAY	PD	プロファイル情報の表示
PROFILE_SAVE	PS	プロファイル情報のファイル出力

### 5.1.10 ウォッチ関連

コマンド名	短縮形	説明
WATCH_ADD	WA	Watch アイテムの追加
WATCH_AUTO_UPDATE	WU	Watch アイテムの自動更新の設定または解除
WATCH_DELETE	WD	Watch アイテムの削除
WATCH_DISPLAY	WI	ウォッチウィンドウの内容の表示

## 5. コマンドライン

コマンド名	短縮形	説明
WATCH_EDIT	WE	Watch アイテムの値の編集
WATCH_EXPAND	WX	Watch アイテムの展開または縮小
WATCH_RADIX	WR	Watch アイテムの表示基数の変更
WATCH_RECORD	WO	Watch アイテムの値更新履歴をファイルに出力
WATCH_SAVE	WS	ウォッチウィンドウの表示内容をファイルに保存

### 5.1.11 スクリプト/ログファイル関連

コマンド名	短縮形	説明
!	-	コメント
ASSERT	-	コンディションのチェック
AUTO_COMPLETE	AC	オートコンプリート機能の有効化 / 無効化
ERASE	ER	コマンドラインウィンドウの内容のクリア
EVALUATE	EV	式の計算
LOG	LO	ロギングファイルの操作
SLEEP	-	コマンド実行の遅延
SUBMIT	SU	コマンドファイルの実行
TCL	-	TCL の情報を表示

### 5.1.12 マップ関連

コマンド名	短縮形	説明
MAP_DISPLAY	MA	メモリリソース設定の表示
MAP_SET	MS	メモリリソースの設定

### 5.1.13 シミュレーションの設定関連

コマンド名	短縮形	説明
EXEC_MODE	EM	実行モードの設定、表示
EXEC_STOP_SET	ESS	割り込み発生時の実行モードの設定、表示

### 5.1.14 標準入出力およびファイル入出力関連

コマンド名	短縮形	説明
SIMULATEDIO_CLEAR	SIOC	I/O シミュレーションウィンドウの内容のクリア
TRAP_ADDRESS	TP	I/O シミュレーションアドレスの設定
TRAP_ADDRESS_DISPLAY	TD	I/O シミュレーションアドレス設定の表示
TRAP_ADDRESS_ENABLE	TE	I/O シミュレーション有効化 / 無効化

## 5.1.15 ユーティリティ関連

コマンド名	短縮形	説明
HELP	HE	コマンドラインヘルプの表示
INITIALIZE	IN	デバッグプラットフォームの初期化
QUIT	QU	HEWの終了
RADIX	RA	入力ラディックス(基数)の設定
RESPONSE	RP	ウィンドウリフレッシュ間隔の設定
STATUS	STA	デバッグプラットフォームの状況表示
TOOL_INFORMATION	TO	現在登録されているツールの情報をファイルへ出力

## 5.1.16 プロジェクト/ワークスペース関連

コマンド名	短縮形	説明
ADD_FILE	AF	カレントプロジェクトへのファイル追加
CHANGE_CONFIGURATION	CC	コンフィギュレーションの設定
CHANGE_PROJECT	CP	プロジェクトの設定
CHANGE_SESSION	CS	セッションの設定
CHANGE_SUB_SESSION	CB	同期デバッグが有効な場合のアクティブセッションの変更
CLEAR_OUTPUT_WINDOW	COW	アウトプットウィンドウの各タブの表示内容クリア
CLOSE_WORKSPACE	CW	ワークスペースを閉じる
OPEN_WORKSPACE	OW	ワークスペースのオープン
REFRESH_SESSION	RSE	セッション情報の再読み込み
REMOVE_FILE	REM	カレントプロジェクトからのファイル削除
SAVE_SESSION	SE	現在のセッションをセーブ
SAVE_WORKSPACE	SW	ワークスペースをセーブ
UPDATE_ALL_DEPENDENCIES	UD	カレントプロジェクトのすべての依存関係更新

## 5.1.17 テスト支援機能関連

コマンド名	短縮形	説明
CLOSE_TEST_SUITE	CTS	テストスイートを閉じる
COMPARE_TEST_DATA	CTD	テストデータの比較
OPEN_TEST_SUITE	OTS	テストスイートを開く
RUN_TEST	RT	テストの実行

## 5.2 コマンド一覧 (アルファベット順)

アルファベット順コマンド一覧を示します。  
各コマンドのシンタックスは、ヘルプを参照ください。

表 5-1 コマンド一覧

項番	コマンド名	短縮形	説明
1	!	-	コメント
2	ADD_FILE	AF	カレントプロジェクトへのファイル追加
3	ANALYSIS	AN	性能分析機能の有効化 / 無効化
4	ANALYSIS_RANGE	AR	性能評価関数の設定、表示
5	ANALYSIS_RANGE_DELETE	AD	性能分析範囲の削除
6	ASSEMBLE	AS	アセンブルの実行
7	ASSERT	-	コンディションのチェック
8	AUTO_COMPLETE	AC	オートコンプリート機能の有効化 / 無効化
9	BREAKPOINT	BP	実行命令位置によるブレークポイントの設定
10	BREAK_ACCESS	BA	メモリ範囲のアクセスによるブレーク条件の設定
11	BREAK_CLEAR	BC	ブレークポイントの削除
12	BREAK_CYCLE	BCY	サイクルによるブレーク条件の設定
13	BREAK_DATA	BD	メモリのデータ値によるブレーク条件の設定
14	BREAK_DATA_DIFFERENCE	BDD	メモリのデータ値の変化量(差分)によるブレーク条件の設定
15	BREAK_DATA_INVERSE	BDI	メモリのデータ値の符号反転によるブレーク条件の設定
16	BREAK_DISPLAY	BI	ブレークポイント一覧の表示
17	BREAK_ENABLE	BE	ブレークポイントの有効/無効の切替
18	BREAK_REGISTER	BR	レジスタのデータ値によるブレーク条件の設定
19	BREAK_SEQUENCE	BS	実行順序を指定したブレークポイントの設定
20	BUILD	BU	カレントプロジェクトのビルド
21	BUILD_ALL	BL	カレントプロジェクトのすべてのビルド
22	BUILD_FILE	BF	ファイルのコンパイル
23	BUILD_MULTIPLE	BM	複数プロジェクトのビルド
24	CACHE	-	メモリキャッシュの有効化 / 無効化
25	CHANGE_CONFIGURATION	CC	コンフィギュレーションの設定
26	CHANGE_PROJECT	CP	プロジェクトの設定
27	CHANGE_SESSION	CS	セッションの設定
28	CHANGE_SUB_SESSION	CB	同期デバッグが有効な場合のアクティブセッションの変更
29	CLEAN	CL	ビルドの中間ファイルおよび出力ファイルの削除
30	CLEAR_OUTPUT_WINDOW	COW	アウトプットウィンドウの各タブの表示内容クリア
31	CLOSE_TEST_SUITE	CTS	テストスイートを閉じる
32	CLOSE_WORKSPACE	CW	ワークスペースを閉じる
33	COMPARE_TEST_DATA	CTD	テストデータの比較
34	COVERAGE	CV	カバレッジ測定の有効化 / 無効化
35	COVERAGE_DISPLAY	CVD	カバレッジ情報の表示
36	COVERAGE_LOAD	CVL	カバレッジ情報のロード
37	COVERAGE_RANGE	CVR	カバレッジ範囲の設定
38	COVERAGE_SAVE	CVS	カバレッジ情報のセーブ

## 5. コマンドライン

項番	コマンド名	短縮形	説明
39	DEFAULT_OBJECT_FORMAT	DO	デフォルトオブジェクト(プログラム)フォーマットの設定
40	DISASSEMBLE	DA	逆アセンブル表示
41	ERASE	ER	[コマンドライン]ウィンドウの内容のクリア
42	EVALUATE	EV	式の計算
43	EXEC_MODE	EM	実行モードの設定、表示
44	EXEC_STOP_SET	ESS	割込み発生時の実行モードの設定、表示
45	FILE_LOAD	FL	オブジェクト(プログラム)ファイルのロード
46	FILE_LOAD_ALL	LA	すべてのオブジェクト(プログラム)ファイルのロード
47	FILE_SAVE	FS	メモリ内容のファイルセーブ
48	FILE_UNLOAD	FU	オブジェクト(プログラム)ファイルのアンロード
49	FILE_UNLOAD_ALL	UA	すべてのオブジェクト(プログラム)ファイルのアンロード
50	FILE_VERIFY	FV	ファイル内容とメモリ内容の比較
51	GENERATE_MAKE_FILE	GM	カレントワークスペースの make ファイル作成
52	GO	GO	ユーザプログラムの実行
53	GO_RESET	GR	リセットベクタからのユーザプログラムの実行
54	GO_TILL	GT	テンポラリブレークポイントまでのユーザプログラムの実行
55	HALT	HA	ユーザプログラムの停止
56	HELP	HE	コマンドラインのヘルプ表示
57	INITIALIZE	IN	デバッグプラットフォームの初期化
58	LOG	LO	ロギングファイルの操作
59	MAP_DISPLAY	MA	メモリリソース設定の表示
60	MAP_SET	MS	メモリリソースの設定
61	MEMORY_COMPARE	MC	メモリ内容の比較
62	MEMORY_DISPLAY	MD	メモリ内容の表示
63	MEMORY_EDIT	ME	メモリ内容の変更
64	MEMORY_FILL	MF	指定データによるメモリ内容の一括変更
65	MEMORY_FIND	MI	メモリ内容の検索
66	MEMORY_MOVE	MV	メモリブロックの移動
67	MEMORY_TEST	MT	メモリブロックのテスト
68	OPEN_TEST_SUITE	OTS	テストスイートを開く
69	OPEN_WORKSPACE	OW	ワークスペースのオープン
70	PROFILE	PR	プロファイルの有効化/無効化
71	PROFILE_DISPLAY	PD	プロファイル情報の表示
72	PROFILE_SAVE	PS	プロファイル情報のファイル出力
73	QUIT	QU	HEW の終了
74	RADIX	RA	入力ラディックス(基数)の設定
75	REFRESH_SESSION	RSE	セッション情報の再読み込み
76	REGISTER_DISPLAY	RD	レジスタ値の表示
77	REGISTER_SET	RS	レジスタ値の設定
78	REMOVE_FILE	REM	カレントプロジェクトからのファイル削除
79	RESET	RE	CPU のリセット
80	RESPONSE	RP	ウィンドウリフレッシュ間隔の設定
81	RUN_TEST	RT	テストの実行
82	SLEEP	-	コマンド実行の遅延

## 5. コマンドライン

項番	コマンド名	短縮形	説明
83	SAVE_SESSION	SE	現在のセッションをセーブ
84	SAVE_WORKSPACE	SW	ワークスペースの保存
85	SET_DISASSEMBLY_SOFT_BREAK	SDB	逆アセンブリレベルのソフトウェアブレークポイントの設定または解除
86	SET_SOURCE_SOFT_BREAK	SSB	ソースレベルのソフトウェアブレークポイントの設定または解除
87	SIMULATEDIO_CLEAR	SIOC	I/O シミュレーションウィンドウの内容のクリア
88	STATE_DISASSEMBLY_SOFT_BREAK	TDB	逆アセンブリレベルのソフトウェアブレークポイントの有効化または無効化
89	STATE_SOURCE_SOFT_BREAK	TSB	ソースレベルのソフトウェアブレークポイントの有効化または無効化
90	STATUS	STA	デバッグプラットフォームの状況表示
91	STEP	ST	ステップ実行(命令単位またはソース行単位)
92	STEP_MODE	SM	ステップモードの設定
93	STEP_OUT	SP	PC 位置の関数を終了するまでのステップ実行
94	STEP_OVER	SO	ステップオーバー実行
95	STEP_RATE	SR	ステップ実行速度の設定、表示
96	SUBMIT	SU	コマンドファイルの実行
97	SYMBOL_ADD	SA	シンボルの設定
98	SYMBOL_CLEAR	SC	シンボルの削除
99	SYMBOL_LOAD	SL	シンボル情報ファイルのロード
100	SYMBOL_SAVE	SS	シンボル情報のファイルセーブ
101	SYMBOL_VIEW	SV	シンボルの表示
102	TCL	-	TCL の有効化 / 無効化
103	TOOL_INFORMATION	TO	現在登録されているツールの情報をファイルへ出力
104	TRACE	TR	トレース情報の表示
105	TRACE_ACQUISITION	TA	トレース情報取得の有効/無効の切替
106	TRACE_SAVE	TV	トレース情報をファイルへ出力
107	TRACE_STATISTIC	TST	統計情報解析の実行
108	TRAP_ADDRESS	TP	I/O シミュレーションアドレスの設定
109	TRAP_ADDRESS_DISPLAY	TD	I/O シミュレーションアドレス設定の表示
110	TRAP_ADDRESS_ENABLE	TE	I/O シミュレーションコール有効化 / 無効化
111	UPDATE_ALL_DEPENDENCIES	UD	カレントプロジェクトのすべての依存関係更新
112	WATCH_ADD	WA	Watch アイテムの追加
113	WATCH_AUTO_UPDATE	WU	Watch アイテムの自動更新の設定または解除
114	WATCH_DELETE	WD	Watch アイテムの削除
115	WATCH_DISPLAY	WI	ウォッチウィンドウの内容の表示
116	WATCH_EDIT	WE	Watch アイテムの値の編集
117	WATCH_EXPAND	WX	Watch アイテムの展開または縮小
118	WATCH_RADIX	WR	Watch アイテムの表示基数の変更
119	WATCH_RECORD	WO	Watch アイテムの値更新履歴をファイルに出力
120	WATCH_SAVE	WS	ウォッチウィンドウの表示内容をファイルに保存

---

## 6. メッセージ一覧

---

### 6.1 インフォメーションメッセージ

シミュレータ・デバッガは実行経過をユーザに知らせるため、インフォメーションメッセージを出力します。シミュレータ・デバッガの出力するインフォメーションメッセージを表 6-1に示します。

表 6-1 インフォメーションメッセージ一覧

メッセージ	内容
Break Access (Access Address:H'nnnnnn, Type:xxxx, Access Size:yyyy)	ブレイクアクセス条件が成立して実行を中断しました。付加情報として成立したブレイクアクセス条件(アクセスアドレス(Access Address)、アクセス種別(Type)、およびアクセスサイズ(Access Size))を表示します。
Break Cycle (Cycle:H'nnnnnnnn)	ブレイクサイクル条件が成立して実行を中断しました。付加情報として成立したブレイクサイクル条件(サイクル数(Cycle))を表示します。
Break Data (Access Address:H'nnnnnn, Data:H'mmmm)	ブレイクデータ条件(一致、または不一致)が成立して実行を中断しました。付加情報として成立したブレイクデータ条件(アクセスアドレス(Access Address)、および値(Data))を表示します。
Break Data (Access Address:H'nnnnnn, Previous Data:H'mmmm, Current Data:H'mmmm)	ブレイクデータ条件(符号反転、または差分)が成立して実行を中断しました。付加条件として成立したブレイクデータ条件(アクセスアドレス(Access Address)、前回値(Previous Data)、および今回値(Current Data))を表示します。
Break Register (Register:XX, Value:H'mmmm)	ブレイクレジスタ条件が成立して実行を中断しました。付加情報として成立したブレイクレジスタ条件(レジスタ名(Register)、および値(Value))を表示します。
Break Sequence (PC:H'nnnnnn)	ブレイクシーケンス条件が成立して実行を中断しました。付加情報として成立したブレイクシーケンス条件(最後の命令アドレス(PC))を表示します。
I/O DLL Stop	周辺機能が停止しました。
PC Breakpoint (PC:H'nnnnnn)	ブレイクポイント条件が成立して実行を中断しました。付加情報として成立したブレイクポイント条件(命令アドレス(PC))を表示します。
Sleep	SLEEP 命令により実行を中断しました。
Step Normal End	ステップ実行が正常に終了しました。
Stop	[STOP]ボタンにより実行を中断しました。
Trace Buffer Full	[トレース取得]ダイアログボックスの[トレースバッファ満杯時の動作]で[停止]モードを選択しており、かつトレースバッファが満杯となったので実行を中断しました。

## 6.2 エラーメッセージ

シミュレータ・デバッガはユーザプログラムや操作の誤りをユーザに知らせるため、エラーメッセージを出力します。シミュレータ・デバッガの出力するエラーメッセージを表 6-2に示します。

表 6-2 エラーメッセージ一覧

メッセージ	内容・対策	
Address Error	以下のいずれかの状態になりました。 (1) PC 値が奇数である (2) 内蔵 I/O 空間から命令読み出しを行おうとした (3) ワードデータを (2n) 番地以外からアクセスしようとした (4) ロングワードデータを (2n) 番地以外からアクセスしようとした エラーが発生しないようにユーザプログラムを修正してください。	
Exception Error	例外処理でエラーが発生しました。 エラーが発生しないようにユーザプログラムを修正してください。	
File Open Error	Break の FileInput/Output アクションでファイルオープンに失敗しました。 ファイル指定を見直してください。	
File Input Error	Break の File Input アクションでファイル読み込みに失敗しました。 ファイル指定を見直してください。	
File Output Error	Break の File Output アクションでファイル書出しに失敗しました。 ファイル指定を見直してください。	
Illegal Instruction	以下のいずれかの状態になりました。 (1) 命令ではないコードを実行しようとした (2) MOV.B Rn,@-SP または、MOV.B @SP+,Rn を実行しようとした エラーが発生しないようにユーザプログラムを修正してください。	
Illegal Operation	以下のいずれかの状態になりました。 (1) DAA 命令、DAS 命令で CCR の C フラグ、H フラグと補正前の値の関係が不正である (2) DIVXU 命令、DIVXS 命令でゼロ除算または、オーバフローが発生した エラーが発生しないようにユーザプログラムを修正してください。	
I/O area not exist	I/O 領域を削除しました。 I/O 領域は必ず設定してください。	
I/O DLL Illegal Interrupt Information (errNum=2xx)	割り込み情報が不正です。詳細情報を[errNum]に示します。割り込み情報を修正してください。	
	errNum	説明
	200	指定したベクタが範囲外です
201	指定した優先度が範囲外です	
I/O DLL Memory Access Error (errNum=0xx, Address=0XXXXXXXXX)	周辺機能へのメモリアクセスでエラーが発生しました。詳細内容を[errNum]、エラー発生アドレスを[Address]に示します。エラー情報に従ってプログラムを修正してください。	
	errNum	説明
	001	指定したアドレスが範囲外です
	002	指定した領域にメモリがありません
	003	必要なメモリが確保できません
	004	指定したデータサイズが範囲外です
005	指定したアドレスにアクセスできません	



メッセージ	内容・対策	
I/O DLL Register Access Error (errNum=1xx, RegisterName=xxxx)	周辺機能へのレジスタアクセスでエラーが発生しました。詳細内容を [errNum]、エラー発生レジスタを[RegisterName]に示します。エラー情報に従ってプログラムを修正してください。	
	errNum	説明
	100	レジスタの記述に誤りがあります
	101	指定したデータ値が不正です
Memory Access Error (Address:H'nnnnnn)	以下のいずれかの状態になりました。付加情報としてエラーの発生したアドレス(Address)を表示します。 (1) 確保していないメモリ領域をアクセスしようとした (2) 書き込み不可属性を持つメモリへの書き込みを行おうとした (3) 読み出し不可属性を持つメモリからの読み出しを行おうとした (4) メモリが存在しない領域をアクセスしようとした (5) EEPMOV 命令以外で EEPROM へ書き込みを行おうとした メモリの確保、属性変更を行うか、当該メモリアccessが発生しないようにユーザプログラムを修正してください。	
System Call Error	I/O シミュレーションエラーが発生しました。 レジスタR0,R1およびパラメータブロックの内容の誤りを修正してください。	
The memory resource has not been set up	メモリマップ範囲外にメモリリソースを設定しました。 エラーが発生しないようにメモリリソース設定を修正してください。	

## 6. メッセージ一覧

---

---

## 7. チュートリアル

---

### 7.1 デバッグの準備

シミュレータ・デバッガの主な特徴をサンプルプログラムを用いて説明します。

#### 注意

本章の使用例(図)の内容は、ご使用になるコンパイラのバージョンによって変わってくる場合がありますのでご了承ください。

#### 7.1.1 サンプルプログラム

サンプルプログラムは、10個のランダムデータを昇順にソートした後、降順にソートするCプログラムに基づいています。

サンプルプログラムでは、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。
- printf関数を用いて、ランダムデータ、ソートしたデータを表示します。

サンプルプログラムは、HEWのデモンストレーションプログラムを用います。

#### 7.1.2 サンプルプログラムの作成

下記に注意してHEWのデモンストレーションプログラムを作成してください。

- 「新規ワークスペースを作成する」では[Project Type]に”Demonstration”を指定してください。
- [CPUシリーズ:]は”2600”を、[ターゲット:]は”H8S/2600N Simulator”を選択してください。
- コンフィギュレーションは、ツールバーより”SimDebug\_H8S-2600N”をビルド前に選択してください。
- セッションは、ツールバーより”SimSessionH8S-2600N”を選択してください。
- 本デモンストレーションプログラムでは周辺機能(タイマ)は利用しないので、セッション切換え時に表示される[周辺機能シミュレーションの設定]ダイアログボックスは[このダイアログを表示しない]チェックボックスチェック後に[OK]ボタンをクリックしてください。

デバッグ機能の説明のため、Demonstrationは最適化なしの設定になっています。最適化の設定は変

更しないください。

## 7.2 デバッグのための設定

### 7.2.1 メモリリソースの確保

開発しているアプリケーションを動作させるためにメモリリソースの確保が必要です。デモンストラーションプロジェクトでは、自動的にメモリリソースを確保しますので、設定を確認してください。

- [基本設定]メニューから [シミュレータ -> メモリリソース...]を選択し、現在のメモリリソースを表示してください。

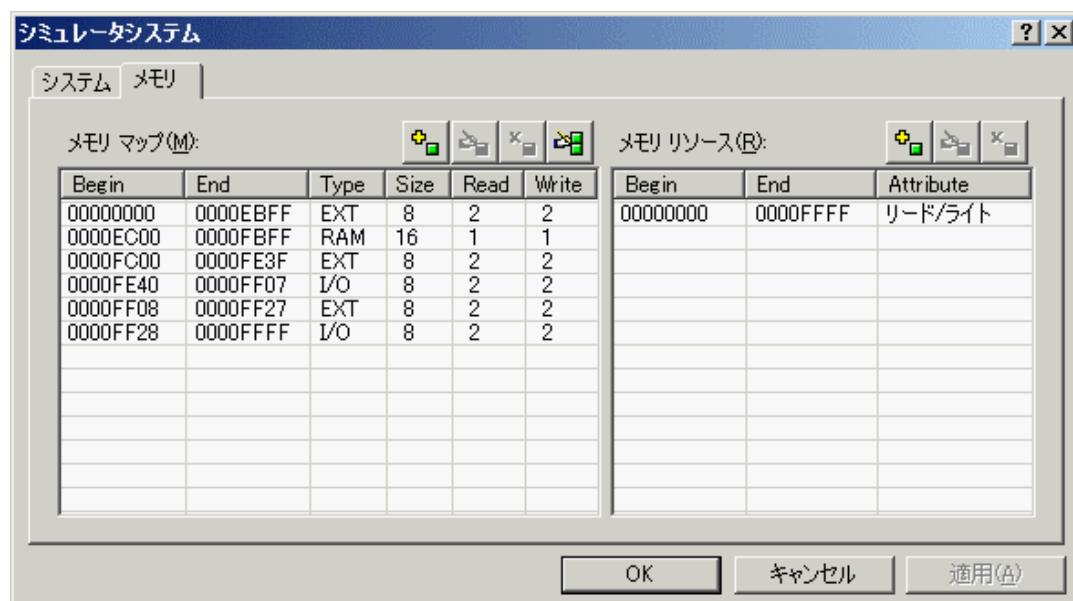


図 7-1 シミュレータシステムダイアログボックス (メモリページ)

プログラム領域およびスタック領域として H'00000000 から H'0000FFFF を読み出し/書き込み可能領域として確保しています。

- [OK]ボタンをクリックしてダイアログボックスを閉じてください。

メモリリソースは、[H8S,H8/300 Standard Toolchain]ダイアログボックスの[デバッガ]ページでも参照/変更ができます。おたがいの変更は反映されます。

## 7.2.2 サンプルプログラムのダウンロード

デモンストレーションプロジェクトでは、ダウンロードするサンプルプログラムを自動的に設定しますので、設定を確認してください。

- [デバッグ]メニューから [デバッグの設定...]を選択して、[デバッグの設定]ダイアログボックスを開いてください。

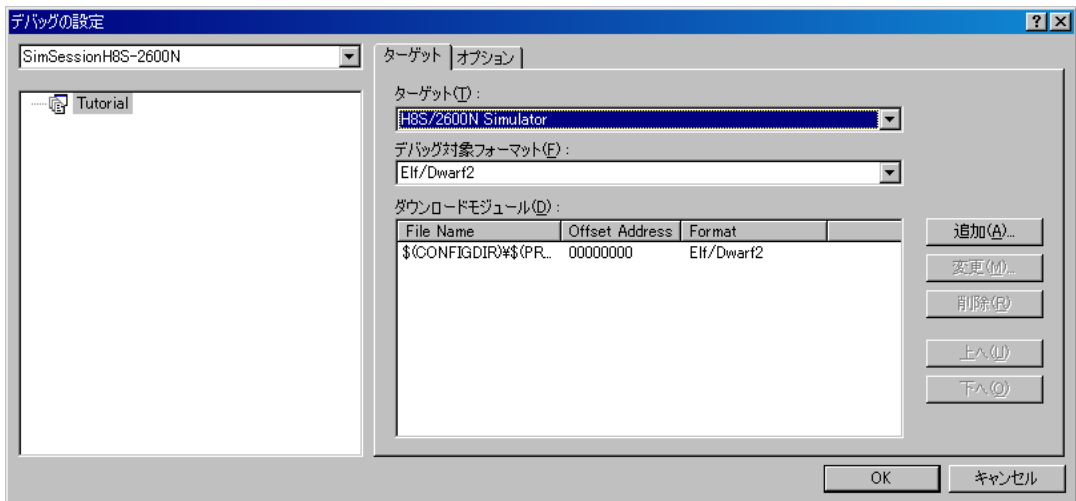


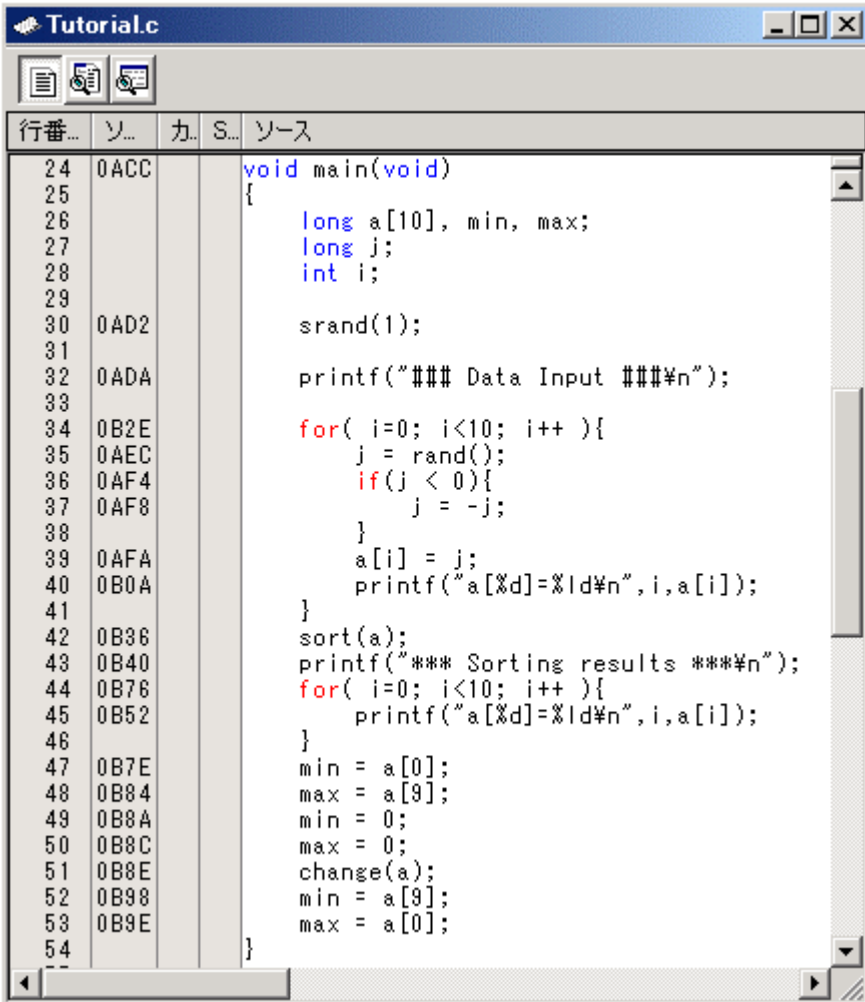
図 7-2 デバッグの設定ダイアログボックス

- [ダウンロードモジュール]に設定しているファイルがダウンロードするファイルです。
- [OK]ボタンをクリックして[デバッグの設定]ダイアログボックスを閉じてください。
- [デバッグ]メニューから [ダウンロード -> All Download Modules]を選択して、サンプルプログラムをダウンロードしてください。

## 7.2.3 ソースプログラムの表示

HEW では、ソースレベルでプログラムをデバッグできます。ソースプログラム ("Tutorial.c") を表示してください。

- [ワークスペース]ウィンドウの Tutorial.c をダブルクリックして[ソース]ウィンドウを開いてください。



```
void main(void)
{
    long a[10], min, max;
    long j;
    int i;

    srand(1);

    printf("### Data Input ###\n");

    for( i=0; i<10; i++ ){
        j = rand();
        if(j < 0){
            j = -j;
        }
        a[i] = j;
        printf("a[%d]=%ld\n",i,a[i]);
    }
    sort(a);
    printf("### Sorting results ###\n");
    for( i=0; i<10; i++ ){
        printf("a[%d]=%ld\n",i,a[i]);
    }
    min = a[0];
    max = a[9];
    min = 0;
    max = 0;
    change(a);
    min = a[9];
    max = a[0];
}
```

図 7-3 ソースウィンドウ (ソースプログラムの表示)

## 7.2.4 PCブレークポイントの設定

[ソース]ウィンドウによって、ブレークポイントを簡単に設定できます。以下のようにして sort 関数のコール箇所にブレークポイントを設定します。

- sort 関数コールを含む行にカーソルを移動して右クリックし、ポップアップメニューの [ブレークポイントの挿入/削除] を選択してください。

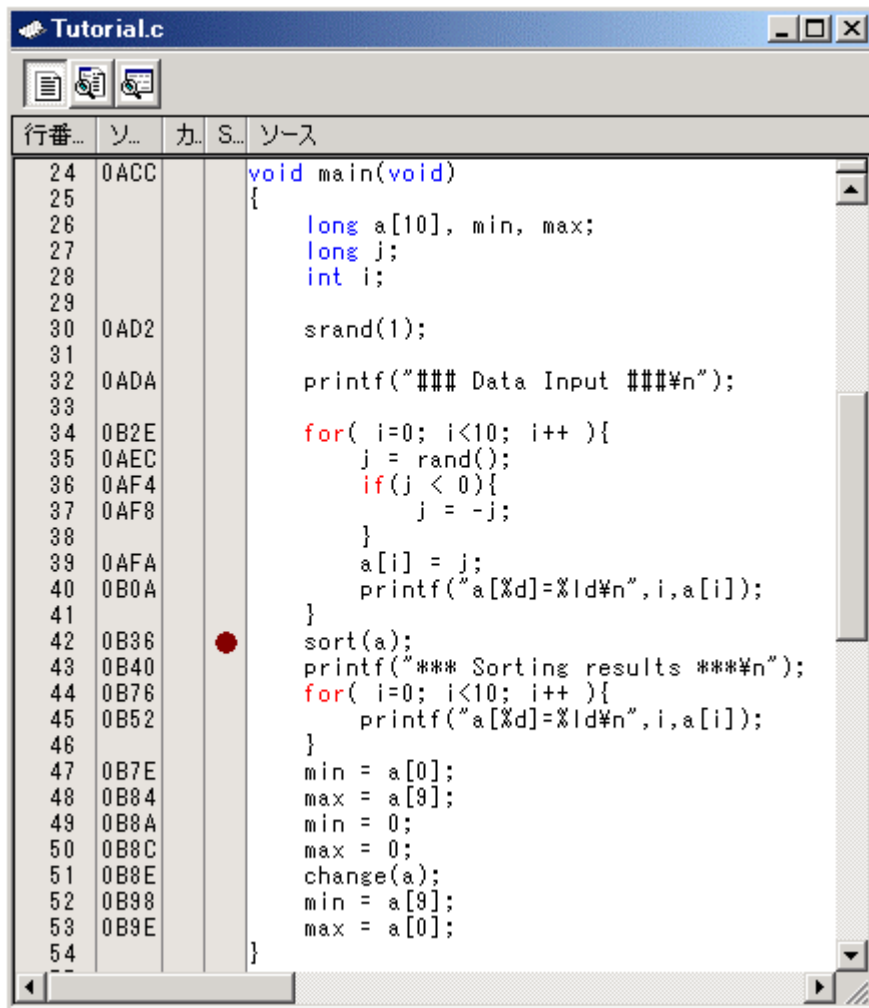


図 7-4 ソースウィンドウ (ブレークポイントの設定)

sort 関数コールを含む行に `sort(a);` を表示し、そのアドレスに PC ブレークポイントを設定したことを示します。

## 7.2.5 プロファイラの設定

- [表示]メニューから [パフォーマンス -> プロファイル]を選択し、[プロファイル]ウィンドウを開いてください。

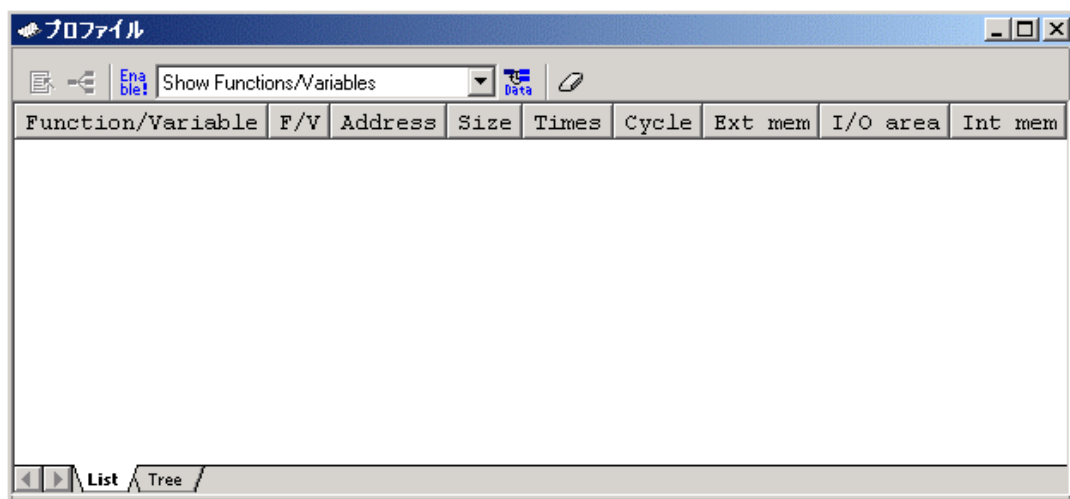


図 7-5 プロファイルウィンドウ

- [プロファイル]ウィンドウ上で右クリックしてポップアップメニューを表示し、[有効]を選択してプロファイラ情報取得を有効にしてください。



## 7.2.6 I/Oシミュレーションの設定

デモンストレーションプロジェクトでは、自動的にI/Oシミュレーションを設定しますので、設定を確認してください。

- [基本設定]メニューから [シミュレータ -> システム]を選択して、[シミュレータシステム]ダイアログボックスを開いてください。

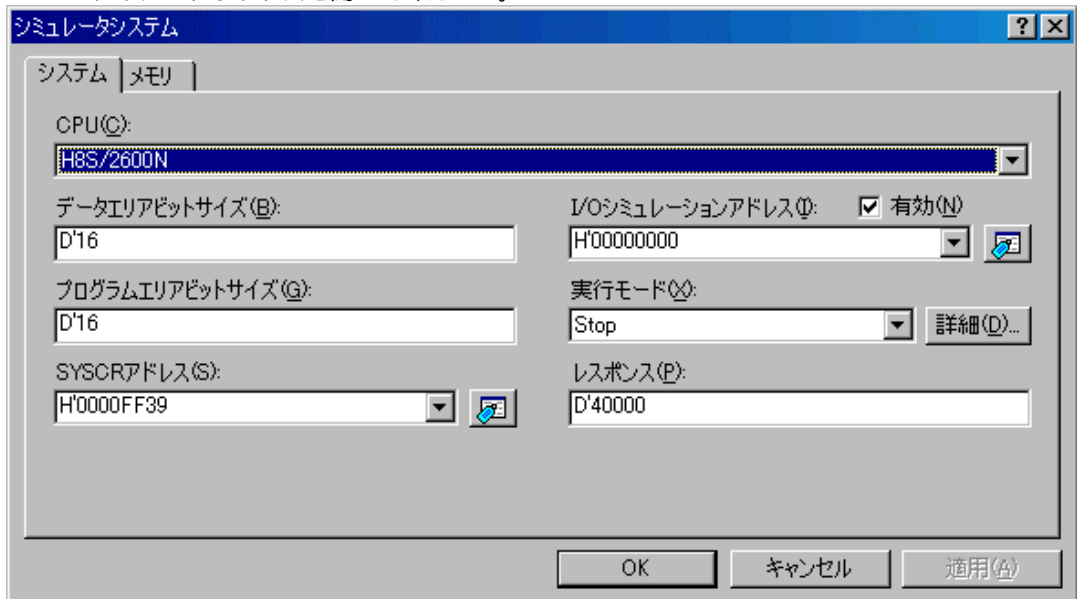


図 7-6 シミュレータシステムダイアログボックス (システムページ)

- [I/O シミュレーションアドレス]の [有効]にチェックがあることを確認してください。
- [OK]ボタンをクリックして I/O シミュレーションを有効にしてください。
- [表示]メニューから [CPU -> I/O シミュレーション]を選択し、[I/O シミュレーション]ウィンドウを開いてください。[I/O シミュレーション]ウィンドウを開かなければ、I/O シミュレーションが有効になりません。

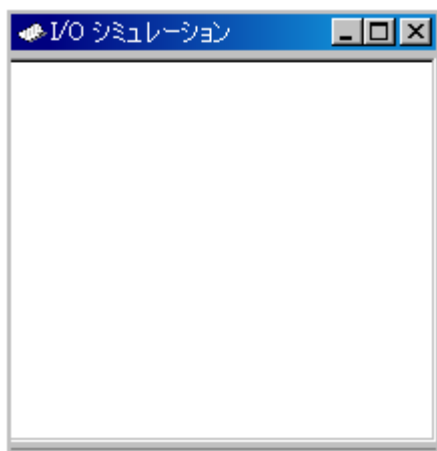


図 7-7 I/O シミュレーションウィンドウ

### 7.2.7 トレース情報取得条件の設定

- [表示]メニューから [コード -> トレース]を選択し、[トレース]ウィンドウを開いてください。さらに、[トレース]ウィンドウ上で右クリックしてポップアップメニューを表示し、[設定...]を選択してください。

以下の[トレース取得]ダイアログボックスを表示します。

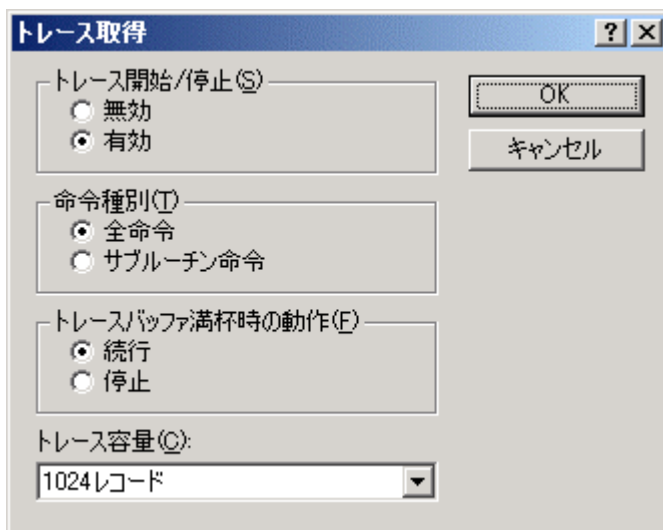


図 7-8 トレース取得ダイアログボックス

- [トレース取得]ダイアログボックスの [トレース開始 / 停止]を [有効]に設定し、[OK]ボタンをクリックしてトレース情報取得を有効にしてください。

## 7.2.8 スタックポインタ、プログラムカウンタの設定

プログラムを実行するために、プログラムカウンタをリセットベクタから設定してください。サンプルプログラムのリセットベクタには PC 値として H'400 が書いてあります。

- [デバッグ]メニューから [CPU のリセット]を選択するか、またはツールバー上の[CPU リセット]ボタンをクリックしてください。

リセットベクタからプログラムカウンタに H'400 を設定します。



図 7-9 CPU リセットボタン

## 7.3 デバッグ開始

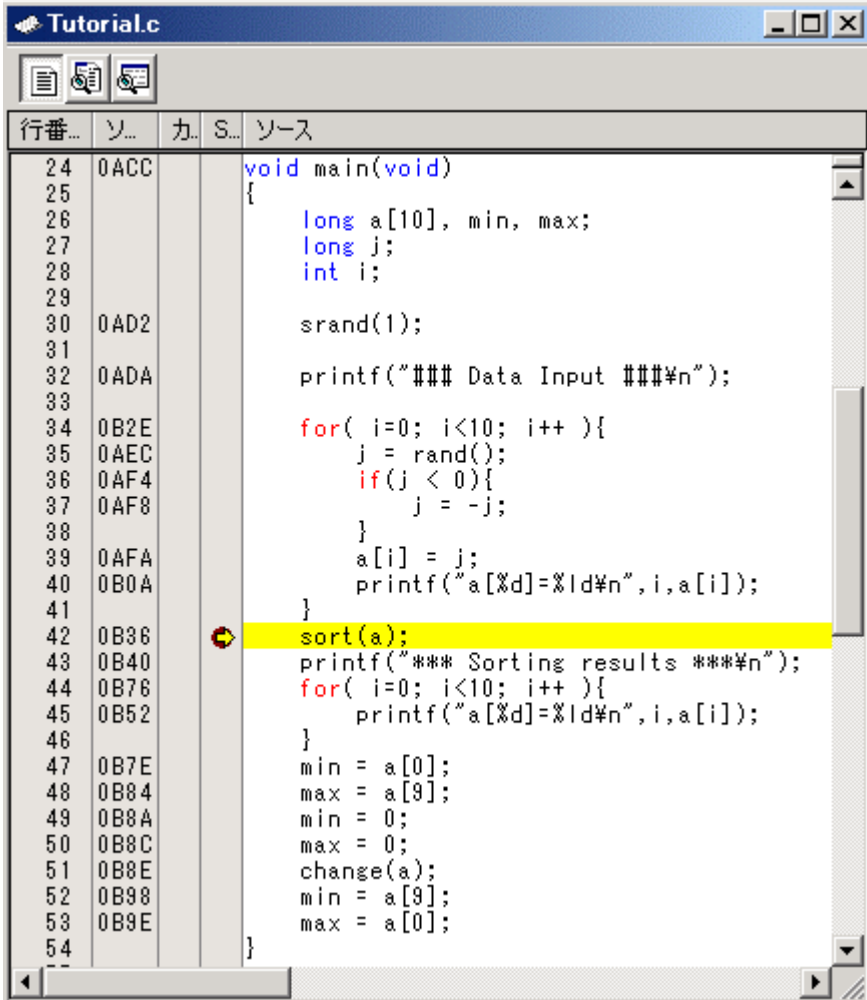
### 7.3.1 プログラムの実行

- プログラムを実行するには、[デバッグ]メニューから [実行]を選択するか、またはツールバー上の[実行]ボタンをクリックしてください。



図 7-10 実行ボタン

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します。プログラムが停止した位置を示すために [ソース]ウィンドウに を表示します。また 停止要因として[アウトプット]ウィンドウに”PC Breakpoint (PC:H'0B36)”を表示します。



The screenshot shows a code editor window titled "Tutorial.c". The window contains C code for a program that generates random numbers and sorts them. The code is displayed in a table-like format with columns for line number, source code, and assembly code. The line number column ranges from 24 to 54. The source code column contains the following code:

```
void main(void)
{
    long a[10], min, max;
    long j;
    int i;

    srand(1);

    printf("### Data Input ###\n");

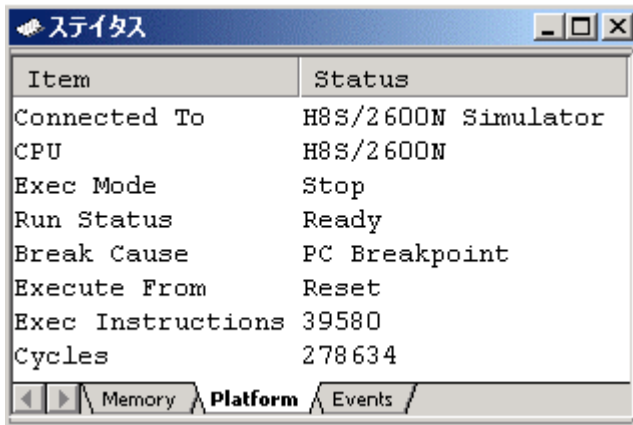
    for( i=0; i<10; i++ ){
        j = rand();
        if(j < 0){
            j = -j;
        }
        a[i] = j;
        printf("a[%d]=%ld\n",i,a[i]);
    }
    sort(a);
    printf("### Sorting results ###\n");
    for( i=0; i<10; i++ ){
        printf("a[%d]=%ld\n",i,a[i]);
    }
    min = a[0];
    max = a[9];
    min = 0;
    max = 0;
    change(a);
    min = a[9];
    max = a[0];
}
```

The line `sort(a);` at line 42 is highlighted in yellow. A mouse cursor is visible over the line. The assembly code column contains hexadecimal values corresponding to the source code lines.

図 7-11 ソースウィンドウ (ブレーク状態)

[ステータス]ウィンドウで停止要因が確認できます。

- [表示]メニューから [CPU -> ステータス]を選択し、[ステータス]ウィンドウを開いてください。さらに、[ステータス]ウィンドウのうち [Platform]シートを表示してください。



The screenshot shows a window titled 'ステータス' (Status) with a table of system information. The table has two columns: 'Item' and 'Status'. Below the table are three tabs: 'Memory', 'Platform', and 'Events'. The 'Platform' tab is currently selected.

Item	Status
Connected To	H8S/2600N Simulator
CPU	H8S/2600N
Exec Mode	Stop
Run Status	Ready
Break Cause	PC Breakpoint
Execute From	Reset
Exec Instructions	39580
Cycles	278634

図 7-12 ステータスウィンドウ

これは、以下の実行内容を表示します。

ブレークの原因は PCブレーク

リセットからの実行

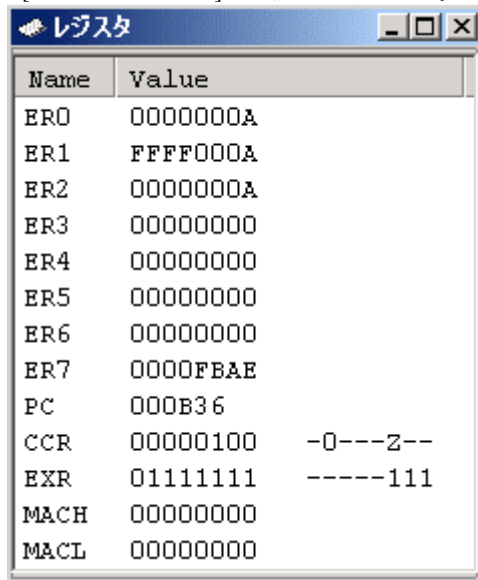
GOコマンドでの実行命令数は39,580命令

パイプラインリセットからの実行サイクル数は278,634サイクル

## 7. チュートリアル

[レジスタ]ウィンドウでレジスタの値が確認できます。

- [表示]メニューから [CPU -> レジスタ]を選択してください。



Name	Value
ER0	0000000A
ER1	FFFF000A
ER2	0000000A
ER3	00000000
ER4	00000000
ER5	00000000
ER6	00000000
ER7	0000FBAE
PC	000B36
CCR	00000100 -0---Z--
EXR	01111111 -----111
MACH	00000000
MACL	00000000

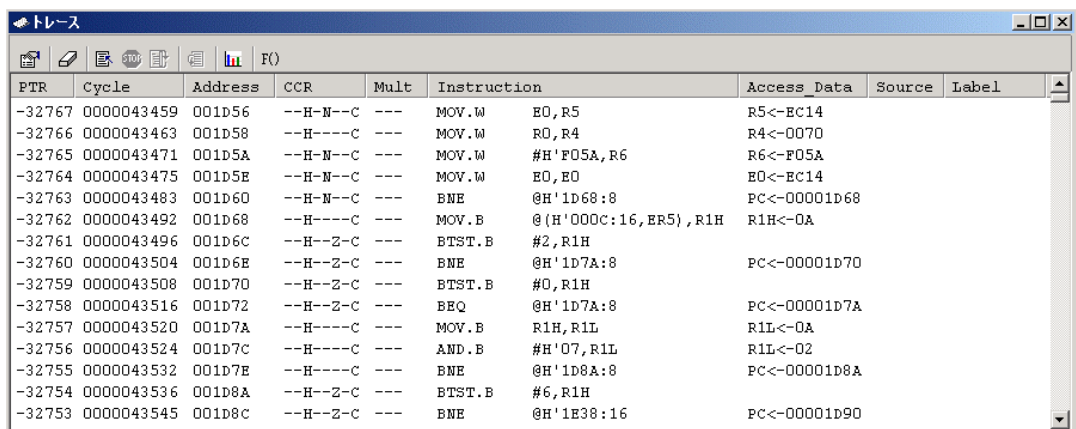
図 7-13 レジスタウィンドウ

プログラム停止時の各レジスタの値を確認することができます。

### 7.3.2 トレースバッファの使い方

トレースバッファを使って、命令実行の履歴を知ることができます。

- [表示]メニューから [コード -> トレース]を選択し、[トレース]ウィンドウを開いてください。ウィンドウの最上部までスクロールアップしてください。



PTR	Cycle	Address	CCR	Mult	Instruction	Access Data	Source	Label
-32767	0000043459	001D56	--H-N--C	---	MOV.W E0,R5	R5<-EC14		
-32766	0000043463	001D58	--H----C	---	MOV.W R0,R4	R4<-0070		
-32765	0000043471	001D5A	--H-N--C	---	MOV.W #H'F05A,R6	R6<-F05A		
-32764	0000043475	001D5E	--H-N--C	---	MOV.W E0,E0	E0<-EC14		
-32763	0000043483	001D60	--H-N--C	---	BNE @H'1D68:8	PC<-00001D68		
-32762	0000043492	001D68	--H----C	---	MOV.B @(H'000C:16,ER5),R1H	R1H<-0A		
-32761	0000043496	001D6C	--H--Z-C	---	BTST.B #2,R1H			
-32760	0000043504	001D6E	--H--Z-C	---	BNE @H'1D7A:8	PC<-00001D70		
-32759	0000043508	001D70	--H--Z-C	---	BTST.B #0,R1H			
-32758	0000043516	001D72	--H--Z-C	---	BEQ @H'1D7A:8	PC<-00001D7A		
-32757	0000043520	001D7A	--H----C	---	MOV.B R1H,R1L	R1L<-0A		
-32756	0000043524	001D7C	--H----C	---	AND.B #H'07,R1L	R1L<-02		
-32755	0000043532	001D7E	--H----C	---	BNE @H'1D8A:8	PC<-00001D8A		
-32754	0000043536	001D8A	--H--Z-C	---	BTST.B #6,R1H			
-32753	0000043545	001D8C	--H--Z-C	---	BNE @H'1E38:16	PC<-00001D90		

図 7-14 トレースウィンドウ (トレース情報の表示)

### 7.3.3 トレース検索の実行

最初に、[トレース]ウィンドウ上で右クリックしてポップアップメニューを表示し、[トレース検索...]を選択して、[トレース検索]ダイアログボックスを開いてください。

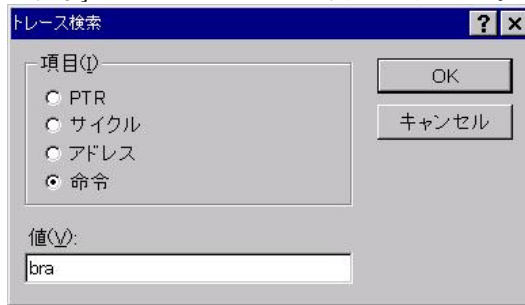


図 7-15 トレース検索ダイアログボックス

検索項目 [項目]と検索内容 [値]を設定して [OK]ボタンをクリックすると、トレース検索を実行します。該当するトレース情報があった場合、該当する最初の行を強調表示します。同じ検索内容 [値]でトレース検索を続ける場合は、[トレース]ウィンドウ上で右クリックしてポップアップメニューを表示し、[次を検索]を選択してください。次に該当する行を強調表示します。

PTR	Cycle	Address	CCR	Mult	Instruction	Access_Data	Source	Label
-32739	0000043649	00092E	----N---	---	SUB.W #H'0004,R7	R7<-FA8E		
-32738	0000043653	000932	----Z--	---	MOV.W R1,E2	E2<-0000	...	
-32737	0000043657	000934	----N---	---	MOV.W E0,R2	R2<-EC35		
-32736	0000043661	000936	-----	---	MOV.W R0,E1	E1<-0001		
-32735	0000043669	000938	----N---	---	MOV.W #H'EC37,R0	R0<-EC37		
-32734	0000043673	00093C	----N---	---	ADD.W E1,R0	R0<-EC38		
-32733	0000043678	00093E	-----	---	MOV.B @E00,R1L	R1L<-1A		
-32732	0000043682	000940	-----	---	AND.B #H'06,R1L	R1L<-02		
-32731	0000043686	000942	-----	---	MOV.B R1L,R1L	R1L<-02		
-32730	0000043694	000944	-----	---	BEQ @H'0960:8	Pc<-00000946		
-32729	0000043698	000946	----Z--	---	MOV.W E2,R3	R3<-0000		
-32728	0000043702	000948	----Z--	---	MOV.W R3,R3	R3<-0000		
-32727	0000043710	00094A	----Z--	---	BEQ @H'095C:8	Pc<-0000095C		
-32726	0000043714	00095C	----Z--	---	MOV.W E2,R0	R0<-0000	...	
-32725	0000043722	00095E	----Z--	---	BRA @H'0964:8	Pc<-00000964		

図 7-16 トレースウィンドウ (検索結果)

### 7.3.4 I/Oシミュレーションの確認

printf 関数で表示したランダムデータを[I/O シミュレーション]ウィンドウで確認することができます。

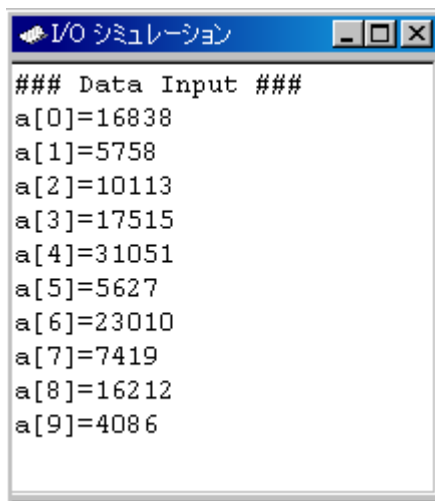


図 7-17 I/O シミュレーションウィンドウ

- [I/O シミュレーション]ウィンドウは閉じないでください。

### 7.3.5 ブレークポイントの確認

プログラムに設定した全てのブレークポイントのリストを[イベントポイント]ウィンドウで確認することができます。

- [表示]メニューから [コード -> イベントポイント]を選択してください。

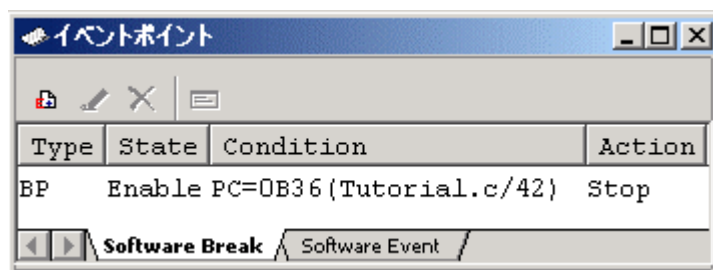


図 7-18 イベントポイントウィンドウ

[イベントポイント]ウィンドウによって、ブレークポイントの設定、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ができます。

[イベントポイント]ウィンドウを閉じてください。



### 7.3.6 変数の参照

プログラムで使用する変数の値を[ウォッチ]ウィンドウで確認することができます。たとえば、プログラムの始めに宣言した long 型の配列 a を見る場合：

- [表示]メニューから [シンボル -> ウォッチ]を選択し[ウォッチ]ウィンドウを表示してください。さらに、[ウォッチ]ウィンドウ上で右クリックしてポップアップメニューを表示し、[シンボル登録...]を選択してください。

以下の [シンボル登録]ダイアログボックスを表示します。



図 7-19 シンボル登録ダイアログボックス

- 配列 “a” をタイプし、[OK]ボタンをクリックします。

[ウォッチ]ウィンドウに、long 型の配列 a を表示します。

配列 a の前にあるシンボル+をクリックすると、配列を拡張して表示します。

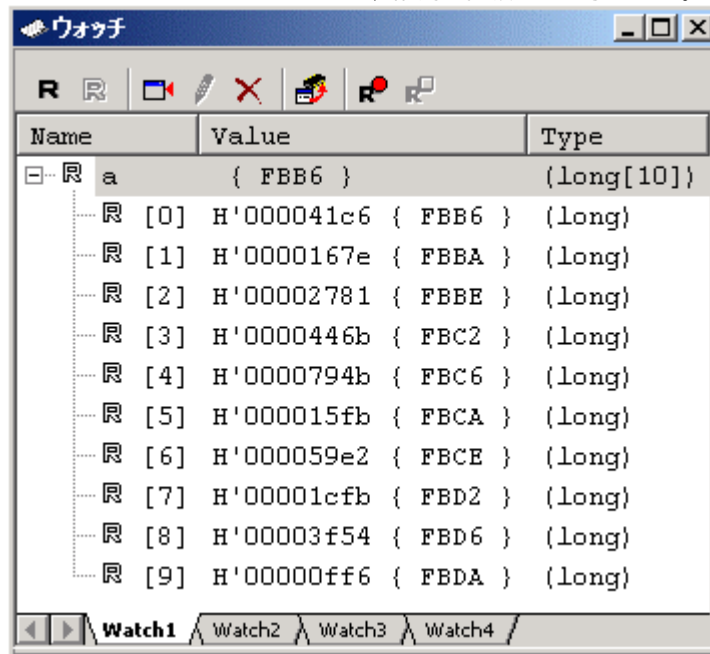


図 7-20 ウォッチウィンドウ

- [ウォッチ]ウィンドウを閉じてください。

## 7.3.7 プログラムのステップ実行

シミュレータ・デバッガは、プログラムのデバッグに有効な各種のステップメニューを備えています。

メニュー	説明
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）。
ステップオーバー	関数コールを1ステップとして、ステップ実行します。
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における次のステートメントで停止します。
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。

## (1) [ステップイン]の実行

[ステップイン]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 7-21 ステップインボタン

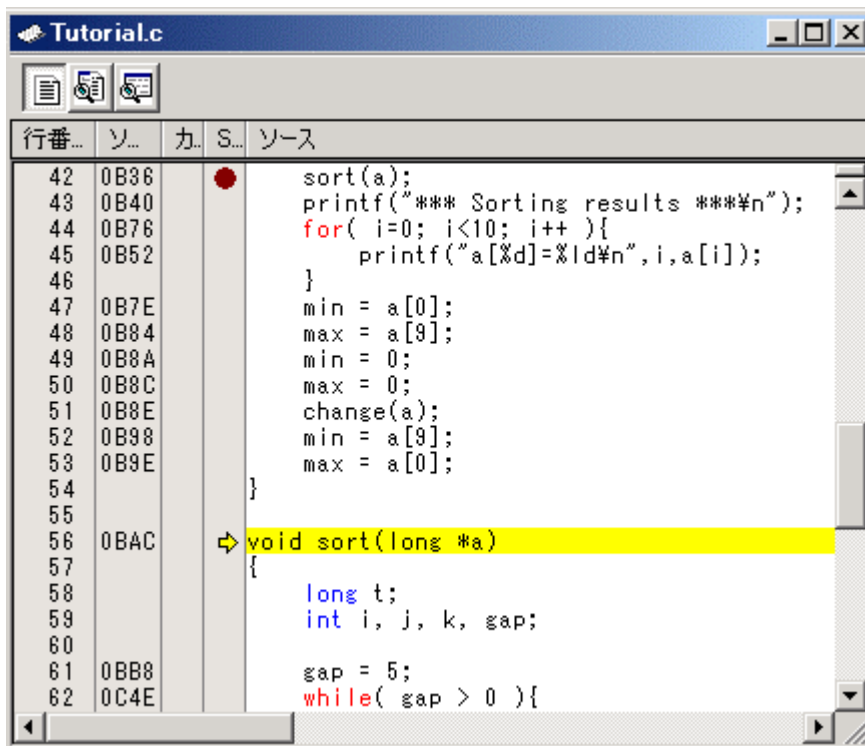


図 7-22 ソースウィンドウ (ステップイン)

- [ソース]ウィンドウの PC 位置表示( )が、sort 関数の先頭のステートメントに移動します。
- (2) [ステップアウト]の実行

[ステップアウト]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ステートメントで停止します。

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



図 7-23 ステップアウトボタン

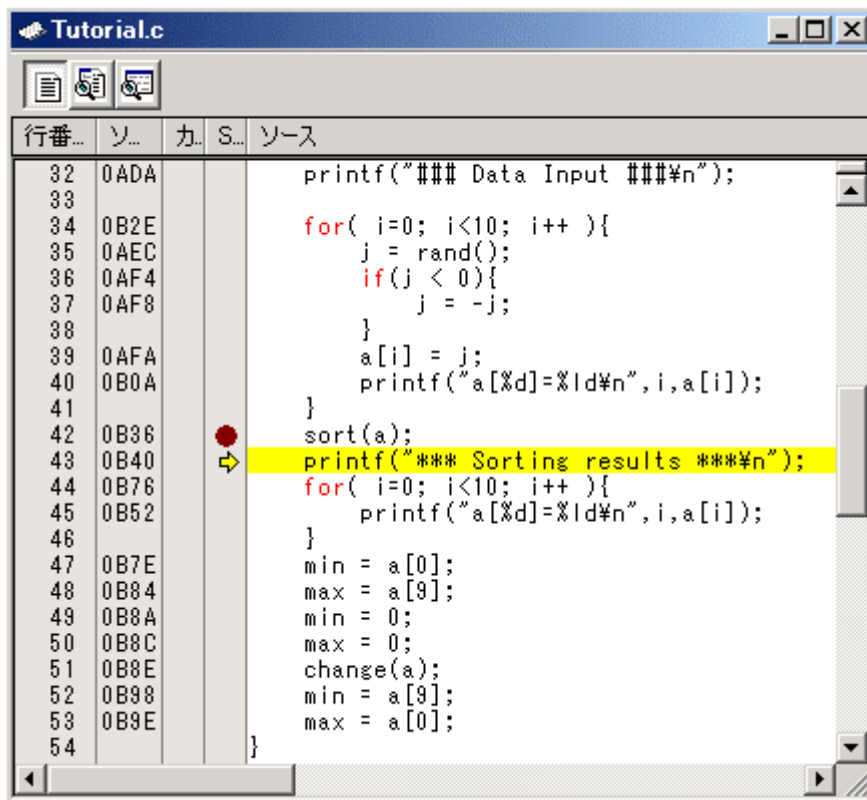


図 7-24 ソースウィンドウ (ステップアウト)

## (3) [ステップオーバー]の実行

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

Printf 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 7-25 ステップオーバーボタン

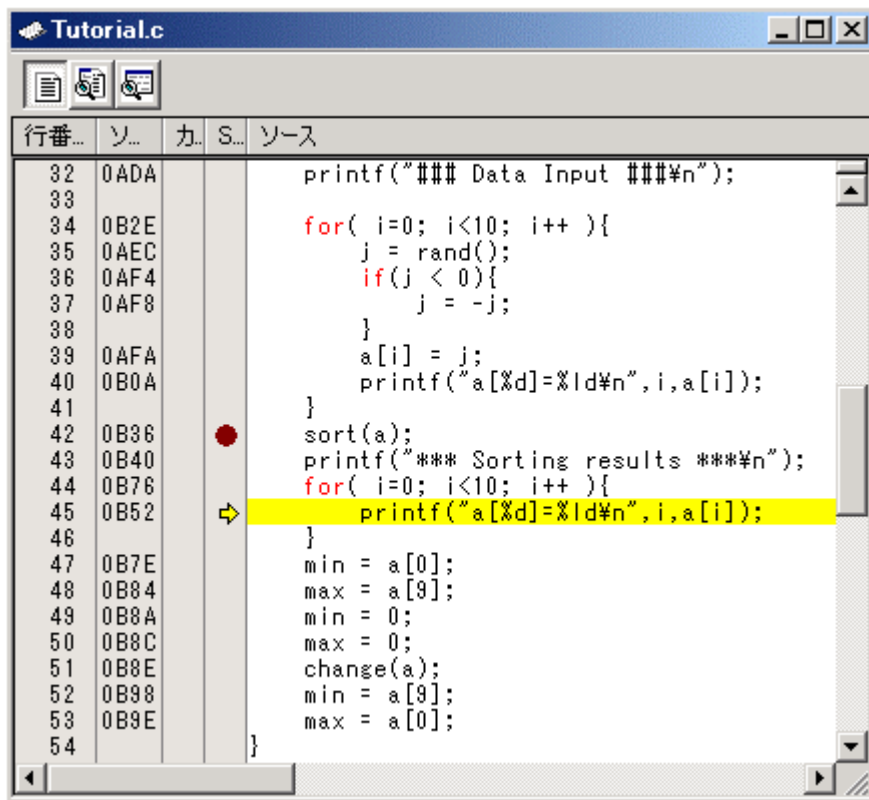


図 7-26 ソースウィンドウ (ステップオーバー)

printf 関数を実行すると、[I/O シミュレーション]ウィンドウに\*\*\* Sorting results \*\*\*を表示します。

### 7.3.8 プロファイラ情報の確認

プロファイラ情報を[プロファイル]ウィンドウで確認することができます。

- [実行]ボタンをクリックして現在の PC から継続実行すると、SLEEP 命令を実行して停止します。

#### (1) [List]シート

プロファイラ情報をリスト形式で表示します。

- [表示]メニューから [パフォーマンス -> プロファイル]を選択し、[プロファイル]ウィンドウを開いてください。[List]シートを表示します。

Function/Variable	F/V	Address	Size	Times	Cycle	Ext mem	I/O area	Int mem
__ctype	V	2138	H'00000001	1	0	0	0	0
__nfiles	V	20D6	H'00000002	8	0	0	0	0
__alocbuf	F	1F78	H'0000008c	1	192	0	0	13
__strlen	F	1EE6	H'00000014	40	4840	0	0	192
__flshbuf	F	1D52	H'000000F0	255	123582	0	0	8408
\$DIVUL\$3	F	1D16	H'00000000	230	27140	0	0	690
__fputc	F	1C40	H'000000D6	255	73159	0	0	3824
__flopen	F	1B18	H'00000128	3	1656	15	0	51
__fclose	F	1A76	H'000000A2	6	1107	0	0	69
1A5C	F	1A5C	H'00000000	124	7192	0	0	496
1A46	F	1A46	H'00000000	22	968	0	0	88
19D6	F	19D6	H'00000000	102	40790	143	0	2203
1742	F	1742	H'00000000	40	48080	0	0	2488
__fmtout	F	0E80	H'0000008c2	22	78593	771	0	3577
\$MULL\$3	F	0E68	H'00000000	10	730	0	0	50

図 7-27 List シート (プロファイルウィンドウ)

\_\_fclose 関数を 6 回コールし、実行サイクルは 1107 サイクル、内部メモリを 69 回アクセスしたことがわかります。

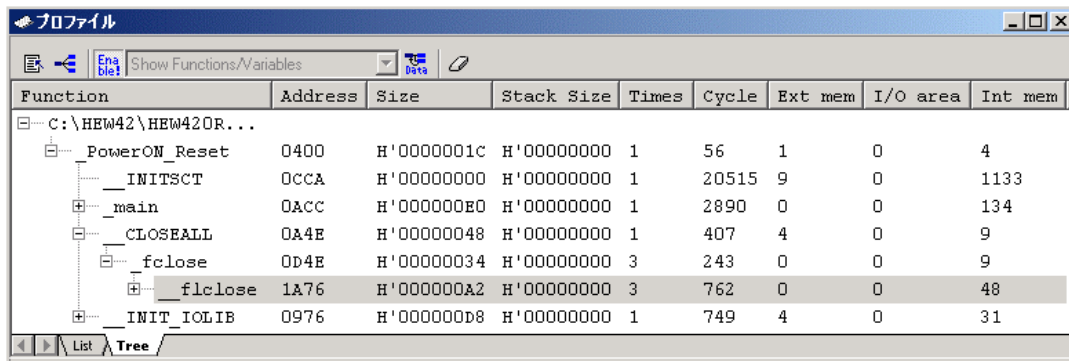
コール回数が多い関数や遅いメモリを多くアクセスする関数など、プログラム性能のクリティカルパスを探することができます。

## 7. チュートリアル

### (2) [Tree]シート

プロファイラ情報をツリー形式で表示します。

- [Tree]シートを選択してください。関数名をダブルクリックすることによりツリー構造を拡張または縮小します。



Function	Address	Size	Stack Size	Times	Cycle	Ext mem	I/O area	Int mem
PowerON_Reset	0400	H'0000001C	H'00000000	1	56	1	0	4
__INITSCT	0CCA	H'00000000	H'00000000	1	20515	9	0	1133
main	0ACC	H'000000E0	H'00000000	1	2890	0	0	134
CLOSEALL	0A4E	H'00000048	H'00000000	1	407	4	0	9
fclose	0D4E	H'00000034	H'00000000	3	243	0	0	9
fclose	1A76	H'000000A2	H'00000000	3	762	0	0	48
INIT_IOLIB	0976	H'000000D8	H'00000000	1	749	4	0	31

図 7-28 Tree シート (プロファイルウィンドウ)

\_\_fclose 関数が fclose 関数から 3 回コールされ、その時の実行サイクルは 762 サイクル、内部メモリを 48 回アクセスしたことがわかります。

## (3) 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]ウィンドウで関数の呼び出し関係を表示します。

- [プロファイル]ウィンドウ上で `_fclose` 関数を選択してから、右クリックしてポップアップメニューを表示し、 [チャート表示]を選択して[プロファイル - チャート]ウィンドウを表示してください。

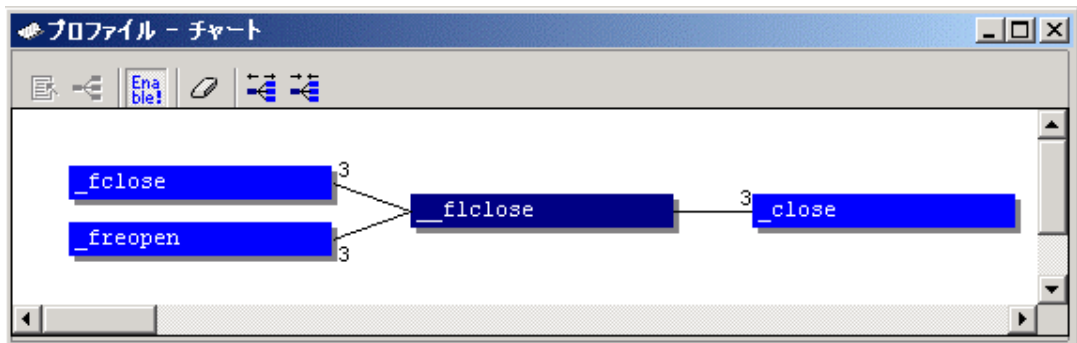


図 7-29 プロファイル - チャートウィンドウ

`_fclose` 関数が `_fclose` 関数から 3 回、 `_freopen` 関数から 3 回コールされたことがわかります。また、 `_close` 関数を 3 回コールしたことがわかります。

以上で、シミュレータ・デバッガを使用したチュートリアルは終了です。





---

ルネサスマイクロコンピュータ開発環境システム  
ユーザーズマニュアル  
H8S、H8/300 シリーズ シミュレータ/デバッガ

発行年月日 2007年7月25日 Rev.4.00

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 (名古屋広小路ブレイス)	(052) 249-3330
関	西	支	社	〒541-0044	大阪府中央区伏見町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)



H8S、H8/300 シリーズ シミュレータ／デバッガ  
ユーザーズマニュアル



ルネサスエレクトロニクス株式会社  
神奈川県川崎市中原区下沼部1753 〒211-8668

RJJ10B0219-0400