

E1/E20エミュレータ

ユーザーズマニュアル別冊

(High-performance Embedded Workshop RXデバッグ編)

対象デバイス
RXファミリ

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

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、
 家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、
 防災・防犯装置、各種安全装置等
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

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

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

目次

1.	E1/E20 エミュレータマニュアル構成	8
2.	デバッグの準備をする	9
2.1	動作環境	9
2.2	High-performance Embedded Workshopを起動する	10
2.3	新規にワークスペースを作成する（ツールチェイン未使用）	11
2.4	新規にワークスペースを作成する（ツールチェイン使用）	14
2.5	既存のワークスペースを開く	17
2.6	エミュレータを接続する	18
2.6.1	エミュレータを接続する	18
2.6.2	エミュレータを再接続する	18
2.7	エミュレータを切断する	18
2.7.1	エミュレータを切断する	18
2.8	High-performance Embedded Workshop を終了する	18
2.9	デバッグ設定を行う	19
2.9.1	ダウンロードモジュールを指定する	19
2.9.2	コマンドラインバッチファイルの自動実行の設定を行う	20
2.10	E1/E20エミュレータデバッグの起動手順	21
3.	ソフトウェア仕様	29
3.1	E1/E20エミュレータ仕様一覧	29
3.2	E1/E20エミュレータとマイコンの相違点	34
3.3	デバッグの設定	36
3.3.1	[起動設定] ダイアログボックス	36
3.3.2	[コンフィグレーションプロパティ] ダイアログボックス	41
4.	E1/E20 エミュレータ機能	50
4.1	機能概要	50
4.2	ダウンロード機能	52
4.3	ソースファイルを開く	52
4.3.1	ソースコードを表示する	52
4.3.2	アセンブリ言語コードを表示する	54
4.3.3	アセンブリ言語コードを修正する	55
4.4	メモリアクセス機能	55
4.4.1	メモリリード/ライト機能	55
4.5	ブレーク機能	57
4.5.1	強制ブレーク機能	57
4.5.2	S/W（ソフトウェア）ブレーク機能	57
4.5.3	オンチップブレーク機能	57
4.6	イベント機能	59
4.6.1	イベントを使用する	59
4.6.2	イベントを追加する	60

4.6.3	イベントを削除する	65
4.6.4	イベントを登録する	67
4.6.5	登録したイベントの削除	71
4.6.6	イベントを都度入力する／再使用する	72
4.6.7	イベントを適用させる	73
4.7	オンチップブレーク機能	74
4.7.1	オンチップブレークポイントを設定する	74
4.7.2	オンチップブレーク設定内容を保存する／読み込む	79
4.8	トレース機能	80
4.8.1	トレース情報を見る	81
4.8.2	トレース情報を取得する	81
4.8.3	トレース結果を表示する	84
4.8.4	オプションメニュー	88
4.8.5	トレース情報取得条件を設定する	90
4.8.6	トレース情報をファイルに保存する	95
4.8.7	トレース情報をファイルから読み込む	95
4.8.8	トレース情報の取得を一時的に停止する	95
4.8.9	トレース情報の取得を再開する	95
4.9	パフォーマンス測定機能	96
4.9.1	パフォーマンスを測定する	96
4.9.2	パフォーマンス測定結果を表示する	96
4.9.3	パフォーマンス測定条件を設定する	99
4.9.4	パフォーマンス測定を開始する	104
4.9.5	パフォーマンス測定条件を解除する	105
4.9.6	パフォーマンス測定結果をクリアする	105
4.9.7	パフォーマンスの計測最大回数について	105
4.9.8	パフォーマンス条件設定の内容を保存する/読み込む	105
4.10	リアルタイムRAMモニタ機能	106
4.10.1	RAM モニタを使用する	106
4.10.2	RAM モニタ範囲を割り当てる	107
4.10.3	モニタ内容を表示する	109
4.10.4	ロストについて	112
4.10.5	ロスト発生の抑止	113
4.11	Start/Stopファンクション機能	115
4.11.1	[Start/Stop ファンクション設定]ダイアログボックスを開く	115
4.11.2	実行するルーチンを指定する	115
4.11.3	Start/Stop ファンクションの制限事項	116
4.11.4	指定ルーチンの記述に関する制限事項	117
4.12	デバッグコンソール機能	118
4.12.1	[デバッグコンソール]ウィンドウを開く	118
4.12.2	低水準インタフェースルーチンについて	118
4.13	ホットプラグイン機能	121
4.13.1	起動手順	121
4.13.2	ホットプラグインの注意事項	127
4.14	グラフ機能	128
4.14.1	[グラフ] ウィンドウを表示する	128
4.15	スタックトレース機能	131
4.16	オンラインヘルプ	131

5.	チュートリアル	132
5.1	はじめに	132
5.2	High-performance Embedded Workshopの起動	133
5.3	エミュレータの接続	133
5.4	チュートリアルプログラムのダウンロード	134
5.4.1	チュートリアルプログラムをダウンロードする	134
5.4.2	ソースプログラムを表示する	135
5.5	S/Wブレークポイントの設定	136
5.6	プログラムの実行	137
5.6.1	CPUのリセット	137
5.6.2	プログラムを実行する	137
5.7	ブレークポイントの確認	139
5.7.1	ブレークポイントを確認する	139
5.8	レジスタ内容の変更	140
5.9	シンボルの参照	141
5.10	メモリ内容の確認	142
5.11	変数の参照	143
5.12	ローカル変数の表示	145
5.13	プログラムのステップ実行	146
5.13.1	ステップインの実行	146
5.13.2	ステップアウトの実行	147
5.13.3	ステップオーバーの実行	148
5.14	プログラムの強制ブレーク	149
5.15	オンチップブレーク機能	150
5.15.1	指定アドレスを実行したときにプログラム実行を停止	150
5.16	メモリにアクセスしたときにプログラム実行を停止	151
5.17	トレース機能	152
5.17.1	フリートレースによるトレース情報の表示	153
5.18	スタックトレース機能	156
5.19	さてつぎは？	157
6.	使用上の注意事項	158
6.1	メモリ	158
6.1.1	I/O レジスタ領域	158
6.1.2	内蔵フラッシュ ROM 領域	158
6.1.3	内蔵フラッシュ ROM へのダウンロード	158
6.1.4	内蔵フラッシュ ROM の書き換え	158
6.1.5	FCU-RAM, FCU ファーム領域	159
6.1.6	デバッガが使用するワーク RAM 領域	159
6.2	メモリウィンドウ	159
6.2.1	コピー、比較、検索機能	159
6.2.2	オプションメニュー	159
6.3	実行操作	160
6.3.1	カーソル位置まで実行	160
6.3.2	条件を指定して実行	160
6.3.3	ステップアウト実行	160
6.4	リセット	161
6.4.1	エミュレータシステムによる操作とマイコンリセットの競合について	161

6.4.2	トレース機能使用時のマイコンリセット	161
6.4.3	リアルタイム RAM モニタ機能使用時のマイコンリセット	161
6.4.4	ユーザプログラム実行中のリセット	161
6.5	IOウィンドウ	162
6.5.1	I/O レジスタファイルのカスタマイズ	162
6.5.2	ベリファイ	162
6.6	トレース機能	163
6.6.1	記録可能なアクセス	163
6.6.2	トレース情報	163
6.7	イベント機能	164
6.7.1	検出可能なアクセス	164
6.7.2	イベント組み合わせ	164
6.7.3	通過回数条件	164
6.7.4	データアクセスイベントのアドレス範囲条件	164
6.7.5	イベント登録について	164
6.7.6	WAIT 命令に対するイベント	165
6.7.7	イベント要因について	165
6.8	ブレーク機能	166
6.8.1	ブレークポイント設定時の注意事項	166
6.9	リアルタイムRAMモニタ機能	167
6.10	パフォーマンス測定機能	168
6.10.1	パフォーマンス測定中のリセットについて	168
6.10.2	パフォーマンス測定中のネスト制限	168
6.10.3	[一度だけ計測] チェック時の注意事項	168
6.11	ダウンロード機能	169
6.11.1	アクセスサイズについて	169
6.12	外部フラッシュメモリへのダウンロード	171
6.13	実行時間	172
6.14	Start/Stopファンクション	172
6.15	ウォッチ機能	172
6.16	デバッグコンソール機能	172
6.17	FINEインタフェース	172
6.18	オプション設定関連レジスタ	173
6.18.1	エンディアン選択レジスタ(MDEB, MDES)について	173
6.18.2	オプション機能選択レジスタ 1(OFS1)の設定について	173
6.19	その他	173
6.19.1	フラッシュメモリ書き換え時のレジスタ値	173
6.19.2	DMAC、DTC に関して	173
6.19.3	高速クロック発振器(HOCO)に関して	174
6.19.4	ロックビット	174
6.19.5	エミュレータの切断について	174
6.19.6	動作周波数について	174
6.19.7	USB ブートプログラムが格納されたマイコンについて	175
6.19.8	クロック操作を許可する設定について	175
6.19.9	MPU 領域のアクセスについて	175

付録 A	メニュー一覧.....	176
付録 B	High-performance Embedded Workshop の注意事項.....	180

1. E1/E20 エミュレータマニュアル構成

E1/E20 エミュレータのマニュアルは、E1/E20 エミュレータユーザーズマニュアルと各マイコンに対応したE1/E20 エミュレータユーザーズマニュアル 別冊で構成されています。

E1/E20 エミュレータのご使用にあたり、それぞれのユーザーズマニュアルを必ずお読みください。

(1) E1/E20エミュレータ ユーザーズマニュアル

E1/E20エミュレータユーザーズマニュアルには、ハードウェア仕様が記載されています。

- E1/E20エミュレータの構成
- E1/E20エミュレータのハードウェア仕様
- E1/E20エミュレータとホストマシンおよびユーザシステムとの接続

(2) E1/E20エミュレータ ユーザーズマニュアル 別冊 (RXユーザシステム設計編)

E1/E20エミュレータユーザーズマニュアル 別冊 (RXユーザシステム設計編)には、ハードウェア設計時に必要なE1/E20エミュレータとの接続例やインタフェース回路が記載されています。

(3) E1/E20エミュレータ ユーザーズマニュアル 別冊 (High-performance Embedded Workshop RXデバッグ編)

E1/E20エミュレータユーザーズマニュアル 別冊 (High-performance Embedded Workshop RXデバッグ編)には、High-performance Embedded Workshop RX E1/E20エミュレータデバッグの機能説明および操作方法が記載されています。

- RX E1/E20エミュレータデバッグの機能説明
- RX E1/E20エミュレータデバッグの操作方法
- RX E1/E20エミュレータデバッグ起動からデバッグ作業までのチュートリアル
- RX E1/E20エミュレータ使用時の注意事項

2. デバッグの準備をする

2.1 動作環境

本エミュレータを使用する場合、表2.1 及び表2.2 に示すホストマシン動作環境を必ず守って使用してください。

表2.1 動作環境 (Windows® XP)

PC環境	
PC本体	IBM PC/AT互換機
OS	Windows® XP 32ビット版 *1 *3
CPU	Pentium 4 1.6GHz 以上を推奨
インタフェース	USB2.0 *2
メモリ	(1GB+ロードモジュールのファイルサイズの10倍) 以上を推奨
マウスなどのポインティングデバイス	ホストマシン本体に接続可能で上記OSに対応している、マウスなどのポインティングデバイス
CDドライブ	エミュレータデバッガをインストールするため、またはユーザーズマニュアルを参照するために必要
ハードディスク	エミュレータデバッガのインストールに600MB以上の空き容量が必要 (スワップ領域を考慮して、さらにメモリ容量の2倍以上(推奨4倍以上)の空き容量をご用意ください)
ディスプレイ解像度	1024×768以上を推奨

表2.2 動作環境 (Windows Vista®, Windows® 7)

PC環境	
PC本体	IBM PC/AT互換機
OS	Windows Vista® 32ビット版 *1 *4 Windows® 7 32/64ビット版 *1
CPU	Pentium 4 3GHz またはCore 2 Duo 1GHz以上を推奨
インタフェース	USB2.0 *2
メモリ	(2GB+ロードモジュールのファイルサイズの10倍)以上を推奨(32ビット) (3GB+ロードモジュールのファイルサイズの10倍)以上を推奨(64ビット)
マウスなどのポインティングデバイス	ホストマシン本体に接続可能で上記OSに対応している、マウスなどのポインティングデバイス
CDドライブ	エミュレータデバッガをインストールするため、またはユーザーズマニュアルを参照するために必要
ハードディスク	エミュレータデバッガのインストールに600MB以上の空き容量が必要 (スワップ領域を考慮して、さらにメモリ容量の2倍以上(推奨4倍以上)の空き容量をご用意ください)
ディスプレイ解像度	1024×768以上を推奨

*1 : Windows および Windows Vista は、米国 Microsoft Corporation の米国およびその他の国における商標または登録商標です。その他すべての会社名および製品名は、各社の登録商標または商標です。

*2 : USB インタフェースは、すべてのホストマシン、USB デバイス、USB ハブの組み合わせでの動作を保証するものではありません。

*3 : 64 ビット版の Windows® XP には対応していません。

*4 : 64 ビット版の Windows Vista®には対応していません。

2.2 High-performance Embedded Workshopを起動する

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

- (1) ホストマシンとエミュレータ、ユーザシステムを接続し、エミュレータとユーザシステムの電源を入れてください。
- (2) [スタート] メニューの[プログラム] から [Renesas] -> [High-performance Embedded Workshop] -> [High-performance Embedded Workshop] を選択してください。
[ようこそ!] ダイアログボックスが表示されます。

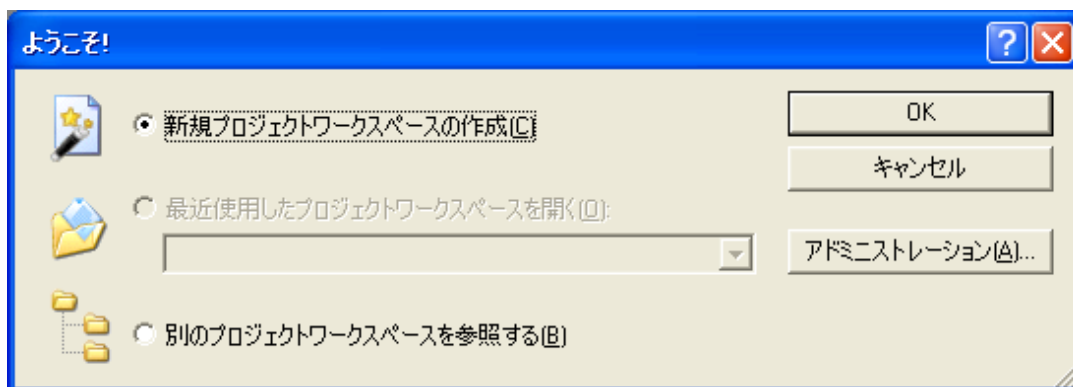


図2.1 [ようこそ!] ダイアログボックス

起動方法を以下から選択してください。

- 新規プロジェクトワークスペースを作成する
ワークスペースを新規作成する場合に選択します。
- 最近使用したプロジェクトワークスペースを開く
開いたワークスペースの履歴が表示されます。
使用した既存のワークスペースをそのまま使用する場合に選択します。
- 別のプロジェクトワークスペースを参照する
既存のワークスペースを使用する場合に選択します。

2.3 新規にワークスペースを作成する（ツールチェーン未使用）

ツールチェーンを使用する場合と使用しない場合では新規プロジェクトワークスペースの作成手順が異なります。

本製品には、ツールチェーンは含まれていません。ツールチェーンは C/C++コンパイラパッケージがインストールされている環境にて使用することができます。

以下の方法で、新規ワークスペースを作成してください。

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

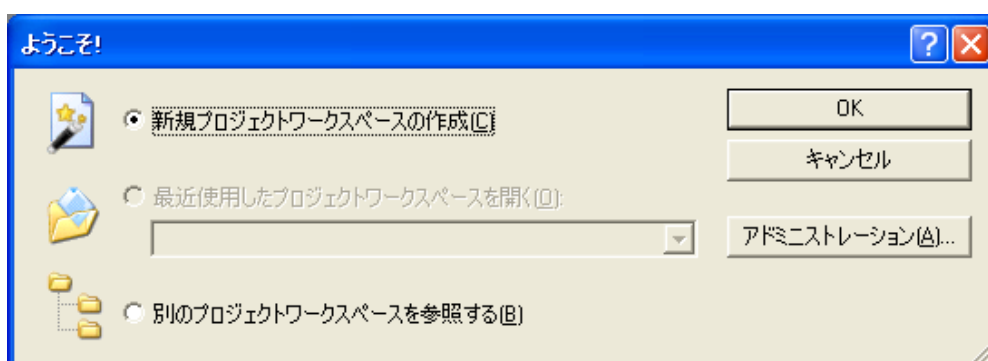


図2.2 新規プロジェクトワークスペースの作成

- (2) Project Generatorを開始します。

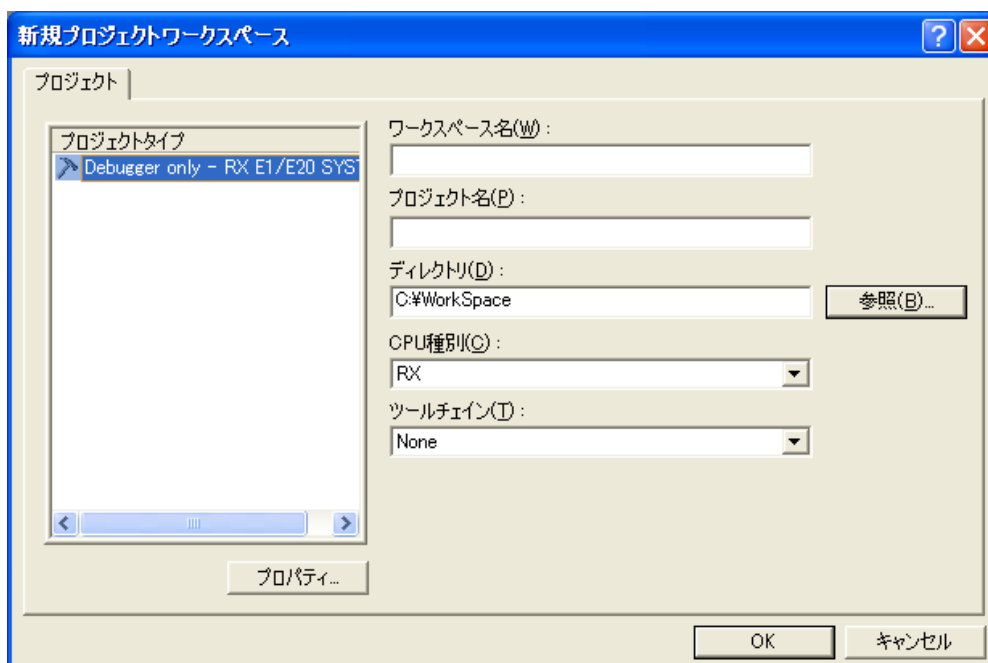


図2.3 [新規プロジェクトワークスペース] ダイアログボックス

- [ワークスペース名]: ワークスペース名を入力してください。
- [プロジェクト名]: プロジェクト名を入力してください。
 ワークスペース名と同じで良い場合は、入力する必要はありません。
- [ディレクトリ]: ワークスペースを作成するディレクトリを入力してください。
 [参照...]ボタンをクリックしてワークスペースを作成するディレクトリを選択することも
 できます。
- [CPU 種別]: 該当する CPU ファミリを選択してください。

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

[OK] ボタンをクリックしてください。

(3) デバッガターゲットを選択します。

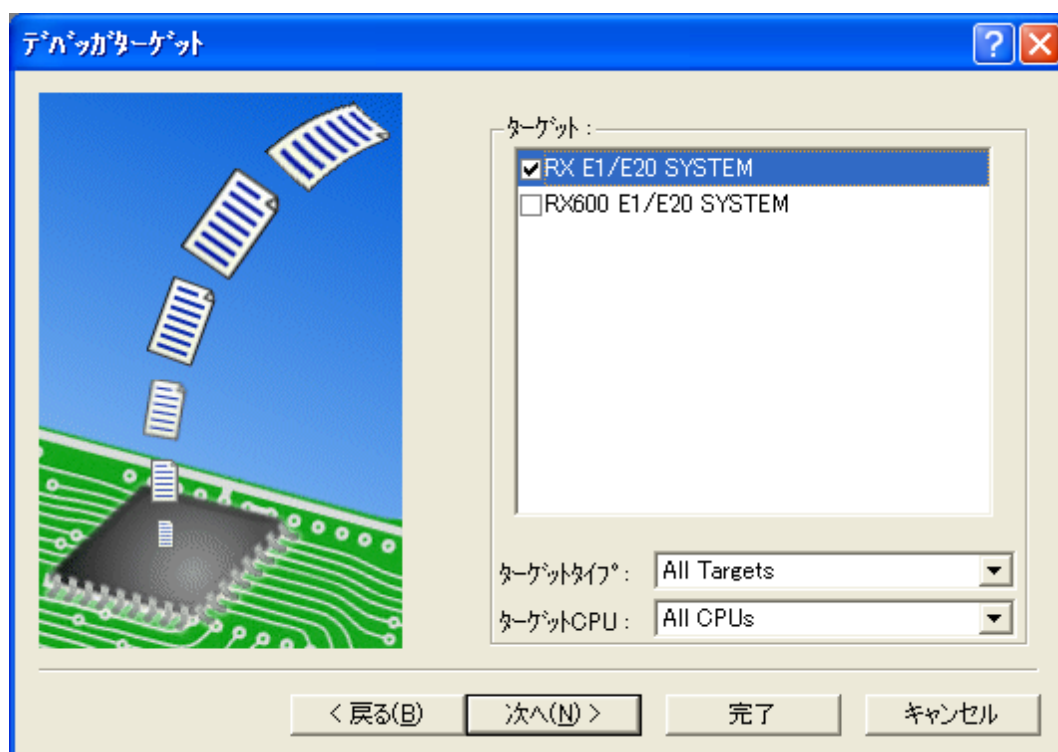


図2.4 [デバッガターゲット] ダイアログボックス

RX ファミリ向けターゲットプラットフォームの“RX E1/E20 SYSTEM”にチェックし、[次へ] ボタンをクリックしてください。

“RX600 E1/E20 SYSTEM”は以前のバージョンとの互換性を維持するために用意されています。こちらは選択しないでください。

(4) コンフィグレーション名を設定します。

コンフィグレーションとは、エミュレータ以外の High-performance Embedded Workshop の状態を保存するファイルです。



図2.5 [デバッガオプション] ダイアログボックス

複数のターゲットプラットフォームをチェックした場合は、[次へ] ボタンをクリックし、チェックしたターゲットプラットフォームごとにコンフィグレーション名を設定します。

コンフィグレーション名の設定が完了したら、エミュレータデバッガに関する設定は完了です。

[完了] ボタンをクリックすると、[概要] ダイアログボックスが表示されます。

[OK] ボタンをクリックすると、High-performance Embedded Workshop が起動します。

(5) 起動後、エミュレータを接続してください。

2.4 新規にワークスペースを作成する（ツールチェーン使用）

以下の方法で、新規ワークスペースを作成してください。

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

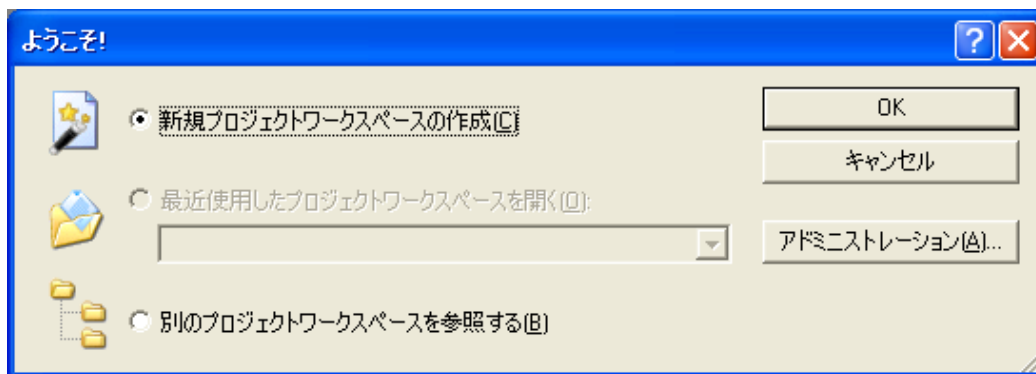


図2.6 [ようこそ!] ダイアログボックス

- (2)Project Generatorを開始します。

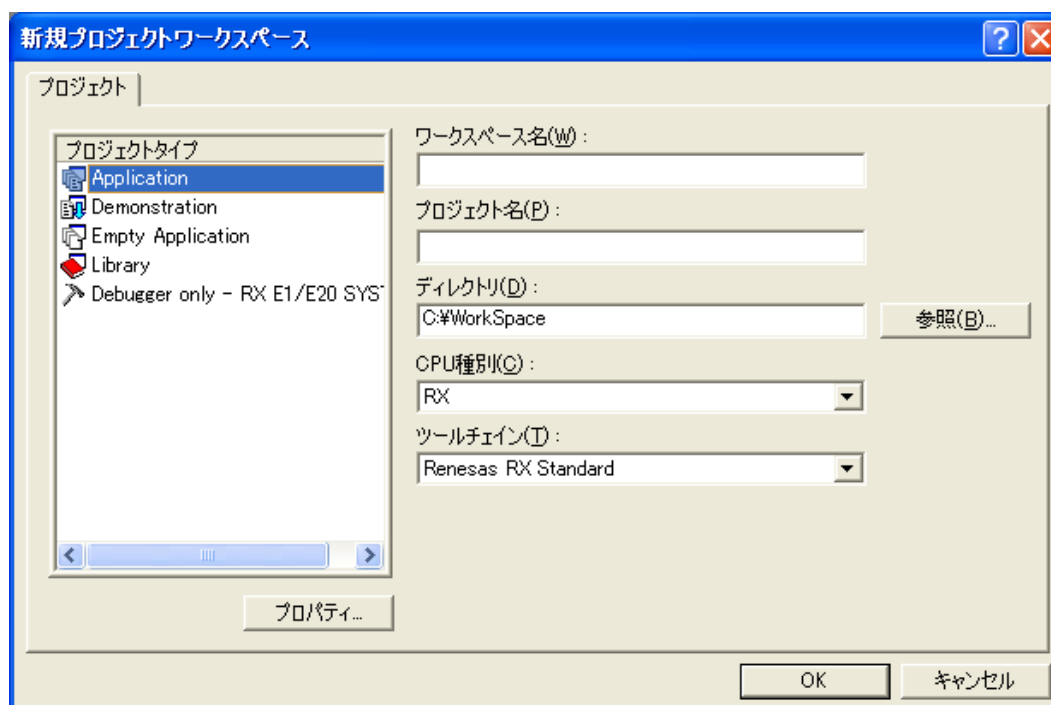


図2.7 [新規プロジェクトワークスペース] ダイアログボックス

- [ワークスペース名]: ワークスペース名を入力してください。
- [プロジェクト名]: プロジェクト名を入力してください。
 ワークスペース名と同じで良い場合は、入力する必要はありません。
- [ディレクトリ]: ワークスペースを作成するディレクトリを入力してください。
 [参照...]ボタンをクリックしてワークスペースを作成するディレクトリを選択することもできます。
- [CPU 種別]: 該当する CPU ファミリを選択してください。
- [ツールチェーン]: ツールチェーンを使用する場合は該当するツールチェーンを選択してください。
 使用しない場合、[None] を選択してください。

上記項目の設定が完了したら[OK] ボタンを押してください。

(3) ツールチェーンのCPU設定、オプション設定、その他必要な設定を行ってください。

(4) デバッガターゲットを選択します。



図2.8 [新規プロジェクト-8/10-デバッガ] ダイアログボックス

使用するターゲットプラットフォームにチェックし、[次へ] ボタンをクリックしてください。

デバッガを使用する場合は、RX ファミリ向けターゲットプラットフォームの“RX E1/E20 SYSTEM”にチェックしてください。

“RX600 E1/E20 SYSTEM”は以前のバージョンとの互換性を維持するために用意されています。こちらは選択しないでください。

(5) コンフィグレーション名を設定します。



図2.9 [新規プロジェクト-9/10-デバッグオプション] ダイアログボックス

複数のターゲットプラットフォームをチェックした場合は、[次へ] ボタンをクリックし、チェックしたターゲットプラットフォームごとにコンフィグレーション名を設定します。

コンフィグレーション名の設定が完了したら、エミュレータデバッグに関する設定は完了です。

[完了] ボタンをクリックすると、[概要] ダイアログボックスが表示されます。

[OK] ボタンをクリックすると、High-performance Embedded Workshop が起動します。

(6) 起動後、エミュレータを接続してください。

2.5 既存のワークスペースを開く

以下の方法で、既存ワークスペースを開きます。

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

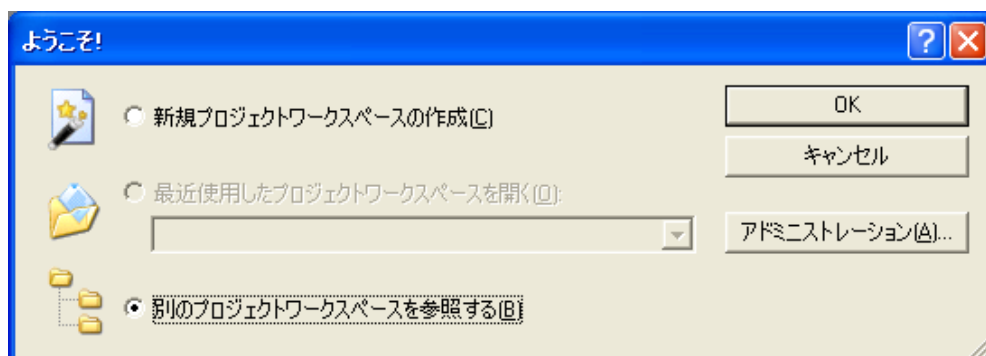


図2.10 [ようこそ!] ダイアログボックス

- (2) [ワークスペースを開く] ダイアログボックスが開きます。

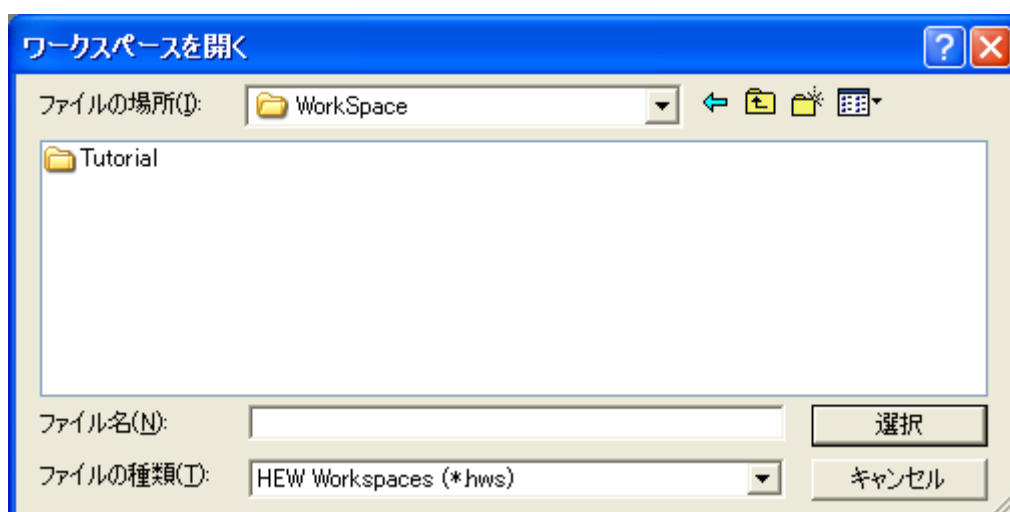


図2.11 [ワークスペースを開く] ダイアログボックス

ワークスペースが作成されているディレクトリを指定し、ワークスペースファイル（拡張子.hws）を選択して [選択] ボタンをクリックしてください。

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

指定したワークスペースがエミュレータに接続された状態を保存していた場合には、エミュレータへの接続が自動で行われます。

指定したワークスペースがエミュレータに接続されていない状態を保存していた場合に、エミュレータの接続を行う場合は、「2.6 エミュレータを接続する」を参照してください。

2.6 エミュレータを接続する

2.6.1 エミュレータを接続する

エミュレータの接続には、以下の方法があります。

(1)エミュレータ起動時の設定を行ってから接続する方法

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


[デバッグの設定] ダイアログボックスでターゲットを選択して[OK] ボタンを押すとエミュレータが接続されます。

(2)セッションファイルのロード

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

2.6.2 エミュレータを再接続する


エミュレータ切断状態時に、以下のいずれかの方法で再接続を行うことができます。

- (1) [デバッグ -> 接続] を選択します。
- (2) [接続] ツールバーボタン()をクリックします。
- (3) [コマンドライン] ウィンドウで "connect" コマンドを入力します。

2.7 エミュレータを切断する

2.7.1 エミュレータを切断する

起動中のエミュレータを、以下のいずれかの方法で接続解除することができます。

- (1) [デバッグ -> 接続解除] を選択します。
- (2) [接続解除] ツールバーボタン()をクリックします。
- (3) [コマンドライン] ウィンドウで "disconnect" コマンドを入力します。

2.8 High-performance Embedded Workshop を終了する

[ファイル -> アプリケーションの終了] を選択すると、High-performance Embedded Workshop 自体を終了します。

終了前には、セッション保存確認のメッセージボックスが表示されます。セッションを保存する場合は、[はい] ボタンをクリックしてください。

2.9 デバッグ設定を行う

ダウンロードモジュールの登録、コマンドラインバッチファイルの自動実行の設定、ダウンロード時のオプション設定等を行います。

2.9.1 ダウンロードモジュールを指定する

[デバッグ -> 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きます。

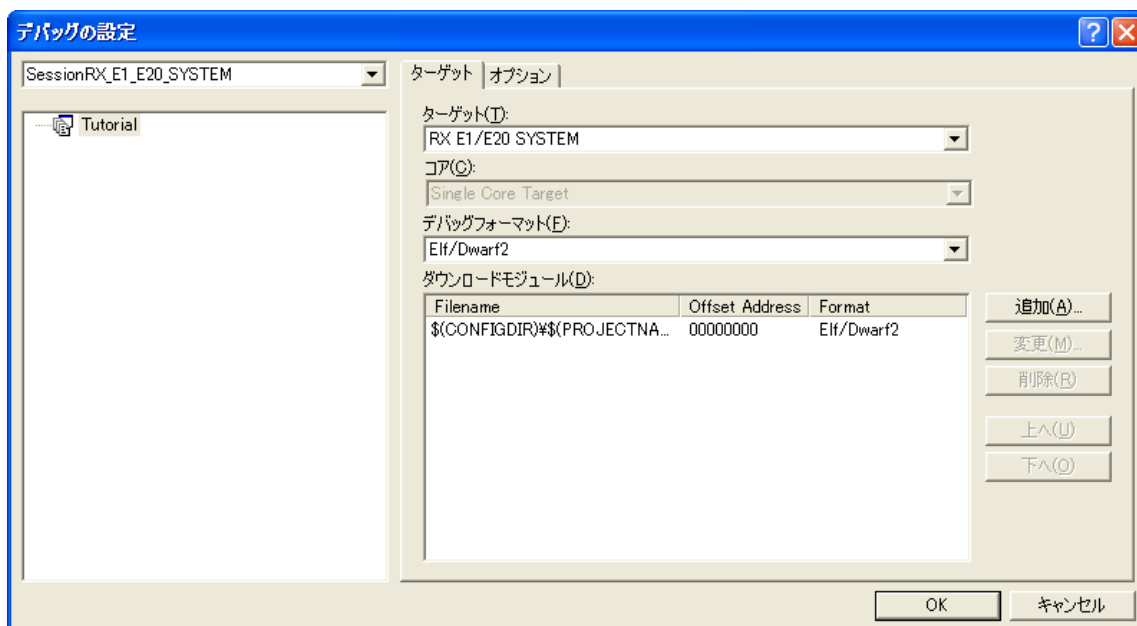


図2.12 [デバッグの設定] ダイアログボックス

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

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

【注】 この時点ではプログラムのダウンロードはされていません。ダウンロード方法については、「4.2 ダウンロード機能」を参照してください。

2.9.2 コマンドラインバッチファイルの自動実行の設定を行う

[オプション] タブをクリックしてください。

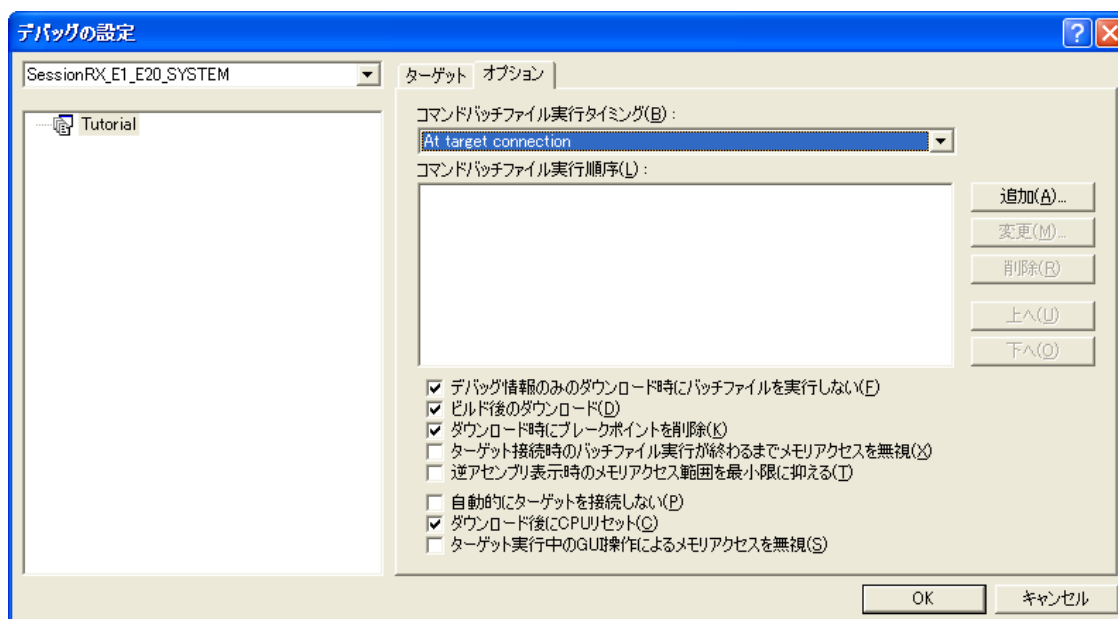


図2.13 [デバッグの設定] ダイアログボックス

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

- エミュレータ接続時 (At target connection)
- ダウンロード直前 (Before download of modules)
- ダウンロード直後 (After download of modules)
- リセット直後 (After reset)

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

2.10 E1/E20 エミュレータデバッグの起動手順

本章では、High-performance Embedded Workshop を起動し、E1/E20 エミュレータがユーザシステム上のマイコンと正しく接続されることをチェックします。

ここでは、本製品に添付されているチュートリアル用のワークスペースを使用してチェックを行います。

最初に以下の内容を確認してください。

- ユーザシステムの電源がOFFになっていることを確認してください。
- ユーザインタフェースケーブルの片方をE1/E20エミュレータのユーザ側コネクタに接続し、もう片方をユーザシステム側のエミュレータ接続コネクタに接続します。
- USBインタフェースケーブルでホストマシンとE1/E20エミュレータを接続してください。

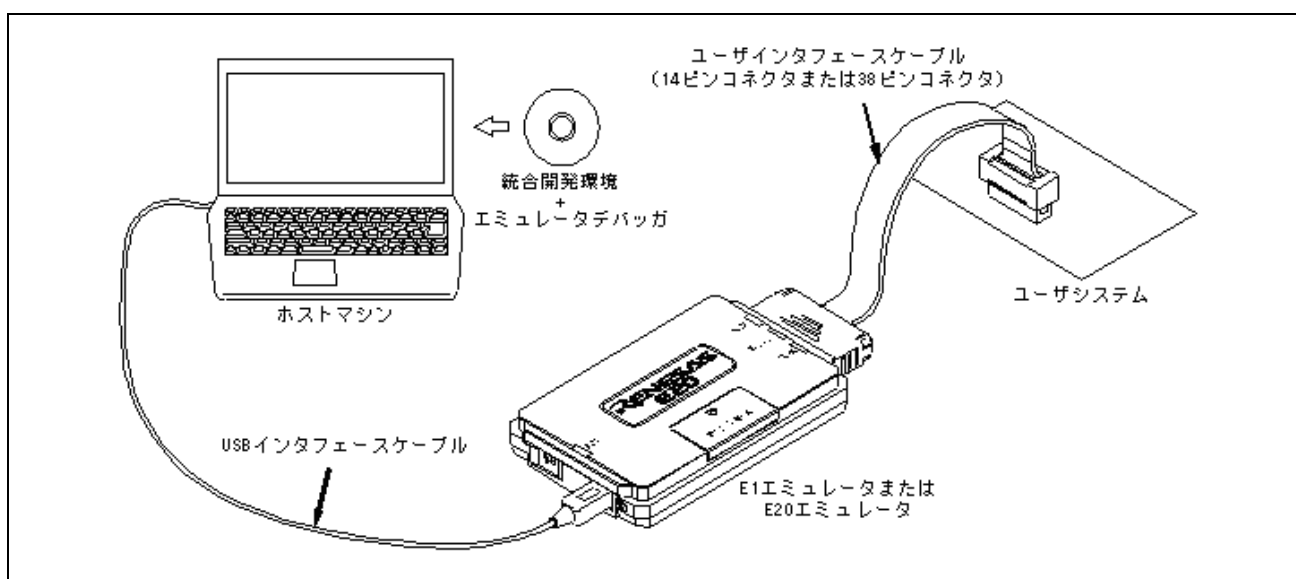


図2.14 システムチェックの構成

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

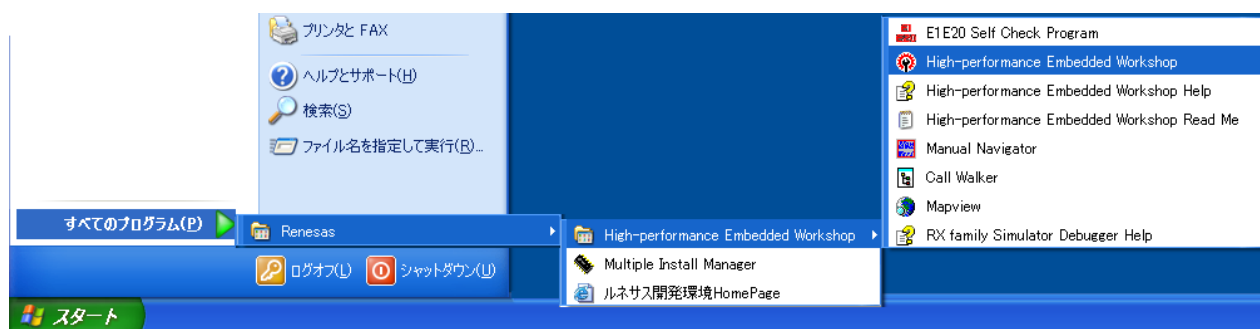


図2.15 [スタート] メニュー

(2) [ようこそ!]ダイアログボックスが表示されます。

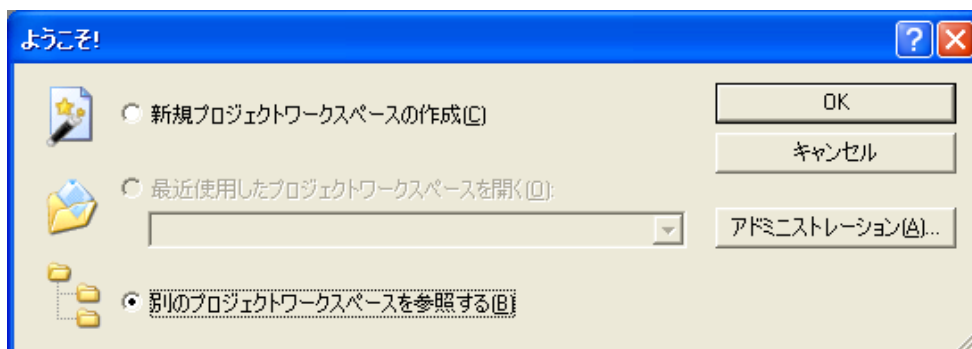


図2.16 [ようこそ!]ダイアログボックス

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

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

<OSがインストールされているドライブ> ¥Workspace¥Tutorial¥E1E20¥RXxxx¥Tutorial_LittleEndian
xxxは対象の製品グループを示します。

ディレクトリの指定後、以下のファイル (Tutorial.hws) を選択し[選択]ボタンを押してください。

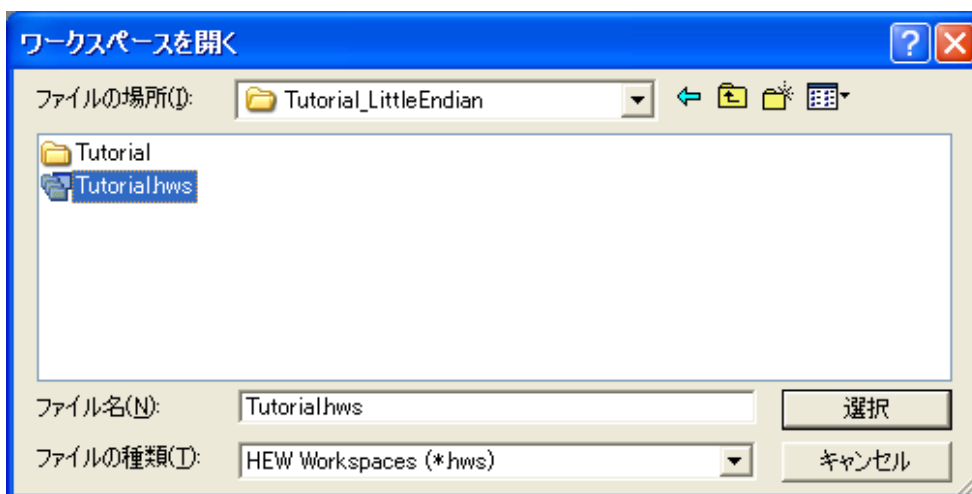


図2.17 [ワークスペースを開く] ダイアログボックス

(3) [起動設定] ダイアログボックスが表示されます。

本ダイアログボックスでは、ターゲットマイコンに関する設定および起動と通信設定を行います。各項目を設定して[OK] ボタンを押してください。

なお、[起動設定] ダイアログボックスの詳細説明は、「3.3.1 [起動設定] ダイアログボックス」を参照してください。

また、ホットプラグインについては、「4.13 ホットプラグイン機能」を参照してください。

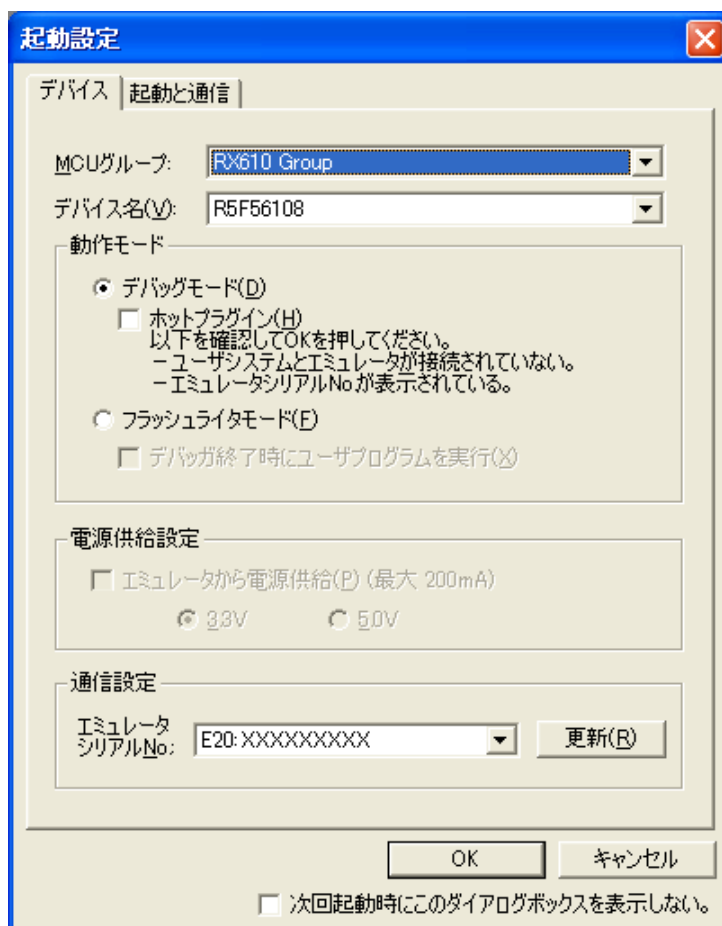


図2.18 [起動設定] ダイアログボックス-[デバイス] ページ

【留意事項】

[起動設定] ダイアログボックスで表示される内容は、マイコンごとに異なる場合があります。

以下に示す2つの注意事項は、E1エミュレータのみの内容です。E20エミュレータでは電源供給機能をサポートしていません。

【エミュレータの電源供給機能を使用してユーザシステムに電源供給する場合の注意事項】

もし、外部電源からユーザシステム上に電源供給していない状態で[電源供給設定] の項目にチェックを入れ忘れた場合は、以下のダイアログボックスが表示されます。

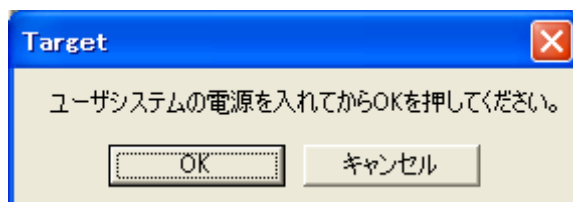


図2.19 電源供給確認ダイアログボックス

[OK] ボタンを押すと[電源] ダイアログボックスが表示されます。各項目にチェックを入れて[OK] ボタンを押してユーザシステムに電源供給してください。

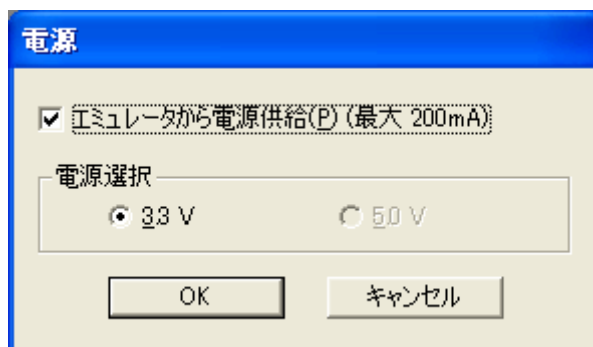


図2.20 [電源] ダイアログボックス

【外部電源を使用してユーザシステムに電源供給する場合の注意事項】

もし、外部電源からユーザシステム上に電源供給している状態で[電源供給設定] の項目にチェックを入れた場合は、下記のダイアログボックスが表示されます。

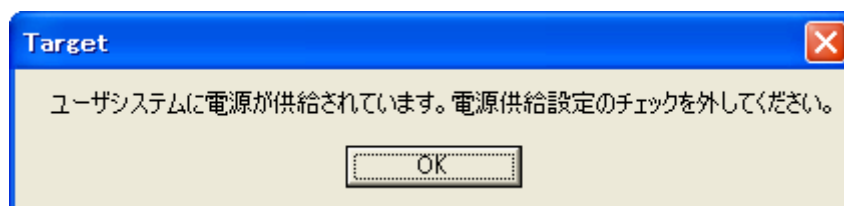


図2.21 ユーザシステム側の電源供給確認ダイアログボックス

[OK] ボタンを押すと[電源] ダイアログボックスが表示されます。[エミュレータから電源供給] チェックボックスのチェックを外して[OK] ボタンを押してエミュレータ側から電源供給しないようにしてください。

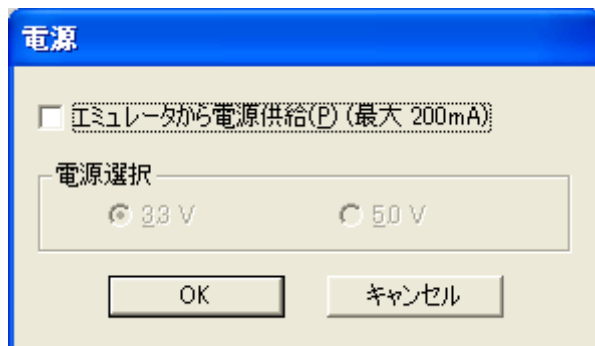


図2.22 [電源]ダイアログボックス

(4) [起動設定] ダイアログボックスの設定後に[OK] ボタンを押すと[接続中...] ダイアログボックスが表示されます。



図2.23 [接続中...] ダイアログボックス

- (5) エミュレータファームウェアのバージョンが古い場合は、[ファームウェアの書き換え] ダイアログボックスが表示されます。

[OK] ボタンを押した場合、エミュレータファームウェアの書き換えを開始します。一度エミュレータファームウェアの書き換えが開始されると途中でキャンセルすることはできませんのでご注意ください。

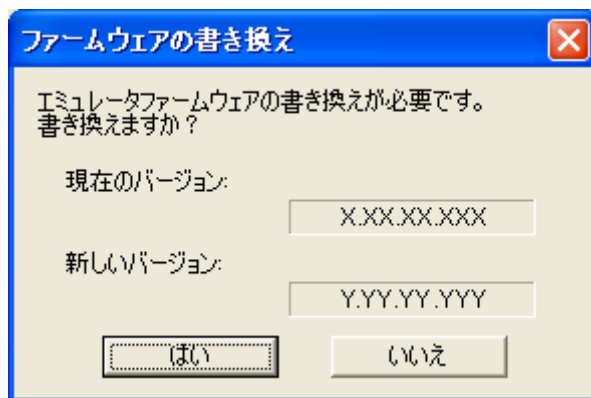


図2.24 [ファームウェアの書き換え] ダイアログボックス

エミュレータファームウェアの書き換え中は、[ダウンロード] プロGRESSバーが表示されます。書き換えが終了すると、エミュレータの起動処理を再開します。

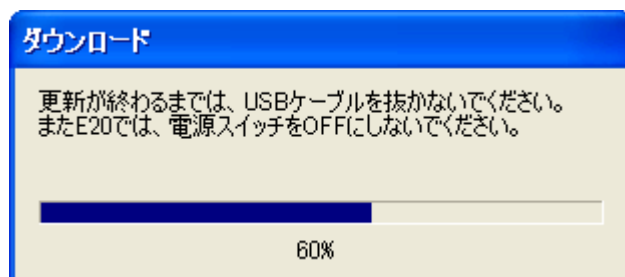


図2.25 [ダウンロード] プロGRESSバー

注意

書き込みが終了するまで絶対にUSBインタフェースケーブルを抜かないでください。
E1/E20 エミュレータの破壊につながります。

- (6) ターゲットマイコンの内蔵フラッシュメモリにIDコードが設定されている場合は、[IDコード確認] ダイアログボックスが表示されます。

設定したIDコードを入力して[OK] ボタンを押してください。

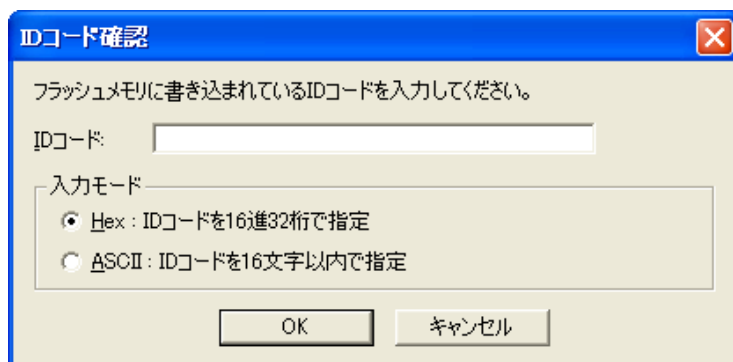


図2.26 [IDコード確認] ダイアログボックス

- (7) 次に[コンフィグレーションプロパティ] ダイアログボックスが表示されます。

本ダイアログボックスでは、エミュレータおよびデバッグ機能に関する設定を行います。各項目を設定して[OK] ボタンを押してください。

なお、[コンフィグレーションプロパティ] ダイアログの詳細説明は、「3.3.2 [コンフィグレーションプロパティ] ダイアログボックス」を参照してください。

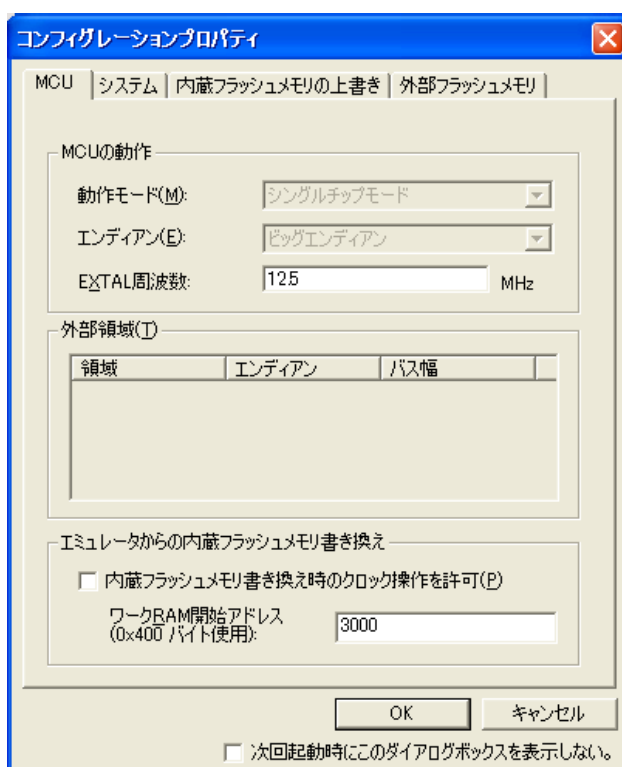


図2.27 [コンフィグレーションプロパティ] ダイアログボックス—[MCU] ページ

【注意事項】

[コンフィグレーションプロパティ] ダイアログボックスで表示される内容は、マイコンごとに異なる場合があります。

- (8) [コンフィグレーションプロパティ] ダイアログボックスの設定をして[OK] ボタンを押した後に High-performance Embedded Workshop の[Output] ウィンドウに”Connected”と表示されたら、E1/E20 エミュレータの起動は完了です。

もし、[Output] ウィンドウに”Connected”以外が表示された場合は、表 2.3 を参照してください。

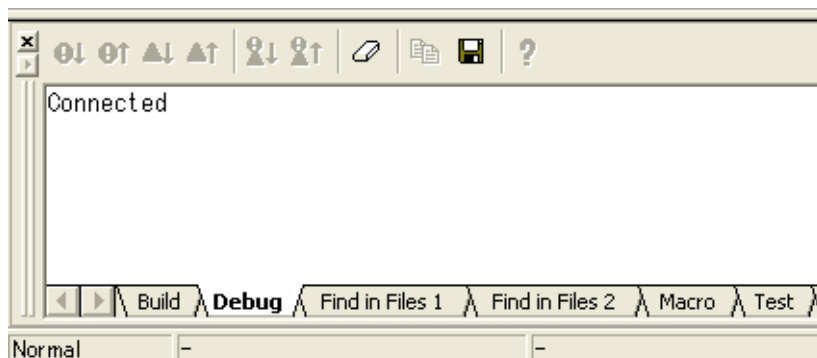


図2.28 [Output] ウィンドウ

【注】 ユーザプログラムが既に内蔵フラッシュメモリにダウンロードされている場合、E1/E20 エミュレータ起動後にはユーザプログラムのデバッグ情報がないため、そのままではソースレベルデバッグが行えません。ソースレベルデバッグを行うためには、E1/E20 エミュレータ起動後にデバッグ情報を含んだダウンロードモジュールファイルをダウンロードしてください。

表2.3 [Output] ウィンドウのエラーメッセージ

エラーメッセージ	対策
LEVEL0 のバージョンが一致しません。	ご使用のエミュレータを使用する場合はデバッグソフトウェアのバージョンアップが必要です。弊社ホームページよりダウンロードしてください。
デバイス ID コードが選択したデバイスのもので一致しません。 起動時に指定したデバイス名をご確認ください。	起動設定で指定されたデバイス名と接続対象の MCU が異なっている可能性があります。 起動設定で指定しているデバイス名をご確認ください。
電源電圧が 5.9V を超えています。 ユーザシステムの設定を確認してください。	ユーザシステムに供給されている電源電圧が規定値を超えている可能性があります。 ユーザシステムの設定をご確認ください。
JTAG 通信エラーが発生しました。 JTAG クロックを下げてお試しください。	下記内容をご確認ください。 <ul style="list-style-type: none"> ユーザインタフェースケーブルの接触不良 マイコンの端子接触不良 JTAG クロックが高すぎる／低すぎる
シリアル通信源クロックの切り替えにより正常に通信できなくなりました。 ボーレートを下げて再接続をお試しください。	下記内容をご確認ください。 <ul style="list-style-type: none"> ユーザインタフェースケーブルの接触不良 マイコンの端子接触不良 FINE ボーレートが高すぎる／低すぎる

3. ソフトウェア仕様

3.1 E1/E20 エミュレータ仕様一覧

表 3.1 にRX610、RX621、RX62N、RX62T、RX62G、RX630、RX631、RX63N、RX63Tグループ使用時におけるE1/E20 エミュレータの仕様一覧を記し、表 3.2 にRX210、RX21A、RX220 グループ使用時におけるE1/E20 エミュレータの仕様一覧を示します。

また、設定可能なイベント数やオンチップブレーク機能、トレース機能、およびパフォーマンス機能の設定条件は、ターゲットマイコンによって仕様が異なります。シリーズマイコン別に機能仕様をまとめた一覧表を表 3.3 に示します。

表3.1 RX610, RX621, RX62N, RX62T, RX62G, RX630, RX631, RX63N, RX63TグループでのE1/E20エミュレータ仕様

項目	E1	E20
対応マイコングループ	RXファミリ RX600 シリーズ RX610、RX621、RX62N、RX62T、 RX62G、RX630、RX631、RX63N、 RX63T グループ	←
対応動作モード【注1】	シングルチップモード、 内蔵ROM有効拡張モード、 内蔵ROM無効拡張モード、 ユーザブートモード	←
S/W ブレーク	最大 256 点	←
イベント (DMAC/DTC によるアクセスは検出不可)		
イベント数【注2】	実行アドレス: 8 点 データアクセス: 4 点	←
通過回数【注3】	最大 256 回	←
オンチップブレーク		
実行前 PC ブレーク【注2】	最大 8 点	←
イベント組み合わせ【注2】	OR / AND(累積) / シーケンシャル	←
その他	トレースフルブレーク	←
トレース (DMAC/DTC によるアクセスは記録不可)	内蔵トレース (最大 256 分岐またはサイクルの分岐情報、データアクセス情報を取得)	<ul style="list-style-type: none"> 内蔵トレース (最大 256 分岐またはサイクルの分岐情報、データアクセス情報を取得) 外部トレース出力 (約 2M 分岐またはサイクルの分岐情報、データアクセス情報を取得)
パフォーマンス測定	最大 2 点の実行サイクル、または、 実行回数 (32bit カウンタ×2 本、または 64bit カウンタ×1 本)	←

表 3.1 RX610, RX621, RX62N, RX62T, RX62G, RX630, RX631, RX63N, RX63T グループでの E1/E20 エミュレータ仕様(つづき)

項目	E1	E20
リアルタイム RAM モニタ (DMAC/DTC によるアクセスは 監視不可)	なし	4K バイト(1K バイト×4 ブロック) リード・ライトの最終アクセス履 歴を表示
デバッグコンソール	printf の出力をアウトプットウィンドウに表示	←
ユーザインタフェース	14 ピンコネクタ	38 ピンコネクタ、または 14 ピンコネクタ【注 4】
ホストマシンとのインタフェース	USB2.0(フルスピード/ハイスピード) 【注 5】	←
ユーザシステムとの接続	付属のユーザインタフェースケーブルで接続	←
エミュレータ用電源	不要(USB バスパワーのため、ホストマシン より供給)	←
電源供給機能【注 6】	ユーザシステムに 3.3V、または 5.0V 供給可 能(最大 200mA)	なし
外部フラッシュメモリへの ダウンロード	あり	←
ホットプラグイン 【注 7】【注 8】	あり	←
通信インタフェース 【注 9】【注 10】	JTAG、および FINE インタフェース	←

- 【注】 1. 選択可能な動作モードはマイコンによって異なります。
2. イベントはオンチップブレイク、トレースで共用されます。同時に複数の機能で同一のイベントを使用することはできません。
3. 通過回数はいずれかひとつのイベントに対してのみ指定可能です。
4. 外部トレース出力機能とリアルタイム RAM モニタ機能を使用しない場合は、E20 エミュレータでも 14 ピンコネクタを使用できます。その場合、38 ピン/14 ピン変換アダプタが必要です。
5. USB1.1 対応のホストマシンにも接続可能です。
6. E1 エミュレータを用いて信頼性を要求した書き込みを行う場合は、電源供給機能は使用せず、マイコン仕様に合致した電源を、ユーザシステムに別途供給してください。なお、量産工程などでの書き込みを行う場合は、FDT を使用してください。
7. E1 エミュレータにてホットプラグイン機能を使用する場合、別売のホットプラグアダプタ (R0E000010ACB00) をご購入してください。
8. FINE インタフェースによるホットプラグイン機能は対応していません。
9. ターゲットマイコンによって使用可能なインタフェースは異なります。インタフェースについては、ご使用のマイコンに関連したハードウェアマニュアルを参照してください。なお、RX610, RX621, RX62N, RX62T, RX62G グループは FINE インタフェースを使用できません。
10. FINE インタフェースは、FINEC, MD/FINED 端子を用いた 2 線式のみをサポートしています。

表3.2 RX210, RX21A, RX220グループでのE1/E20エミュレータ仕様

項目	E1	E20
対応マイコングループ	RX ファミリ RX200 シリーズ RX210, RX21A, RX220 グループ	←
対応動作モード	シングルチップモード、内蔵 ROM 有効拡張モード、 内蔵 ROM 無効拡張モード、ユーザブートモード	←
S/W ブレーク	最大 256 点	←
イベント (DMAC/DTC によるアクセスは検出不可)		
イベント数【注 1】	実行アドレス: 4 点 データアクセス: 2 点	←
通過回数	なし	←
オンチップブレーク		
実行前 PC ブレーク【注 1】	最大 4 点	←
イベント組み合わせ【注 1】	OR / AND(累積)/ シーケンシャル	←
その他	トレースフルブレーク	←
トレース (DMAC/DTC による アクセスは記録不可)	内蔵トレース(最大 64 分岐またはサイクルの分岐情報、 データアクセス情報を取得)	←
パフォーマンス測定	1 点の実行サイクル(24bit カウンタ×1 本)	←
リアルタイム RAM モニタ (DMAC/DTC によるアクセスは 監視不可)	なし	←
デバッグコンソール	printf の出力をアウトプットウィンドウに表示	←
ユーザインタフェース	14 ピンコネクタ	←【注 4】
ホストマシンとのインタフェース	USB2.0(フルスピード/ハイスピード)【注 2】	←
ユーザシステムとの接続	付属のユーザインタフェースケーブルで接続	←
エミュレータ用電源	不要(USB バスパワーのため、ホストマシンより供給)	←
電源供給機能【注 3】	ユーザシステムに 3.3V、または 5.0V 供給可能 (最大 200mA)	なし
外部フラッシュメモリへのダウン ロード	あり	←
ホットプラグイン	なし	←
通信インタフェース【注 5】	FINE インタフェース	←

- 【注】 1. イベントはオンチップブレーク、トレースで共用されます。同時に複数の機能で同一のイベントを使用することはできません。
2. USB1.1 対応のホストマシンにも接続可能です。
3. E1 エミュレータを用いて信頼性を要求した書き込みを行う場合は、電源供給機能は使用せず、マイコン仕様に合致した電源を、ユーザシステムに別途供給してください。なお、量産工程などでの書き込みを行う場合は、FDT を使用してください。
4. E20 エミュレータを使用する場合、38 ピン/14 ピン変換アダプタが必要です。
5. FINE インタフェースは、MD/FINED 端子を用いた 1 線式をサポートしています。

表3.3 ターゲットマイコンに対する機能仕様一覧

機能項目		RX600 シリーズマイコン	RX200 シリーズマイコン
イベント			
検出対象		CPU バスのみ	←
実行 アドレス	設定数	8 点	4 点
データ アクセス	設定数	4 点	2 点
	検出条件	アドレス条件指定： アドレスマスク，比較条件， アドレス範囲 (1 点のみ) データ条件指定： リード/ ライト，アクセスサイズ， データマスク，比較条件	アドレス条件指定： アドレスマスク，比較条件 データ条件指定： リード/ ライト，アクセスサイズ， データマスク，比較条件
通過回数条件		あり (1 点のみ)	なし
イベント 組み合わせ	条件	OR / AND (累積) / シーケンシャル	←
	設定数	OR / AND (累積) : 最大 8 点 シーケンシャル： 7 段階(順方向)+リセットポイント	OR / AND (累積) : 最大 6 点 シーケンシャル： 3 段階(順方向)+リセットポイント
トレース			
検出対象		CPU バスのみ	←
表示モード		BUS / DIS / SRC (混合表示可能)	←
外部 トレース	取得容量	1M / 2M / 4M / 8M / 16M / 32M Byte より選択可能	なし
	トレースフル ブ레이크	あり	なし
	トレース操作	プログラム実行中のトレース取得一時停止 およびリスタート	なし
内蔵 トレース	取得容量	最大 256 分岐	最大 64 分岐
	トレースフル ブ레이크	あり	あり
	トレース操作	なし	なし
取得モード		フル / フリー	←
出力モード		CPU 実行優先/ トレース出力優先/ 外部出力しない(内蔵トレースバッファ使用)	外部出力しない (内蔵トレースバッファ使用)
データ種別		分岐/ 分岐+データアクセス/ データアクセス	分岐元のみ/ 分岐/ 分岐元+時間/ データアクセス/ データアクセス+時間
トレース 抽出	条件	データアクセスイベント組み合わせ[OR]	←
	拡張設定	アクセス抽出設定： 例外命令/ FPU 命令/ ビット操作命令/ 論理演算命令/ 算術命令/ ストリング命令/ スタック命令 / 転送命令	なし
開始条件		イベント組み合わせ[OR / AND(累積) / シーケンシャル]によるトレース取得開始	←
終了条件		イベント組み合わせ[OR]によるトレース 取得終了	←

表 3.3 ターゲットマイコンに対する機能仕様一覧(つづき)

機能項目	RX600 シリーズマイコン	RX200 シリーズマイコン
パフォーマンス		
カウンタクロック	ICLK	←
カウンタ CH	2CH (32bit) または 1CH (64bit)	1CH (24bit)
計測項目	経過サイクル / 経過サイクル(スーパバイザモード) / 例外・割り込み処理サイクル / 割り込み処理サイクル / 例外処理サイクル / 有効命令発行回数 / 例外・割り込み受付回数 / 例外受付回数 / 割り込み受付 / イベントマッチ回数	経過サイクル
サイクルを時間情報で表示	あり	←
1 度だけ計測	あり	←
関数情報からのイベント追加	あり	←
64bit カウンタ	あり	なし
開始条件	イベント組み合わせ(OR) による パフォーマンス開始	←
終了条件	イベント組み合わせ(OR) による パフォーマンス終了	←

3.2 E1/E20 エミュレータとマイコンの相違点

(1) レジスタ初期値

E1/E20エミュレータは、システム起動時に汎用レジスタやコントロールレジスタの一部を初期化していますのでご注意ください（表3.4）。なお、マイコンの初期値は不定です。

表3.4 E1/E20エミュレータでのレジスタ初期値

状態	レジスタ名	E1/E20 エミュレータ
E1/E20 エミュレータ 起動時	R0(SP)	00000000h
	R1~R15	00000000h
	USP	00000000h
	ISP	00000000h
	PSW	00000000h
	PC	パワーオンリセットベクタテーブル中の PC の値
	INTB	00000000h
	BPSW	00000000h
	FINTV	00000000h
	FPSW 【注 2】	00000100h
	ACC 【注 1】	00000000h

- 【注】 1. ACC レジスタの b15-b0 は、常に“0”が読み出されます。B15-b0 への書き込みは無視されます。
2. RX210, RX21A, RX220 グループマイコンには、浮動少数点ステータスワード(FPSW)はありません。

【留意事項】

[レジスタ]ウィンドウにてレジスタ値を変更した場合、実際にレジスタに反映されるのはユーザプログラムの実行開始直前になります。REGISTER_SET コマンドによる変更も同様です。

[レジスタ]ウィンドウや REGISTER_SET コマンドによりレジスタ値を変更した後にユーザプログラムの実行をせず CPU リセットした場合は変更したレジスタ値は無効となります。

(2) 低消費電力状態

E1/E20エミュレータ使用時は、低消費電力状態の解除要因の他に、[Stop]ボタンによっても状態が解除され、ブレイクします。

(3) リセット信号

ユーザプログラム停止中に端子リセットによるリセットが発生した場合、JTAG接続ではリセット信号がマスクされます。なお、FINE接続ではマスクされずMCUがリセットされます。

【留意事項】

RES#、WAIT#端子が“Low”状態のままユーザプログラムをブレイクしないでください。タイムアウトエラーが発生します。また、ブレイク中に WAIT#端子が“Low”固定状態になると、メモリアクセス時にタイムアウトエラーが発生します。

(4) ダイレクトメモリアクセスコントローラ (DMAC)

DMACはブレイク中でも停止しません。このため、転送要求が発生すると、DMA転送を実行します。使用上の注意事項の「6.19.2 DMAC、DTCIに関して」も参照してください。

(5) WDTの使用について

ブレーク中は、WDTのカウントは停止します。プログラム実行再開時に、カウントも再開します。

(6) クロック発生回路関連レジスタの変更とデバッグ機能の競合

ユーザプログラム実行中にクロック発生回路の関連レジスタの値をメモリウィンドウ等から変更しないでください。クロック発生回路の関連レジスタについては、ご使用になるマイコンのハードウェアマニュアルを参照してください。

(7) ユーザ端子の占有

以下の端子はユーザポートとマルチプレクスになっているため、デバッグ時にはユーザが使用することができません。マルチプレクスになっているユーザポートについては、ご使用になるマイコンのハードウェアマニュアルを参照してください。

- TRST#
- TCK
- TMS
- TDI
- TDO
- FINEC

なお、RX621, RX62Nグループマイコンは、TDOが割り当てられている端子設定をオープンドレイン出力に設定すると、TDOとして使用しているにも関わらずオープンドレイン出力となります。デバッグ時は、P26をオープンドレイン出力に設定しないでください。

以下の端子はE20を使用して38ピンコネクタを経由してユーザシステムに接続する場合に、ユーザシステムで使用することができません。38ピン/14ピン変換アダプタを使用する場合は、ユーザポートとして使用できます。

- TRCLK
- TRSYNC
- TRDATA[3:0]

(8) マスカブル割り込みに関して

- ユーザプログラム停止中（ランタイムデバッグ中を含む）であっても、タイマなどの機能も動作しています。ユーザプログラム停止中（ランタイムデバッグ中を含む）は、エミュレータで割り込みを禁止しているため、マスカブル割り込みの要求が発生しても受け付けられません。この割り込み要求は、ユーザプログラムの実行を開始した直後に受け付けられます。
- ユーザプログラム停止中（ランタイムデバッグ中を含む）は、周辺I/Oの割り込み要求が受け付けられませんのでご注意ください。

3.3 デバッガの設定

エミュレータ接続時、[起動設定] ダイアログボックス および [コンフィグレーションプロパティ] ダイアログボックスが表示されます。ここで、エミュレータ全体に関わるオプションを選択してください。

デバッグの対象となるターゲットマイコン等は、起動時一度のみしか設定できませんのでご注意ください。

3.3.1 [起動設定] ダイアログボックス

[起動設定] ダイアログボックスは、ターゲットマイコンに関する設定を行うためのダイアログボックスです。設定した内容は、次回起動時にも有効です（但し、ホットプラグインおよび電源供給設定は除く）。

本ダイアログボックスは、エミュレータ起動後に、[基本設定 -> エミュレータ -> デバイスセッティング...] を選択して再オープンさせることができますが、起動後は設定を変更できません。

(1) エミュレータおよびターゲットマイコンに関する初期設定を行う

[起動設定] ダイアログボックスの[デバイス] ページで、エミュレーションするターゲットマイコンとエミュレーションモードを指定します。

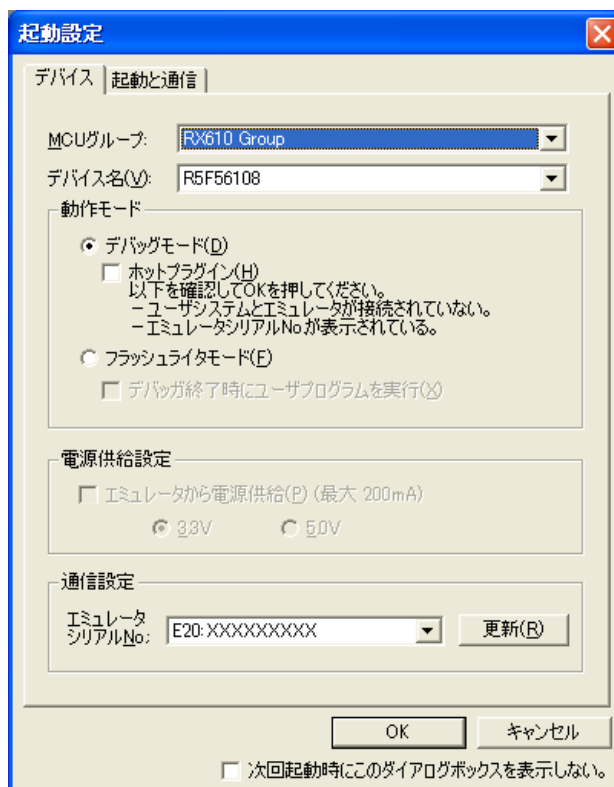


図3.1 [起動設定] ダイアログボックス - [デバイス] ページ

【留意事項】

[起動設定] ダイアログボックスで表示される内容は、マイコンごとに異なる場合があります。

ここで設定したターゲットマイコンは、エミュレータ起動後に変更することができません。

ターゲットマイコンを変更する場合、一度エミュレータの起動を解除し、再起動させる必要があります。

(a) ターゲットマイコンを選択する

エミュレーションするターゲットマイコンを選択します。[MCU グループ] プルダウンメニューよりターゲットグループを選択し、[デバイス名] プルダウンメニューよりターゲットマイコンの型名を選択してください。

なお、RX ファミリ向け E1/E20 エミュレータデバッグで選択可能なターゲットマイコンは以下です。

表3.5 ターゲットマイコン一覧表

MCU グループ	デバイス名	通信インタフェース
RX210 Group	R5F52103, R5F52104, R5F52105, R5F52106, R5F52107, R5F52108, R5F5210A, R5F5210B	FINE
RX21A Group	R5F521A6, R5F521A7, R5F521A8	FINE
RX220 Group	R5F52201, R5F52203, R5F52205, R5F52206	FINE
RX610 Group	R5F56104, R5F56106, R5F56107, R5F56108	JTAG
RX621 Group	R5F56216, R5F56217, R5F56218	JTAG
RX62N Group	R5F562N7, R5F562N8	JTAG
RX62T Group	R5F562T6, R5F562T7, R5F562TA	JTAG
RX62G Group	R5F562G7, R5F562GA	JTAG
RX630 Group	R5F56307, R5F56308, R5F5630A, R5F5630B, R5F5630D, R5F5630E	JTAG, FINE
RX631 Group	R5F5631A, R5F5631B, R5F5631D, R5F5631E, R5F5631M, R5F5631N, R5F5631P	JTAG, FINE
RX63N Group	R5F563NA, R5F563NB, R5F563ND, R5F563NE	JTAG, FINE
RX63T Group	R5F563T4, R5F563T5, R5F563T6	JTAG, FINE

(b) 動作モードを選択する

以下のオプションからエミュレーションモードを選択します。

“デバッグモード”、“フラッシュライターモード”

【デバッグモード】

E1/E20をデバッグとして使用する場合に選択します。そのため、本モードでプログラムのダウンロードを行った後に、E1/E20を外してユーザシステム単体で動作させることはできません。

また、本モードでは内蔵フラッシュROMの読み出しを禁止するためのIDコードをマイコンの内蔵フラッシュメモリに書き込むことはできません。

- ホットプラグインを選択する。

ホットプラグインのチェックボックスをチェックした場合、ユーザプログラム実行途中からデバッグを開始することが可能です。ホットプラグインについては、「4.13 ホットプラグイン機能」を参照してください。

【フラッシュライターモード】

E1/E20をライターとして使用する場合に選択します。そのため、本モードではプログラムのデバッグはできません。

また、内蔵フラッシュROMの読み出しを禁止するためのIDコードは本モードでプログラムのダウンロードを行ったときにマイコンの内蔵フラッシュメモリに書き込まれます。

チェックサムは以下の3領域に分け、その領域の全データをバイト単位で加算した値を表示します。

- データフラッシュ領域
- ユーザブート領域
- プログラムROM領域

(c) 電源供給設定を選択する

E1エミュレータからユーザシステムへの電源供給方法を指定します。E1エミュレータからユーザシステムに電源を供給する場合[エミュレータから電源供給 (最大200mA)] チェックボックスにチェックした後、“3.3V”および“5.0V”を選択します。

なお、選択可能な供給電源はターゲットマイコンによって異なります。マイコンの動作電圧については、関連するマイコンのハードウェアマニュアルを参照してください。

また、E20エミュレータは電源供給機能をサポートしていませんので本オプションは選択できません。

本製品からの供給電圧はホストマシンのUSB電源性能に依存するため、精度の保証ができません。本製品を使用して信頼性を要求する書き込みを行う場合は、本製品からの電源供給機能は使用せず、別途、マイコンの書き込み電圧に合致した電源をユーザシステムに供給してください。なお、量産工程などでの書き込みを行う場合は、FDT を使用してください。FDTの詳細については、<http://japan.renesas.com/fdt> を参照してください。

(d) 通信設定を選択する

USBの接続先を設定します。

エミュレータシリアルNo.リストボックスに、USBで接続されているエミュレータの固体識別情報一覧が表示されます。

[更新] ボタンをクリックすると、固体識別情報一覧がリフレッシュされます。

(2) 起動と通信に関する設定を行う

[起動設定] ダイアログボックスの[起動と通信] ページで、選択したターゲットマイコンの動作モードと通信インタフェースを設定します。

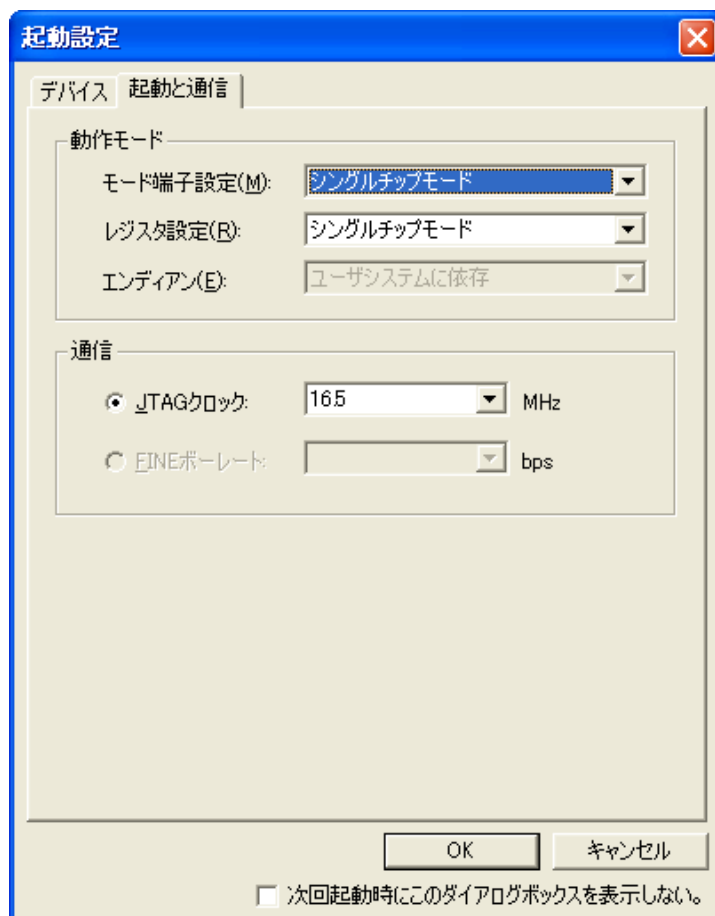


図3.2 [起動設定] ダイアログボックス - [起動と通信] ページ

ここで設定した情報は、エミュレータ起動後に変更することができません。
通信情報を変更する場合、一度エミュレータの起動を解除し、再起動させる必要があります。

(a) モード端子設定を選択する

モード端子による動作モードを以下のオプションから選択します。ご使用のユーザシステムの端子状態に合わせて選択してください。なお、選択可能なオプションはターゲットマイコンによって異なります。

“シングルチップモード”、“ユーザブートモード”

(b) レジスタ設定を選択する

レジスタ設定による動作モードを以下のオプションから選択します。起動後、ユーザプログラムで設定される動作モードに合わせて選択してください。なお、選択可能なオプションはターゲットマイコンによって異なります。

“シングルチップモード”、“内蔵ROM有効拡張モード”、“内蔵ROM無効拡張モード”

(c) エンディアンを選択する

エンディアンを以下のオプションから選択します。なお、MDE端子があるターゲットマイコンを選択した場合、オプションは非アクティブ(グレー)で“ユーザシステムに依存”と表示されます。

“リトルエンディアン”、“ビッグエンディアン”

(d) 通信インタフェースを選択する

通信インタフェースを以下のオプションから選択します。ご使用のユーザシステムに合わせて選択してください。なお、選択可能なオプションはターゲットマイコンによって異なりますので、「表3.5 ターゲットマイコン一覧表」を参照してください。

“JTAGクロック”、“FINEボーレート”

【JTAGクロック】

JTAGクロックのラジオボタンをチェックした場合、エミュレータとユーザシステム間のJTAG通信速度【注】を設定することが可能です。以下のオプション（単位：MHz）から選択します。

“16.5”、“12.38”、“6.188”、“3.094”、“1.547”

【注】 ユーザシステム上での JTAG 信号の配線長や配線方法によっては、選択した JTAG クロック周波数で通信ができない場合があります。この周波数を下げることによって、通信が出来るようになる場合があります。

【FINEボーレート】

FINEボーレートのラジオボタンをチェックした場合、エミュレータとユーザシステム間のFINE通信速度【注】を設定することが可能です。以下のオプション（単位：bps）から選択します。

“2000000”、“750000”、“500000”、“250000”

【注】 ユーザシステム上での FINE 信号の配線長や配線方法によっては、選択した FINE ボーレートで通信ができない場合があります。このボーレートを下げることによって、通信が出来るようになる場合があります。

3.3.2 [コンフィグレーションプロパティ] ダイアログボックス

[コンフィグレーションプロパティ] ダイアログボックスは、エミュレータおよびデバッグ機能に関する設定を行うためのダイアログボックスです。設定した内容は、次回起動時にも有効です。

本ダイアログボックスは、エミュレータ起動後に、[基本設定 -> エミュレータ -> システム] を選択して再オープンさせることができます。

一部のオプションは、起動後に設定を変更することができます。

変更可能なオプションはアクティブ表示、変更不可のオプションは非アクティブ（グレー）で設定内容の表示のみとなります。

(1) エミュレータおよびデバッグ機能に関する設定を行う

[コンフィグレーションプロパティ] ダイアログボックスの[MCU] ページで、ターゲットマイコンに関わる設定を行います。

[起動設定] ダイアログボックスの後に続けて表示されます。

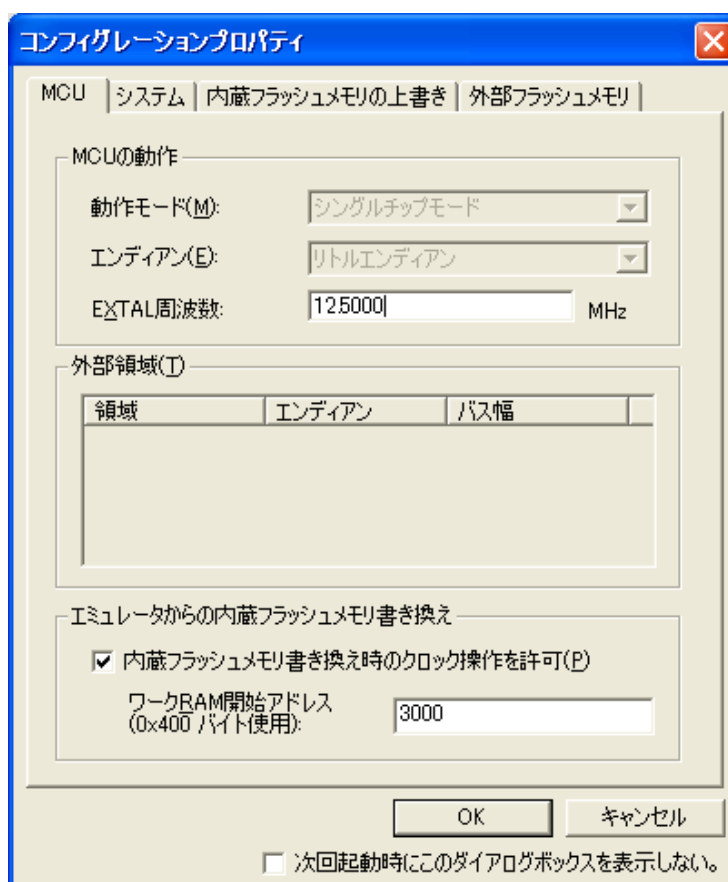


図3.3 [コンフィグレーションプロパティ] ダイアログボックス - [MCU] ページ

【留意事項】

[コンフィグレーションプロパティ] ダイアログボックスで表示される内容は、マイコンごとに異なる場合があります。

(a) 動作モードを表示する

[起動設定] ダイアログボックスの[起動と通信] ページで選択したレジスタ設定の動作モードが表示されます。

(b) エンディアンを表示する

[起動設定] ダイアログボックスの[起動と通信] ページで選択したエンディアン設定情報が表示されます。

なお、MDE端子があるターゲットマイコンを選択した場合は、E1/E20エミュレータがユーザシステムの端子状態を読み取り、そのエンディアン設定情報を表示します。

(c) EXTAL周波数を選択する

ターゲットマイコンに供給されている動作クロックを設定します。

入力値の範囲は、0.0001～99.9999です。この範囲外の値を入力した場合、エラーメッセージが表示されます。

なお、入力値の少数第5位以下は切捨てられます。

また、内蔵クロック発振のみを使用する場合は、EXTAL周波数の入力は不要です。この場合、[EXTAL周波数]チェックボックスのチェックを外してください。なお、内蔵クロック発振が未搭載のターゲットマイコン選択時にはチェックボックスは表示されません。

MCUの動作	
動作モード(M):	シングルチップモード
エンディアン(E):	リトルエンディアン
<input checked="" type="checkbox"/> EXTAL周波数:	125 MHz

図3.4 MCU の動作設定

(d) 外部領域を選択する

外部領域のエンディアンとバス幅を設定します。

レジスタ設定による動作モードが内蔵ROM有効拡張モード、内蔵ROM無効拡張モードの時に設定可能です。

外部領域の設定を行うラインを選択してダブルクリックすると [外部領域] ダイアログボックスが表示され領域毎にエンディアンとバス幅を設定することができます。選択可能なオプションはターゲットマイコンによって異なります。

なお、外部領域はターゲットマイコンによって異なります。マイコンの外部領域については、関連するマイコンのハードウェアマニュアルを参照してください。

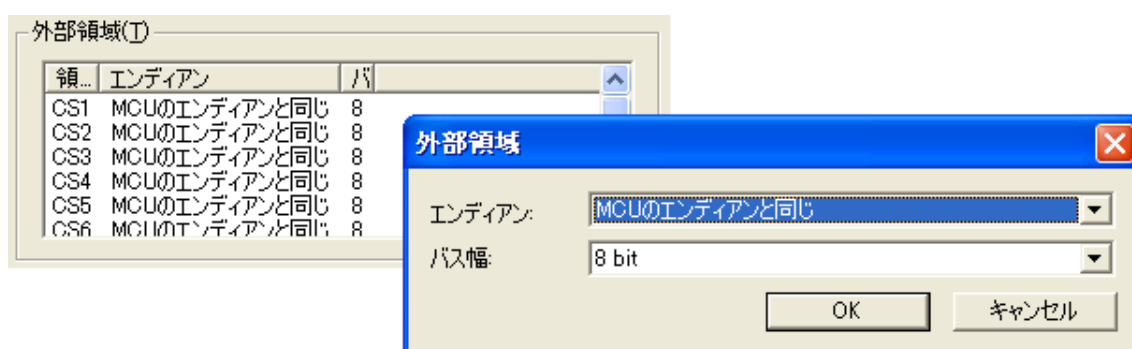


図3.5 外部領域の設定

(e) クロック操作を許可する

内蔵フラッシュメモリ書き換え時、E1/E20エミュレータからのクロック操作を許可します。

本チェックボックスをチェックした場合、内蔵フラッシュメモリ書き換え時のクロック設定がターゲットマイコンの動作保証範囲外であっても、E1/E20エミュレータが動作保証範囲内のクロック設定に変更して書き換えます。

なお、チェックを外した場合、クロック設定によっては正常に内蔵フラッシュメモリを書き換えできない可能性がありますので注意してください。使用上の注意事項の「6.19.8 クロック操作を許可する設定について」も参照してください。

(f) ワークRAMを選択する

デバッガが使用するワークRAMの配置アドレスを選択します。

指定したワークRAMの配置アドレスから指定された使用バイト分は、E1/E20エミュレータデバッガのファームウェアが使用します。メモリ内容の退避・復帰が行われるためユーザプログラムでもこの領域を使用できますが、この領域をDMAまたはDTC機能の転送元、および転送先には指定しないでください。

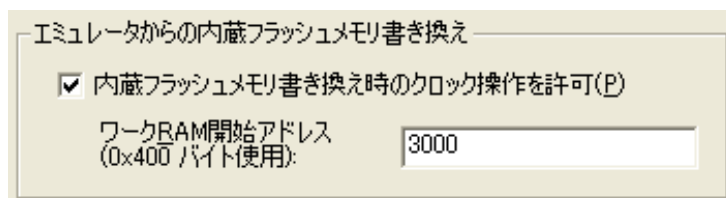


図3.6 エミュレータからの内蔵フラッシュ書き換え設定

(2) システムに関する設定を行う

[コンフィグレーションプロパティ] ダイアログボックスの[システム] ページで、エミュレータシステム全体に関わる設定を行います。

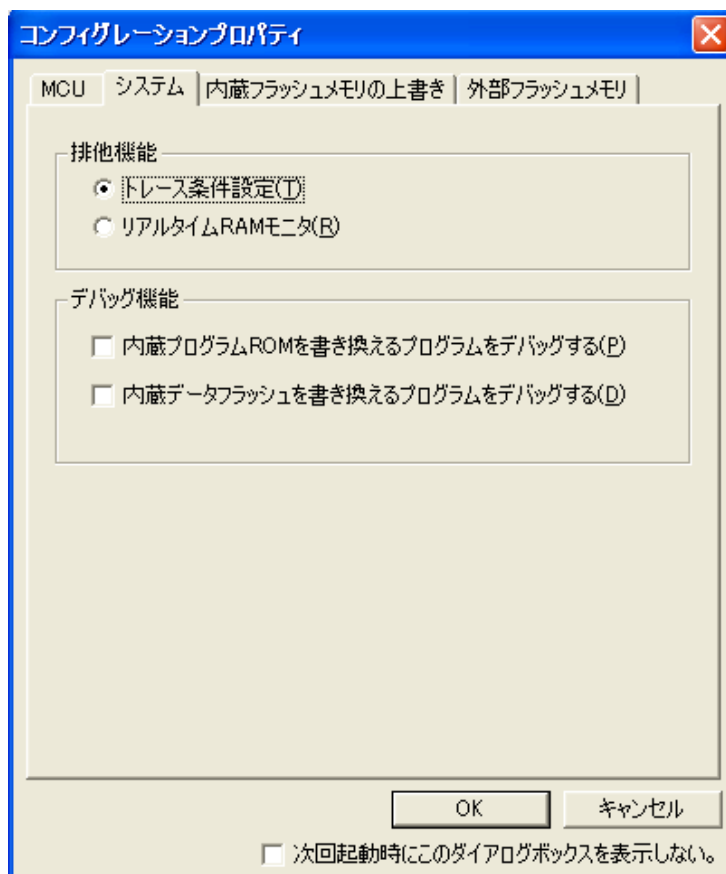


図3.7 [コンフィグレーションプロパティ] ダイアログボックス - [システム] ページ

(a) 排他機能を選択する

トレース条件設定機能、リアルタイムRAMモニタ機能は同時に使用することはできません。どれか一つを選択してください。

初期設定はトレース条件設定です。

なお、E1エミュレータを使用した場合、リアルタイムRAMモニタ機能を選択することはできません。(チェックボックスはグレー表示のままになります。)

(b) デバッグ機能を使用する

- 内蔵プログラムROMを書き換えるプログラムをデバッグする
内蔵プログラムROM領域の書き換えを伴うプログラムのデバッグを行う場合はチェックしてください。
- 内蔵データフラッシュを書き換えるプログラムをデバッグする
内蔵データフラッシュ領域の書き換えを伴うプログラムのデバッグを行う場合はチェックしてください。

(3) フラッシュROMの上書き設定を行う

[コンフィグレーションプロパティ] ダイアログボックスの[内蔵フラッシュメモリの上書き] ページで、フラッシュROMの上書き設定を行います。

起動後も[基本設定 -> エミュレータ -> システム] を選択して本ダイアログボックスを開き、設定を変更することができます。

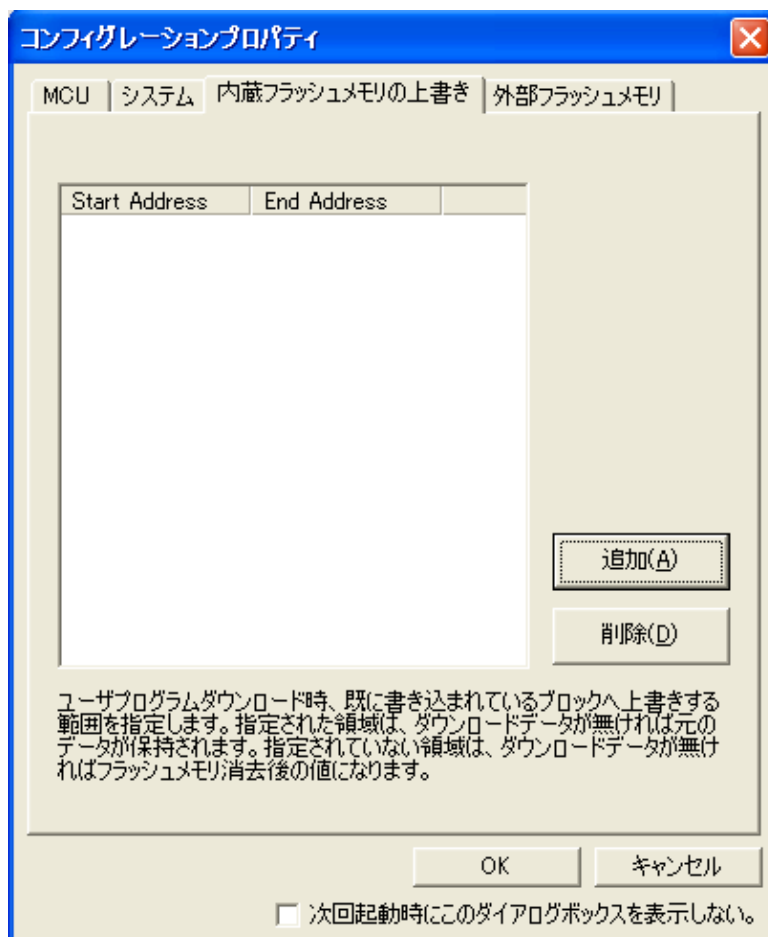


図3.8 [コンフィグレーションプロパティ] ダイアログボックス - [内蔵フラッシュメモリの上書き] ページ

内蔵フラッシュメモリのアドレス範囲を指定して登録します。

最大16個の範囲が設定可能です。

プログラムダウンロード時にダウンロードデータがない領域について、本設定で指定された領域はダウンロード前の値が保持され、指定されていない領域はフラッシュメモリ消去後の値となります。

(a) 内蔵フラッシュメモリの上書き範囲を登録する

[追加] ボタンをクリックすると、[アドレス範囲] ダイアログボックスが開きます。範囲を入力し[OK] ボタンをクリックすると、指定した範囲情報がリストに表示されます。

アドレスはブロック境界に自動的に補正され、リストに設定されます。

また、[起動設定] ダイアログボックスの[起動と通信] ページにて動作モードを“ユーザブートモード”に設定した場合、ユーザブート領域も登録可能です。

(b) 内蔵フラッシュメモリの上書き範囲を削除する

リストから削除する上書き範囲情報を選択し、[削除] ボタンをクリックしてください。また、[削除] ボタンの代わりにDeleteキーで削除することもできます。

(c) 内蔵フラッシュメモリの範囲を変更する

リストから上書き範囲情報を選択してダブルクリックすると、[アドレス範囲] ダイアログボックスが表示されます。

[アドレス範囲] ダイアログボックスで、開始アドレス及び終了アドレスの設定が変更できます。

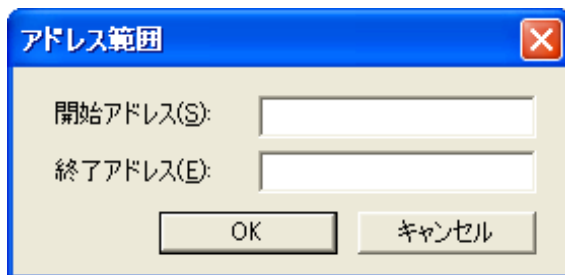


図3.9 [アドレス範囲] ダイアログボックス

(4) 外部フラッシュメモリ情報の登録を行う

[コンフィグレーションプロパティ] ダイアログボックスの[外部フラッシュメモリ] ページで外部フラッシュメモリ情報の登録を行います。マイコンの外部バスアドレス空間に接続されたフラッシュメモリ（外部フラッシュメモリと呼びます）にプログラムをダウンロードする場合に設定します。

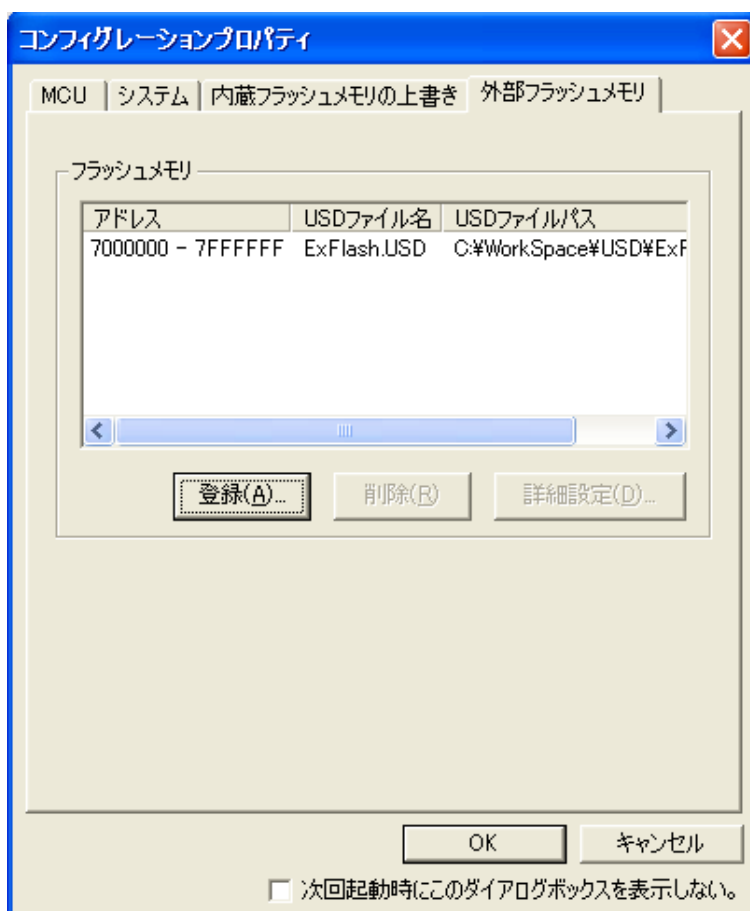


図3.10 [コンフィグレーションプロパティ] ダイアログボックス - [外部フラッシュメモリ] ページ

外部フラッシュメモリは、[外部フラッシュメモリ] ページに USD ファイル（ユーザシステムでの外部フラッシュメモリの接続形態などを定義したファイル）を登録することで認識されます。

最大 4 つの USD ファイルを登録できます。

USD ファイルは、「External Flash Definition Editor」で作成します。USD ファイルの作成方法については、下記 URL に掲載されている「External Flash Definition Editor」のマニュアルを参照してください。

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

(a) 外部フラッシュメモリ情報を登録する

[起動設定] ダイアログボックスの[起動と通信] ページにて動作モードを“内蔵ROM有効拡張モード”、または“内蔵ROM無効拡張モード”に選択することで、[外部フラッシュメモリ] ページの[登録] ボタンが有効となります。

[外部フラッシュメモリ] ページの[登録] ボタンをクリックすると、USDファイルのファイル選択ダイアログが開きます。ファイル選択ダイアログでUSDファイルを指定すると、指定したUSDファイルの情報がフラッシュメモリリストに表示されます。

(b) 外部フラッシュメモリ情報を削除する

フラッシュメモリリストから削除する外部フラッシュメモリ情報を選択し、[削除] ボタンをクリックしてください。

(c) 外部フラッシュメモリの詳細設定を行う

フラッシュメモリリストから詳細設定する外部フラッシュメモリ情報を選択し、[詳細設定] ボタンをクリックすると、[外部フラッシュメモリの詳細設定] ダイアログボックスが表示されます。[詳細設定] ボタンの代わりにダブルクリックでも表示されます。

[外部フラッシュメモリの詳細設定] ダイアログボックスで、外部フラッシュメモリのセクタ（ブロック）ごとの上書き設定を行います。

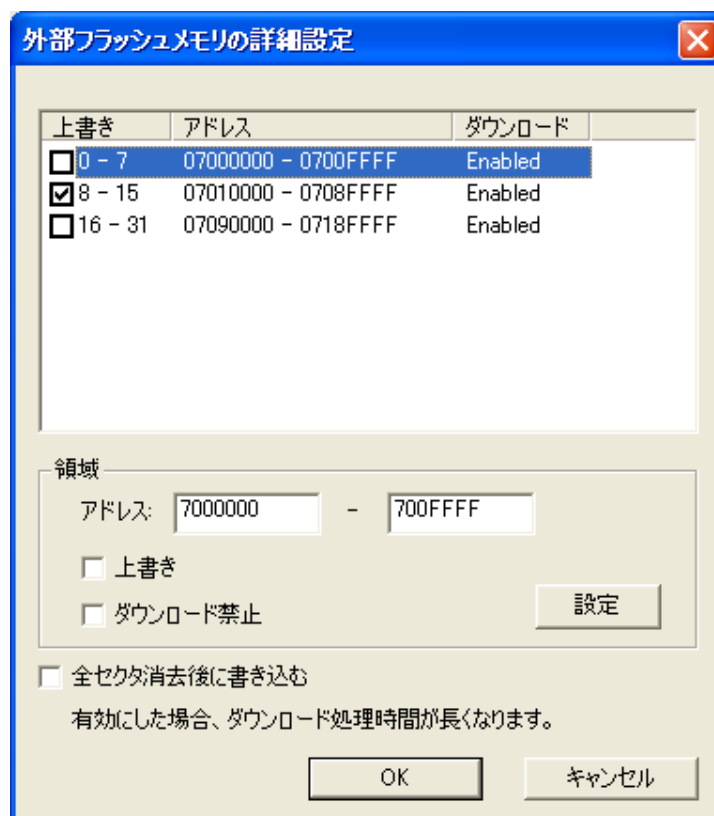


図3.11 [外部フラッシュメモリの詳細設定] ダイアログボックス

選択した外部フラッシュメモリ情報のセクタ毎の設定が、リストに自動表示されます。チェックボックスをチェックすると、選択されたセクタは、ダウンロード時に上書き（マージ）されます。

セクタごとに設定を変更する場合は、アドレス エディットボックスに設定を変更する開始アドレスと終了アドレスを入力し、[設定] ボタンをクリックしてください。その際、[上書き] にチェックをつけた場合、設定したセクタはダウンロード時に消去されずに上書きされます。リストの[ダウンロード] カラムには“Enabled”と表示されます。

また、[ダウンロード禁止] にチェックをつけた場合、設定したセクタへはダウンロードされません。リストの[ダウンロード] カラムには“Disabled”と表示されます。

なお、[全セクタ消去後に書き込む] にチェックすると、ダウンロード時に全セクタを消去します。

4. E1/E20 エミュレータ機能

本章では、E1/E20 エミュレータの機能を紹介します。
各機能の使用方法は、「5 チュートリアル」をご参照ください。

4.1 機能概要

E1/E20 エミュレータの機能概要を表 4.1 に示します。

表4.1 E1/E20エミュレータの機能

項番	項目	機能
1	ユーザプログラム 実行系機能	<ul style="list-style-type: none"> • マイコンが保証する範囲の動作周波数による、プログラム実行 • リセットエミュレーション • Step 機能 <ul style="list-style-type: none"> シングル Step 機能 (1Step : 1 命令) ソースレベル Step 機能 (1Step : ソース 1 行) Step Over 機能 (サブルーチン内はブレークしない) ソースレベル Step Over 機能 (サブルーチン内はブレークしない) Step Out 機能 (PC 実行中のサブルーチンの呼び出し元関数に戻るまで実行) • カーソル位置まで実行 • リセット後実行 • フリー実行 • ブレーク要因の表示
2	リセット機能	<ul style="list-style-type: none"> • ブレーク中、High-performance Embedded Workshop からマイコンへリセット発行
3	トレース機能 【注 2】	<ul style="list-style-type: none"> • 内蔵トレース機能 (マイコン内蔵のトレースバッファにより実現) • 外部トレース出力機能 <ul style="list-style-type: none"> トレース出力優先モード (リアルタイム性なし) CPU 実行優先モード (リアルタイム性あり、但し取得データの欠落 (ロスト) が発生する場合あり) • トレース内容 <ul style="list-style-type: none"> 分岐 (分岐元/分岐先) トレース、データトレース • 検索機能 • フィルタ機能 • バス/逆アセンブリ/C ソースレベルで表示
4	ブレーク機能 【注 2】	<ul style="list-style-type: none"> • S/W ブレーク機能 • オンチップブレーク機能 <ul style="list-style-type: none"> 実行前PCブレーク機能 イベントブレーク機能 (OR/AND (累積) /シーケンシャル/通過回数) トレースフルブレーク機能 • 強制ブレーク機能
5	パフォーマンス測定 機能【注 2】	<ul style="list-style-type: none"> • マイコン内蔵のカウンタにより、Point to Point の経過サイクル数などを測定する機能

項番	項目	機能
6	メモリアクセス機能 【注2】	<ul style="list-style-type: none"> • RAM へのダウンロード • 内蔵フラッシュメモリへのダウンロード • 外部フラッシュメモリへのダウンロード • 1行アセンブル • 逆アセンブル • メモリリード • メモリライト • ユーザプログラム実行中の変数内容の表示自動更新 • フィル • サーチ • ムーブ • メモリデータの保存/読み込み • メモリデータのベリファイ • 変数内容表示 • リアルタイム RAM モニタ • メモリデータの自動更新
7	汎用/制御レジスタ アクセス機能	汎用/制御レジスタのリード/ライト
8	内蔵 I/O レジスタ アクセス機能	内蔵 I/O レジスタのリード/ライト【注1】
9	ソースレベル デバッグ機能	豊富なソースレベルデバッグ機能
10	コマンドライン機能	コマンド入力をサポートしています。 各コマンドを入力順に羅列したファイルを作成すれば、バッチ処理を行うこともできます。
11	ヘルプ機能	各機能の操作方法や、コマンドラインウィンドウから入力できるコマンドのシンタックスを記載しています。
12	ホットプラグイン 機能	実行中のプログラムに E1/E20 を接続し、実行中のプログラム途中からデバッグを開始できる起動方式をサポートしています。

【注】 1. [IO]ウィンドウは、“xxxx.io”に定義されている内容を表示しています。“xxxx.io”の内容を編集することにより、表示するレジスタを追加／削除することができます。

“xxxx.io”に記載すべき内容については、High-performance Embedded Workshop ユーザーズマニュアル「リファレンス 6. I/O ファイルフォーマット」を参照してください。

また、“xxxx.io”は以下ディレクトリ内にあります。（xxxx はマイコングループ名を示します）

<High-performance Embedded Workshop フォルダ>

¥Tools¥Renesas¥DebugComp¥Platform¥E1E20¥E20RX¥IOFiles

2. ターゲットマイコンによっては、使用できない機能や仕様が異なるデバッグ機能があります。

4.2 ダウンロード機能

デバッグするロードモジュールをダウンロードします。

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

[注] プログラムをダウンロードする場合、ロードモジュールとして High-performance Embedded Workspace に登録する必要があります。

4.3 ソースファイルを開く

4.3.1 ソースコードを表示する

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

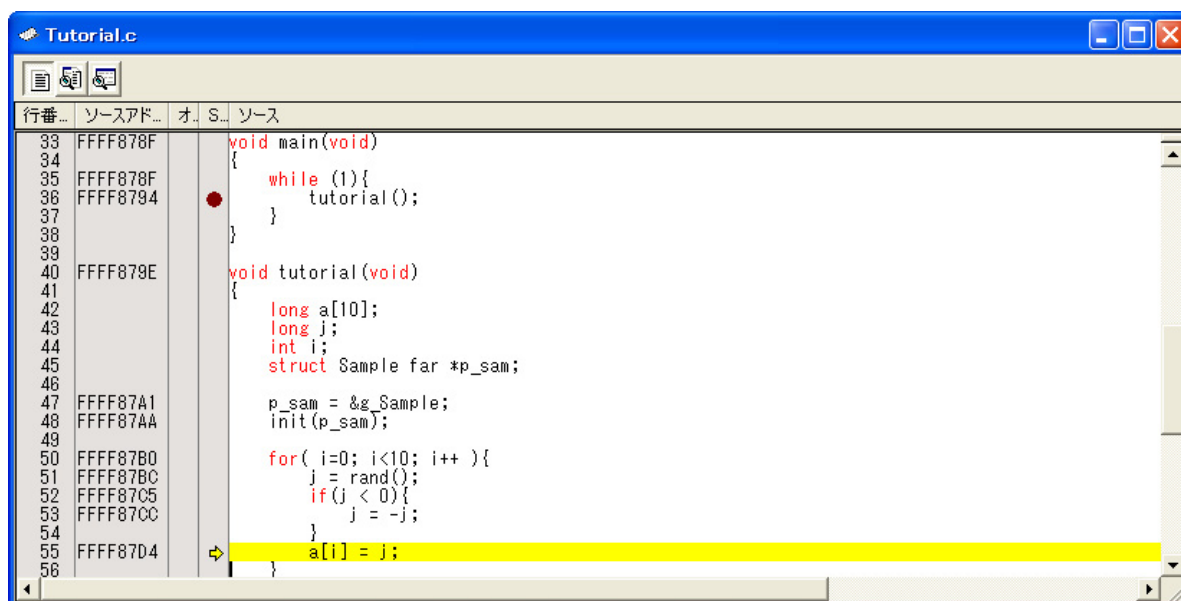


図4.1 [エディタ]ウィンドウ

本ウィンドウでは左端に行情報として下記を表示します。

1 列目	(ソースアドレスカラム)	アドレス情報
2 列目	(オンチップブレイクポイントカラム)	オンチップブレイクポイント情報
3 列目	(S/W ブレイクポイントカラム)	PC、ブックマーク、ブレイクポイント情報

ソースアドレスカラム

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

オンチップブレイクポイントカラム




オンチップブレイクポイントカラムには下記を表示します。

- オンチップブレイクポイントのアドレス条件を設定します。
設定可能なイベントの本数分設定可能です。
本数はマイコンによって異なります。

オンチップブレイクポイントカラムをダブルクリックすることによって、上記のビットマップが現れます。

S/W ブレイクポイントカラム

S/W ブレイクポイントカラムには下記を表示します。

-  ブックマークを設定している
-  PC Break を設定している
-  PC 位置

☞すべてのソースファイルでカラムをオフにするには

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

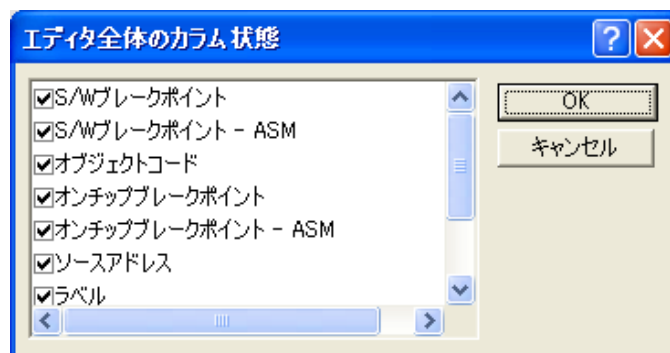



図4.2 [エディタ全体のカラム状態]ダイアログボックス

⇒1つのソースファイルでカラムをオフにするには


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

4.3.2 アセンブリ言語コードを表示する

ソースファイルが開いているときは、ウィンドウ上部の[逆アセンブリ]ツールバーボタンをクリックしてください。開いているソースファイルに対応するアセンブリ言語コードを表示します。

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

[逆アセンブリ]ウィンドウは現在のPCの場所で開きます。また、ディスアセンブルニモニック(可能なときはラベルも一緒に)を表示する[Address], [Code] (オプション)を表示します。

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

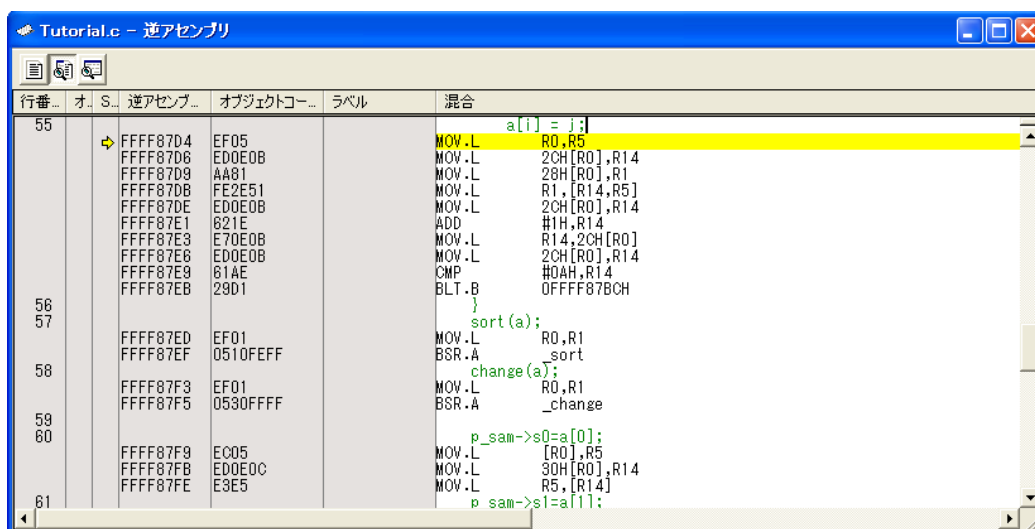


図4.3 [逆アセンブリ]ウィンドウ

4.3.3 アセンブリ言語コードを修正する

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

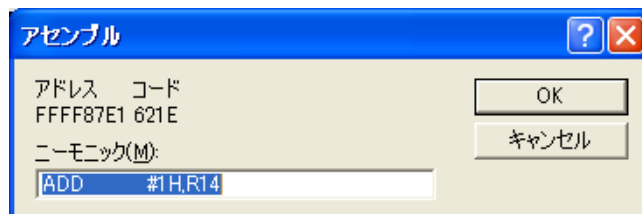


図4.4 [アセンブル]ダイアログボックス

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

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

4.4 メモリアクセス機能

E1/E20 エミュレータには以下のメモリアクセス機能があります。

4.4.1 メモリリード/ライト機能

- [メモリ]ウィンドウ

メモリ内容をウィンドウ表示します。

[メモリ]ウィンドウ OPEN 時に指定したサイズのみリードします。

エミュレータ内にキャッシュを持っていないため、常にリードサイクルが発生します。

また、[メモリ]ウィンドウからライトした場合は、ウィンドウの更新のために、[メモリ]ウィンドウで表示されている範囲のリードが発生します。

[メモリ]ウィンドウを更新したくない場合、ウィンドウ上で右クリックすることで表示されるオプションメニューの[表示固定]を選択すると更新しない設定にすることができます。

- me コマンド

コマンドライン機能です。

指定アドレスを指定サイズでリード、ライトする機能です。

(1) ユーザプログラムのダウンロード機能

ワークスペース内に登録されたロードモジュールをダウンロードできます。

[デバッグ]メニューの[ダウンロード]で、ダウンロードするモジュールを選択できます。

また、ワークスペース内のロードモジュールを右クリックすることによってポップアップメニューが開きますが、このポップアップメニューからもダウンロードを行うことができます。

ダウンロード先は、RAM またはフラッシュメモリです。

本機能では、デバッグ情報などソースレベルデバッグに必要な情報もダウンロードします。

(2) メモリデータのアップロード機能

指定アドレスから指定サイズ分、ファイルに保存することができます。

[メモリ]ウィンドウのポップアップメニューから[保存]を選択してください。

(3) メモリデータのダウンロード機能

ファイルに保存されているメモリ内容をダウンロードできます。

[メモリ]ウィンドウのポップアップメニューから[読み込み]を選択してください。

(4) 変数内容表示

ユーザプログラムの指定した変数の内容を表示します。

(5) リアルタイム RAM モニタ機能

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

(6) メモリデータの自動更新機能

ユーザプログラム実行中に[メモリ]ウィンドウに表示されているメモリ内容を自動更新することができます。更新間隔は変更することが可能です。

(7) そのほかのメモリ操作機能

その他、以下の機能があります。

- メモリフィル機能
- メモリムーブ機能
- メモリセーブ機能
- メモリベリファイ機能
- メモリサーチ機能
- 内蔵I/O表示機能
- ラベル名、変数名とその内容を表示する機能

詳細につきましてはオンラインヘルプを参照してください。

4.5 ブレーク機能

E1/E20 エミュレータには、強制ブレーク、S/W ブレーク機能、およびオンチップブレーク機能の3種類のブレーク機能があります。

4.5.1 強制ブレーク機能

ユーザプログラムを強制的にブレークする機能です。

4.5.2 S/W（ソフトウェア）ブレーク機能

指定アドレスの先頭命令をデバッグ専用の命令に置き換えてブレークする機能です。

S/W ブレークポイントを設定するとデバッグ専用の命令に置き換えるためにメモリへのライトが発生します（S/W ブレークポイントの設定を解除した場合もデバッグ専用の命令に置き換える前の命令に戻すため同じようにメモリへのライトが発生します）。

S/W ブレークポイントは、High-performance Embedded Workshop の[エディタ] ウィンドウや[逆アセンブリ] ウィンドウの[S/W ブレークポイント] カラムをダブルクリックすることで設定できます。

4.5.3 オンチップブレーク機能

オンチップブレーク機能には、実行前 PC ブレークとイベント（実行 PC イベント、オペランドアクセスイベント）を使用したイベントブレークの他にトレースフルブレークの3種類があります。

RX600 シリーズマイコンの場合、実行 PC イベントを使用したブレークとして8点、オペランドアクセスイベントを使用したブレークとして4点が設定でき、イベント組合せ[OR / AND (累積)] で最大8点まで設定が可能です。

RX200 シリーズマイコンの場合、実行 PC イベントを使用したブレークとして4点、オペランドアクセスイベントを使用したブレークとして2点が設定でき、イベント組合せ[OR / AND (累積)] で最大6点まで設定が可能です。

オンチップブレーク機能の使用方法については「4.7.1 オンチップブレークポイントを設定する」を参照してください。

(1) 実行前 PC ブレーク機能

実行PCイベントを使用して、指定されたアドレス (PC) の命令を実行する前にブレークする機能です。ユーザプログラム実行中も設定することができます。

実行前PCブレークは、High-performance Embedded Workshopの[エディタ]ウィンドウや[逆アセンブリ]ウィンドウの[オンチップブレークポイント] カラムをダブルクリックすることで設定できます。[オンチップブレークポイント] カラムについては「4.2 ダウンロード機能」を参照してください。

また、[オンチップブレーク条件設定] ダイアログボックスからも設定することができます。

(2) イベントブレーク機能

イベントの組み合わせが可能なブレーク機能で、イベント条件には命令フェッチ（実行PCイベントを使用）/データアクセス（オペランドアクセスイベントを使用）を設定することができます。イベントの組み合わせはOR/AND（累積）/シーケンシャル/通過回数の設定が可能です。ユーザプログラム実行中も設定することができます。

[オンチップブレーク条件設定]ダイアログボックスで設定することができます。

なお、RX200シリーズマイコンでは、通過回数の設定はできません。

(3) トレースフルブレーク機能

トレースバッファが満杯になるとブレークする機能です。ユーザプログラム実行中は設定変更できません。

以下にブレーク機能の一覧表を示します。

表4.2 ブレーク機能一覧

ブレーク機能	設定ポイント数		ブレーク条件	フラッシュの書き換え	プログラム実行中
	RX600 シリーズ マイコン	RX200 シリーズ マイコン			
S/W ブレーク	256 点	←	指定アドレス	あり	設定不可
実行前 PC ブレーク	8 点	4 点	指定アドレス	なし	設定可能
イベントブレーク (命令フェッチ)	8 点	4 点	指定アドレス	なし	設定可能
イベントブレーク (データアクセス)	4 点	2 点	データアクセス	なし	設定可能
トレースフルブレーク	—		トレースバッファ が満杯	なし	設定不可

4.6 イベント機能

4.6.1 イベントを使用する

イベントとは、プログラム実行中に起こる事象の組み合わせを指しています。

本デバッガでは、設定したイベントを、ブレーク機能、トレース機能、およびパフォーマンス機能の条件として使用することができます。

イベントの設定数や設定条件はマイコンによって異なります。各機能仕様については、「表 3.3 ターゲットマイコンに対する機能仕様一覧」を参照してください。

作成したイベントは、後で再使用するために登録しておくことができます。

(a) イベントのタイプ

イベントには、以下のタイプがあります。

表4.3 イベントタイプ一覧

実行アドレス	指定アドレスの命令を CPU が実行した場合に、イベントを検出します。 実行前 PC ブレークポイントに指定された場合、指定アドレスの命令実行直前にイベント成立となります。 それ以外の用途に指定された場合、指定アドレスの命令を実行後にイベント成立となります。
データアクセス	指定アドレス、または指定アドレス範囲にアクセスした場合にイベントを検出します。

(b) イベントの組み合わせ

2 つ以上のイベントを使用して、以下の組み合わせ条件を指定することができます。

表4.4 イベントの組み合わせ一覧

OR	指定されたイベントのうちいずれかひとつでも成立した時点で条件成立
AND (累積)	時間軸に関係なく、指定されたすべてのイベントが成立すると条件成立
シーケンシャル	指定された順序で、指定されたイベントが成立すると条件成立

4.6.2 イベントを追加する

以下のいずれかの方法で、イベントを追加してください。

- 各設定（オンチップブ레이크条件設定、トレース条件設定、およびパフォーマンス条件設定）ダイアログボックスによる追加
- [エディタ] ウィンドウのオンチップブ레이크ポイントカラムによる追加（オンチップブ레이크条件設定のみ）
- 他ウィンドウからのドラッグ&ドロップによる追加
- コマンドラインからの追加

(a) 各設定ダイアログボックスからイベントを追加する

ここでは、[オンチップブ레이크条件設定] ダイアログボックスからイベントを追加する方法を説明します。

- ① [表示 -> イベント -> オンチップブ레이크条件設定] を選択して、[オンチップブ레이크条件設定] ダイアログボックスを表示します。[オンチップブ레이크] ページ内で条件と組み合わせを設定した後、設定した内容に対応するページが表示されます。ここでは、[OR] ページを選択した場合のイベント追加方法について説明しますので、[オンチップブ레이크条件設定] ダイアログボックス内の [イベントブ레이크] チェックボックスにチェックを入れて”OR”を選択してください。OR以外にページとして表示される実行前PC、AND、およびシーケンシャルについては、「4.7.1 オンチップブ레이크ポイントを設定する」を参照してください。
[追加] ボタンをクリックするか、入力したいラインを選択してダブルクリックします。

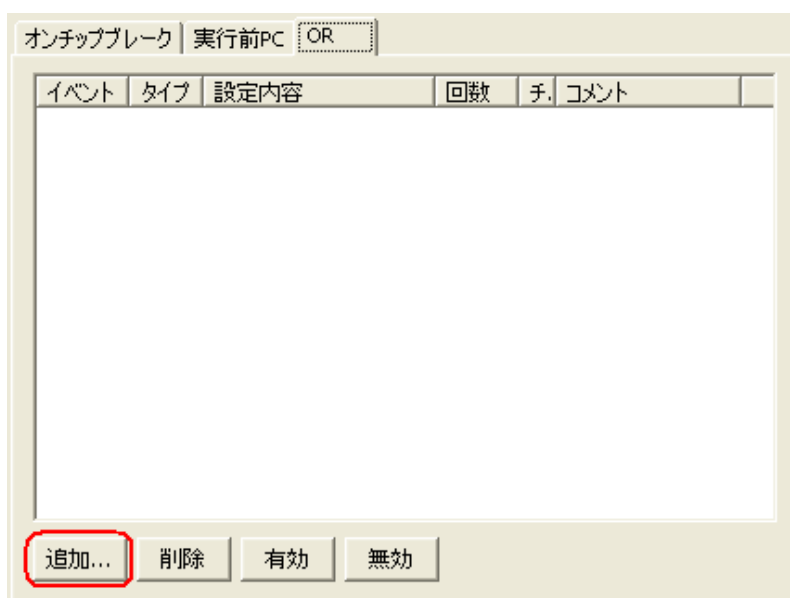


図4.5 [オンチップブ레이크条件設定] ダイアログボックス

- ② [イベント設定] ダイアログボックスが表示されるので、イベント詳細条件を設定して[OK]ボタンをクリックします。

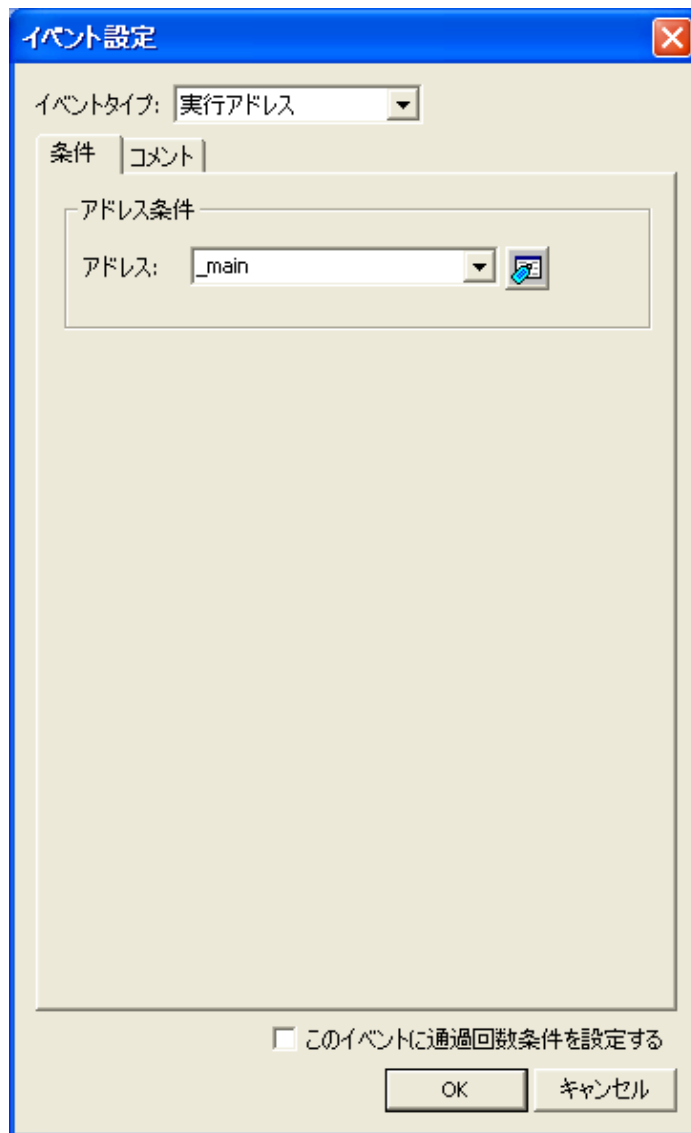


図4.6 イベント追加

ターゲットマイコンが RX600 シリーズマイコンの場合、イベントの通過回数条件（1-256 回）を設定できません。

イベントの通過回数を設定する場合は、[このイベントに通過回数条件を設定する] チェックボックスにチェックします。

[回数] ページが表示されるので、イベントの通過回数を設定して[OK] ボタンをクリックします。

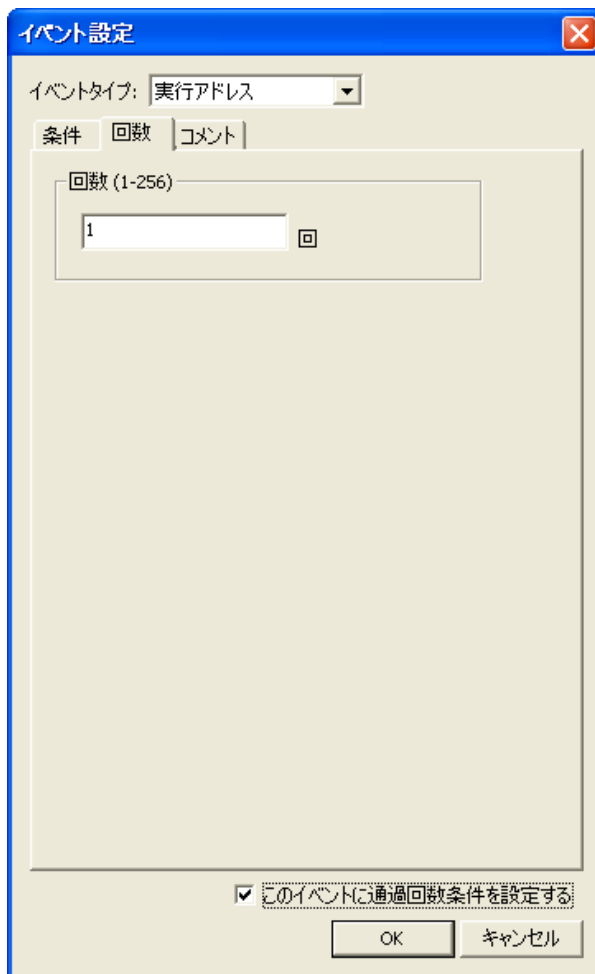


図4.7 通過回数の設定

③ 指定した位置に、イベントが追加されます。

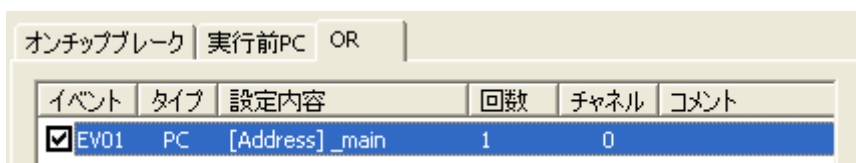


図4.8 イベント追加履歴

④ イベントを作成したときに、イベントが設定可能な最大ポイント数を超えた場合は、イベントは無効な状態で追加されます。

(b) [エディタ] ウィンドウのオンチップブレイクポイントカラムからイベントを追加する

【オンチップブレイクポイントを追加する場合】

- ① [エディタ] ウィンドウのオンチップブレイクポイントカラムをダブルクリックすると、指定したアドレスへの実行を条件としたオンチップブレイクポイントを設定することができます。
⇒ 実行アドレス条件

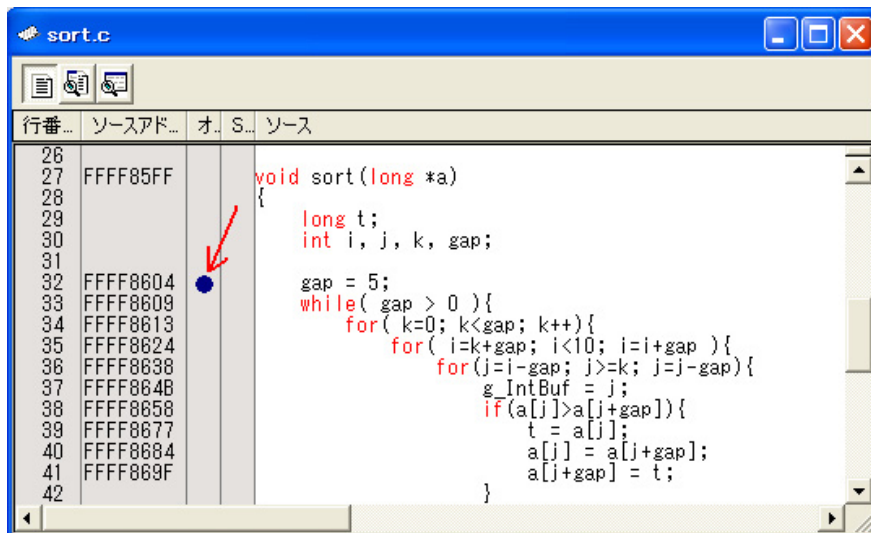


図4.9 [エディタ] ウィンドウ

【注】 [オンチップブレイク条件設定] ダイアログボックスで設定変更後に[適用]ボタンをクリックしないで、設定内容を有効にしていない（タイトルバーの最後に「*」が付いている）場合、[エディタ] ウィンドウのオンチップブレイクポイントカラムでオンチップブレイクポイントを設定できません。

(c) ドラッグ&ドロップでイベントを追加する

【[エディタ] ウィンドウの変数名、関数名をドラッグ&ドロップする場合】

- ② 変数名を[オンチップブレーク条件設定] ダイアログボックス内のイベント設定欄にドラッグ&ドロップすると、その変数へのアクセスを条件としたイベントを設定できます。
⇒ データアクセス条件

このとき、変数のサイズを自動的にデータアクセスイベントの条件に設定します。

イベントとして登録できる変数は、グローバル変数、スタティック変数で、1バイト、2バイトあるいは4バイトのサイズの変数のみです。関数内のスタティック変数はイベントとして登録できません。

また、条件と組み合わせ設定に実行前 PC ブレーク ([実行前 PC] ページ) を選択している場合、イベントタイプに実行アドレス条件しか設定できないため、ドラッグ&ドロップによる変数の登録はできません。

- ③ 関数名を[オンチップブレーク条件設定] ダイアログボックス内のイベント設定欄にドラッグ&ドロップすると、その関数の先頭アドレスへの実行アドレス条件としてイベントを設定できます。
⇒ 実行アドレス条件

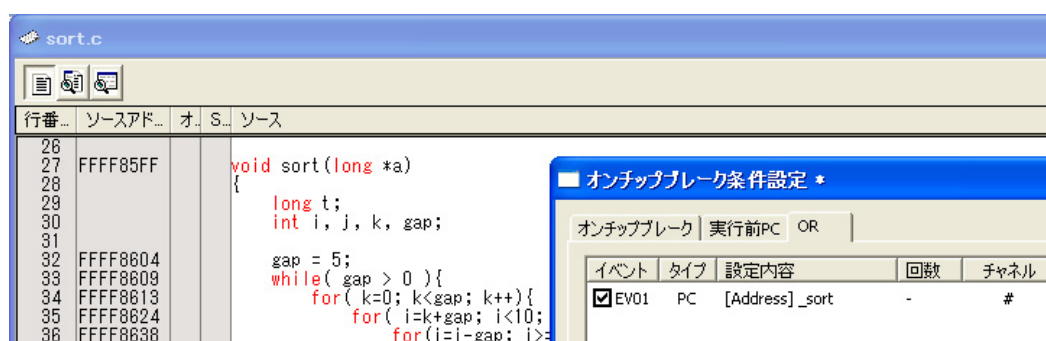


図4.10 ドラッグ&ドロップによるイベント追加

【[メモリ] ウィンドウのアドレス範囲をドラッグ&ドロップする場合】

RX600 シリーズマイコンの場合、[メモリ] ウィンドウのメモリ内容を選択して[オンチップブレーク条件設定] ダイアログボックス内のイベント設定欄にドラッグ&ドロップすると、選択したメモリ内容のアドレス範囲を条件としてデータアクセスイベントを設定できます。

⇒ データアクセス条件

条件と組み合わせ設定に実行前 PC ブレーク ([実行前 PC] ページ) を選択している場合、イベントタイプに実行アドレス条件しか設定できないため、[メモリ] ウィンドウのアドレス範囲をドラッグ&ドロップしてイベントを登録することはできません。

【[ラベル] ウィンドウのラベルをドラッグ&ドロップする場合】

[ラベル] ウィンドウのラベルを選択して[オンチップブレーク条件設定] ダイアログボックス内のイベント設定欄にドラッグ&ドロップすると、選択したラベルへの実行を条件としたイベントを設定します。

⇒ 実行アドレス条件

4.6.3 イベントを削除する

以下の方法で、イベントを削除してください。

- 各設定（オンチップブレイク条件設定、トレース条件設定、およびパフォーマンス条件設定）ダイアログボックスによる削除

(a) 各設定ダイアログボックスからイベントを削除する

ここでは、[オンチップブレイク条件設定] ダイアログボックスからイベントを削除する方法を説明します。

- ① 1点を削除する場合、[オンチップブレイク条件設定] ダイアログボックス内のイベント設定欄（[OR] ページ内）で削除したい行を選択して[削除] ボタンをクリックします。（[削除] ボタンの代わりに Ctrl+Del キーで削除することもできます）
選択したイベントが、イベント設定欄から削除されます。

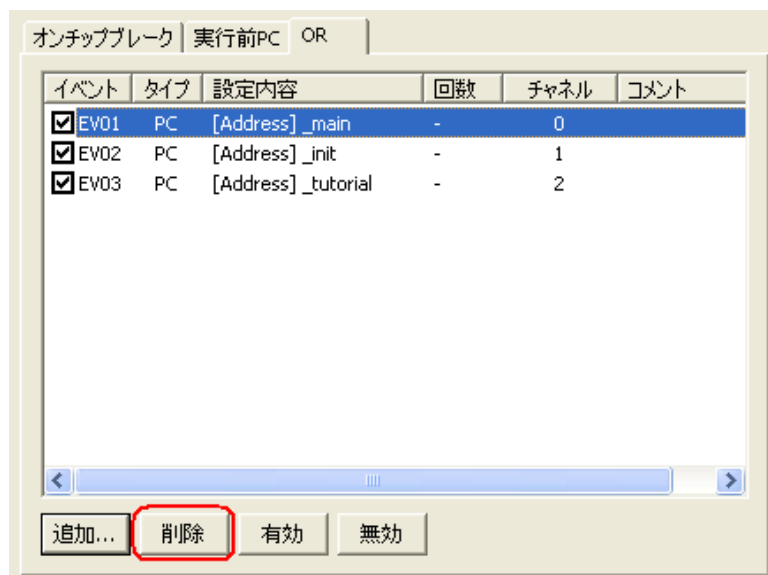


図4.11 イベントの削除

- ② 複数のイベントを削除する場合、[オンチップブ레이크条件設定] ダイアログボックス内のイベント設定欄で削除したい行を Shift キーまたは Ctrl キーを押しながら選択して、[削除] ボタンをクリックします。([削除] ボタンの代わりに Ctrl+Del キーで削除することもできます)
選択したイベントが、イベント設定欄から削除されます。

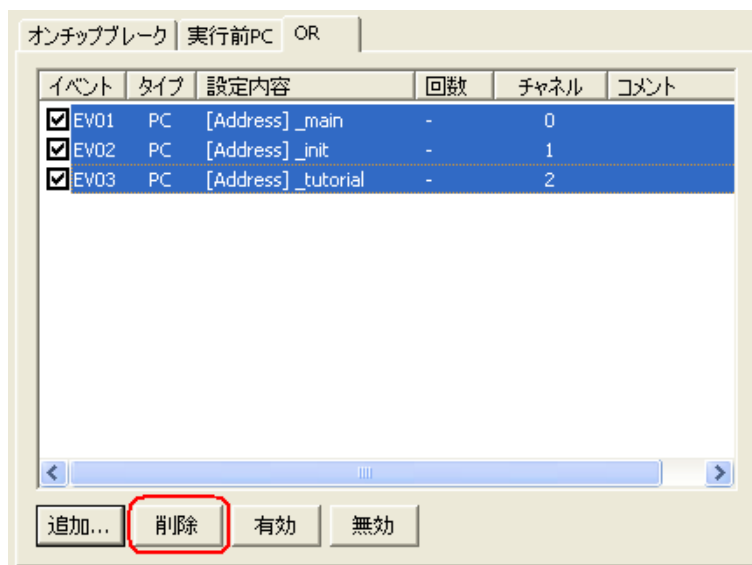


図4.12 複数イベントの削除

4.6.4 イベントを登録する

イベントの登録とは、[登録イベント一覧] ダイアログボックスにイベントを入れることです。

登録したイベントは、後で再使用（Re-use）することができます。

以下のいずれかの方法で、イベントを登録してください。イベントは 256 個まで登録できます。

- 各設定（オンチップブ레이크条件設定、トレース条件設定、およびパフォーマンス条件設定）ダイアログボックスによる登録
- 各設定ダイアログボックスからのドラッグ&ドロップによる追加
- [登録イベント一覧] ダイアログボックスによる登録

(a) 各設定ダイアログボックスからイベントを登録する

ここでは、[オンチップブ레이크条件設定] ダイアログボックスからイベントの登録方法を説明します。

- ① [オンチップブ레이크条件設定] ダイアログボックス内のイベント設定欄 ([OR] ページ内) の[追加] ボタンをクリックして[イベント設定] ダイアログボックスを開きます。

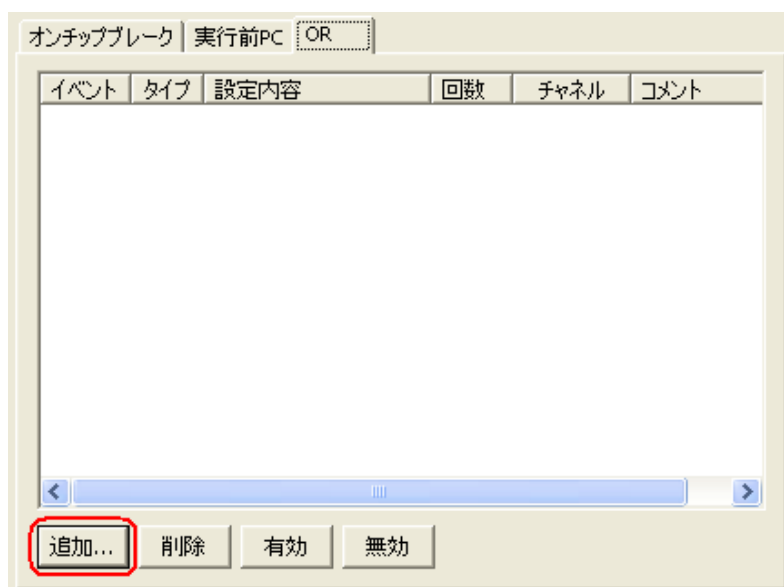


図4.13 [オンチップブ레이크条件設定] ダイアログボックス

[イベント設定] ダイアログボックスで [コメント] ページを表示して、[このイベントを登録する] チェックボックスをチェックしてください。

必要に応じてコメントを付けてください。

[登録イベント一覧] ダイアログボックスで登録内容およびコメントを確認することができます。

[OK] ボタンをクリックします。

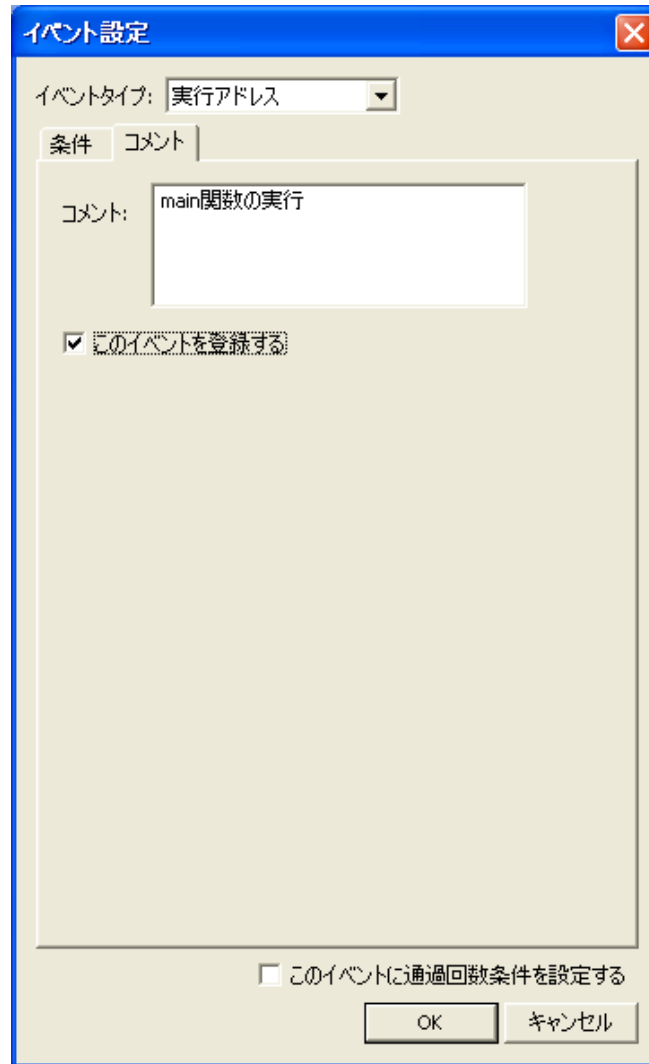


図4.14 イベントの登録

- ② 指定した位置にイベントが追加され、かつ、[登録イベント一覧] ダイアログボックスにも登録されます。

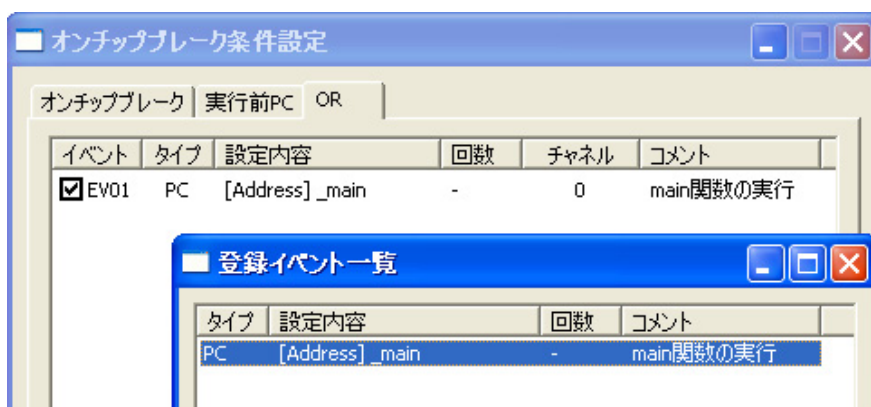


図4.15 イベントの登録履歴

[登録イベント一覧] ダイアログボックスは、[オンチップブレイク条件設定] ダイアログボックスの下方にある[登録イベント一覧...] ボタンをクリックすると開くことができます。

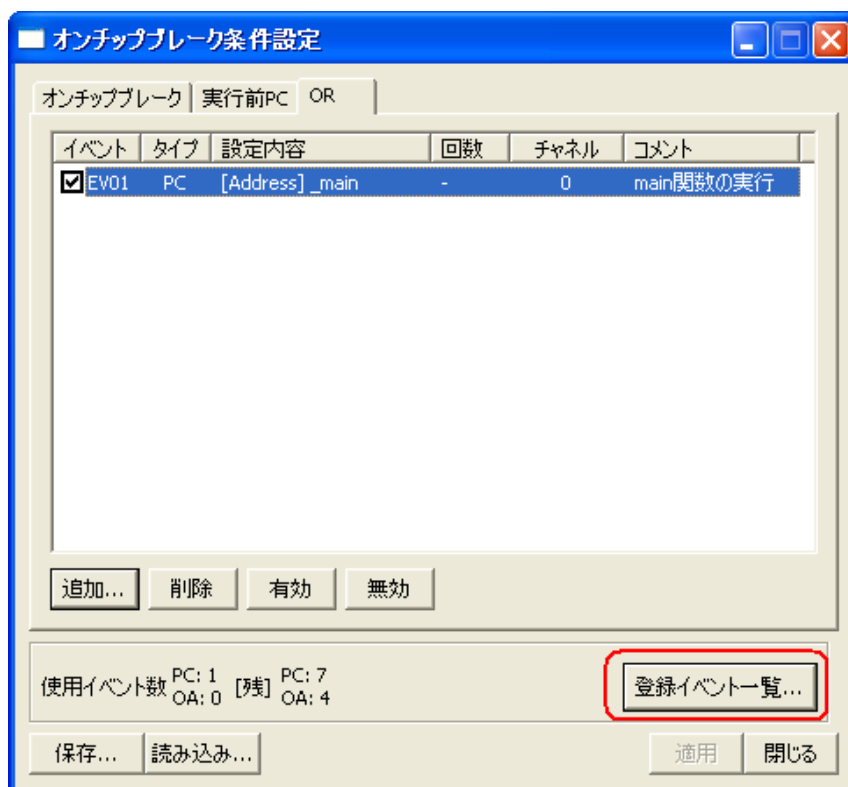


図4.16 [登録イベント一覧] ダイアログボックスの表示方法

(b) ドラッグ&ドロップでイベントを登録する

作成したイベントを[オンチップブレイク条件設定] ダイアログボックス内のイベント設定欄から[登録イベント一覧] ダイアログボックスにドラッグ&ドロップして登録することもできます。

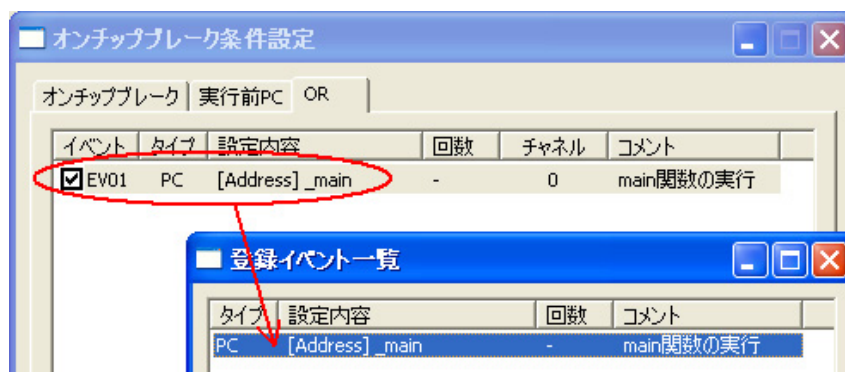


図4.17 ドラッグ&ドロップによるイベント登録

(c) [登録イベント一覧] ダイアログボックスからイベントを登録する

[追加...] ボタンをクリックすると、[イベント設定] ダイアログボックスが表示されるのでイベントを作成します。

ここから作成したイベントは[登録イベント一覧] ダイアログボックスに追加されます。

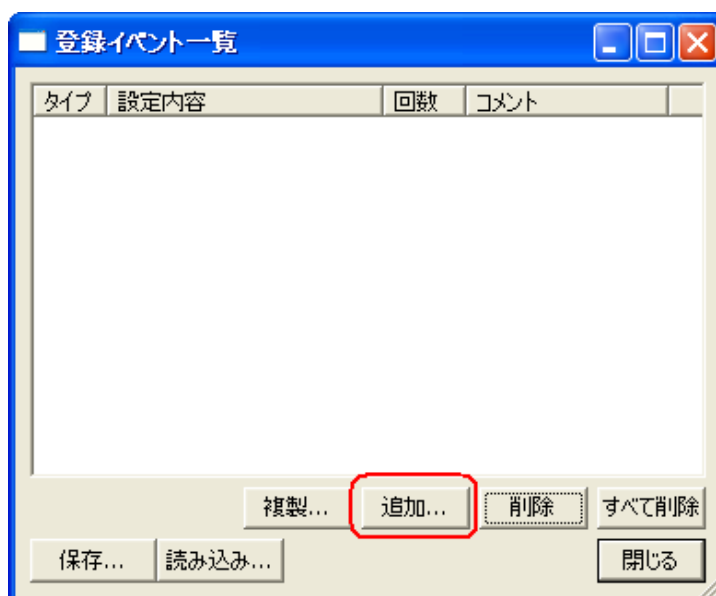


図4.18 [登録イベント一覧] ダイアログボックス

4.6.5 登録したイベントの削除

[オンチップブ레이크条件設定] ダイアログボックスの下方にある[登録イベント一覧...] ボタンをクリックして[登録イベント一覧] ダイアログボックスを開いてください。

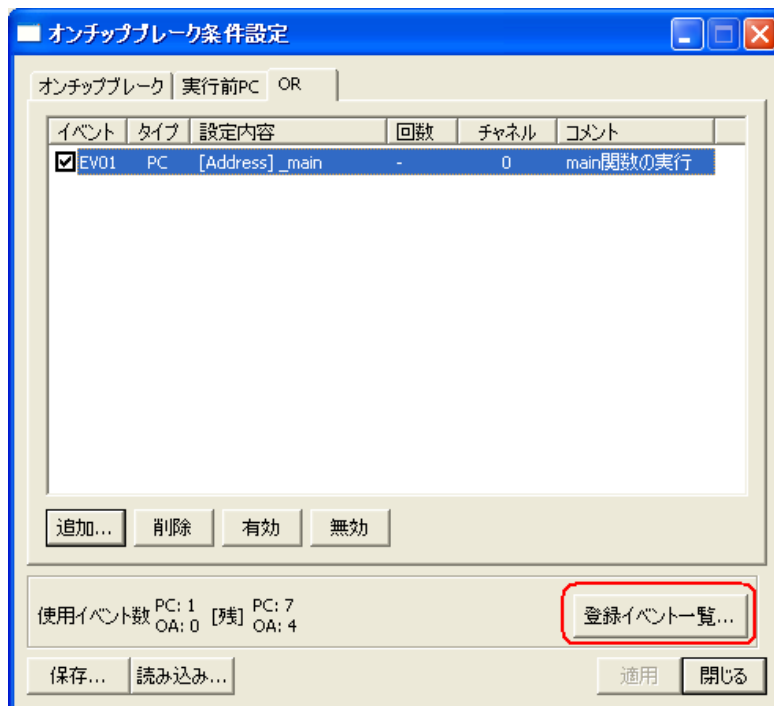


図4.19 [登録イベント一覧] ダイアログボックスの表示方法

1点を削除する場合、[登録イベント一覧] ダイアログボックスで削除したい行を選択して[削除] ボタンをクリックしてください。([削除] ボタンの代わりに Ctrl+Del キーで削除することもできます) 選択したイベントが、[登録イベント一覧] ダイアログボックスから削除されます。すべてのイベントを削除する場合、[すべて削除] ボタンをクリックしてください。

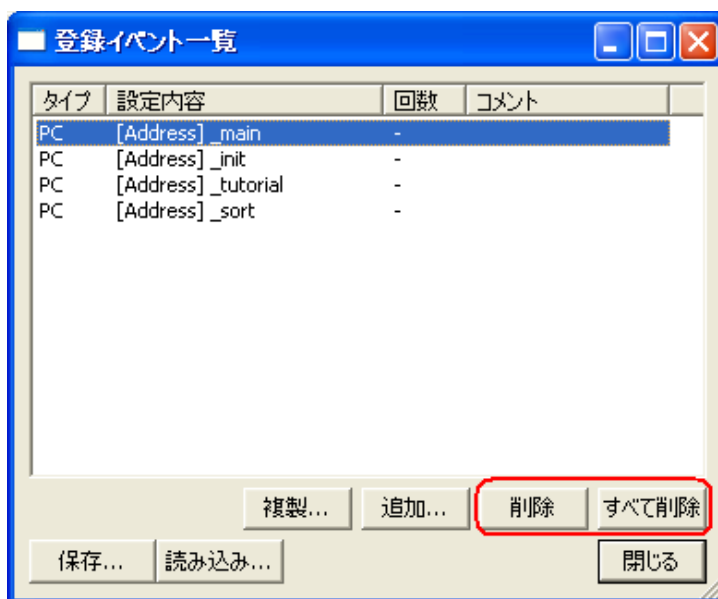


図4.20 登録イベントの削除

4.6.6 イベントを都度入力する／再使用する

各機能へイベントを設定する方法は2通りあります。

一つ目は、それぞれの設定ダイアログボックス上でその都度イベントを作成する方法です。

二つ目は、[登録イベント一覧] ダイアログボックスの中から使用したい条件をひとつ選び、設定したい条件エリアにドラッグ&ドロップする方法です。

前者を 都度入力、後者を 再使用 (Re-use) と呼びます。

- 都度入力

一度しか使用しない条件です。

作成したイベントを「登録なし」で使用します。

イベント使用后 (変更、削除を行った場合) に、設定は残りません。

[エディタ] ウィンドウのカラムダブルクリック等で簡易作成されたイベントは、都度入力となります。

- 再使用

[登録イベント一覧] ダイアログボックスに登録されているイベントは、各機能の条件設定エリアにドラッグ&ドロップすることで、再使用することができます。

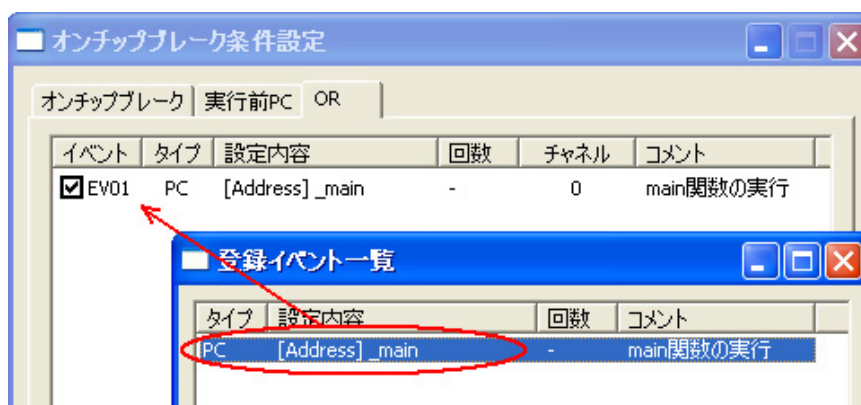


図4.21 イベント再使用の概略図

(a) 複数機能へのドラッグ&ドロップ

[登録イベント一覧] ダイアログボックスのひとつのイベントを、複数の機能へドラッグ&ドロップすることが可能です。

ドラッグ&ドロップ後にイベント内容を変更した場合、[登録イベント一覧] ダイアログボックス側には反映されません。

(b) [登録イベント一覧] ダイアログボックスへの重複登録

同一設定内容のイベントの場合でも、重複して登録することが可能です。

4.6.7 イベントを適用させる

イベントを作成した後、設定を有効にするには [適用] ボタンをクリックします。

[適用] ボタンがクリックされるまで、設定内容は有効になりません。

[オンチップブレーク条件設定] ダイアログボックス、[トレース条件設定] ダイアログボックス、[パフォーマンス条件設定] ダイアログボックスのうち、いずれかのタイトルバーの最後に「*」が付いている場合は、設定が編集集中であることを示しています。編集集中は、[エディタ] ウィンドウのイベントカラム、またはコマンドラインから、設定変更を行うことはできません。

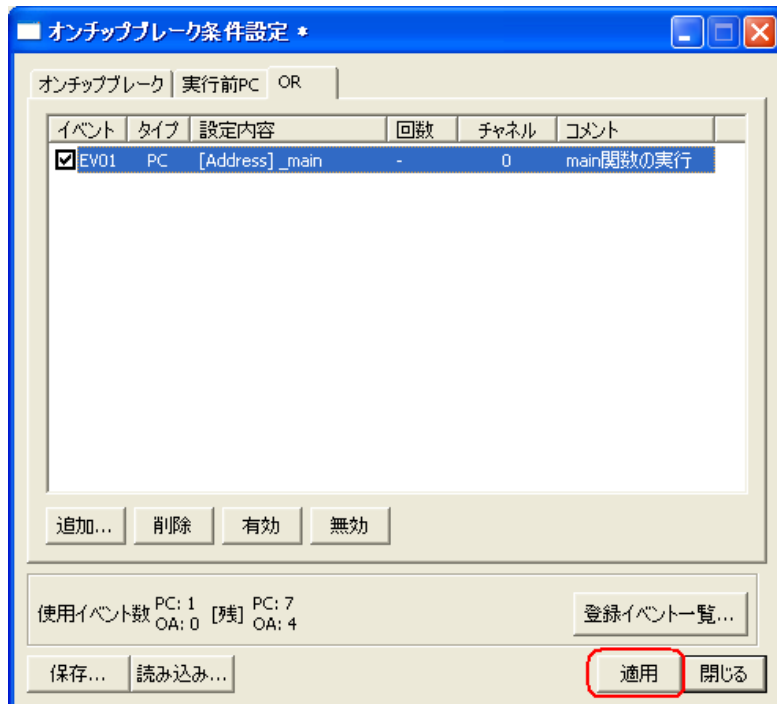


図4.22 設定内容の適用

4.7 オンチップブレーク機能

オンチップブレークは、設定されたイベントを検出した時点で、ユーザプログラムの実行を停止させる機能です。

4.7.1 オンチップブレークポイントを設定する

(a) オンチップブレークポイントを設定する

オンチップブレークポイントは、実行前PC、イベント組み合わせ（OR、AND（累積）、シーケンシャル）および その他の条件を設定できます。

実行前PC、イベント組み合わせ（OR、AND（累積）、シーケンシャル）およびその他の条件はすべて同時に設定することもできますし、どれかひとつに限定することもできます。

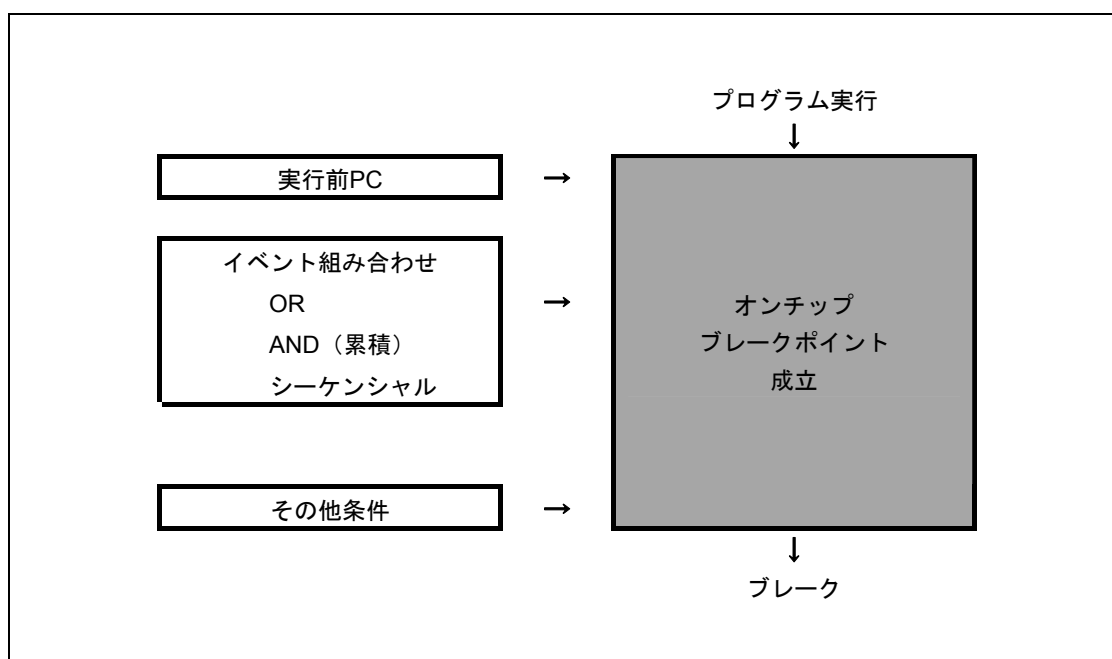


図4.23 オンチップブレーク概要

オンチップブレークポイントを設定する場合は、[表示 -> イベント -> オンチップブレーク条件設定] を選択して[オンチップブレーク条件設定] ダイアログボックスを開いてください。

(b) 実行前PCブレークを設定する

実行前PCブレークは常に有効になっています。

実行前PCブレークを無効にしたい場合、登録されているイベントを削除または、無効にします。

実行前PCブレークは、イベントに指定したアドレスの命令が実行される直前にブレークします。

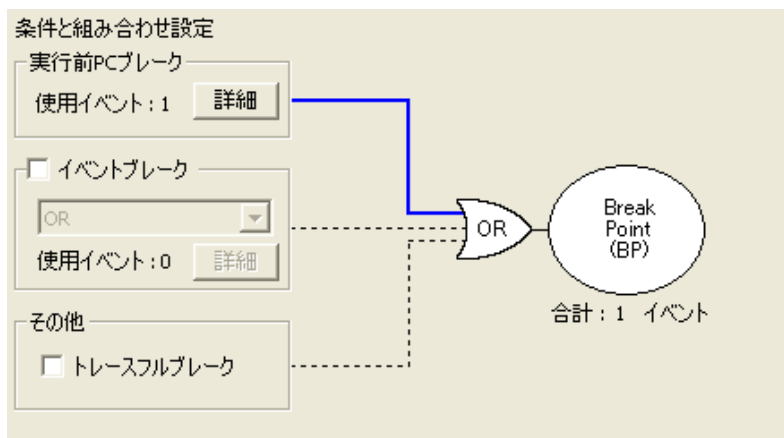


図4.24 [オンチップブレーク条件設定] ダイアログボックス_実行前 PC ブレーク

実行前 PC ブレーク欄の[詳細] ボタンをクリックするとイベント設定欄 ([実行前 PC] ページ) が表示されます。

[追加] ボタンをクリックすると[イベント設定] ダイアログボックスが表示され、実行前 PC ブレークのイベント（実行アドレス条件）を指定することができます（ラベルによる指定も可能です）。
（実行前 PC ブレークのイベントタイプは”実行アドレス”のみです。）

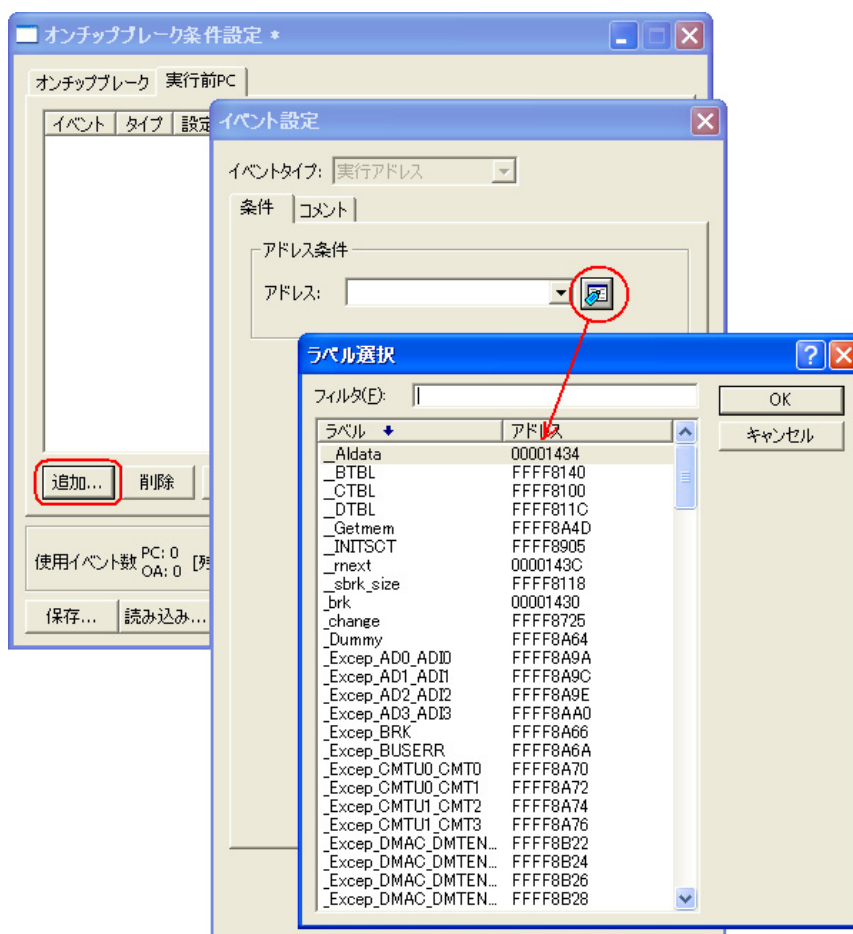


図4.25 実行前 PC ブレークの設定

タイトルバーの最後に「*」が付いている場合は、設定が有効になっていません。イベント設定後に[適用] ボタンをクリックして設定を有効にしてください（設定が有効になると「*」が消えます）。

(c) イベントブレークを設定する

[OR]、[AND(累積)]、[シーケンシャル] のうち、どれかひとつを選択できます。

イベントブレークを有効にしたい場合、イベントブレークの左にあるチェックボックスにチェックをつけます。イベントブレークを無効にしたい場合、イベントブレークの左にあるチェックをはずします。

デフォルトは、イベントブレーク無効です。

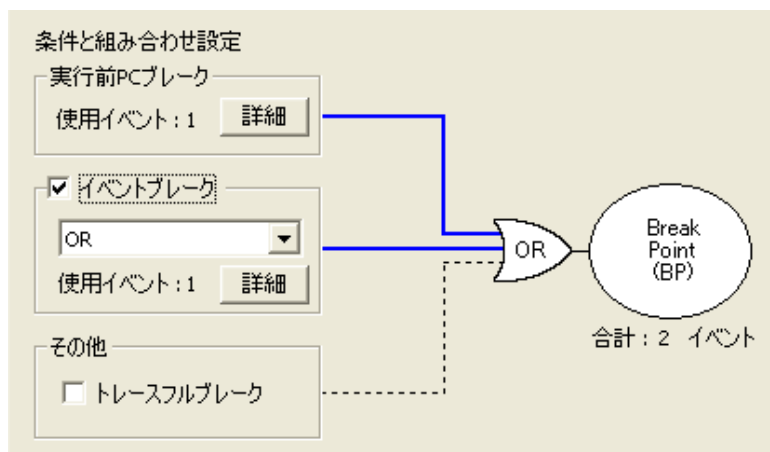


図4.26 [オンチップブレーク条件設定] ダイアログボックス_イベントブレーク

表4.5 イベントブレーク項目一覧

	種類	内容
1	OR	設定されたイベントのうちどれか一つが成立するとブレークポイントが成立します。
2	AND (累積)	時間軸に関係なく、設定されているすべてのイベントが成立すると、ブレークポイントが成立します。
3	シーケンシャル	設定されたイベントが指定された順序で成立すると、ブレークポイントが成立します。 RX600 シリーズマイコン： 7 段階（順方向）+ リセットポイント(R) 【注】 RX200 シリーズマイコン： 3 段階（順方向）+ リセットポイント(R) 【注】

【注】 リセットポイント(R)：リセットポイントに設定したイベントが成立した場合、その時点までの成立したイベントが全てクリアされます。

各条件のリストに表示されているイベントは、Ctrl + Del キーで削除することができます。

イベントブレーク欄の[詳細] ボタンをクリックするとイベント設定欄（OR を選択した場合は[OR] ページ）が表示されます。

[追加] ボタンをクリックすると[イベント設定] ダイアログボックスが表示され、イベントブレークのイベント（実行アドレス条件、またはデータアクセス条件）を指定することができます（アドレス条件はラベルによる指定も可能です）。

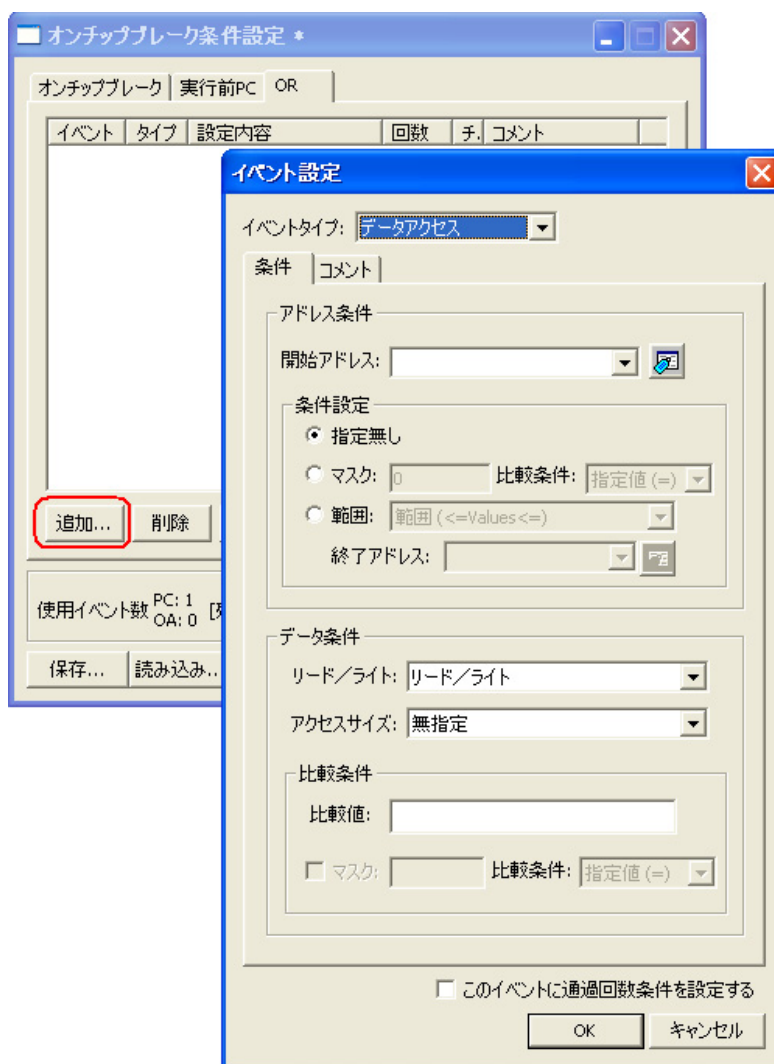


図4.27 イベントブレークの設定

タイトルバーの最後に「*」が付いている場合は、設定が有効になっていません。イベント設定後に[適用] ボタンをクリックして設定を有効にしてください（設定が有効になると「*」が消えます）。

(d) その他条件を設定する

以下のその他条件の検出をブレークポイントとして使用するかどうか設定します。

- トレースフルブレーク

4.7.2 オンチップブレーク設定内容を保存する／読み込む

(a) オンチップブレーク設定内容を保存する

[オンチップブレーク設定] ダイアログボックスの [保存...] ボタンをクリックしてください。

[名前を付けて保存] ダイアログボックスが開きます。

保存先ファイル名を指定してください。ファイル名の拡張子は `.hev` です。省略した場合は、拡張子 `.hev` が付加されます。

(b) オンチップブレーク設定内容を読み込む

[オンチップブレーク設定] ダイアログボックスの [読み込み...] ボタンをクリックしてください。

[ファイルを開く] ダイアログボックスが開きます。

読み込むファイル名を指定してください。

読み込みを行った時、読み込み前のオンチップブレーク設定は破棄して、読み込んだ設定で再設定します。

[オンチップブレーク設定] ダイアログボックスの [適用] ボタンをクリックすると、読み込んだオンチップブレーク設定が確定します。

4.8 トレース機能

E1/E20 エミュレータデバッガでは、表 4.6 に示す内蔵トレースおよび外部トレースの 2 種類のトレース機能が使用できます。

表4.6 トレース機能一覧

機能	内蔵トレース	外部トレース出力
分岐トレース機能	可	可
データトレース機能	可	可

なお、外部トレース出力機能が使用できる製品は以下ですので、ご注意ください。

表4.7 製品名と外部トレース出力機能対応表

製品名 (製品型名)	外部トレース出力機能使用
E1 (R0E000010KCE00)	使用できません。
E20 (R0E000200KCT00)	使用できます。

内蔵トレースおよび外部トレース出力の設定は、[トレース] ウィンドウの[トレース条件設定] ダイアログボックスで行います。

(1) 内蔵トレース機能

内蔵トレース機能は、マイコンに内蔵されているトレースバッファを使用して実現します。

イベント機能 (実行 PC イベント、オペランドアクセスイベント) により、CPU バスを対象とした分岐トレース、データトレースが可能です。

RX600 シリーズマイコンの場合、最大 256 分岐 or サイクル分の分岐元/分岐先アドレス情報、データアクセス情報を取得することができます。

RX200 シリーズマイコンの場合、最大 64 分岐 or サイクル分の分岐元/分岐先アドレス情報、データアクセス情報を取得することができます。ただし、データアクセス情報を取得するためには、トレース抽出機能にてアドレス条件を設定する必要があります。

また、取得したトレース結果はトレースウィンドウ上でバス/逆アセンブリ/C ソースレベルで表示することができます。

なお、各機能仕様については、「表 3.3 ターゲットマイコンに対する機能仕様一覧」を参照してください。

(2) 外部トレース出力機能

マイコンがトレース端子から出力するトレース情報を使用する機能です。E20 はマイコンのトレース端子が出力する情報を記録することで大容量トレース機能を実現します。E20 エミュレータを 38 ピンコネクタに接続した場合に使用できます。トレース内容は内蔵トレース機能と同等で分岐トレース、データトレースが可能です。約 2M 分岐 or サイクルの大容量トレースが可能です。

外部トレース出力機能には、トレース出力優先モード (フルトレースモード) と CPU 実行優先モード (リアルタイムトレースモード) の 2 つのモードを搭載しています。

(a) トレース出力優先モード（フルトレースモード）

全てのトレース情報を全く欠落（ロスト）させることなく、外部に出力することができるモードです。

トレース出力優先モードでは、欠落（ロスト）なくトレース情報を出力させるため、外部への出力量を上回るトレース情報がトレース回路に入力された場合、それ以上のトレース情報が入力されないようにCPUおよびバスを強制的に停止させて、マイコン内部で処理中の全てのトレース情報を外部へ出力した後、再びトレース入力を許可します。よって、トレース出力優先モードではリアルタイム性が損なわれる可能性があります。

(b) CPU 実行優先モード（リアルタイムトレースモード）

トレース情報をリアルタイムに外部へ出力するためのモードです。

CPU実行優先モードでは、リアルタイム性を優先させるため、外部への出力量を内部からの入力量が上回ったとき、トレース情報が欠落（ロスト）する可能性があります。

4.8.1 トレース情報を見る

トレースとは、ユーザプログラム実行中に、サイクルごとのバス情報を取得し、トレースメモリ内に保持する機能です。

トレースを使用して、アプリケーションの実行の流れを追跡し、問題発生ポイントを調べることができます。

4.8.2 トレース情報を取得する

本デバッガは、トレース情報の取得条件を設定しない場合、デフォルトで無条件に全分岐情報をトレース取得します。（トレース取得モード = フリー）

フリーモードではユーザプログラムの実行開始と共にトレース取得を開始し、ユーザプログラムの停止によりトレース取得を停止します。

取得したトレース情報を [トレース] ウィンドウに表示します。

Cycle	Label	Address	Source Address	Destination Address	Data	Size	R/W	BUS Master	Type	BCN	Branch	Channel	TimeStamp (Count)
-0000063		FFFF87C0	-----	FFFF87C0	-----	----	-	CPU	DESTINATION	-----	-----	-----	-----
-0000062		-----	-----	-----	-----	----	-	CPU	BCND	11	-----	-----	-----
-0000061	_rand	FFFF88E9	-----	FFFF88E9	-----	----	-	CPU	DESTINATION	-----	-----	-----	-----
-0000060		FFFF87C0	-----	FFFF87C0	-----	----	-	CPU	DESTINATION	-----	-----	-----	-----
-0000059		-----	-----	-----	-----	----	-	CPU	BCND	11	-----	-----	-----
-0000058	_rand	FFFF88E9	-----	FFFF88E9	-----	----	-	CPU	DESTINATION	-----	-----	-----	-----
-0000057		FFFF87C0	-----	FFFF87C0	-----	----	-	CPU	DESTINATION	-----	-----	-----	-----
-0000056		-----	-----	-----	-----	----	-	CPU	BCND	11	-----	-----	-----
-0000055	_rand	FFFF88E9	-----	FFFF88E9	-----	----	-	CPU	DESTINATION	-----	-----	-----	-----
-0000054		FFFF87C0	-----	FFFF87C0	-----	----	-	CPU	DESTINATION	-----	-----	-----	-----
-0000053		-----	-----	-----	-----	----	-	CPU	BCND	11	-----	-----	-----
-0000052	_rand	FFFF88E9	-----	FFFF88E9	-----	----	-	CPU	DESTINATION	-----	-----	-----	-----
-0000051		FFFF87C0	-----	FFFF87C0	-----	----	-	CPU	DESTINATION	-----	-----	-----	-----
-0000050		-----	-----	-----	-----	----	-	CPU	BCND	11	-----	-----	-----
-0000049	_rand	FFFF88E9	-----	FFFF88E9	-----	----	-	CPU	DESTINATION	-----	-----	-----	-----
-0000048		FFFF87C0	-----	FFFF87C0	-----	----	-	CPU	DESTINATION	-----	-----	-----	-----
-0000047		-----	-----	-----	-----	----	-	CPU	BCND	11	-----	-----	-----
-0000046	_rand	FFFF88E9	-----	FFFF88E9	-----	----	-	CPU	DESTINATION	-----	-----	-----	-----
-0000045		FFFF87C0	-----	FFFF87C0	-----	----	-	CPU	DESTINATION	-----	-----	-----	-----

図4.28 [トレース] ウィンドウ

以下にバス表示の場合の表示項目一覧を示します。

表4.8 表示項目一覧

カラム	内容
Cycle	トレースメモリ内のサイクル番号。最後に取得されたサイクル番号を0とし、古いサイクルにさかのぼって、順に-1、-2と番号が小さくなります。
Label	アドレスに対応するラベルを表示します。ラベルが設定されている場合のみ表示します。
Address	アドレス値を表示します。アドレスが存在しない場合は空白になります。
Source Address	分岐元のアドレス値を表示します。該当の値がない場合は“－”を表示します。
Destination Address	分岐先のアドレス値を表示します。該当の値がない場合は“－”を表示します。
Data	バイト、ワード、ロングワードのデータを16進数で表示します。 アクセスされなかったデータバスは“－”もしくは空白を表示します。
Size	アクセスサイズを表示します。 BYTE:バイト WORD:ワード LONG:ロングワード 該当の値がない場合は“－”を表示します。
R/W	データバスの状態を表示します。 W:ライトアクセス (バスのデータライトの行を赤茶色の文字で表示します。) R:リードアクセス (バスのデータリードの行を緑色の文字で表示します。) バスのデータライト/リード以外の行は黒色で表示します。 該当の値がない場合は“－”を表示します。
BUS Master	トレース事象を発生させたバスマスタを表示します。 CPU:CPU アクセス 該当の値がない場合は“－”を表示します。
Type	トレース情報の種別を表示します。 BCND:条件分岐 BRANCH: 分岐命令の分岐元 DESTINATION: 分岐命令の分岐先 BRANCH/DESTINATION: 分岐命令の分岐元/分岐先 MEMORY: オペランドアクセス STANDBY: スタンバイ情報 LOST:トレース情報が失われたことを示します 該当の値がない場合は“－”を表示します。
BCN	条件分岐成立情報を表示します。条件分岐が成立すると1、不成立だと0を表示します。 最大で15分岐分を1行で表示します。 該当の値がない場合は“－”を表示します。

表 4.8 表示項目一覧 (つづき)

カラム	内容
Branch Type	予約領域です。“-”を表示します。
Channel	オペランドアクセスがどのイベントマッチで出力されたか表示します。 (なし) : イベントマッチではないオペランドアクセス 0: Channel0 イベントマッチでのオペランドアクセス 1: Channel1 イベントマッチでのオペランドアクセス 2: Channel2 イベントマッチでのオペランドアクセス 3: Channel3 イベントマッチでのオペランドアクセス 該当の値がない場合は“-”を表示します。
TimeStamp (Count)	タイムスタンプを表示します。 RX600 シリーズマイコンは、0-19bit までを 10 進数で表示します。 RX200 シリーズマイコンは、0-23bit までを 10 進数で表示します。 オーバフローは検出できません。 トレースの TimeStamp 値は、マイコンによってカウントソースが異なります。 【RX610, RX621, RX62N, RX62T, RX62G グループマイコン】 EXTAL * 8 ≤ 100 MHz の場合 TimeStamp のカウント周波数 = EXTAL * 8 EXTAL * 8 > 100 MHz の場合 TimeStamp のカウント周波数 = EXTAL * 4 【RX630, RX631, RX63N, RX63T グループマイコン】 EXTAL 使用時 : 選択クロックソース / 2 または、選択クロックソース * 1 (SCKCR.ICK を 1 分周に設定した場合) EXTAL 未使用時 : 選択クロックソース * 1 【RX210, RX21A, RX220 グループマイコン】 ICLK (パフォーマンスカウンタを使用)

[トレース] ウィンドウ内の不要なカラムは非表示にすることができます。

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

4.8.3 トレース結果を表示する

トレース結果は、[トレース] ウィンドウで参照します。

トレース結果は、以下の表示形式で表示できます。

表示形式は、[トレース] ウィンドウのポップアップメニューの [表示モード] で切り替えることができます。

トレース結果表示は、バス表示、逆アセンブル表示、およびソース表示と、これらの混合表示ができます。

(a) バス表示

トレースサイクルごとのバス情報を参照できます。表示内容は、ご使用のマイコン、エミュレータシステムに依存します。バス情報に加えて、逆アセンブル情報、ソース行情報、およびデータアクセス情報を混合表示できます。

バス表示は、ポップアップメニューから [表示モード -> BUS] を選択します。

Cycle	Label	Address	Source Address	Destination Address	Data	Size	R/W	BUS Master	Type	BCN	Branch	Channel	TimeStamp (Count)
-00000063		FFFF87C0	-----	FFFF87C0	-----	----	-	CPU	DESTINATION	11	-----	-----	-----
-00000062		FFFF87C0	-----	FFFF87C0	-----	----	-	CPU	BCND	11	-----	-----	-----
-00000061	_rand	FFFF88E9	-----	FFFF88E9	-----	----	-	CPU	DESTINATION	11	-----	-----	-----
-00000060		FFFF87C0	-----	FFFF87C0	-----	----	-	CPU	DESTINATION	11	-----	-----	-----
-00000059		FFFF87C0	-----	FFFF87C0	-----	----	-	CPU	BCND	11	-----	-----	-----
-00000058	_rand	FFFF88E9	-----	FFFF88E9	-----	----	-	CPU	DESTINATION	11	-----	-----	-----
-00000057		FFFF87C0	-----	FFFF87C0	-----	----	-	CPU	DESTINATION	11	-----	-----	-----
-00000056		FFFF87C0	-----	FFFF87C0	-----	----	-	CPU	BCND	11	-----	-----	-----
-00000055	_rand	FFFF88E9	-----	FFFF88E9	-----	----	-	CPU	DESTINATION	11	-----	-----	-----
-00000054		FFFF87C0	-----	FFFF87C0	-----	----	-	CPU	DESTINATION	11	-----	-----	-----
-00000053		FFFF87C0	-----	FFFF87C0	-----	----	-	CPU	BCND	11	-----	-----	-----
-00000052	_rand	FFFF88E9	-----	FFFF88E9	-----	----	-	CPU	DESTINATION	11	-----	-----	-----
-00000051		FFFF87C0	-----	FFFF87C0	-----	----	-	CPU	DESTINATION	11	-----	-----	-----
-00000050		FFFF87C0	-----	FFFF87C0	-----	----	-	CPU	BCND	11	-----	-----	-----
-00000049	_rand	FFFF88E9	-----	FFFF88E9	-----	----	-	CPU	DESTINATION	11	-----	-----	-----
-00000048		FFFF87C0	-----	FFFF87C0	-----	----	-	CPU	DESTINATION	11	-----	-----	-----
-00000047		FFFF87C0	-----	FFFF87C0	-----	----	-	CPU	BCND	11	-----	-----	-----
-00000046	_rand	FFFF88E9	-----	FFFF88E9	-----	----	-	CPU	DESTINATION	11	-----	-----	-----
-00000045		FFFF87C0	-----	FFFF87C0	-----	----	-	CPU	DESTINATION	11	-----	-----	-----

図4.29 バス表示時の[トレース]ウィンドウ

- ① サイクル表示領域：トレースサイクルを表示します。ダブルクリックすると、表示サイクルを変更するためのダイアログボックスが表示されます。
- ② ラベル表示領域：アドレスバス情報に対応するラベルを表示します。
- ③ バス情報表示領域：トレースバス情報を表示します。
- ④ 取得済みトレース計測結果の範囲：現在取得されているトレース計測結果の範囲を表示します。
- ⑤ 先頭行のサイクル：表示先頭行のサイクルを表示します。
- ⑥ 先頭行のアドレス：表示先頭行のアドレスを表示します。

(b) 逆アセンブル表示

実行した機械語命令を参照できます。逆アセンブル情報に加えて、ソース行情報、データアクセス情報を混合表示できます。

逆アセンブル表示は、ポップアップメニューから [表示モード -> DIS] を選択します。

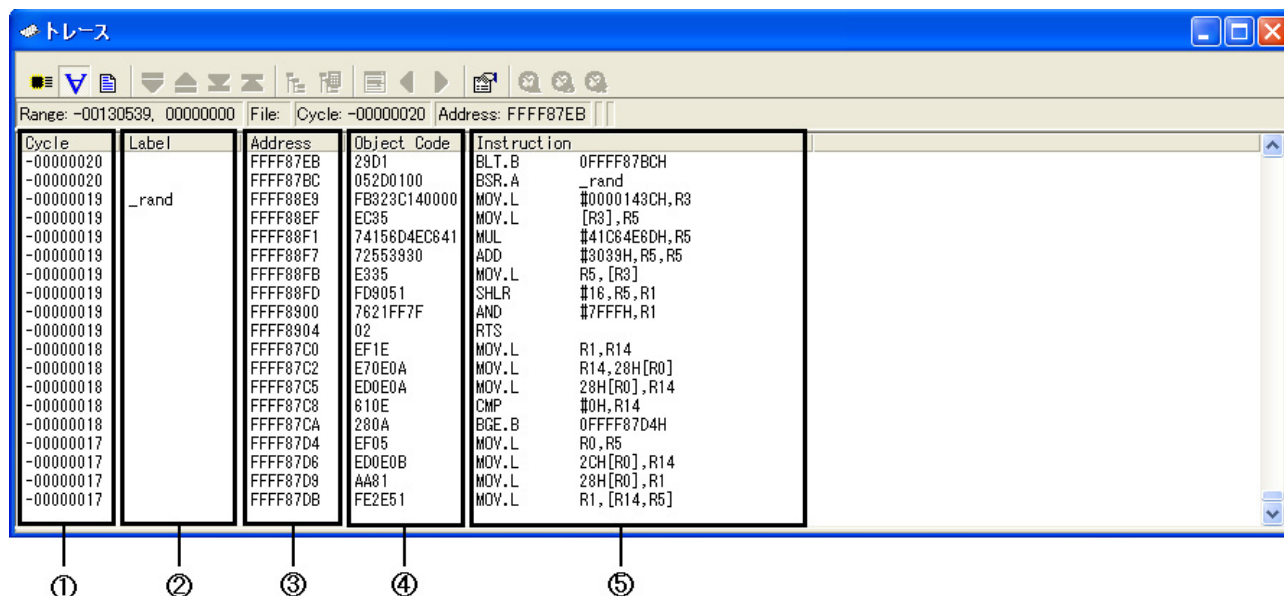


図4.30 逆アセンブル表示時の[トレース]ウィンドウ

- ① サイクル表示領域：トレースサイクルを表示します。ダブルクリックすると、表示サイクルを変更するためのダイアログボックスが表示されます。
- ② ラベル表示領域：命令のアドレスに対応するラベルを表示します。
- ③ アドレス表示領域：命令に対応するアドレスを表示します。ダブルクリックすると、アドレスを検索するためのダイアログボックスが表示されます。
- ④ オブジェクトコード表示領域：命令のオブジェクトコードを表示します。
- ⑤ 命令表示領域：命令を表示します。

(c) ソース表示

ソースプログラムの実行経路を参照できます。

実行経路は、現在のトレースサイクルから順方向、または逆方向にトレースデータ内をソースステップして確認できます。

トレース計測が終了した時点で計測結果を表示します。トレース計測が再開されると、ウィンドウ表示はクリアされます。

ソース表示は、ポップアップメニューから [表示モード -> SRC] を選択します。

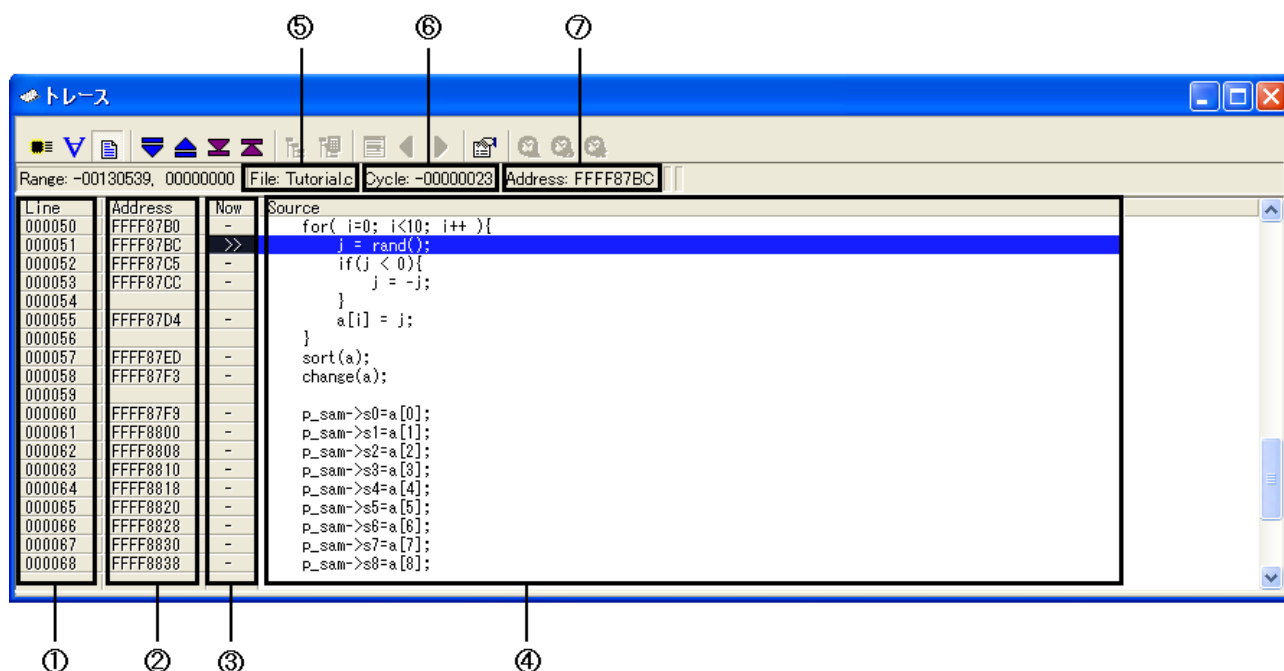


図4.31 ソース表示時の[トレース]ウィンドウ

- ① 行番号表示領域：表示されているファイルの行番号情報を表示します。
- ② アドレス表示領域：ソース行に対応するアドレスを表示します。ダブルクリックすると、アドレスを検索するためのダイアログボックスが表示されます。
- ③ 参照サイクル表示領域：現在参照しているサイクルには">>"が表示されます。また、ソース行に対応するアドレスが存在する場合は"- "が表示されます。
- ④ ソース表示領域：ソースファイルを表示します。
- ⑤ ファイル名：現在表示中のソースファイル名を表示します。
- ⑥ 参照サイクル：現在参照中のサイクルを表示します。
- ⑦ 参照アドレス：現在参照中のサイクルに対応するアドレスを表示します。

(d) 混合表示

バス表示、逆アセンブル表示、ソース表示の混合表示ができます。

ポップアップメニューの[表示モード -> BUS] を選択した後、[表示モード -> DIS] を選択すると、バスと逆アセンブルの混合表示ができます。

同様の方法で、バスとソース、逆アセンブルとソース、バスと逆アセンブルとソースの混合表示ができます。

バスと逆アセンブルの混合表示にした後、バス表示のみの表示に戻すには、再度ポップアップメニューの[表示モード -> DIS] を選択することでバス表示になります。

Cycle	Label	Address	Source	Destina	Data	Size	R/W	BUS Mas	Type	BCN	Branch	Channel	TimeSta
-00000023	FFFF87CA	BGE.B		0FFFF87D4H		----	-	CPU	BCND	11	-----	-	
	FFFF87D4	MOV.L	R0, R5										
	FFFF87D6	MOV.L	2CH[R0], R14										
	FFFF87D9	MOV.L	28H[R0], R1										
	FFFF87DB	MOV.L	R1, [R14, R5]										
	FFFF87DE	MOV.L	2CH[R0], R14										
	FFFF87E1	ADD	#1H, R14										
	FFFF87E3	MOV.L	R14, 2CH[R0]										
	FFFF87E6	MOV.L	2CH[R0], R14										
	FFFF87E8	CMP	#0AH, R14										
	FFFF87EB	BLT.B		0FFFF87BCH									
	FFFF87BC	BSR.A	_rand										
-00000022	_rand	FFFF88E3		FFFF88E		----	-	CPU	DESTINA		-----	-	
	FFFF88E9	MOV.L	#0000149CH, R3										
	FFFF88EF	MOV.L	[R3], R5										
	FFFF88F1	MUL	#41C64E6DH, R5										
	FFFF88F7	ADD	#3039H, R5, R5										
	FFFF88FB	MOV.L	R5, [R3]										
	FFFF88FD	SHLR	#16, R5, R1										
	FFFF8900	AND	#7FFFH, R1										
	FFFF8904	RTS											
-00000021		FFFF87C0		FFFF87C		----	-	CPU	DESTINA		-----	-	
	FFFF87C0	MOV.L	R1, R14										
	FFFF87C2	MOV.L	R14, 28H[R0]										

図4.32 バスと逆アセンブルの混合表示時の[トレース]ウィンドウ

4.8.4 オプションメニュー

ウィンドウ内でマウスの右ボタンをクリックすると以下のポップアップメニューを表示します。これらのメニューの主要な機能は、ツールバーのボタンにも割り付けられています。

表4.9 オプションメニュー

メニュー名	機能	
表示モード	BUS	バス (BUS) 情報を表示します。
	DIS	逆アセンブリ (DIS) 情報を表示します。
	SRC	ソース (SRC) 情報を表示します。
トレース操作	順方向ステップ	現在のサイクルから順方向にソースステップします。 (ソースモード時のみ有効)
	逆方向ステップ	現在のサイクルから逆方向にソースステップします。 (ソースモード時のみ有効)
	指定行まで順方向実行	現在のサイクルから順方向に、指定カーソル位置まで実行します。 (ソースモード時のみ有効)
	指定行まで逆方向実行	現在のサイクルから逆方向に、指定カーソル位置まで実行します。 (ソースモード時のみ有効)
	停止 【注】	ユーザプログラム実行中にトレース取得が行われているときのみ、機能します。トレース取得を中止し、トレース表示を更新します。 内蔵トレースモードでは使用できません。
	リスタート 【注】	トレース取得を再開します。ユーザプログラム実行中、一時的にトレース取得を停止したときのみ、機能します。 内蔵トレースモードでは使用できません。
検索	検索	[検索] ダイアログボックスを表示します。特定のトレース情報を検索します。
	前方検索	[検索...] で指定したパターンに一致する、前のサイクルを検索します。
	後方検索	[検索...] で指定したパターンに一致する、次のサイクルを検索します。
オートフィルタ	オートフィルタ機能の有効/無効を切り換えます。	
トレース設定	トレース条件を設定する、[トレース条件設定]ダイアログを表示します。	
表示カラムの設定	表示カラムを設定する[表示カラムの設定]ダイアログを表示します。	

表 4.9 オプションメニュー(つづき)

メニュー名	機能	
ファイル	ソースファイル編集	ソースモードで表示しているソースファイルを、 [エディタ] ウィンドウに表示します。
	ソースファイル表示	ソースモードの場合、表示するソースファイルを選択 します。ファイルセレクションダイアログボックスを オープンします。
	保存	トレースデータをファイルに保存します。バイナリか テキスト形式の保存が選択可能です。バイナリ形式の 場合、すべてのサイクルを保存します。 テキスト形式の場合、ウィンドウに表示されている内 容を保存します。テキスト形式の場合、指定されたサ イクルの範囲の保存もできます。 また、タブ区切り形式で保存できます（混合モードの ようにバスと逆アセンブル、ソースが混在する場合、 バス表示をタブ区切り、逆アセンブル、ソース表示は スペース区切りになります）。
	読み込み	保存したトレースデータを読み込みます。 バイナリ形式で保存したファイルのみ読み込みでき ます。
ツールバー表示	ツールバーの表示/非表示を切り換えます。	
ツールバーのカスタマイズ...	ツールバーをカスタマイズします。	
ドッキングビュー	ウィンドウをドッキングします。	
非表示	ウィンドウを非表示にします。	

【注】 RX200シリーズマイコンは、内蔵トレースのみサポートのため、本オプションメニューは使用できません。

4.8.5 トレース情報取得条件を設定する

トレースバッファは有限であるため、バッファがいっぱいになった場合は最も古いトレースデータから順に新しいデータを上書きします。

トレース情報の取得条件を設定することにより、有用なトレース情報のみを取得し、トレースバッファを有効に活用することができます。

トレース情報の取得条件はポップアップメニューから [トレース設定...] を選択して表示される [トレース条件設定] ダイアログボックスで設定します。

なお、トレース取得容量や設定可能なトレース取得条件はマイコンによって異なります。「表 3.3 ターゲットマイコンに対する機能仕様一覧」を参照してください。

(a) トレース取得条件を設定する

最初に、トレース取得条件設定を行ってください。

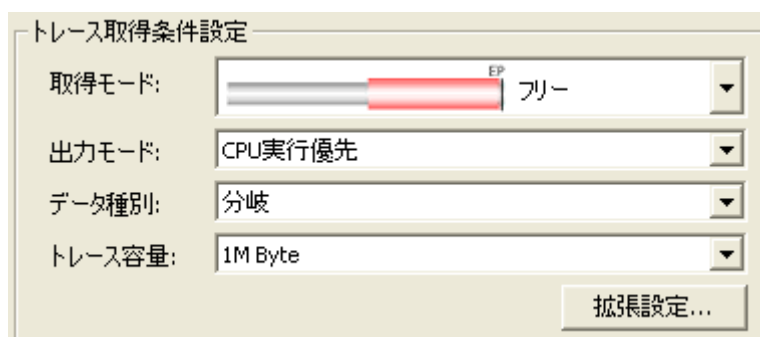


図4.33 [トレース取得条件設定] ダイアログボックス

トレース取得条件設定には、以下の4種類があります。

表4.10 トレース取得条件設定一覧

	パラメータ		内容
①	取得モード	<ul style="list-style-type: none"> ・フリー ・フル 	<p>トレース取得モードを設定します。</p> <p>フリー プログラムを停止するか、トレース取得停止条件が成立するまでトレース取得を続けます。</p> <p>フル プログラムの開始もしくはトレース取得開始条件が成立してからトレースバッファが一杯になった時点でトレース取得を停止します。</p>
②	出力モード 【注1】	<ul style="list-style-type: none"> ・CPU 実行優先 ・トレース優先 ・外部出力しない 	<p>トレース取得データの出力モードを設定します。</p> <p>CPU 実行優先 トレース出力よりも CPU 実行を優先します。出力されるトレース情報にロスが発生する場合があります。</p> <p>トレース優先 CPU 実行よりもトレース出力を優先します。トレース出力のために CPU 実行が停止するため、リアルタイム性が損なわれます。</p> <p>外部出力しない トレース出力を行わず、マイコン内蔵バッファのみを使用します。</p>

表 4.10 トレース取得条件設定一覧(つづき)

	パラメータ		内容
③	データ種別	<ul style="list-style-type: none"> ・分岐 ・分岐+データアクセス ・データアクセス ・分岐元のみ ・分岐元+時間 ・データアクセス+時間 	<p>トレース取得データの種類を設定します。</p> <p>【RX600 シリーズマイコン】</p> <p>分岐： プログラム実行中に発生した分岐処理の分岐元、分岐先アドレス情報をトレースします。</p> <p>分岐+データアクセス： 分岐トレース、およびデータアクセストレースを行います。</p> <p>データアクセス： プログラム実行中に成立したイベントのデータ情報をトレースします。</p> <p>【RX200 シリーズマイコン】</p> <p>分岐： プログラム実行中に発生した分岐処理の分岐元、分岐先アドレス情報をトレースします。</p> <p>データアクセス【注2】： プログラム実行中に成立したイベントのデータ情報をトレースします。</p> <p>分岐元のみ： プログラム実行中に発生した分岐処理の分岐元アドレス情報をトレースします。</p> <p>分岐元+時間： プログラム実行中に発生した分岐処理の分岐元アドレス情報とタイムスタンプをトレースします。</p> <p>データアクセス+時間【注2】： プログラム実行中に成立したイベントのデータ情報とタイムスタンプをトレースします。</p>
④	トレース容量 【注3】	1M、2M、4M、8M、 16M、32M Byte	<p>トレース取得データをエミュレータに保存するバッファ容量を設定します。</p> <p>1M、2M、4M、8M、16M、32M Byte が設定できます。</p>

- 【注】
1. RX200 シリーズマイコンでは、本パラメータは非アクティブ(グレー表示)で“外部出力しない”と表示されません。
 2. RX200 シリーズマイコンでは、トレース抽出条件で設定されたデータアクセスのみトレースできます。
 3. RX200 シリーズマイコンでは、本パラメータは非アクティブ(グレー表示)です。

RX600 シリーズマイコンの場合、拡張設定ボタンをクリックすると、[トレース] ウィンドウの表示内容の設定ができます。表示したい情報のチェックボックスを有効に設定してください。

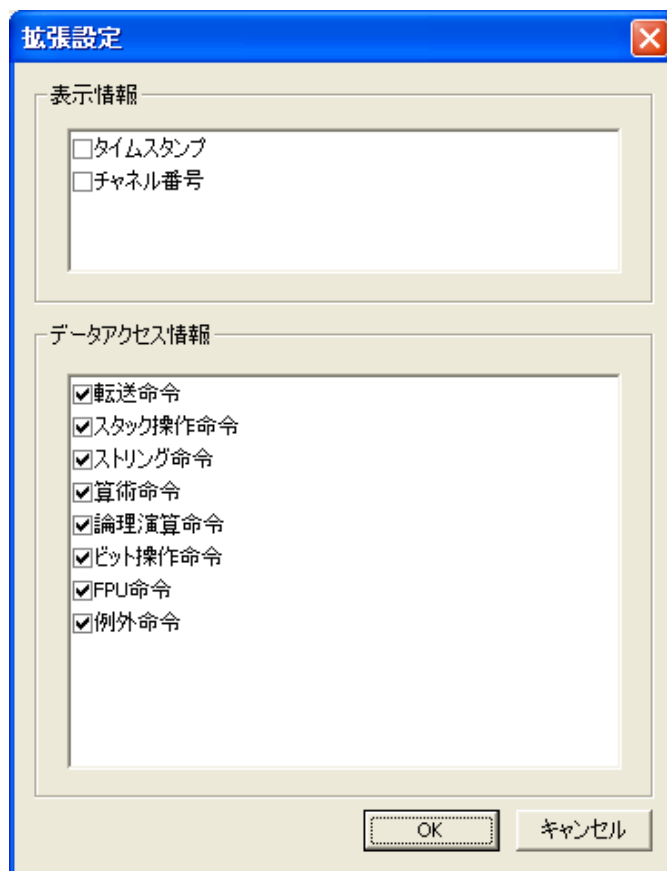


図4.34 [拡張設定] ダイアログボックス

(b) トレース条件を設定する

トレース取得の開始条件、停止条件および抽出条件をそれぞれ設定することができます。トレース条件を設定しない場合は、プログラムの実行から、停止までの内容をトレースします。

条件と組み合わせ設定

トレース取得開始条件(SP)

プログラム実行開始 開始条件: OR 詳細
使用イベント: 0

トレース取得停止条件(EP)

プログラム実行停止 停止条件: OR 詳細
使用イベント: 0

トレース抽出条件

全てのデータアクセス 抽出条件: OR 詳細
使用イベント: 0

図4.35 [トレース条件設定] ダイアログボックス

トレース条件の設定には、以下の3種類があります。

表4.11 トレース条件設定一覧

設定項目	内容
① トレース取得開始条件(SP)	<p>トレース取得の開始トリガをイベントの組み合わせで指定できます。 トレース取得開始条件を設定しない場合は、プログラム実行がトレース取得開始となります。</p> <p>OR 設定されたイベントのうちどれかが成立するとトレース取得開始条件が成立します。</p> <p>AND(累積) 時間軸に関係なく、設定されているすべてのイベントが成立すると、トレース取得開始条件が成立します。</p> <p>シーケンシャル 設定されたイベントが指定された順序で成立すると、トレース取得開始条件が成立します。</p> <p>【RX600 シリーズマイコン】 7段階（順方向）+ リセットポイント(R) 【注1】</p> <p>【RX200 シリーズマイコン】 3段階（順方向）+ リセットポイント(R) 【注1】</p>
② トレース取得停止条件(EP)	<p>トレース取得の停止トリガをイベント（ORのみ）で指定できます。 トレース取得停止条件を設定しない場合は、プログラム停止がトレース取得停止となります。</p>
③ トレース抽出条件 【注2】	<p>トレース抽出のトリガをイベント（ORのみ）で指定できます。</p>

- 【注】
- リセットポイント(R): リセットポイントで設定したイベントが成立した場合、その時点まで成立したイベントを全てクリアします。
 - RX200シリーズマイコンでは、抽出条件(OR)指定で非アクティブ(グレー)表示となります。

トレース条件の設定と、トレース取得条件設定の取得モード（フリー、フル）の組み合わせにより、トレース取得内容が変わります。以下に各設定時のトレース取得内容の違いについて示します。

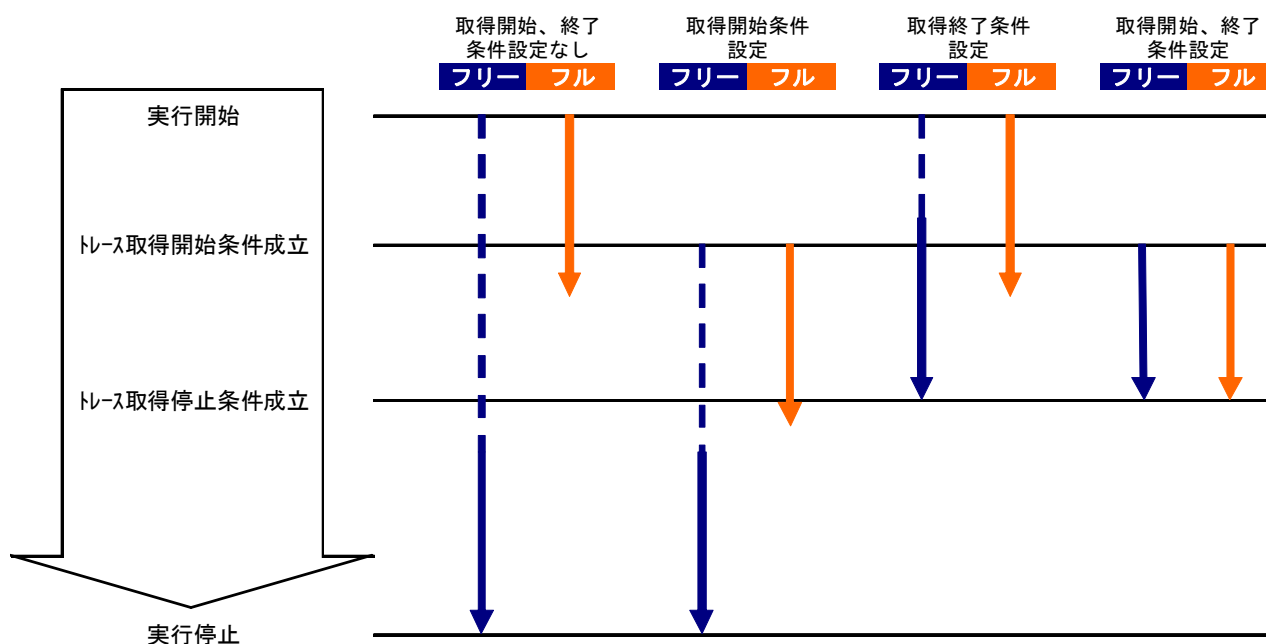


図4.36 トレース取得条件設定と、トレース条件設定によるトレース取得内容の違い

(c) トレース設定内容を保存する／読み込む

① トレース設定内容を保存する

[トレース設定] ダイアログボックスの [保存...] ボタンをクリックしてください。

[名前を付けて保存] ダイアログボックスが開きます。

保存先ファイル名を指定してください。ファイル名の拡張子は .tev です。省略した場合は、拡張子 .tev が付加されます。

② トレース設定内容を読み込む

[トレース設定] ダイアログボックスの [読み込み...] ボタンをクリックしてください。


[ファイルを開く] ダイアログボックスが開きます。

読み込むファイル名を指定してください。

読み込みを行った時、読み込み前のトレース設定は破棄して、読み込んだ設定で再設定します。

[トレース条件設定] ダイアログボックスの [適用] ボタンをクリックすると、読み込んだトレース設定が確定します。

4.8.6 トレース情報をファイルに保存する

トレース情報をファイルに保存するには、ポップアップメニューから [ファイル -> 保存...] を選択するか、[保存...] ツールバーボタン()をクリックします。

[トレース] ウィンドウに表示しているトレース情報をバイナリ形式、またはテキスト形式で保存します。

(a) バイナリ形式で保存する

バイナリ形式で保存するには、[ファイル -> 保存...] メニューで表示されるダイアログボックスの [ファイルの種類] のリストボックスで、 [Trace Data File: Memory Image (*.rtt)] を選択します。


バイナリ形式で保存する場合、すべてのサイクルを保存します。このファイルは、[トレース] ウィンドウへ読み込むことができます。

(b) テキスト形式で保存する

テキスト形式で保存するには、[ファイル -> 保存...] メニューで表示されるダイアログボックスの [ファイルの種類] のリストボックスで、 [Text Files : Save Only (*.txt)] を選択します。

テキスト形式で保存する場合、保存するサイクル範囲を指定することができます。このファイルは保存のみ可能で、[トレース] ウィンドウへの読み込みはできません。

4.8.7 トレース情報をファイルから読み込む

トレース情報をファイルから読み込むには、ポップアップメニューから [ファイル -> 読み込み...] を選択するか、[読み込み...] ツールバーボタン()をクリックします。


バイナリ形式で保存したトレース情報ファイルを指定してください。現在のトレース結果は上書きされません。

バイナリ形式で保存したファイルを読み込む場合、[トレース設定...] メニューで表示される [トレース条件設定] ダイアログボックスで保存した時のトレースモードにしてから読み込んでください。

テキスト形式で保存したトレース情報ファイルは読み込むことができません。

4.8.8 トレース情報の取得を一時的に停止する

ユーザプログラム実行中に一時的にトレース情報の取得を停止するには、[トレース] ウィンドウのポップアップメニューから


[トレース操作 -> 停止] を選択するか、[停止] ツールバーボタン()をクリックします。

トレース取得を中止し、トレース表示を更新します。

ユーザプログラムを停止せずにトレース情報の取得のみ停止し、トレース情報を確認する場合などに使用します。

内蔵トレースモードでは使用できません。

4.8.9 トレース情報の取得を再開する

ユーザプログラム実行中、一時的にトレース情報の取得を停止した場合、再度トレース情報の取得を再開するには、[トレース] ウィンドウのポップアップメニューから [トレース操作 -> リスタート] を選択するか、[リスタート] ツールバーボタン()をクリックします。

内蔵トレースモードでは使用できません。

4.9 パフォーマンス測定機能

4.9.1 パフォーマンスを測定する

パフォーマンス測定機能とは、ユーザが指定した最大2区間に対して、それぞれのトータル実行時間または通過回数を計測し、実行時間やサイクル数を表示する機能です。

本機能は、マイコン内蔵のカウンタにより、Go-Break間、もしくはカウント開始イベント成立時からカウント終了イベント成立時までの測定の他、ユーザプログラム実行中の再計測をサポートしています。カウント開始イベント、カウント終了イベントには命令フェッチ条件、もしくはデータアクセス条件が設定できます。

パフォーマンス測定機能は、ターゲットマイコンのパフォーマンス測定機能により実行時間を測定するため、ユーザプログラムの実行を妨げることはありません。

パフォーマンス測定条件は、プログラム実行中は操作することはできません。

RX600 シリーズマイコンの場合、計測項目は、計測指定なし、経過サイクル、経過サイクル(スーパバイザモードサイクル)、例外・割り込み処理サイクル、例外処理サイクル、割り込み処理サイクル、有効命令発行回数、例外・割り込み受付回数、例外受付回数、割り込み受付回数、イベントマッチ回数の11項目があります。

32ビットカウンタ×2ch (ch0 および ch1 を別々に使用)、または64ビットカウンタ×1ch (ch0 および ch1 の32ビットカウンタを連結)の2つの方式をサポートしています。

RX200 シリーズマイコンの場合、計測項目は、経過サイクルのみであり、24ビットカウンタ×1chをサポートしています。

なお、各機能仕様については、「表 3.3 ターゲットマイコンに対する機能仕様一覧」を参照してください。

4.9.2 パフォーマンス測定結果を表示する

[パフォーマンス解析] ウィンドウに測定結果を表示します。

[パフォーマンス解析] ウィンドウを開くには、[表示 -> パフォーマンス -> パフォーマンス解析]を選択するか、[パフォーマンス解析] ツールバーボタン()をクリックしてください。[パフォーマンス解析方法の選択] ダイアログボックスが開きますので、[OK] ボタンを押してください。

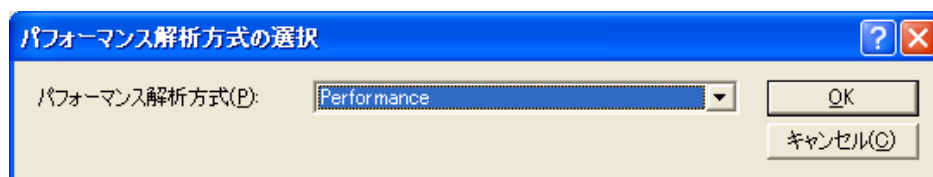


図4.37 [パフォーマンス解析方法の選択]ダイアログボックス

[パフォーマンス解析] ウィンドウが表示されます。

本ウィンドウをマウスの右ボタンでクリックすることによって開くポップアップメニューより、[設定...] を選択してください。[パフォーマンス条件設定] ダイアログボックスが開きます。

本ダイアログボックスで測定するイベントや測定条件を設定することができます。

条件設定後、[OK] ボタンを押し、ユーザプログラムを実行すると、実行終了後にユーザが設定した条件の実行時間、およびサイクル数を表示します。

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

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

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

No	Condition	Time (h:m:s.ms.us.ns)	Count (Decimal)
1	Not use	---:---:---.---.---	-
2	Not use	---:---:---.---.---	-

図4.38 [パフォーマンス解析]ウィンドウ

表示内容は以下のとおりです。

表4.12 カラムと表示内容一覧

カラム	内容
No	<p>[パフォーマンス条件設定] ダイアログボックスで設定する計測区間の番号です。</p> <p>【RX600 シリーズマイコン】 チャンネル番号:32bit×2ch モードの場合は 1,2、64bit×1ch モードの場合は 1 のみを表示します。</p> <p>【RX200 シリーズマイコン】 チャンネル番号:24bit×1ch モードで 1 のみを表示します。</p>
Condition	<p>[パフォーマンス条件設定] ダイアログボックスの計測項目の内容を表示します。</p> <p>【RX600 シリーズマイコン】</p> <ul style="list-style-type: none"> 未使用時:"Not use" 経過サイクル: "Execution cycle" 経過サイクル(スーパーバイザモードサイクル): "Execution cycle (Supervisor mode)" 例外・割り込み処理サイクル:"Exception and interrupt cycle" 例外処理サイクル:"Exception cycle" 割り込み処理サイクル:"Interrupt cycle" 有効命令発行回数:"Execution count" 例外・割り込み受付回数:"Exception and interrupt count" 例外受付回数:"Exception count" 割り込み受付回数:"Interrupt count" イベントマッチ回数:"Event match count" <p>【RX200シリーズマイコン】</p> <ul style="list-style-type: none"> 経過サイクル: "Execution cycle"
Time (h:m:s.ms.us.ns)	<p>計測された実行時間の累積時間を表示します。</p> <p>【RX600 シリーズマイコン】</p> <ul style="list-style-type: none"> 経過サイクル、経過サイクル(スーパーバイザモードサイクル)、例外・割り込み処理サイクル、例外処理サイクル、割り込み処理サイクルの場合は Count を時間表示に直し表示します。 ユーザが設定した周波数と測定した Count を用いて値を決定します。 なお、計算後の値が 65536 時間以上となる場合、0h0m0s0ms0us0ns と表示されます。 有効命令発行回数、例外・割り込み受付回数、例外受付回数、割り込み受付回数、イベントマッチ回数の場合は"---:---:---.---.---.---"を表示します。 <p>【RX200シリーズマイコン】</p> <ul style="list-style-type: none"> Count を時間表示に直し表示します。 ユーザが設定した周波数と測定した Count を用いて値を決定します。 なお、計算後の値が 65536 時間以上となる場合、0h0m0s0ms0us0ns と表示されます。
Count (Decimal)	<p>測定した Count を 10 進数で表示します。カウンタがオーバーフローした場合は"overflow"を表示します。</p>

4.9.3 パフォーマンス測定条件を設定する

[パフォーマンス解析] ウィンドウで、条件を設定したい区間番号の行を選択して、ポップアップメニューから[設定...] を選択してください。

[パフォーマンス条件設定] ダイアログボックスを表示します。

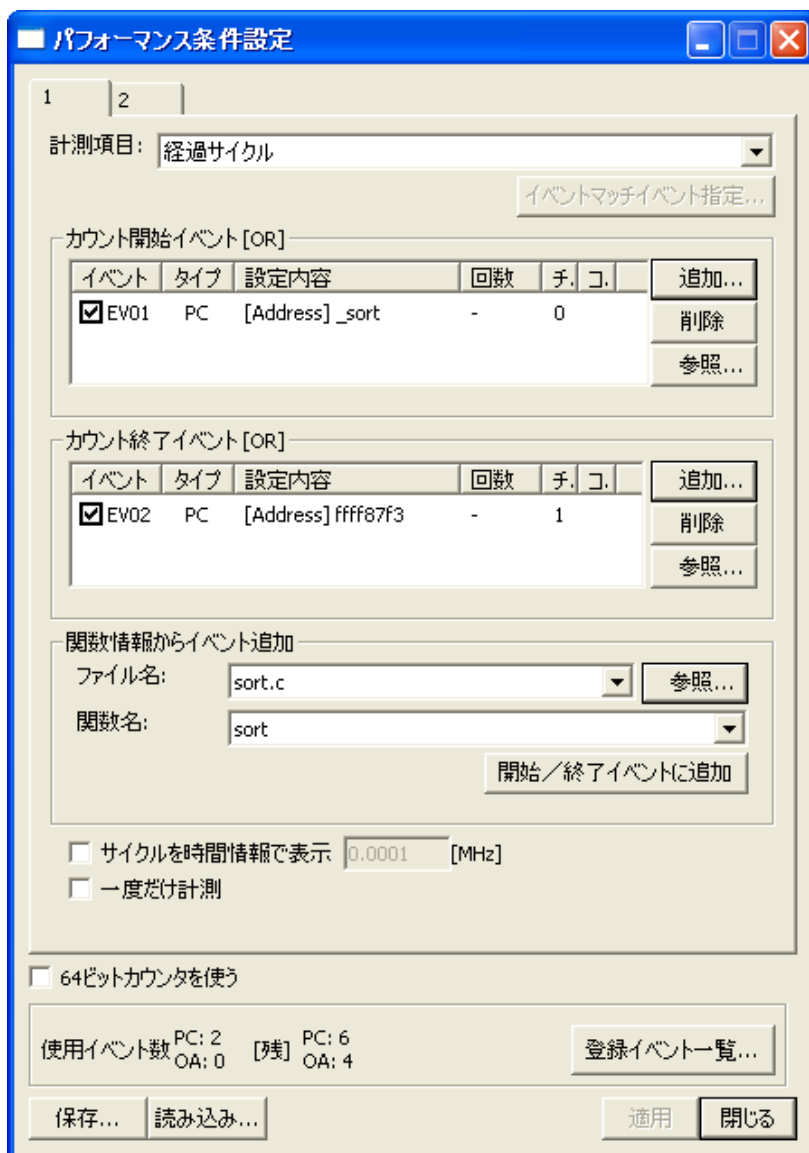


図4.39 [パフォーマンス条件設定] ダイアログボックス

タイトルバーの最後に「*」が付いている場合は、設定が有効になっていません。イベント設定後に[適用] ボタンをクリックして設定を有効にしてください（設定が有効になると「*」が消えます）。

(a) 測定条件を設定する

RX600シリーズマイコンの場合、1区間につき、32ビットの回数測定カウンタを使用し、2区間の測定が可能です。また1区間のみを64ビットの回数測定カウンタに設定して使用することが可能です。どちらか1区間を64ビットカウンタに設定した場合、もう1区間の設定は無効になります。

RX200シリーズマイコンの場合、24ビットの回数測定カウンタで1区間のみ測定が可能です。ひとつの区間につき、ひとつの測定条件を選択します。

区間の設定には、イベントを使用します

表4.13 測定条件のモード一覧

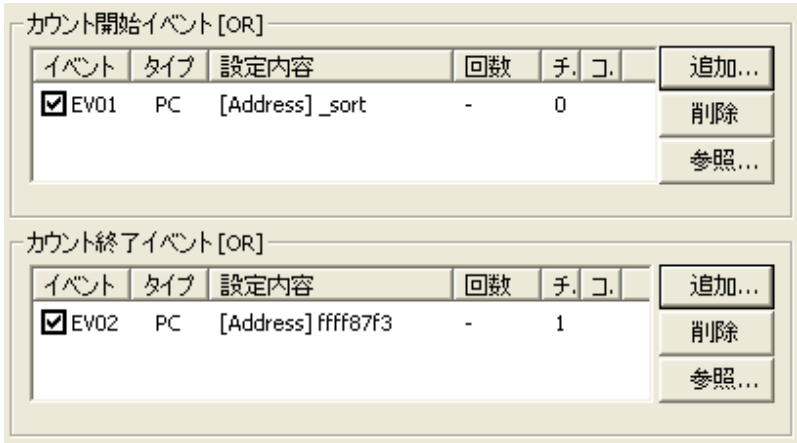
計測項目	計測内容
測定指定なし	計測しません。
【処理サイクル計測】 経過サイクル、 経過サイクル(スーパバイザ モードサイクル)、 例外・割り込み処理サイク ル、 例外処理サイクル、 割り込み処理サイクル 【処理回数計測】 有効命令発行回数、 例外・割り込み受付回数、 例外受付回数、 割り込み受付回数	開始イベントの成立から終了イベントの成立まで計測します。  <p style="text-align: center;">図4.40 イベント点間</p> <p><開始イベント>と<終了イベント>で設定された範囲の実行サイクルおよび処理回数を測定します。測定時間は、<開始イベント>の成立で測定を開始し、<終了イベント>の成立で測定を中断します。</p> <p>処理回数は、設定範囲の<開始イベント>と<終了イベント>間に成立した処理回数をカウントします。</p> <p>[カウント開始イベント]: 1イベントまたは複数イベントを設定することができます。カウント開始イベントを設定しなかった場合は、ユーザプログラムの実行で測定が開始されます。</p> <p>[カウント終了イベント]: 1イベントまたは複数イベントを設定することができます。カウント終了イベントを設定しなかった場合は、ユーザプログラムの停止で測定が中断されます。</p>

表 4.13 測定条件のモード一覧 (つづき)

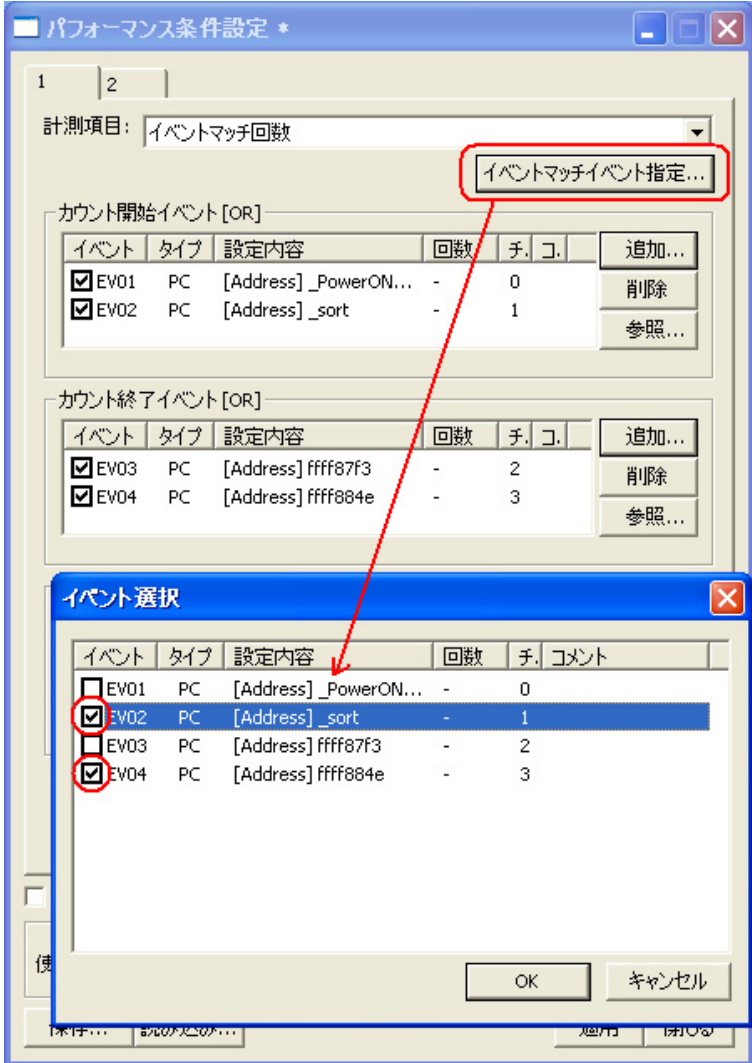
計測項目	計測内容
イベントマッチ回数	<p>開始イベントの成立から終了イベントの成立までのイベント成立回数を計測します。</p>  <p>図4.41 イベント間のイベントマッチ回数</p> <p><開始イベント>と<終了イベント>で設定された範囲内の、イベントマッチ回数測定に指定されたイベントの成立回数をカウントします。</p> <p>[カウント開始イベント]:1 イベントまたは複数イベントを設定することができます。</p> <p>[カウント終了イベント]:1 イベントまたは複数イベントを設定することができます。</p> <p>イベントマッチ回数測定に使用するイベントは、トレース条件設定、オンチップブ레이크条件設定、およびパフォーマンス条件設定で設定されているイベントから選択します。</p> <p>イベントマッチ回数測定に指定するイベントは、 [イベントマッチイベント指定...] ボタンをクリックして表示される[イベント選択] ダイアログボックスから設定することができます。</p>

表 4.13 測定条件のモード一覧 (つづき)

計測項目	計測内容
イベントマッチ回数	<p>イベントマッチ回数計測で指定したイベントは共有状態となります。 共有状態のイベントは有効/無効の変更はできません。 イベント関連の操作により共有状態が解除され、イベントマッチ回数計測の独立したイベントとなる場合があります。</p> <p>イベントマッチ回数計測の独立したイベントを削除する場合は、[イベント選択] ダイアログボックスで指定イベントのチェックボックスを外し、[パフォーマンス条件設定] ダイアログボックスで設定変更して状態を必ず更新してください。 上記操作のチェックボックスを外すと同時に指定イベントの表示は消えます。 なお、その後のキャンセルは受け付けません。</p> <p>[イベント選択]ダイアログボックスでは、同一名のイベントが複数表示される場合があります。</p>

【注】 関数の実行時間を計測するには、イベント 2 点間を使用してください。

開始イベントとして関数の先頭アドレスへの実行を設定し、終了イベントとして、関数の出口 (return 文のところ) への実行を設定してください。出口が複数ある場合は、すべての出口に対する実行条件を終了イベントとして設定してください。

(b) 計測項目

以下にパフォーマンス測定機能で選択できる計測項目の一覧を示します。

選択名は[パフォーマンス解析]ウィンドウの[Condition]に表示します。

表4.14 計測項目一覧

選択名	Condition	選択項目
計測指定無し	Not use	パフォーマンス測定項目を設定しません。
経過サイクル	Execution cycle	指定期間の経過サイクル数 (ICLK) を測定項目に設定します。
経過サイクル (スーパーバイザモードサイクル)	Execution cycle (Supervisor mode)	スーパーバイザモードにいる間の経過サイクル数 (ICLK) を計測項目に設定します。
例外・割り込み処理サイクル	Exception and interrupt cycle	例外処理を含む割り込み処理に要したサイクル数 (ICLK) を計測項目に設定します。
例外処理サイクル	Exception cycle	例外処理に要したサイクル数 (ICLK) を計測します。
割り込み処理サイクル	Interrupt cycle	割り込み処理に要したサイクル数 (ICLK) を計測します。
有効命令発行回数	Execution count	実行の完了した命令の個数を計測します。
例外・割り込み受付回数	Exception and interrupt count	例外を含む割り込みを受け付けた回数を計測します。
例外受付回数	Exception count	例外が発生した回数を計測します。
割り込み受付回数	Interrupt count	割り込みを受け付けた回数を計測します。
イベントマッチ回数	Event match count	設定したイベントに実際にマッチした回数を計測します。

パフォーマンス測定での例外処理／割り込み処理には、以下マイコンの例外事象が含まれます。

【例外処理】

- 未定義命令例外
- 特権命令例外
- 浮動小数点例外
- 無条件トラップ (INT命令、BRK命令)

【割り込み処理】

- ノンマスクابل割り込み
- 割り込み (マスクابل)
- リセット

(c) 関数情報からイベントを追加する【注】

ソースファイルから関数名を指定して、パフォーマンス測定の開始、終了イベントを設定できます。

[参照] ボタンをクリックしてソースファイルを選択後、関数名を設定し、[開始/終了イベントに追加]のボタンをクリックすると、[開始イベント]、[終了イベント]に関数の先頭、終了アドレスの実行イベントが設定されます。

【注】 この設定では、関数の開始番地に開始イベントを、終了番地に終了イベントを設定します。終了番地が関数の出口（通常 RTS 命令）ではない場合は、正しく計測ができません。終了番地が関数の出口ではない場合は、手動で関数出口の番地を終了イベントに設定してください。

(d) 時間情報を表示する

計測項目をサイクルに選択して[サイクルを時間情報で表示] チェックボックスにチェックした場合、[周波数] エディットボックスにて入力した周波数とパフォーマンス計測結果のカウント値から時間情報を算出して表示します。

入力値の範囲は、0.0001～999.999です。この範囲外の値を入力した場合、エラーメッセージが表示されます。

なお、入力値の有効桁数は6桁であり、整数部が2桁以下の場合は少数第5位以下を切捨て、整数部が3桁の場合は少数第4位以下を切り捨てます。

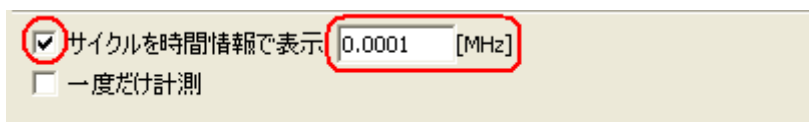


図4.42 時間情報の表示

4.9.4 パフォーマンス測定を開始する

ユーザプログラムを実行すると、設定したパフォーマンス測定条件にしたがい自動的にパフォーマンス測定を開始します。

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

ユーザプログラムを停止後、測定条件を変えずに再実行した場合、測定時間は前回の測定値に加算されます。

測定を最初からやり直したい場合は、実行前にポップアップメニューから[データクリア] を選択して計測結果をクリアしてください。

また、[一度だけ計測] チェックボックスをチェックした場合、再実行の測定時間は前回の測定値に加算されず、実行毎の測定結果が表示されます。

4.9.5 パフォーマンス測定条件を解除する

[パフォーマンス解析] ウィンドウ上で削除したい測定条件を選択して、ポップアップメニューから[設定...]を選択して [パフォーマンス条件設定] ダイアログボックスを表示します。

[パフォーマンス条件設定] ダイアログボックスで解除したい条件を計測指定なしにして設定してください。

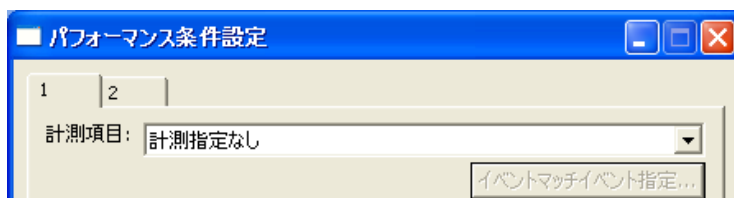


図4.43 パフォーマンス測定条件の削除

4.9.6 パフォーマンス測定結果をクリアする

[パフォーマンス解析] ウィンドウ上でクリアしたい区間を選択して、ポップアップメニューから [データクリア] を選択すると、選択された区間の測定結果をクリアします。

すべての測定結果をクリアする場合は、ポップアップメニューから [全てのデータをクリア] を選択します。

4.9.7 パフォーマンスの計測最大回数について

RX600 シリーズマイコンの場合、各区間のパフォーマンスの回数測定は 32 ビットのカウンタで構成されており、最大 4,294,967,295 回まで測定できます。1 区間を 64 ビットのカウンタに設定した場合、最大 18,446,744,073,709,551,615 回まで測定できます。

RX200 シリーズマイコンの場合、パフォーマンスの回数測定は 24 ビットのカウンタで構成されており、最大 16,777,215 回まで測定できます。

パフォーマンス測定中にカウンタがオーバーフローした場合は、[パフォーマンス解析] ウィンドウには"overflow"と表示されます。

64 ビットカウンタを使用する場合は、[パフォーマンス条件設定] ダイアログボックス内の[64 ビットカウンタを使う] チェックボックスにチェックを入れてください。

4.9.8 パフォーマンス条件設定の内容を保存する/読み込む

(a) パフォーマンス条件設定の内容を保存する

[パフォーマンス条件設定] ダイアログボックスの [保存...] ボタンをクリックしてください。

[名前を付けて保存] ダイアログボックスが開きます。

保存先ファイル名を指定してください。ファイル名の拡張子は .pev です。省略した場合は、拡張子 .pev が付加されます。

(b) パフォーマンス条件設定の内容を読み込む

[パフォーマンス条件設定] ダイアログボックスの [読み込み...] ボタンをクリックしてください。

[ファイルを開く] ダイアログボックスが開きます。

読み込むファイル名を指定してください。

読み込みを行った時、読み込み前のパフォーマンス条件設定は破棄して、読み込んだ設定で再設定します。[パフォーマンス条件設定] ダイアログボックスの [適用] ボタンをクリックすると、読み込んだトレース設定が確定します。

4.10 リアルタイムRAMモニタ機能

4.10.1 RAMモニタを使用する

ユーザプログラム実行中にメモリ内容をモニタするには、[RAM モニタ] ウィンドウを使用します。RAM モニタ機能とは、モニタ範囲に割り当てられた領域のメモリ内容とアクセス状況を、ユーザプログラムの実行を妨げることなく、リアルタイムに記録、参照する機能です。[RAM モニタ] ウィンドウ上でアクセス状況（リード／ライト／ロスト発生／エラー検出）を色分け表示します。

RAM モニタはデバッガの外部トレース機能を使用して実現しているため、外部トレース機能が使用出来ないE1 エミュレータでは使用することが出来ません。

RAM モニタ機能を使用する場合は、[コンフィグレーションプロパティ] ダイアログボックスの排他機能をリアルタイム RAM モニタに設定してください。

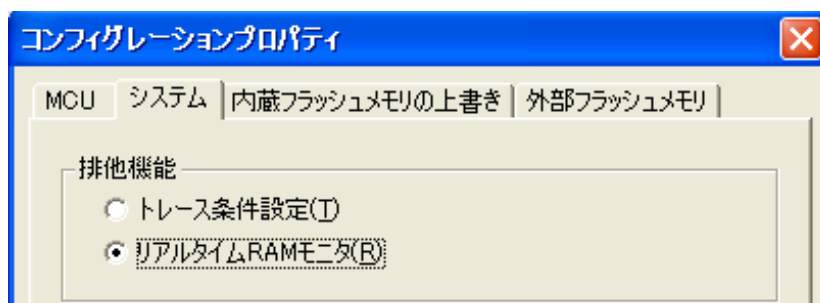


図4.44 [コンフィグレーションプロパティ]ダイアログボックス

[コンフィグレーションプロパティ] ダイアログボックスの排他機能を、リアルタイム RAM モニタに設定すると一部のトレース機能が使用できなくなります。下記にリアルタイム RAM モニタ使用時のトレース機能の制限事項を示します。

表4.15 トレース機能の制限事項一覧

機能名	制限事項内容
イベント設定	開始、終了イベントは使用不可、トレース抽出イベントのみ使用可
取得モード	フルモードは使用不可、フリーモードのみ使用可
データ種別	分岐、分岐+データトレースは使用不可、データトレースのみ使用可
トレース容量	1MBのみ使用可
トレース表示情報	タイムスタンプ、スタックポインタ、チャンネル番号は指定不可

4.10.2 RAMモニタ範囲を割り当てる

4096 バイトの RAM モニタ領域を備えています。

この RAM モニタ領域は任意の連続したアドレス、または、1024 バイト単位で 4 ブロックに分割して配置することができます。

この設定は、どの領域を監視するかを設定するものです。この状態では、全てのデータアクセスに対してトレース出力を行っているため、ロスト発生の可能性があります。さらにトレースの抽出条件を指定することで最適な RAM モニタ監視を行うことができます。

「4.10.5 ロスト発生の抑止」を参照してください。

RAM モニタ領域の設定は[RAM モニタ領域設定] ダイアログボックスで設定します。

[RAM モニタ領域設定] ダイアログボックスを開くには [表示 -> CPU -> RAM モニタ] を選択し[RAM モニタ] ウィンドウを表示させ、[RAM モニタ領域の設定] ツールバーボタンをクリックしてください。

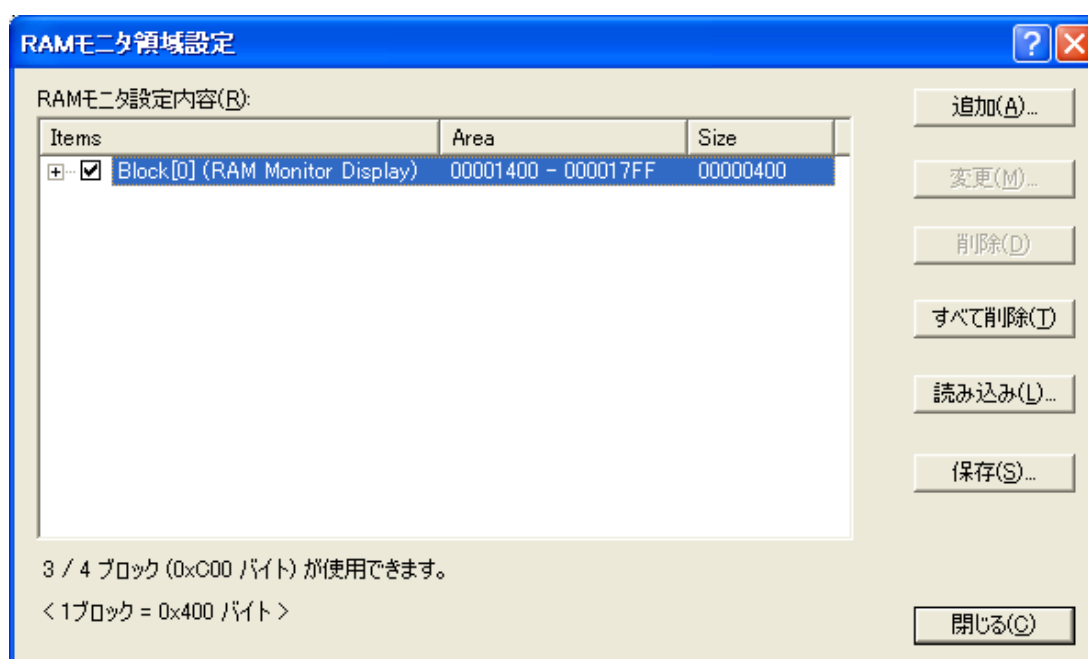


図4.45 [RAM モニタ領域設定] ダイアログボックス

(a) RAMモニタブロックを設定する

[追加] ボタンをクリックすると、RAMモニタ領域に割り当てるアドレスを設定するダイアログボックスが開きます。

RAMモニタブロックは1ブロック1024バイトです。RAMモニタ領域に割り当てるアドレスが1024バイト未満の範囲を指定しても、開始アドレスから1024バイトまでの領域がRAMモニタ領域に割り当てられます。

また、開始アドレスから1024バイトを超える範囲の領域を設定した場合は、自動で複数のRAMモニタブロックを割り当てます。

なお、ユーザプログラム実行中に本処理を行うとメモリのリードサイクルが発生するためリアルタイム性は損なわれます。

(b) RAMモニタブロックを変更する

登録されているRAMモニタブロックを選択して[変更] ボタンをクリックしてください。

RAMモニタ領域に割り当てるアドレスを変更するダイアログボックスが開きます。

なお、ユーザプログラム実行中に本処理を行うとメモリのリードサイクルが発生するためリアルタイム性は損なわれます。

(c) RAMモニタブロックを削除する

登録されているRAMモニタブロックを選択して[削除] ボタンをクリックしてください。

登録されている全てのRAMモニタブロックを削除する場合、[すべて削除] ボタンをクリックしてください。

(d) RAMモニタ領域設定内容を保存する

[RAMモニタ領域設定] ダイアログボックスの[保存] ボタンをクリックしてください。

[名前を付けて保存] ダイアログボックスが開きます。

保存先ファイル名を指定してください。ファイルの拡張子は .rbi です。省略した場合は、拡張子 .rbi が付加されます。

(e) RAMモニタ領域設定内容を読み込む

[RAMモニタ領域設定] ダイアログボックスの[読み込み] ボタンをクリックしてください。


[ファイルを開く] ダイアログボックスが開きます。

読み込むファイル名を指定してください。

読み込みを行った時、読み込み前のRAMモニタ領域設定は破棄して、読み込んだ設定で再設定します。

[RAMモニタ領域設定] ダイアログボックスの [適用] ボタンをクリックすると、読み込んだRAMモニタ領域設定が確定します。

4.10.3 モニタ内容を表示する

[RAM モニタ] ウィンドウを開くには、[表示 -> CPU -> RAM モニタ] を選択するか、[CPU] ツールバーボタン()をクリックしてください。

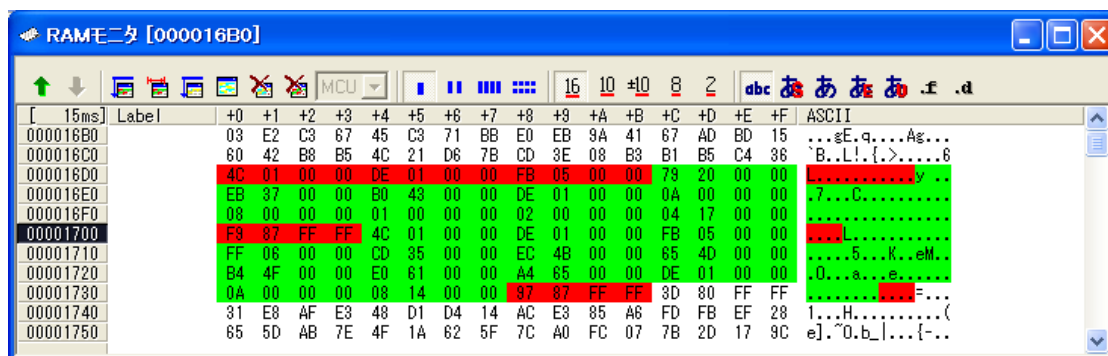


図4.46 [RAM モニタ] ウィンドウ

アクセス属性により以下の背景色で表示します。

リードアクセス、ライトアクセスは最終履歴を表示します。

RAM モニタ機能は、CPU バスアクセスのデータを取得して、その内容を表示します。したがって、ユーザプログラムによる書き込み/読み出しが行われなかった領域（例えば、他のデバイスから外部メモリに書き込み/読み出しをする場合など）の変化、およびDMA/DTC 機能でのアクセスによる変化は表示されません。

RAM モニタ領域の表示データ長が1バイト単位以外の場合、そのデータの1バイト単位でメモリに対するアクセス属性が異なる場合があります。このように1つのデータの中でアクセス属性が異なる場合は、そのデータが括弧に囲まれて表示されます。また、この時の背景色は、そのデータの1バイト目のアクセス属性を示します。

(0001) E4BF 84BA 399F 68AD


表示の更新間隔は、動作状況(以下の要因)によって指定した更新間隔より長くなる場合があります。

- ホストマシンの性能/負荷状況
- 通信インタフェース
- ウィンドウのサイズ(メモリ表示範囲)や表示枚数

表4.16 アクセス属性の背景色一覧

アクセス属性		背景色
リードアクセス		緑色
ライトアクセス		赤色
エラー検出時	未初期化メモリ (ライトアクセスしていない領域をリードアクセスした)	黄色
	未参照メモリ (ライトアクセスした領域をリードアクセスしていない)	水色
アクセスなし		白色
ロスト発生後アクセスなし		灰色

(a) RAMモニタの更新間隔を設定する

[RAMモニタ] ウィンドウのポップアップメニューから[表示更新間隔設定] を選択するか、ツールボタン () をクリックすると、[表示更新間隔設定] ダイアログボックスが表示されます。

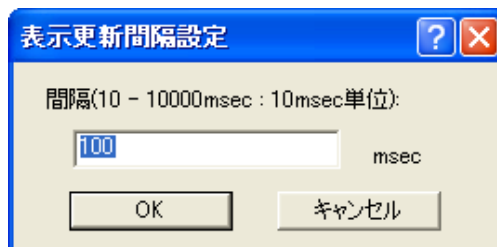



図4.47 [表示更新間隔設定] ダイアログボックス

更新間隔はウィンドウごとに指定することができます。初期値は 100ms です。

(b) 初期化抜け検出を設定する

[RAMモニタ] ウィンドウのポップアップメニューから [初期化抜け検出設定] を選択するか、ツールバーボタン () をクリックすると[初期化抜け検出] ダイアログボックスが表示されます。

本ダイアログボックスで登録した初期化抜け検出領域は、ライトアクセスしていない領域をリードアクセスした場合に「初期化抜け」と判断して黄色でエラー出力表示します。また、ライトアクセスした領域を一度もリードアクセスしていない場合に「未参照」と判断して水色でエラー出力表示します。

初期化抜け検出の登録領域は、[RAMモニタ領域設定] ダイアログボックスの登録リストにも「Uninitialization」と表示されます。

なお、ユーザプログラム実行中は、初期化抜け検出領域を設定することはできません。

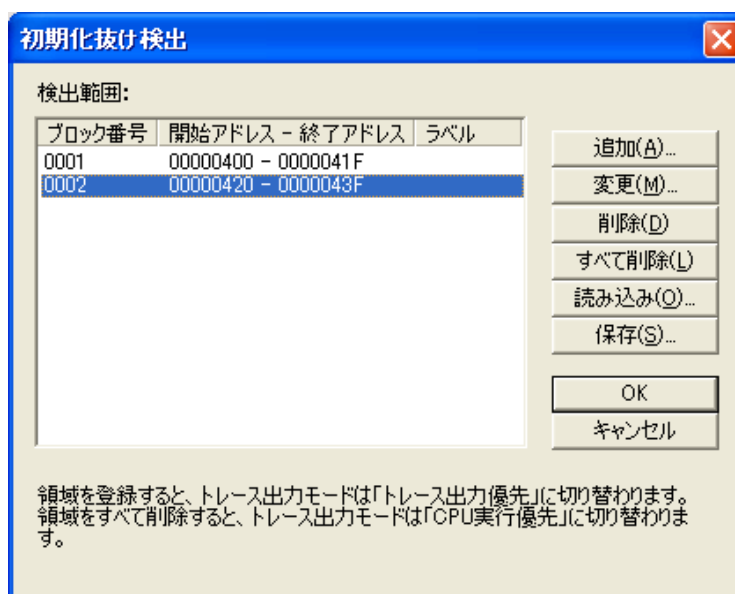


図4.48 [初期化抜け検出] ダイアログボックス

- エラー検出領域の追加

新規にエラー検出領域を登録する場合、[追加] ボタンをクリックしてください。

[初期化抜け検出設定] ダイアログが表示されます。

エラー検出したい領域をセッション名、変数名、またはアドレス範囲から登録してください。

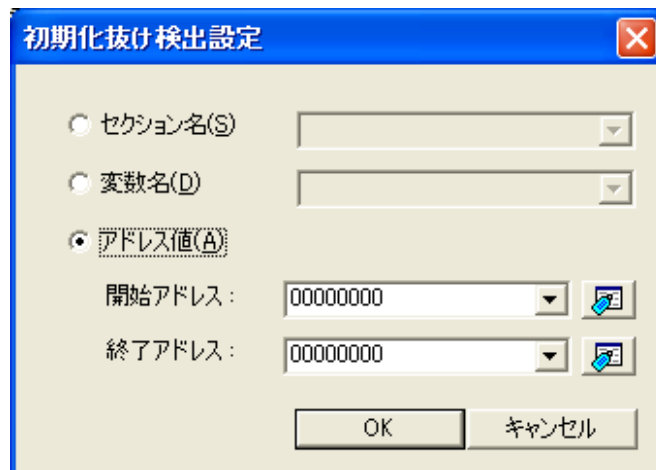


図4.49 [初期化抜け検出設定] ダイアログボックス

- エラー検出領域の変更

登録されているエラー検出領域を選択して[変更] ボタンをクリックしてください。

[初期化抜け検出設定] ダイアログボックスが開きます。

- エラー検出領域の削除

登録されているエラー検出領域を選択して[削除] ボタンをクリックしてください。

登録されている全てのエラー検出領域を削除する場合、[すべて削除] ボタンをクリックしてください。

- エラー検出領域設定内容の読み込み

[初期化抜け検出] ダイアログボックスの[読み込み] ボタンをクリックしてください。

[ファイルを開く] ダイアログボックスが開きます。

読み込むファイル名を指定してください。

読み込みを行った時、読み込み前のエラー検出領域設定は破棄して、読み込んだ設定で再設定します。


- エラー検出領域設定内容の保存

[初期化抜け検出] ダイアログボックスの[保存] ボタンをクリックしてください。

[名前を付けて保存] ダイアログボックスが開きます。

保存先ファイル名を指定してください。ファイルの拡張子は .uni です。省略した場合は、拡張子 .uni が付加されます。

(c) エラー検出を表示する

[RAMモニタ] ウィンドウのポップアップメニューから [エラー検出表示] を選択するか、ツールバーボタン () をクリックしてください。[初期化抜け検出] ダイアログボックスにて登録した検出範囲が [RAMモニタ] ウィンドウで抽出表示されます。このとき、リードアクセスおよびライトアクセスの表示はされません。

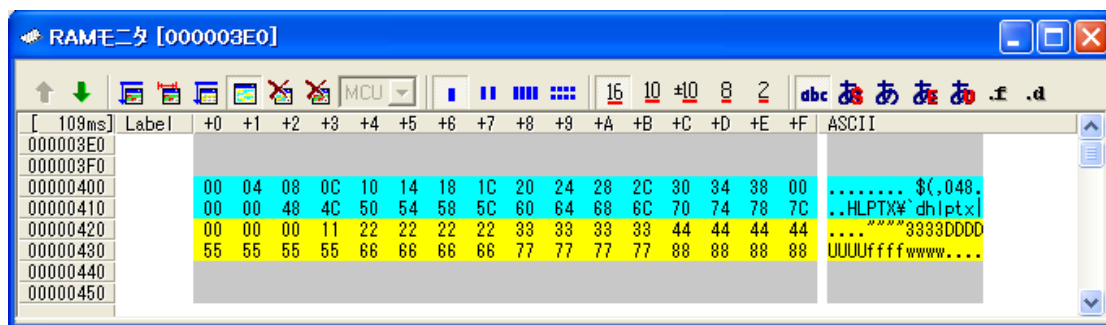




図4.50 [RAMモニタ] ウィンドウ(エラー検出表示)

(d) エラー検出情報をクリアする

[RAMモニタ] ウィンドウのポップアップメニューから [エラー検出情報の消去] を選択するか、ツールバーボタン () をクリックすると、RAMモニタ領域すべての未初期化メモリ、未参照メモリの検出情報をクリアします。

ユーザプログラム実行中に本処理を行うとメモリのリードサイクルが発生するためリアルタイム性は損なわれます。

(e) RAMモニタのアクセス履歴をクリアする

[RAMモニタ] ウィンドウのポップアップメニューから [アクセス情報の消去] を選択するか、ツールバーボタン () をクリックすると、RAMモニタ領域すべてのアクセス履歴を消去します。

ユーザプログラム実行中に本処理を行うとメモリのリードサイクルが発生するためリアルタイム性は損なわれます。

4.10.4 ロストについて

トレース出力でロストが発生すると、RAMモニタウィンドウで表示される内容と実際のメモリ内容が一致しなくなります。ロストが発生したデータアクセスを特定することはできません。

RAMモニタウィンドウでは、ロストが発生するとアクセス履歴をすべて灰色 (ロストを示す) で表示し、ロスト後に発生したデータアクセスから再度アクセス履歴を保持します。なお、RAMモニタの更新間隔よりも短い間隔でロストが発生するプログラムではRAMモニタウィンドウのアクセス履歴は常に灰色となる可能性があります。この場合、「4.10.5 ロスト発生時の抑止」をご参照ください。

4.10.5 ロスト発生を抑止

以下の設定によって、RAM モニタ中のアクセスデータのロスト発生を抑えることができます。

(a) トレース優先出力を選択する

[表示 -> イベント -> トレース条件設定] を選択し[トレース条件設定] ダイアログボックスを開き、出力モードを[トレース出力優先] に設定してください。トレース出力優先モードに設定すると、RAM モニタ中のロストは発生しなくなりますが、トレース出力を優先させるために、CPUが一時的に停止する場合がありますため、プログラム実行のリアルタイム性は損なわれます。

(b) トレース抽出機能を使用する

「4.10.2 RAMモニタ範囲を割り当てる」の設定だけでは、全てのメモリアクセスに対するトレースデータ出力が行われているため、モニタ対象の範囲でロストが発生していなくても、それ以外の領域でロストが発生してしまうと、[RAMモニタ] ウィンドウ上ではロストと表示されます。

このため、トレース抽出機能によりトレース出力するデータを、監視したいデータ範囲のみに絞り込むことで、ロストの発生を減らすことが可能です。なお、[RAMモニタ] ウィンドウでは、監視範囲外のデータ範囲はグレー表示になります。

この場合、プログラム実行のリアルタイム性は維持されますが、モニタできる範囲は小さくなります。

ただし、絞り込んだ範囲のデータアクセスが頻繁に発生し、ロストを起こす場合は、この方法でもロストを抑止することはできません。

トレース抽出条件は、[表示 -> イベント -> トレース条件設定] により表示される[トレース条件設定] ダイアログボックスで設定できます。

以下に設定例を示します。

設定例：0x1700～0x171F番地のデータを監視したい場合、「4.10.2 RAMモニタ範囲を割り当てる」の設定で、0x1400～0x17FF番地の設定をした後に、トレース抽出機能で、0x1700～0x171F番地のみ出力するように設定します。

- (1) [トレース条件設定] ダイアログでトレース抽出条件の[抽出条件] ラジオボタンを選択してください。

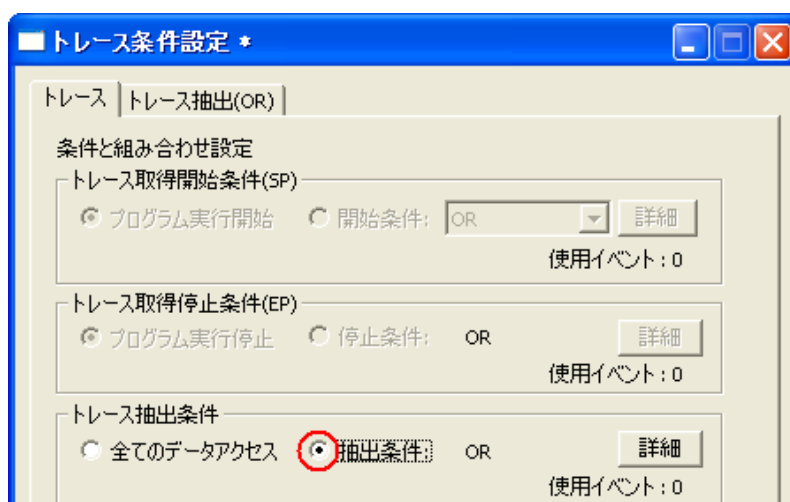


図4.51 トレース抽出機能

- (2) [トレース抽出] ページを選択して[追加] ボタンを押してください。[イベント設定] ダイアログボックスが表示されるので下図の通りに設定後、[OK] ボタンを押してください。イベント設定後は、必ず[適用] ボタンを押してください。

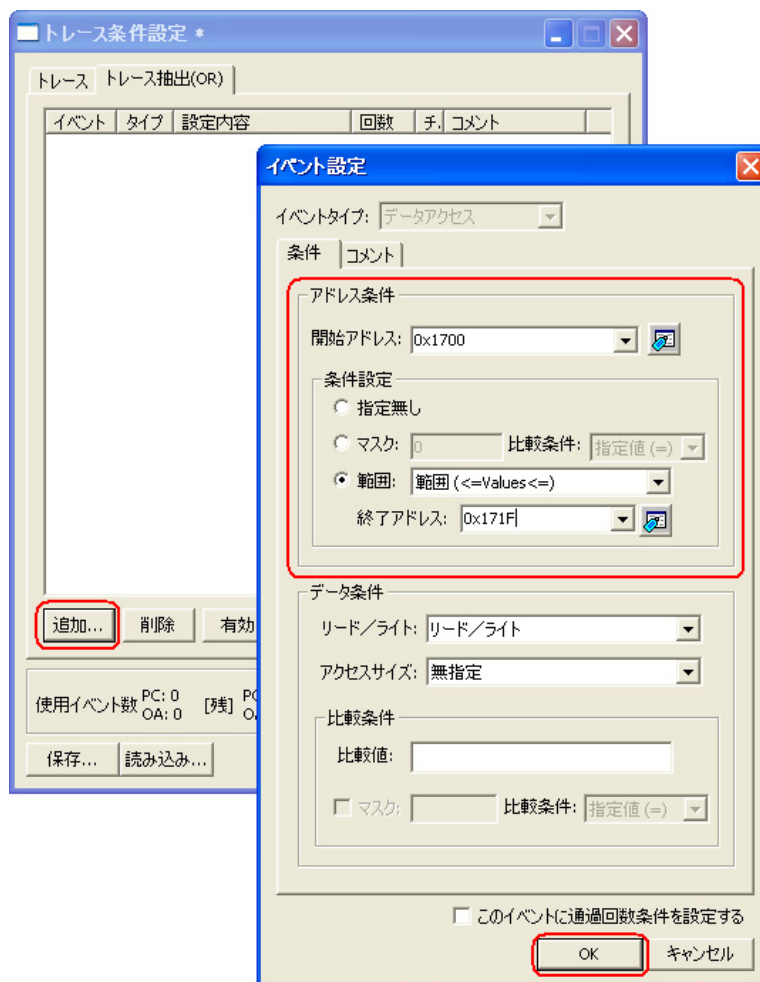


図4.52 アドレス条件の設定

- (3) トレース抽出条件で指定した監視範囲(0x1700~0x171F番地)のトレースデータが[RAMモニタ] ウィンドウに表示されます。

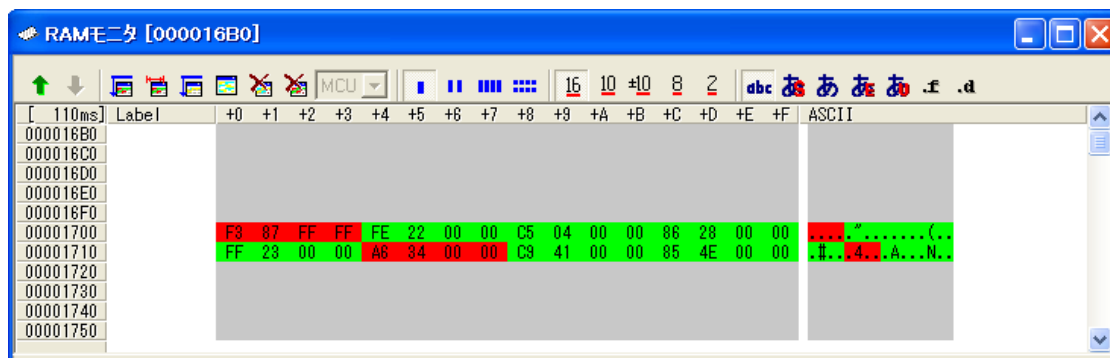


図4.53 [RAM モニタ] ウィンドウ(監視範囲指定)

4.11 Start/Stopファンクション機能

Start/Stop ファンクションは、ユーザプログラムの実行開始直前および停止直後にユーザプログラムの指定ルーチンを実行する機能です。

ユーザプログラムの実行や停止に同期してユーザシステムの制御を行いたい場合に、任意の指定ルーチンを設定します。

4.11.1 [Start/Stopファンクション設定]ダイアログボックスを開く

ユーザプログラムの実行開始直前および停止直後に実行するルーチンは、[Start/Stop ファンクション設定] ダイアログボックスで指定します。

メニューから、[基本設定 -> エミュレータ -> Start/Stop ファンクション設定...] を選択すると[Start/Stop ファンクション設定] ダイアログボックスがオープンします。

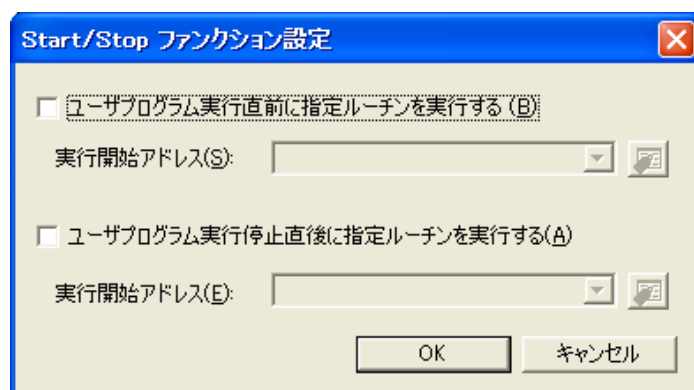


図4.54 [Start/Stop ファンクション設定] ダイアログボックス

4.11.2 実行するルーチンを指定する

ユーザプログラム実行直前に実行するルーチンと停止直後に実行するルーチンは個別に指定が可能です。

[ユーザプログラム実行直前に指定ルーチンを実行する] チェックボックスをチェックすると、同チェックボックスの下側にある[実行開始アドレス] コンボボックスで指定したルーチンがユーザプログラム実行直前に実行されます。

[ユーザプログラム実行停止直後に指定ルーチンを実行する] チェックボックスをチェックすると、同チェックボックスの下側にある[実行開始アドレス] コンボボックスで指定したルーチンがユーザプログラム実行停止直後に実行されます。

4.11.3 Start/Stopファンクションの制限事項

Start/Stop ファンクションには以下の制限があります。

- Start/Stopファンクション使用時は、以下のデバッグ機能は使用しないでください。
 - a) 指定ルーチンのプログラム領域へのメモリ設定およびダウンロード
 - b) 指定ルーチンのプログラム領域へのブレークポイント設定
- 指定ルーチン実行時は、割り込みスタックポインタが指し示す4バイトをエミュレータ側の制御により使用します。
- 指定ルーチンで使用する汎用レジスタおよびフラグについては以下の制限があります。

表4.17 レジスタおよびフラグの制限事項

レジスタ/フラグ名	制限事項
ISP レジスタ	指定ルーチン終了時に、指定ルーチン実行開始時の値に戻してください。
Uフラグ	指定ルーチンを実行中はユーザモードへの切り替えは禁止になります。
Iフラグ	指定ルーチンを実行中は割り込み禁止になります。
PMフラグ	指定ルーチンを実行中はユーザモードへの切り替えは禁止になります。

- 指定ルーチン実行時は、以下のデバッグ機能は無効です。
 - a) トレース
 - b) ブレーク関連
 - c) RAMモニタ
 - d) パフォーマンス
 - e) 指定ルーチン内のイベント設定
- 指定ルーチン実行時は、ノンマスカブル割り込みを常に禁止にします。
- 指定ルーチン実行後にユーザプログラムを開始する場合、マイコンの状態は以下のようになります。

表4.18 ユーザプログラム開始時のマイコンの状態

マイコン資源	状態
マイコン汎用レジスタ	ユーザプログラム停止時の状態またはユーザがレジスタウィンドウで設定したレジスタ値となります。指定ルーチン実行後のレジスタ内容は反映されません。
マイコン空間のメモリ	指定ルーチン実行後のメモリアクセスが反映されます。
マイコン周辺機能	指定ルーチン実行後のマイコン周辺機能動作が継続されます。

4.11.4 指定ルーチンの記述に関する制限事項

指定ルーチンの記述に関して以下の制限があります。

- 指定ルーチン内でスタックを使用する場合は、必ず割り込みスタックを使用してください。
- 指定ルーチン処理の終了は、リターンサブルーチン命令を記述してください。
- 1回の指定ルーチンの処理時間は100ms以内に終了させてください。指定ルーチン内でクロック停止状態を継続させた場合などはエミュレータの制御ができなくなる可能性があります。
- 指定ルーチン実行開始時のレジスタ値は不定です。指定ルーチン内でレジスタ値の初期設定をしてください。
- 指定ルーチンは開始時にスーパーバイザモードで実行します。ユーザモードに切り替えないでください。
- JTAGインタフェースの場合、Startファンクション実行後、約20msでユーザプログラムを実行します。
(CPUクロック：100MHz, JTAGクロック：16.5MHz)
- JTAGインタフェースの場合、ユーザプログラム実行後から、約20msでStopファンクションを実行します。
(CPUクロック：100MHz, JTAGクロック：16.5MHz)
- FINEインタフェースの場合、Startファンクション実行後、約30msでユーザプログラムを実行します。
(CPUクロック：100MHz, FINEボーレート：2000000bps)
- FINEインタフェースの場合、ユーザプログラム実行後から、約40msでStopファンクションを実行します。
(CPUクロック：100MHz, FINEボーレート：2000000bps)

4.12 デバッグコンソール機能

デバッグコンソール機能は、ユーザプログラムの標準入出力をサポートする機能です。


標準入出力は、[デバッグコンソール] ウィンドウ上で行います。

ユーザプログラム中の低水準インタフェースルーチン(`_charput`、`_charget`) を置き換えることで実現します。

【留意事項】

入出力機能を利用する場合は、必ず [デバッグコンソール] ウィンドウをオープンしておいてください。

4.12.1 [デバッグコンソール]ウィンドウを開く

[デバッグコンソール] ウィンドウを開くには、[表示->CPU->デバッグコンソール] を選択するか、[デバッグコンソール] ツールバーボタン()をクリックします。

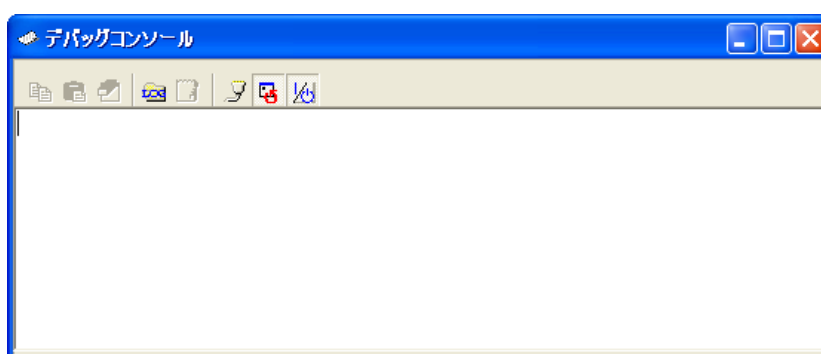


図4.55 [デバッグコンソール]ウィンドウ

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。

4.12.2 低水準インタフェースルーチンについて

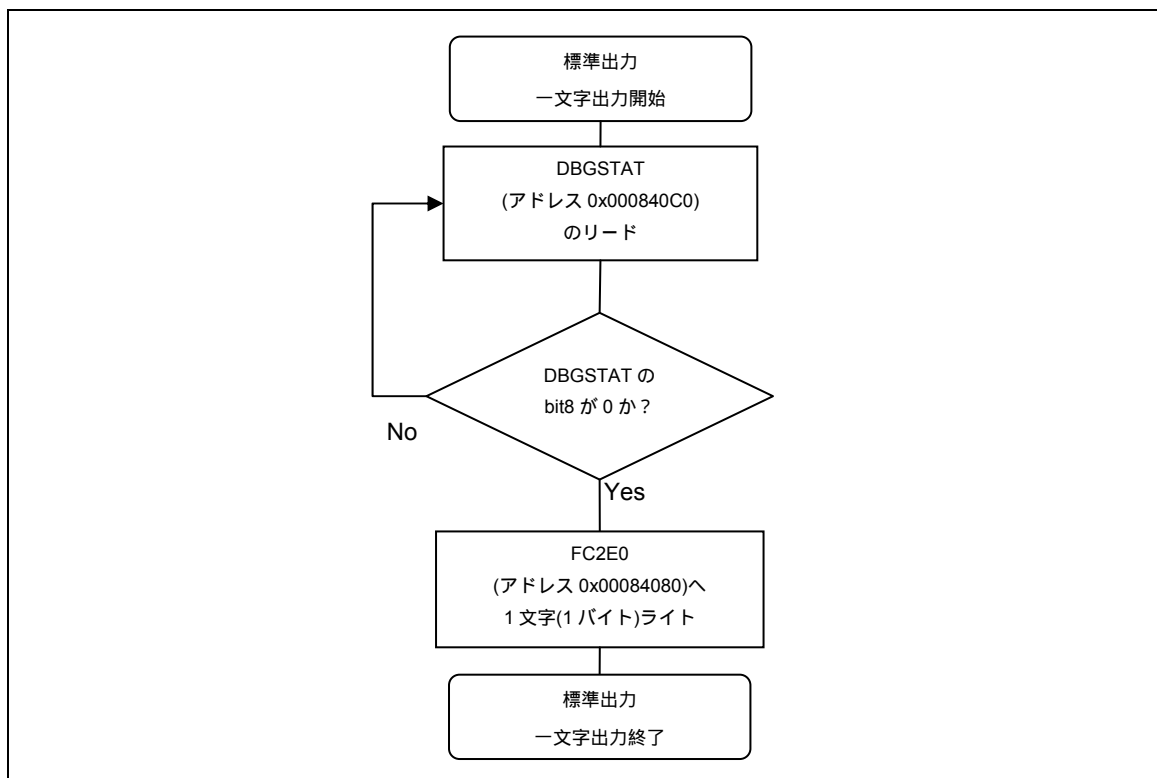
標準入出力を C/C++プログラムで使用する場合は、低水準インタフェースルーチンを作成します。

デバッグコンソール機能を実現するためには、この低水準インタフェースルーチンから呼ばれる、一文字の入出力を行う関数 `charput`、`charget(_charput, _charget)` をデバッグコンソール機能用のプログラムに置き換える必要があります。

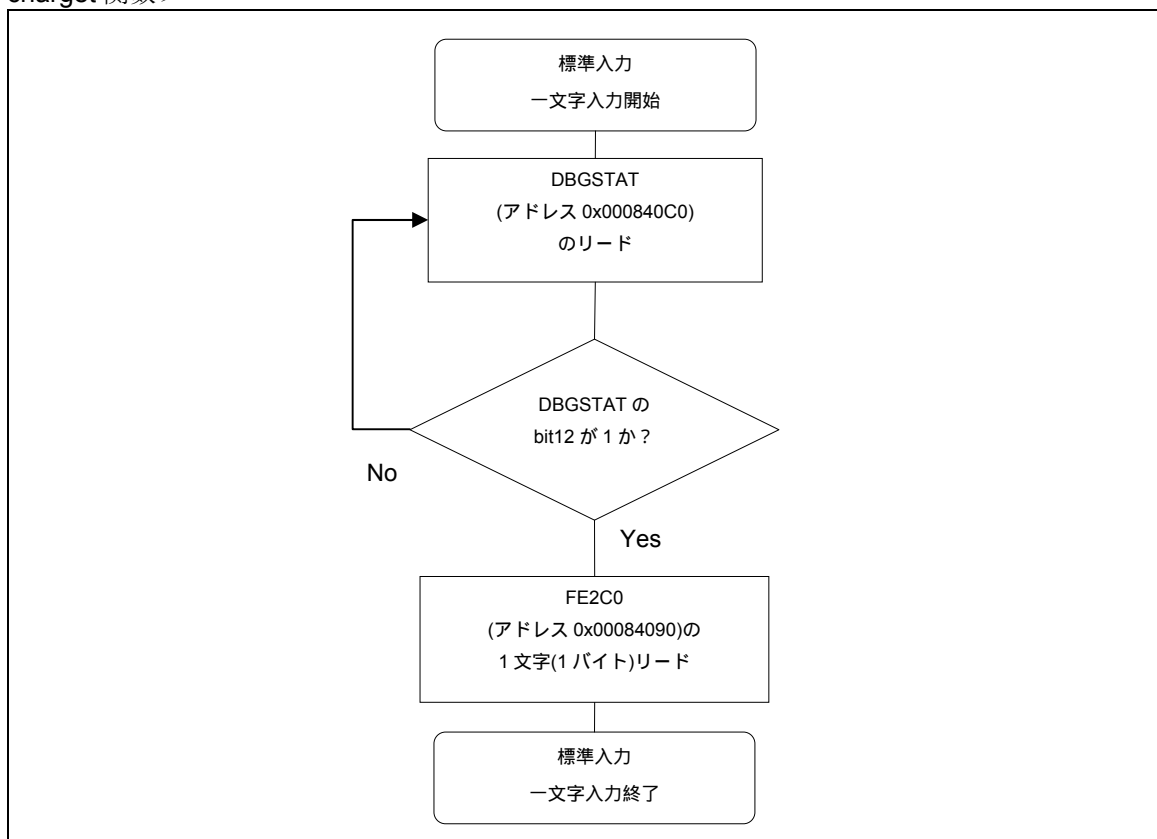
デバッグコンソール機能用に修正する場合、特定のアドレスに対して一文字の入出力を行う関数に変更する必要があります。次頁に処理フローとサンプルプログラムを示します。

【注】 サンプルプログラムを含め、この処理フローに従ったプログラムはデバッガのデバッグコンソール機能用のプログラムになります。エミュレータを接続しない場合、デバッグコンソール機能用のプログラムをユーザプログラムに含めないでください。

charput 関数フロー



charget 関数フロー



デバッグコンソール機能用サンプルプログラム

```

;-----
;
; FILE      :lowlvl.src
; DATE      :Wed, Jul 01, 2009
; DESCRIPTION      Program of Low level
; CPU TYPE      RX
;
;-----

.GLB      _charput
.GLB      _charget

FC2E0    .EQU 00084080h
FE2C0    .EQU 00084090h
DBGSTAT  .EQU 000840C0h
RXFL0EN  .EQU 00001000h
TXFL0EN  .EQU 00000100h
.SECTION P,CODE
;-----
; _charput:
;-----
_charput:
.STACK   _charput = 00000000h
__C2ESTART:  MOV.L #TXFL0EN,R3
             MOV.L #DBGSTAT,R4
__TXLOOP:   MOV.L [R4],R5
             AND   R3,R5
             BNZ   __TXLOOP
__WRITEFC2E0: MOV.L #FC2E0,R2
             MOV.L R1,[R2]
__CHARPUTEXIT:RTS
;-----
; _charget:
;-----
_charget:
.STACK   _charget = 00000000h
__E2CSTART: MOV.L #RXFL0EN,R3
             MOV.L #DBGSTAT,R4
__RXLOOP:   MOV.L [R4],R5
             AND   R3,R5
             BZ    __RXLOOP
__READFE2C0: MOV.L #FE2C0,R2
             MOV.L [R2],R1
__CHARGETEXIT:RTS
;-----
.END

```


4.13 ホットプラグイン機能

ホットプラグインは、実行中のユーザプログラムにE1/E20エミュレータを接続し、実行中のプログラムをデバッグする起動方式です。

なお、E1エミュレータでホットプラグインを使用する場合、E1エミュレータとユーザシステムの間で別売の“ホットプラグアダプタ”(R0E000010ACB00)を接続する必要がありますので、ご注意ください。

以下にホットプラグイン機能の対応表を示します。

表4.19 製品名とホットプラグイン機能対応表

製品名 (製品型名)	ホットプラグイン機能使用
E1 (R0E000010KCE00)	使用できません。
E1 (R0E000010KCE00) + ホットプラグアダプタ (R0E000010ACB00)	使用できます。
E20 (R0E000200KCT00)	使用できます。

4.13.1 起動手順

以下にホットプラグインでの起動手順を記します。指示があるまでエミュレータとユーザシステムを接続しないでください。

- (1) ユーザシステムとエミュレータを接続しない状態で[スタート]メニューより[プログラム -> Renesas -> High-performance Embedded Workshop -> High-performance Embedded Workshop]を選択してご使用のワークスペースを選択してください。
- (2) [起動設定]ダイアログボックスが表示されます。[動作モード]の[デバッグモード]を選択して[ホットプラグイン]チェックボックスをチェックしてください。ユーザシステムとエミュレータが接続されていないこと、および[通信設定]のエミュレータシリアルNo.が表示されていることを確認して[OK]ボタンをクリックしてください。
 なお、[ホットプラグイン]チェックボックスの状態は、次回起動時に無効となります。
 また、その他の設定については「3.3.1 [起動設定]ダイアログボックス」を参照してください。

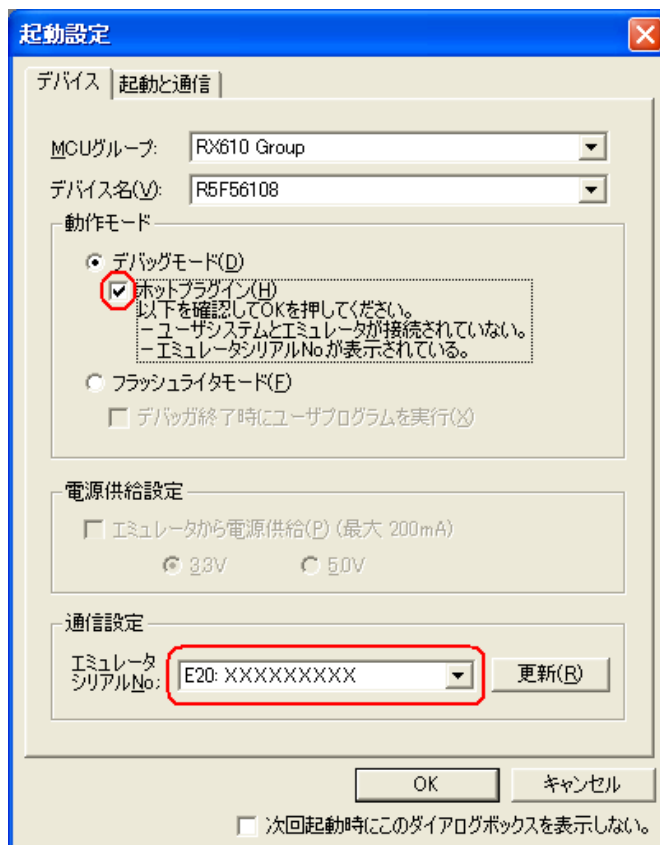


図4.56 ホットプラグインでの[起動設定] ダイアログボックス-[デバイス] ページ

【注意事項】

[起動設定]ダイアログボックスで表示される内容は、マイコン毎に異なる場合があります。

- (3) [Connect] ダイアログボックスが表示されます。ご使用のエミュレータの接続方法に従ってユーザシステムとエミュレータを接続してください。なお、ファームウェアの書き換えが発生する場合は、本ダイアログボックスが表示される前に実行されます。ファームウェアの書き換えについては「(5) [ファームウェアの書き換え] ダイアログボックス」を参照してください。

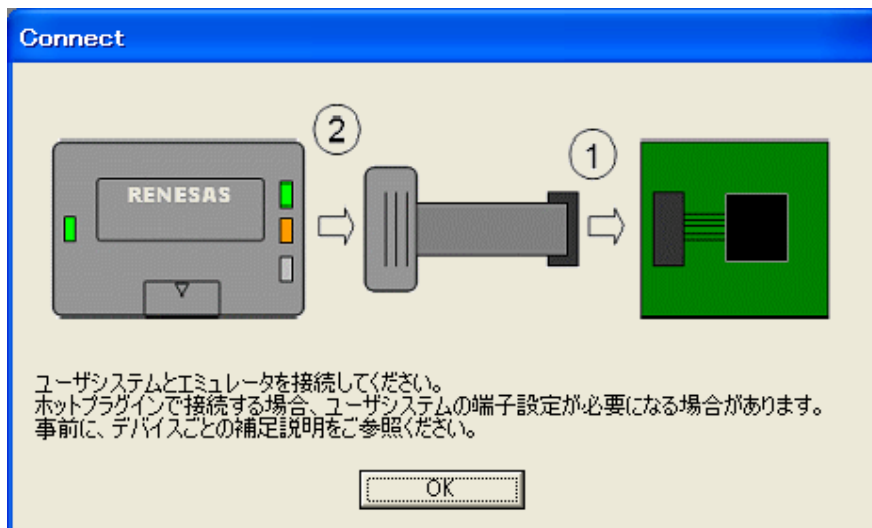


図4.57 [Connect] ダイアログボックス(E20 エミュレータ)

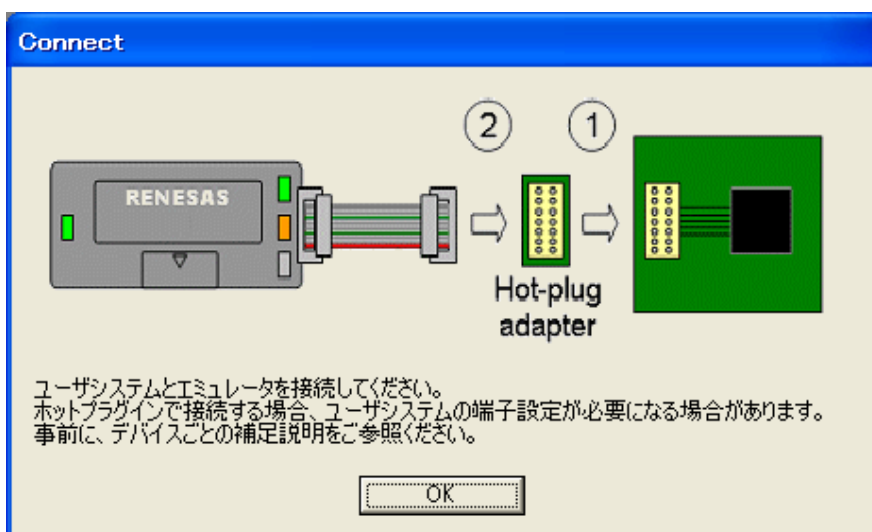


図4.58 [Connect] ダイアログボックス(E1 エミュレータ)

- (4) [ID コード確認] ダイアログボックスが表示されます。ターゲットマイコンの内蔵フラッシュメモリのセキュリティのため、設定した ID コード【注】を入力してください。なお、ID コードを設定していない場合は、入力モード(Hex / ASCII) に合わせて指定桁数分“0xFF”を入力してください。

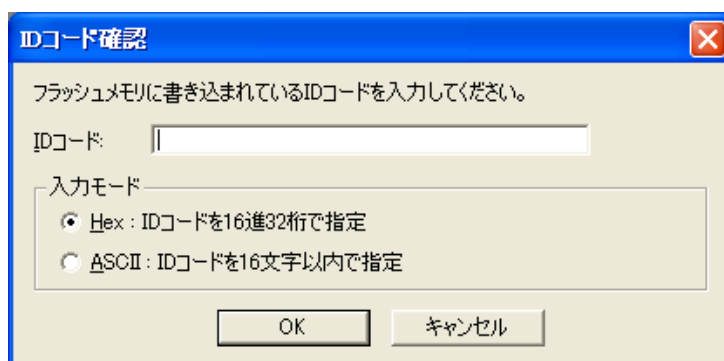


図4.59 ホットプラグインでの[ID コード確認] ダイアログボックス

【注】 ID コードについては、ご使用のマイコンのハードウェアマニュアルを参照してください。

- (5) [接続中...] ダイアログボックスが表示されます。



図4.60 ホットプラグインでの[接続中...] ダイアログボックス

- (6) [コンフィグレーションプロパティ] ダイアログボックスが表示されます。エミュレータおよびデバッグ機能に関する設定を行い、[OK] ボタンをクリックしてください。
 なお、各設定については「3.3.2 [コンフィグレーションプロパティ] ダイアログボックス」を参照してください。

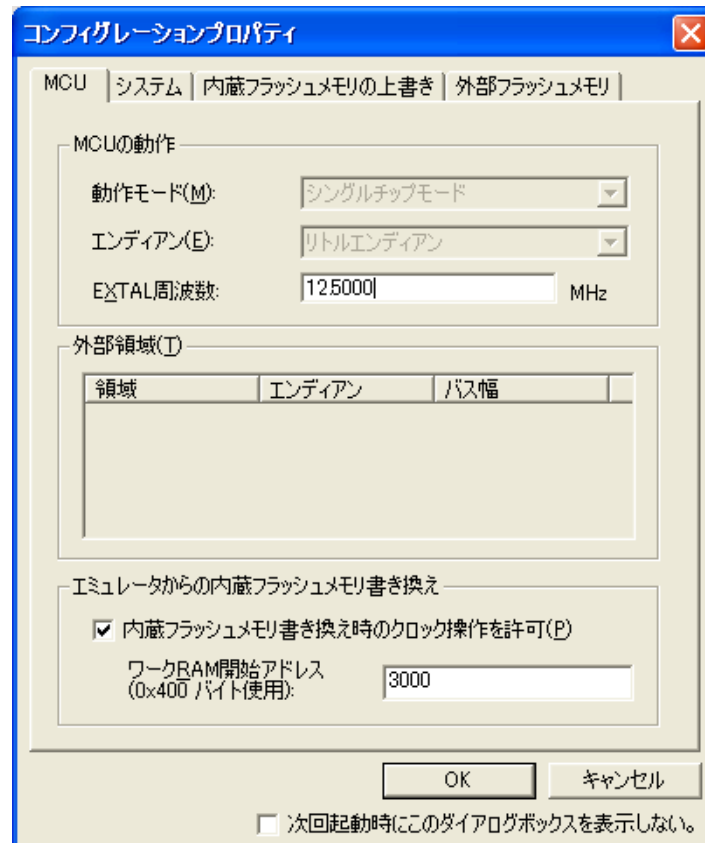


図4.61 ホットプラグインでの[コンフィグレーションプロパティ] ダイアログボックス—[MCU] ページ

【注意事項】

[コンフィグレーションプロパティ] ダイアログボックスで表示される内容は、マイコン毎に異なる場合があります。

- (7) High-performance Embedded Workshop の[Output] ウィンドウに”Connected”が表示され、以下のメッセージボックスが表示されます。[OK] ボタンを押してダウンロードモジュールファイルのデバッグ情報のみをダウンロードし、[実行] ボタンを押してください。

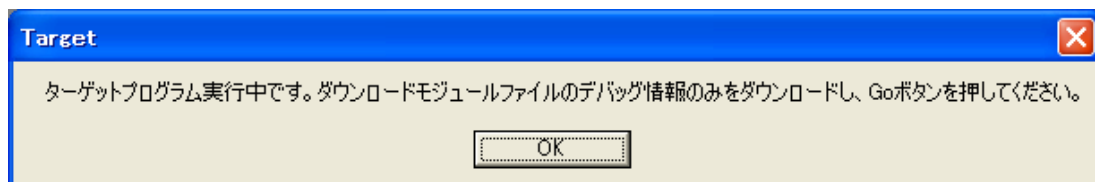


図4.62 ホットプラグイン接続完了時のメッセージボックス

【留意事項】

1. エミュレータとユーザシステムが接続された状態で、[起動設定] ダイアログボックスの[ホットプラグイン] チェックボックスをチェックして[OK] ボタンをクリックすると以下のエラーメッセージが表示されます。

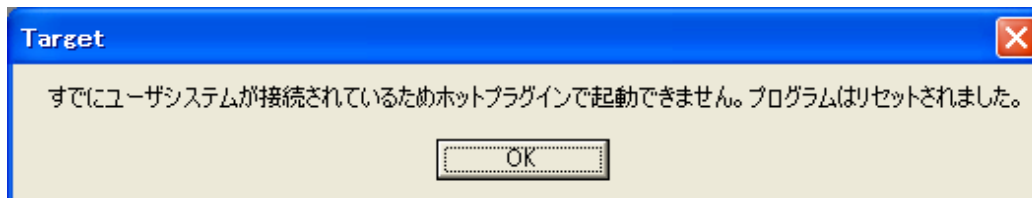


図4.63 [ユーザシステム接続エラー] ダイアログボックス

2. エミュレータとユーザシステムが接続されていない状態で、[Connect] ダイアログボックスの[OK] ボタンをクリックすると以下のエラーメッセージが表示されます。

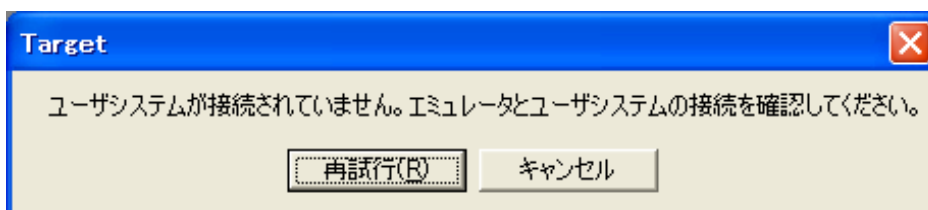


図4.64 [ユーザシステム未接続エラー] ダイアログボックス

3. ユーザシステム上に外部から電源が供給されていない場合、以下のエラーメッセージが表示されます。

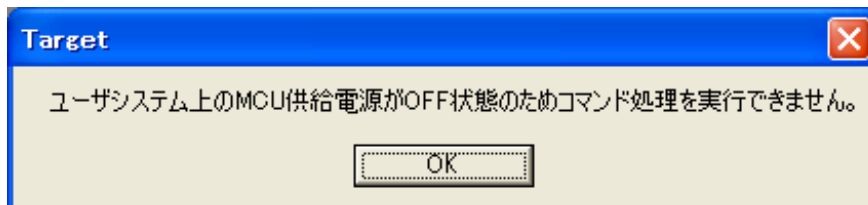


図4.65 [コマンド処理エラー] ダイアログボックス

4.13.2 ホットプラグインの注意事項


ホットプラグインには以下の注意事項があります。

- ホットプラグイン接続完了後、[実行] ボタンを押すまでの間はイベントの設定変更、ブレークポイント設定、およびその他のデバッグ機能を使用しないでください。また、セッション情報に保存されたソフトウェアブレークポイントはホットプラグイン接続では設定されません。
- ホットプラグイン接続完了後、[実行] ボタンを押すまでの間に[コマンドライン] ウィンドウおよび[メモリ] ウィンドウにてメモリ比較を実施しないでください。いずれかで実施した場合、正しいメモリ比較結果が表示されません。
- ホットプラグイン接続完了後、最初に[実行] ボタンを押してから停止するまでの間はトレース機能を使用できません。
- ホットプラグイン接続完了後、ステップイン、ステップオーバー等の実行操作でも[実行] ボタンを押したときと同じ動作になります。
- ホットプラグインで起動した直後は、[レジスタ] ウィンドウの表示と実際のレジスタ値は異なります。
- ホットプラグインで起動する場合、リアルタイム RAM モニタを使用できません。（[コンフィグレーションプロパティ] ダイアログボックスにてグレー表示になります。）
- ホットプラグインでは、ID コードを確認する為、ユーザプログラムを一時的に約 800 μ s 停止させます。（CPU クロック：100MHz, JTAG クロック:16.5MHz）
- FINE インタフェースによるホットプラグイン起動は対応していません。
- RX630, RX631, RX63N, RX63T グループマイコンを用いたホットプラグインでは、マイコンに書かれたエンディアン選択レジスタ(MDEB, MDES) のエンディアン設定値と[起動設定] ダイアログボックスで設定するエンディアンを必ず合わせてください。エンディアン選択レジスタ(MDEB, MDES) については、関連するマイコンのハードウェアマニュアルを参照してください。また、フラッシュ書き換えプログラムが動作中にホットプラグインはできません。

4.14 グラフ機能

メモリ内容の変化をグラフ表示するには、[グラフ] ウィンドウを使用します。
[グラフ] ウィンドウでは登録したシンボルの値をグラフィカルに表示します。

4.14.1 [グラフ] ウィンドウを表示する

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

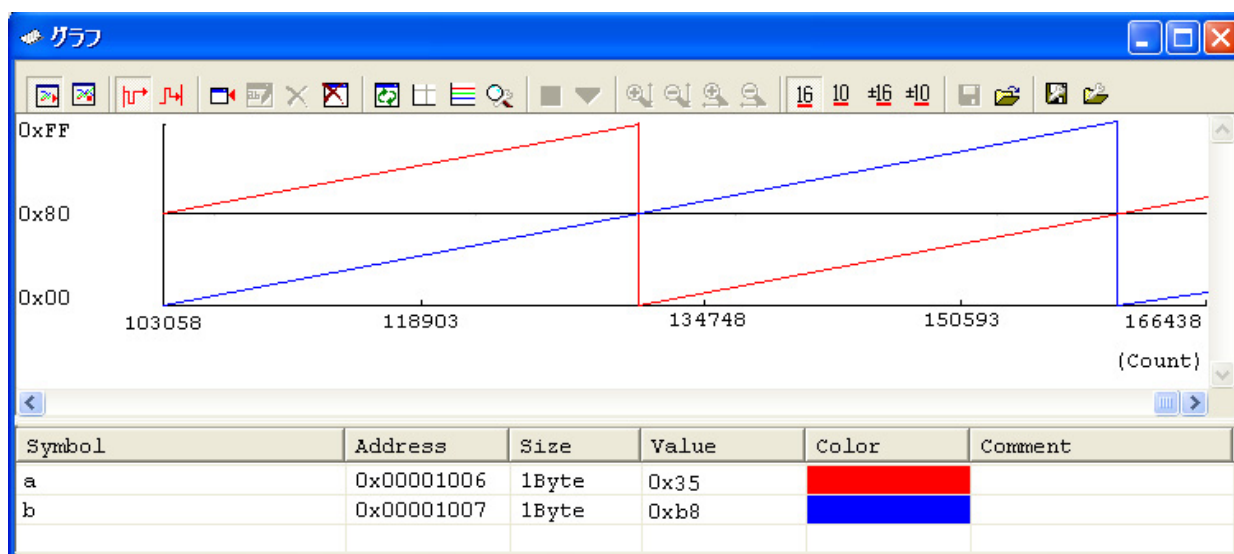




図4.66 [グラフ] ウィンドウ

(1) モードの選択

[グラフ] ウィンドウには、サンプリングモードと、トレースモードがあります。[グラフ] ウィンドウの上部ウィンドウ内でマウスの右ボタンをクリックしてポップアップメニューを表示し、[モード]-[サンプリング]、[モード]-[トレース]を選択するか、[グラフ] ウィンドウのツールバーボタン( )で切り替えを行います。

サンプリングモードは、プログラム実行中にサンプリング周期に設定された時間が経過するたびに値が計測され表示が行われます。トレースモードは、トレースメモリの情報を解析し表示を行います。

トレースモードにはフルと、フリーの2つの方式があります。フルモードは、トレースフルもしくは、プログラム停止のタイミングで、トレースメモリを解析し、表示が行われます。フリーモードは、プログラム停止のタイミングで、トレースメモリを解析し、表示が行われます。

トレースモードは、トレースメモリの解析に時間を要するため、表示タイミングでは全ての解析を行いません。右スクロールや、縮小操作を行ったタイミングで表示位置までの解析が行われます。

登録シンボルは、サンプリングモードと、トレースモードで共通ではありません。使用するモード別に設定を行う必要があります。


【注】 サンプルモードおよびトレースフリーモードでは、最大保持データを超えると過去データから順に削除されます。

登録シンボルは、サンプルモードで各[グラフ] ウィンドウ毎に最大10個、トレースモードで全[グラフ] ウィンドウ合わせて最大4個です。

[コンフィグレーション]プロパティダイアログの[システム] ページにある排他機能が、[リアルタイム RAM モニタ]で接続された場合、トレースモードは使用できません。

[コンフィグレーション]プロパティダイアログの[システム] ページにある排他機能を[トレース条件設定]で接続し、サンプルモードで使用した場合、リアルタイムRAMモニタではなく通常のメモリから値を取得します。

(2) シンボルの登録

[グラフ] ウィンドウのポップアップメニューから[シンボル追加]を選択するか、[グラフ] ウィンドウのツールバーボタン()をクリックすると以下のダイアログボックスが表示されます。

また、[エディタ] ウィンドウから[グラフ] ウィンドウの下部ウィンドウに評価可能な文字列をドラック&ドロップすることでも登録可能です。

トレースモードでシンボルを登録した場合、一部のトレース機能が使用できなくなります。

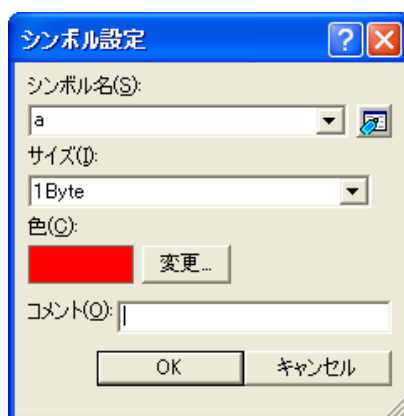


図4.67 [シンボル設定] ダイアログボックス

(3) サンプルング周期の設定


[グラフ] ウィンドウがサンプルモードの時にデータ計測する間隔を設定します。[グラフ] ウィンドウのポップアップメニューから[サンプルング周期設定...]を選択するか、[グラフ] ウィンドウのツールバーボタン()をクリックすると以下のダイアログボックスが表示されます。




図4.68 [サンプリング周期設定] ダイアログボックス

サンプリング周期は、ウィンドウごとに指定することができます。

10ms単位で、10ms～10000msの範囲を指定できます。デフォルトは100msです。

(4) 拡大縮小率の設定

拡大、縮小操作が行われたときの、縦横それぞれの拡大率と縮小率を設定します。

[グラフ] ウィンドウのポップアップメニューから[拡大縮小率設定...]を選択するか、[グラフ] ウィンドウのツールバーボタン()をクリックすると以下のダイアログボックスが表示されます。

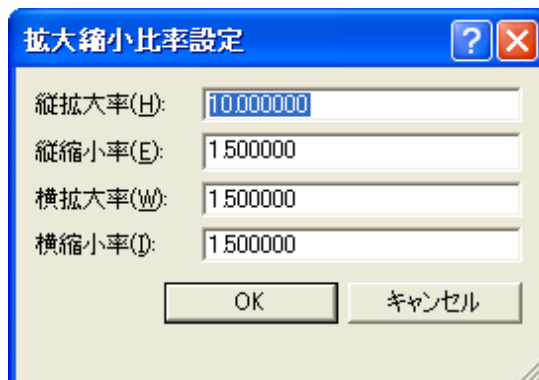


図4.69 [拡大縮小比率設定] ダイアログボックス


縦横それぞれの拡大縮小操作時のグラフ拡大縮小倍率を入力します。

初期値は1.5倍です。

全設定共通で、1以下の数値を指定した場合は1が設定されます。また、小数点を含む数値を指定した場合、数値によっては丸め誤差が発生し、近似値となる可能性があります。

(5) 境界線の設定

最大4つの境界線を表示するための設定を行います。

[グラフ] ウィンドウのポップアップメニューから[境界線設定...]を選択するか、[グラフ] ウィンドウのツールバーボタン()をクリックすると以下のダイアログボックスが表示されます。

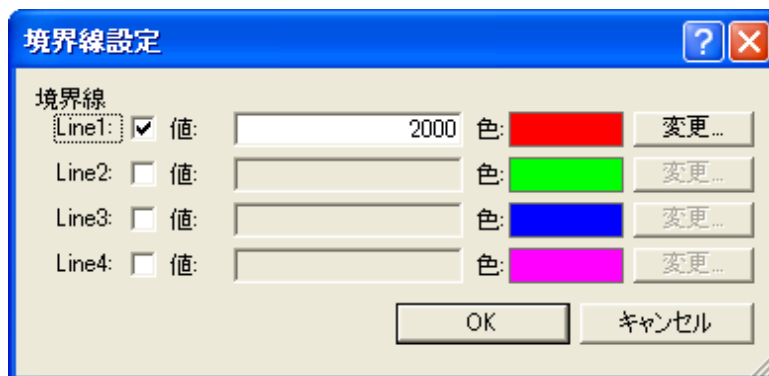


図4.70 [境界線設定] ダイアログボックス

4つの境界線の表示有無を設定します。

表示する境界線については、境界線を表示する値又は、式を入力し、表示色を選択します。

[グラフ] ウィンドウの基数設定が、符号付10進数、符号付16進数の場合は、負の値を設定する事が出来ず。

4.15 スタックトレース機能

E1/E20 エミュレータでは、スタック情報を用いて、現在の PC がある関数がどの関数からコールされているかを表示します。本機能は、Elf/Dwarf2 形式のデバッグ情報を持ったロードモジュールをロードした場合のみ使用できます。

本機能の使用方法については、「5.18 スタックトレース機能」を参照してください。

4.16 オンラインヘルプ

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

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

5. チュートリアル

5.1 はじめに

エミュレータを実際に操作して主な機能を理解していただくためにチュートリアルプログラムを提供しています。本章では、このチュートリアルプログラムを用いて E1/E20 エミュレータの主な機能について説明します。

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

チュートリアルプログラムでは、以下の処理を行います。

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

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

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

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

チュートリアルプログラムは、エミュレータ機能の使用方法、エミュレータデバッガの操作方法を理解していただくために作成したプログラムです。ユーザシステム、ユーザプログラム開発時には、ターゲットマイコンの各種マニュアルをご確認ください。

- 【注】
1. 本プログラムはリトルエンディアンで動作します。ビッグエンディアンで動作させる場合、再コンパイルを行ってください。
 2. 再コンパイルを行った場合、本章で説明しているアドレスとずれることがあります。
 3. 本製品に添付されるチュートリアルプログラムのファイル拡張子、動作アドレスは、マイコンによって異なります。したがって、表示される内容が本章の説明と異なることがあります。

5.2 High-performance Embedded Workshopの起動

「2.10 E1/E20 エミュレータデバッガの起動手順」の手順に従ってワークスペースを開きます。

ディレクトリは以下を指定してください。

OS インストールドライブ¥Workspace¥Tutorial¥E1E20¥RX600¥Tutorial_LittleEndian

ファイルは以下を指定してください。

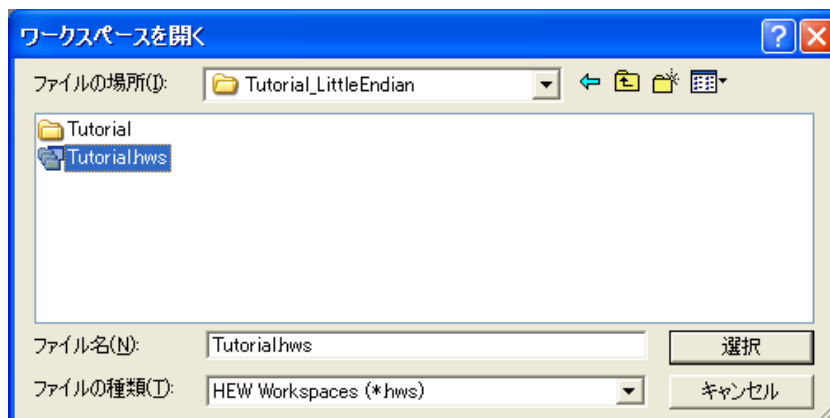


図5.1 [ワークスペースを開く] ダイアログボックス

5.3 エミュレータの接続

エミュレータに接続すると、デバッガをセットアップするためのダイアログボックスが表示されます。このダイアログボックスでデバッガの初期設定を行います。

デバッガのセットアップが終了すると、デバッグできる状態になります。

5.4 チュートリアルプログラムのダウンロード

5.4.1 チュートリアルプログラムをダウンロードする

デバッグしたいオブジェクトプログラムをダウンロードします。なお、ダウンロードするプログラムやダウンロード先のアドレスは使用しているマイコンによって異なります。画面の表示などをご使用のマイコンに合わせて適宜読み替えてください。

[Download modules]の[Tutorial.abs]を選択した状態で右クリックをして、表示されるポップアップメニューから[ダウンロード]を選択します。

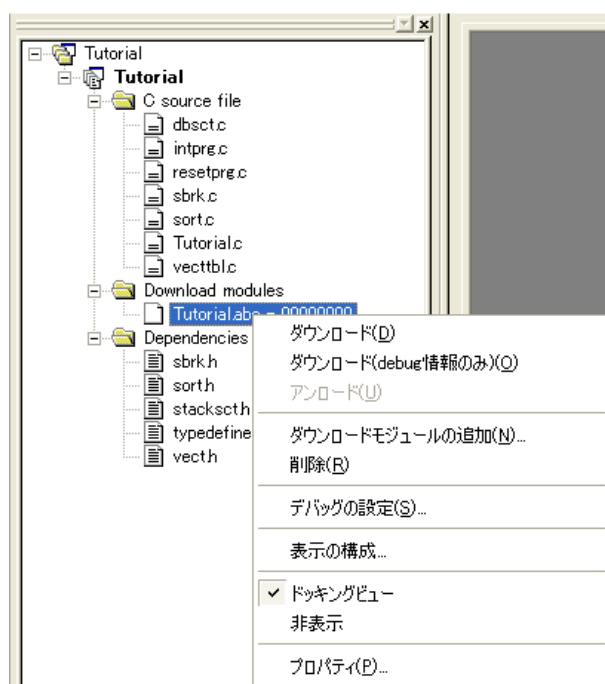


図5.2 チュートリアルプログラムのダウンロード画面

上記操作後、内蔵フラッシュメモリへのダウンロードが開始されます。正常にダウンロードが完了した場合、[アウトプット] ウィンドウに完了メッセージが表示され、[Download modules] の[Tutorial.abs] のアイコン内に矢印が表示されます。

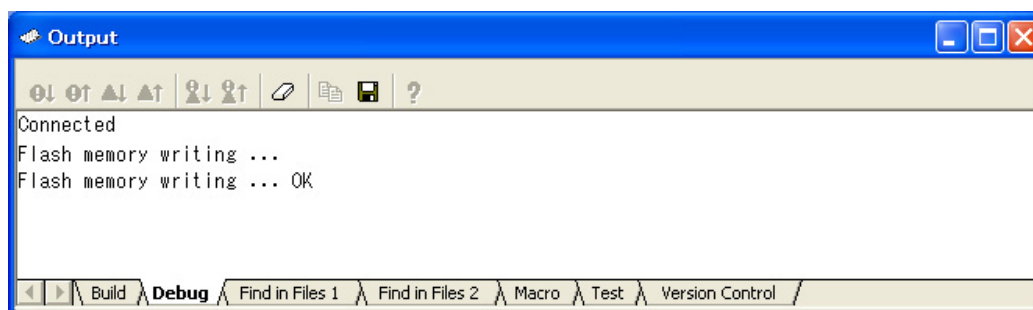


図5.3 チュートリアルプログラムのダウンロード完了画面

なお、[デバッガ] メニューの[デバッグの設定] を選択して表示される[デバッグの設定] ダイアログボックスの[オプション]ページにて[ダウンロード後に CPU リセット] チェックボックスをチェックした場合、ダウンロードする度に CPU リセットが発生します。

5.4.2 ソースプログラムを表示する

High-performance Embedded Workshop では、ソースレベルでプログラムをデバッグできます。
[C source file] の[Tutorial.c] をダブルクリックします。

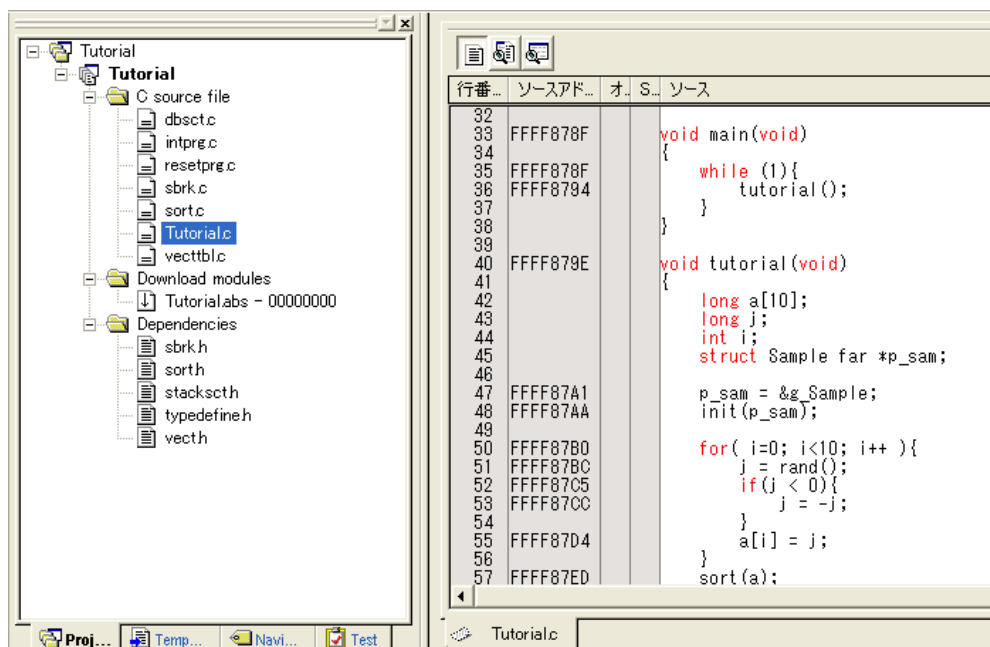


図5.4 [エディタ] ウィンドウ(ソースプログラムの表示)

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

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

5.5 S/Wブレークポイントの設定

簡単なデバッグ機能の1つに S/W ブレーク（ソフトウェアブレーク）ポイントがあります。

[エディタ] ウィンドウにおいて、S/W ブレークポイントを簡単に設定できます。例えば、`sort` 関数のコール箇所にて S/W ブレークポイントを設定する方法を説明します。

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

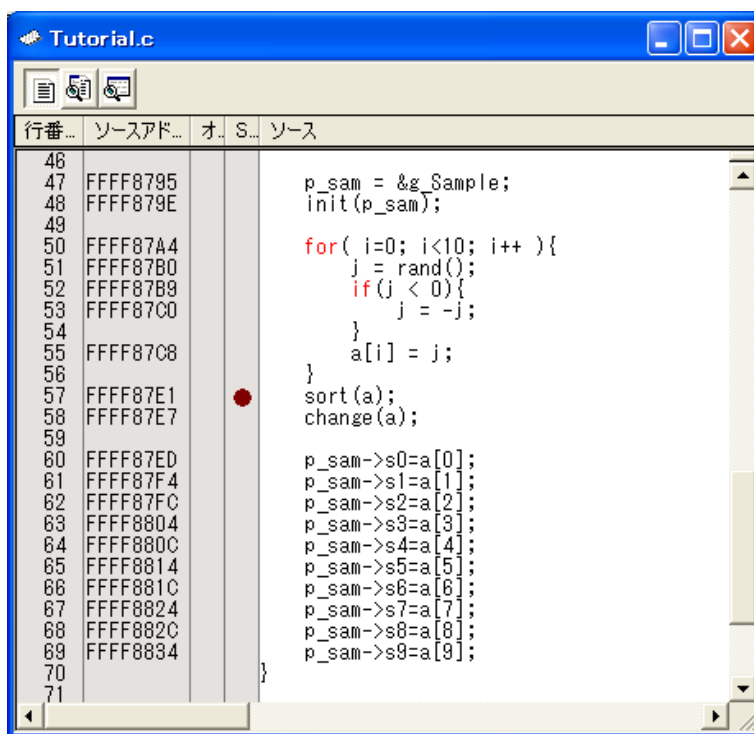



図5.5 [エディタ] ウィンドウ(S/W ブレークポイントの設定)

`sort` 関数を含む行に、赤色の印が表示されます。この表示により S/W ブレークポイントが設定されたことを示します。


5.6 プログラムの実行

プログラムの実行方法について説明します。

5.6.1 CPUのリセット

CPU をリセットする場合は、[デバッグ] メニューから[CPUのリセット] を選択するか、ツールバー上の [CPUのリセット] ボタン を選択してください。

5.6.2 プログラムを実行する

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

プログラムはブレークポイントを設定したところまで実行されます。プログラムが停止した位置を示すために[S/W ブレークポイント] カラム中に矢印が表示されます。

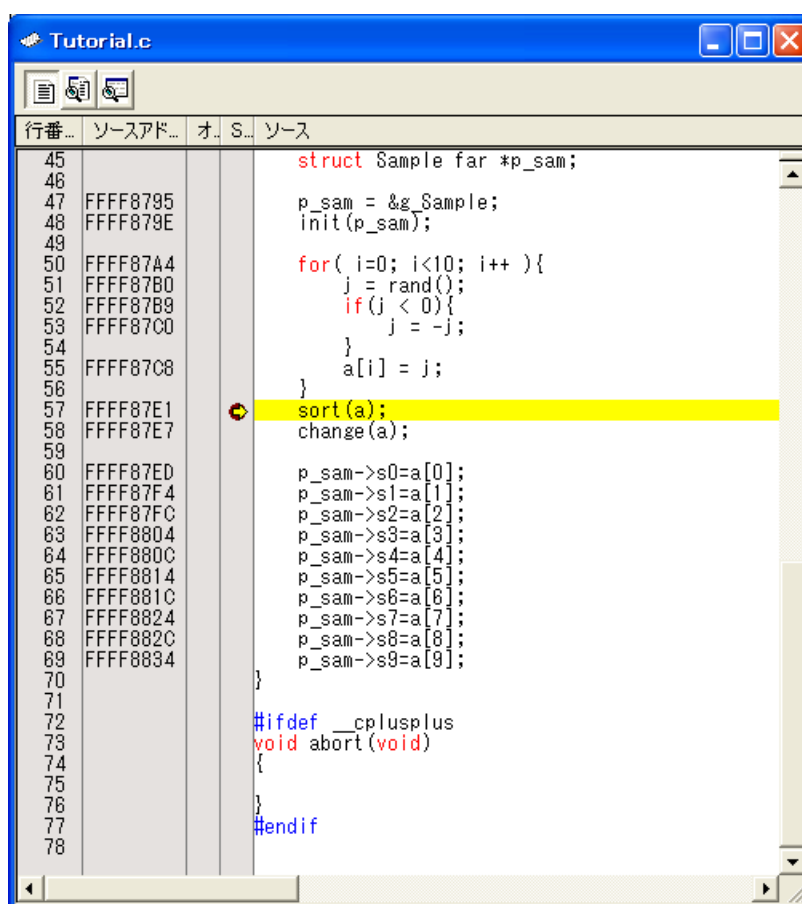



図5.6 [エディタ] ウィンドウ(ブレーク状態)

[ステイタス] ウィンドウで最後に発生したブレイクの要因が確認できます。

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

[ステイタス] ウィンドウが表示されますので、[Platform] シートを開いて確認してください。

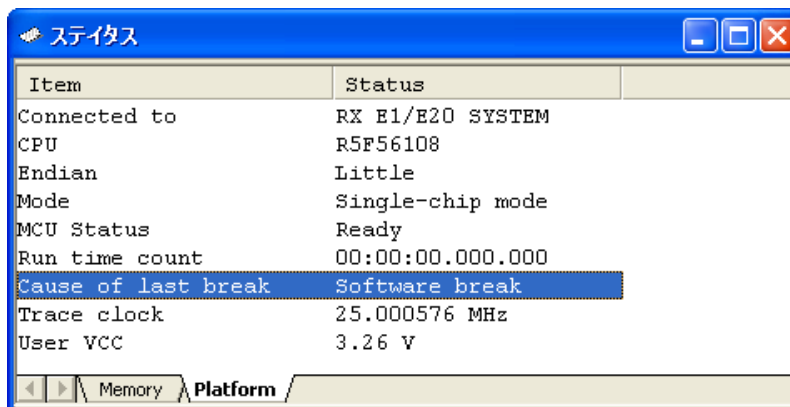


図5.7 [ステイタス] ウィンドウ

【注】 本ウィンドウで表示される内容は、エミュレータごとに異なります。表示内容については、オンラインヘルプを参照してください。

5.7 ブレークポイントの確認

設定したすべての S/W ブレークポイントは、[ブレークポイント] ダイアログボックスで確認することができます。

5.7.1 ブレークポイントを確認する

[編集] メニューの[ソースブレークポイント] を選択してください。 [ブレークポイント] ダイアログボックスが表示されます。

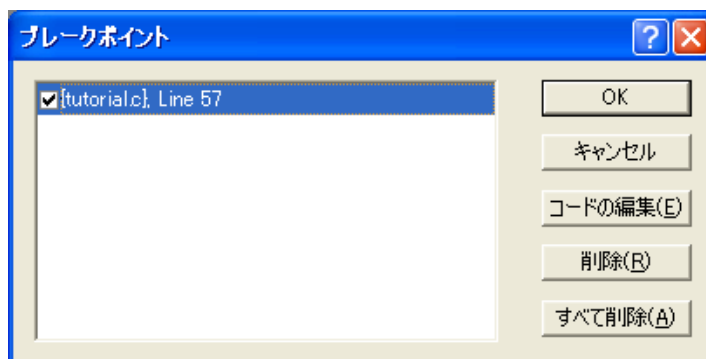



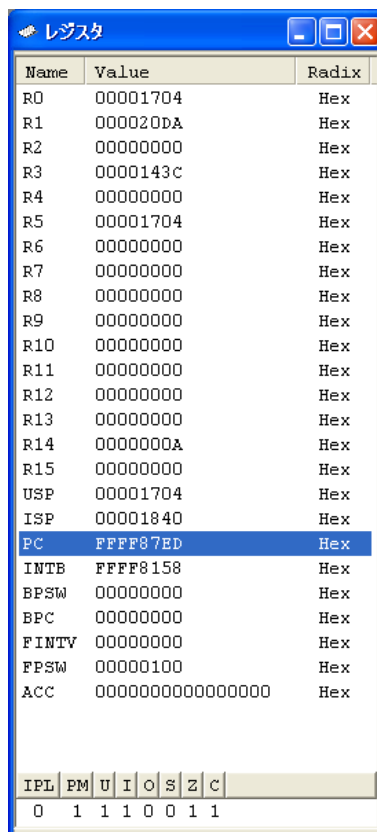
図5.8 [ブレークポイント] ダイアログボックス

このダイアログボックスを使って、ブレークポイントの削除および有効/無効の選択ができます。

【注】 S/W ブレークポイントを 1 点も設定していない場合は、[ブレークポイント] ダイアログボックスを開くことはできません。

5.8 レジスタ内容の変更

[表示 -> CPU -> レジスタ] を選択するか[レジスタ] ツールバーボタン  をクリックすると、[レジスタ] ウィンドウが表示されます。



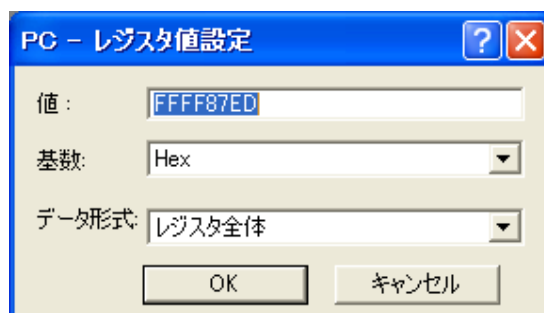
Name	Value	Radix
R0	00001704	Hex
R1	000020DA	Hex
R2	00000000	Hex
R3	0000143C	Hex
R4	00000000	Hex
R5	00001704	Hex
R6	00000000	Hex
R7	00000000	Hex
R8	00000000	Hex
R9	00000000	Hex
R10	00000000	Hex
R11	00000000	Hex
R12	00000000	Hex
R13	00000000	Hex
R14	0000000A	Hex
R15	00000000	Hex
USP	00001704	Hex
ISP	00001840	Hex
PC	FFFF87ED	Hex
INTB	FFFF8158	Hex
BPSW	00000000	Hex
BPC	00000000	Hex
FINTV	00000000	Hex
FPSW	00000100	Hex
ACC	0000000000000000	Hex

IPL | PM | U | I | O | S | Z | C
0 | 1 | 1 | 1 | 0 | 0 | 1 | 1

図5.9 [レジスタ] ウィンドウ

任意のレジスタの内容を変更することができます。

変更するレジスタ行をダブルクリックしてください。ダイアログボックスが表示されますので、変更する値を入力ください。



PC - レジスタ値設定

値:

基数:

データ形式:


OK キャンセル

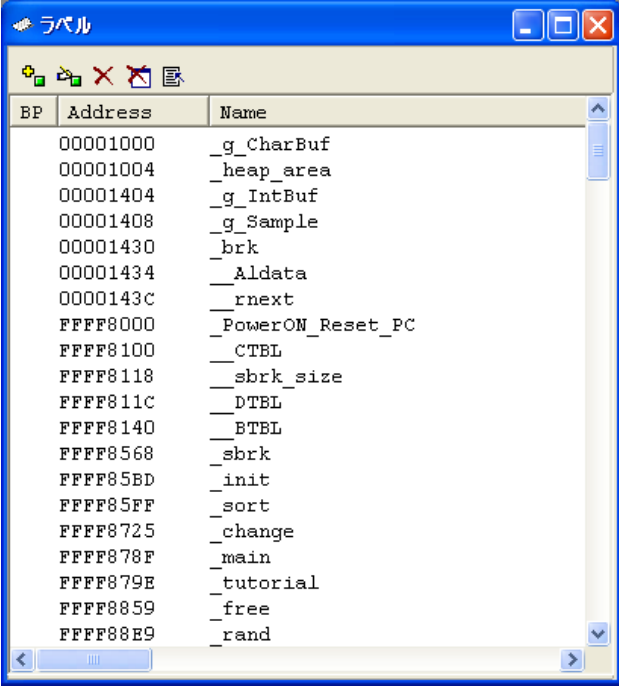
図5.10 [レジスタ値設定] ダイアログボックス(PC)

[注] [レジスタ] ウィンドウで表示されるレジスタ項目は、マイコンごとに異なります。レジスタ項目については、関連するマイコンのハードウェアマニュアルを参照してください。

5.9 シンボルの参照

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

[表示 -> シンボル -> ラベル] を選択するか、[ラベル] ツールバーボタン  をクリックすると、[ラベル] ウィンドウが表示されます。モジュール内のシンボル情報が参照できます。




BP	Address	Name
	00001000	_g_CharBuf
	00001004	_heap_area
	00001404	_g_IntBuf
	00001408	_g_Sample
	00001430	_brk
	00001434	__Alldata
	0000143C	__rnext
	FFFF8000	__PowerON_Reset_PC
	FFFF8100	__CTBL
	FFFF8118	__sbrk_size
	FFFF811C	__DTBL
	FFFF8140	__BTBL
	FFFF8568	__sbrk
	FFFF85BD	__init
	FFFF85FF	__sort
	FFFF8725	__change
	FFFF878F	__main
	FFFF879E	__tutorial
	FFFF8859	__free
	FFFF88E9	__rand

図5.11 [ラベル] ウィンドウ

5.10 メモリ内容の確認

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

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

[表示開始アドレス] エディットボックスに `"_main"` を入力してください。

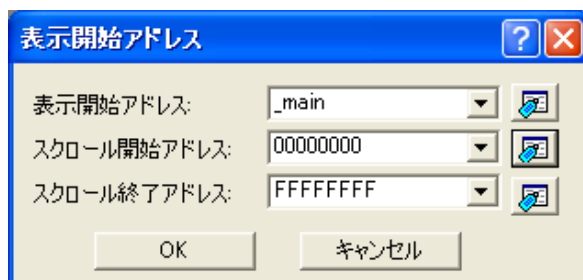


図5.12 [表示開始アドレス] ダイアログボックス

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

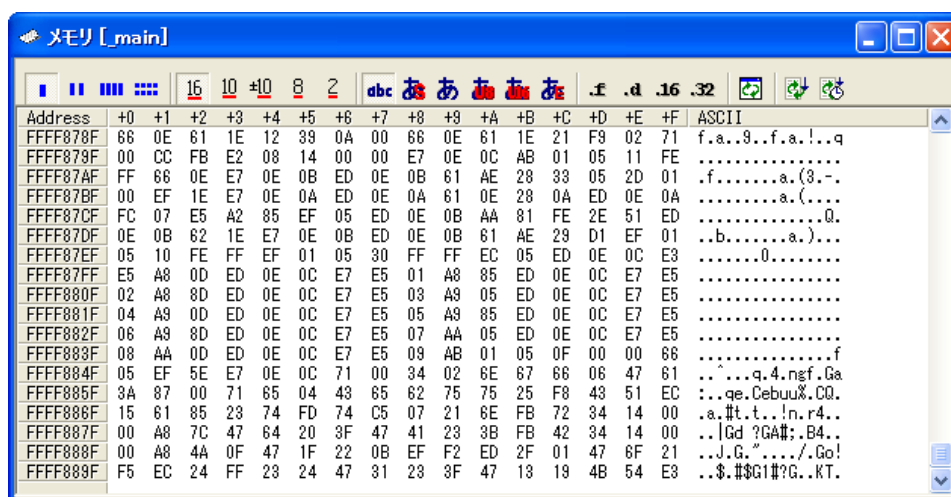


図5.13 [メモリ] ウィンドウ

5.11 変数の参照

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

[エディタ] ウィンドウに表示されている配列 a の左側をクリックし、カーソルを置いてください。

マウスの右ボタンで[インスタントウォッチ] を選択してください。

以下のダイアログボックスが表示されます。

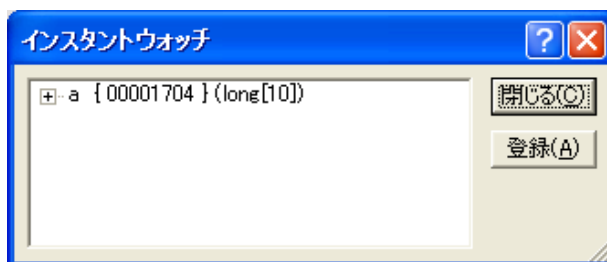


図5.14 [インスタントウォッチ] ダイアログボックス

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

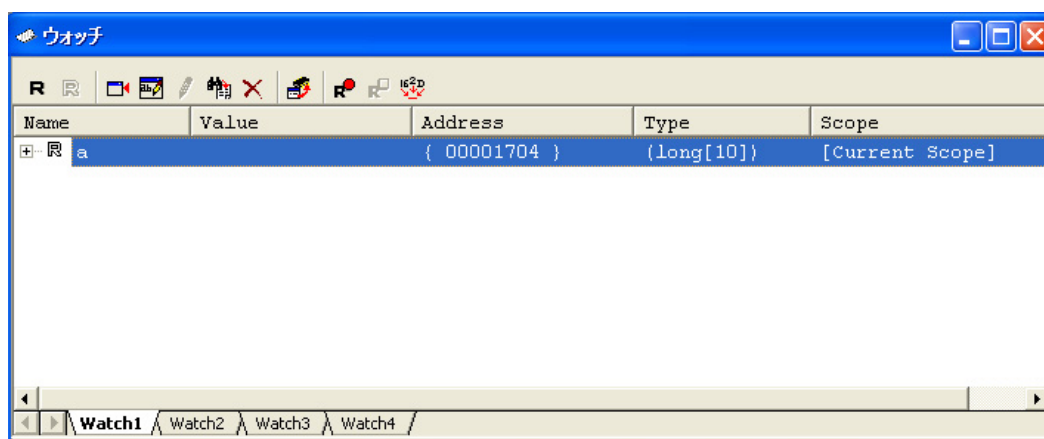


図5.15 [ウォッチ] ウィンドウ(配列の表示)

また、変数名を指定して、[ウォッチ] ウィンドウに変数を加えることもできます。
マウスの右ボタンで[ウォッチ] ウィンドウをクリックし、ポップアップメニューから[シンボル登録] を選択してください。以下のダイアログボックスが表示されます。

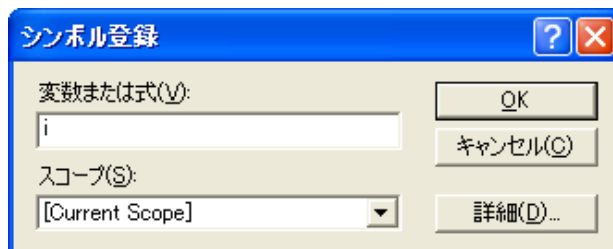


図5.16 [シンボル登録] ダイアログボックス

[変数または式] エディットボックスに変数 `i` を入力し、[OK] ボタンをクリックします。
[ウォッチ] ウィンドウに、`int` 型の変数 `i` が表示されます。

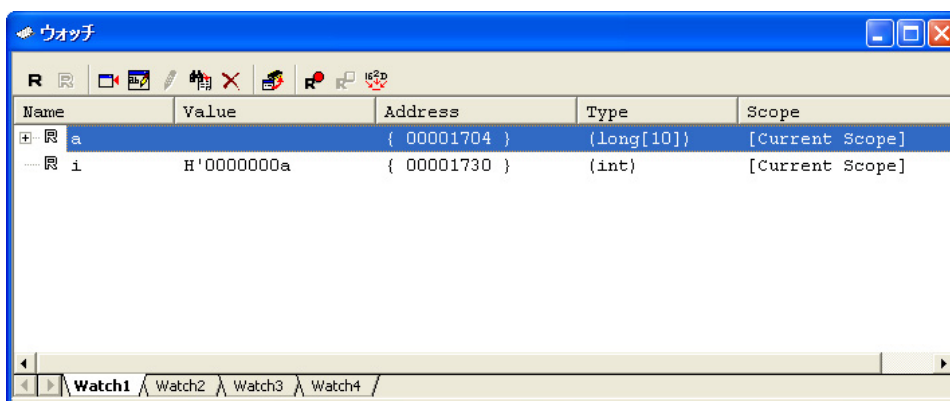


図5.17 [ウォッチ] ウィンドウ(変数の表示)

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

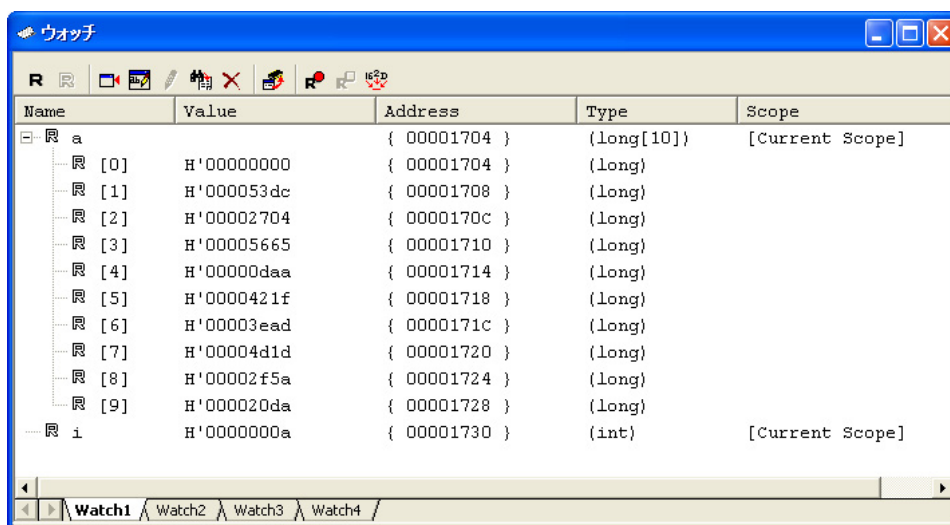



図5.18 [ウォッチ] ウィンドウ(配列要素の表示)

5.12 ローカル変数の表示

[ローカル] ウィンドウを使って関数内のローカル変数を表示させることができます。例として、tutorial 関数のローカル変数を調べます。この関数は、4つのローカル変数 a、j、i、p_sam を宣言します。

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

[ローカル] ウィンドウには、現在のプログラムカウンタ (PC) が指している関数のローカル変数とその値が表示されます。

関数内にローカル変数が存在しない場合、[ローカル] ウィンドウに何も表示されません。

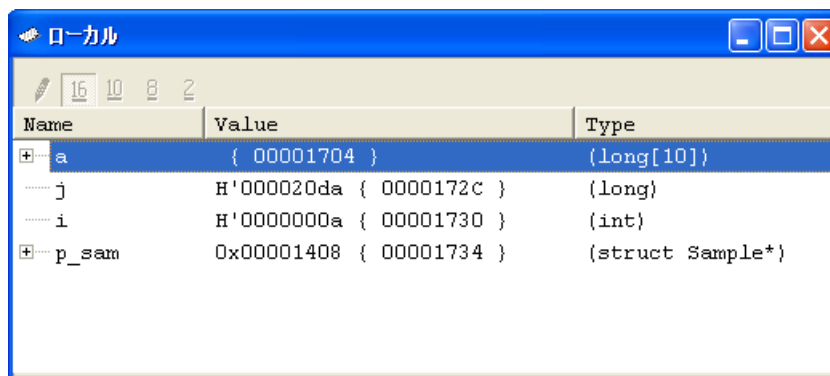


図5.19 [ローカル] ウィンドウ

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

sort 関数実行前と実行後のクラスインスタンス p_sam の要素を参照し、ランダムデータが降順にソートされていることを確認してください。

5.13 プログラムのステップ実行

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

表5.1 ステップオプション

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

5.13.1 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5.20 ステップインボタン

```

sort.c
-----
行番 ソースアド. オ. S. ソース
11 FFFF85C8 p_sam->s1 = 0;
12 FFFF85CC p_sam->s2 = 0;
13 FFFF85D2 p_sam->s3 = 0;
14 FFFF85D8 p_sam->s4 = 0;
15 FFFF85DE p_sam->s5 = 0;
16 FFFF85E4 p_sam->s6 = 0;
17 FFFF85EA p_sam->s7 = 0;
18 FFFF85F0 p_sam->s8 = 0;
19 FFFF85F6 p_sam->s9 = 0;
20
21
22
23 //-----
24 int g_IntBuf;
25 char g_CharBuf;
26 //-----
27
28 FFFF85FF void sort(long *a)
29 {
30     long t;
31     int i, j, k, gap;
32
33     gap = 5;
34     while( gap > 0 ){
35         for( k=0; k<gap; k++){
36             for( i=k+gap; i<10; i=i+gap ){
37                 for( j=i-gap; j>k; j=j-gap ){
38                     g_IntBuf = j;
39                     if( a[j]>a[j+gap] ){
40                         t = a[j];
41                         a[j] = a[j+gap];
42                         a[j+gap] = t;
43                     }
44                     else break;
45                 }
46             }
47             gap = gap/2;
48         }
49         g_CharBuf = (char)g_IntBuf & 0x00FF;
50     }
51 }
52
53 FFFF8725 void change(long *a)
54 {
55     long tmp[10];
56     int i;
57
58     for( i=0; i<10; i++){
59         tmp[i] = a[i];
60     }
61     for( i=0; i<10; i++){
62         a[i] = tmp[9 - i];
63     }
64 }

```

図5.21 [エディタ] ウィンドウ(ステップイン)

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

5.13.2 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のステートメントで停止します。

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



図5.22 ステップアウトボタン

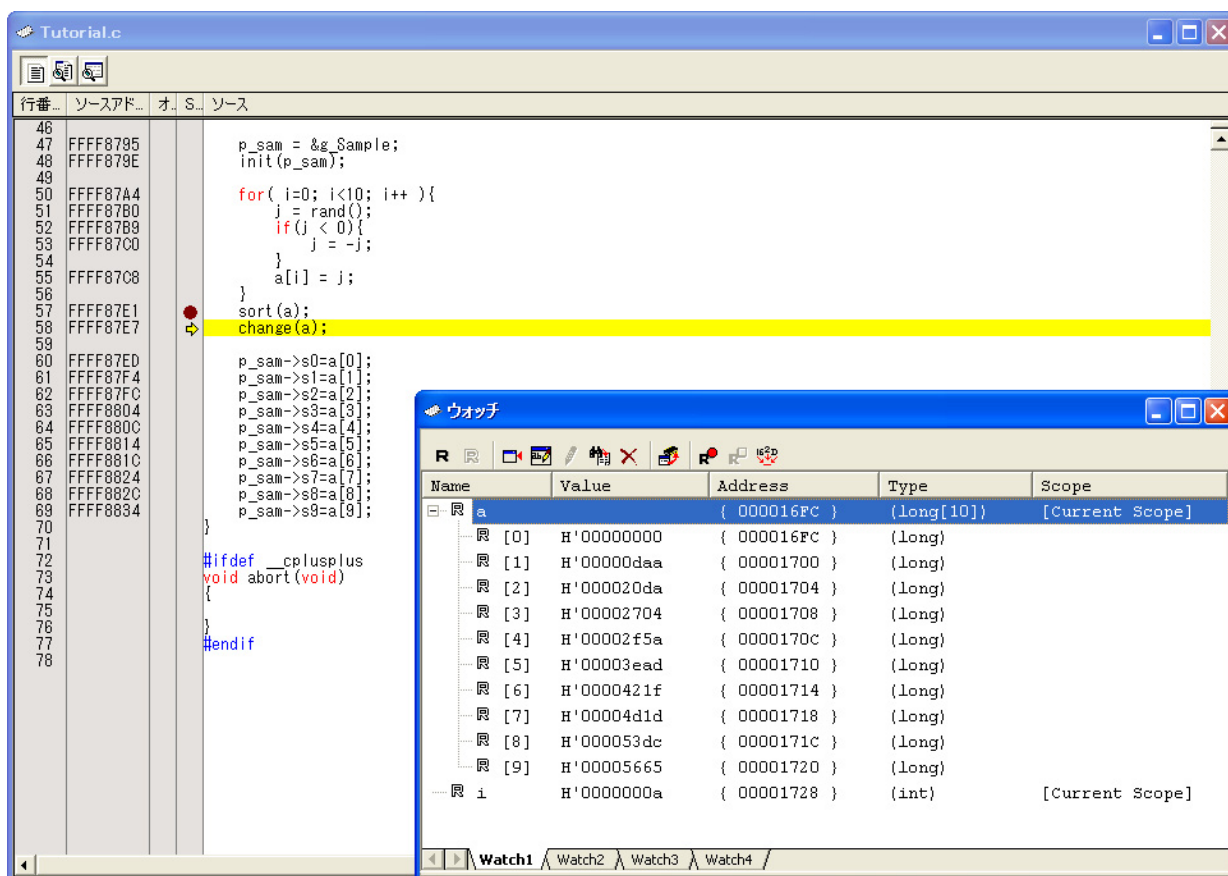


図5.23 [エディタ] ウィンドウ(ステップアウト)

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

5.13.3 ステップオーバの実行

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

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



図5.24 ステップオーバボタン

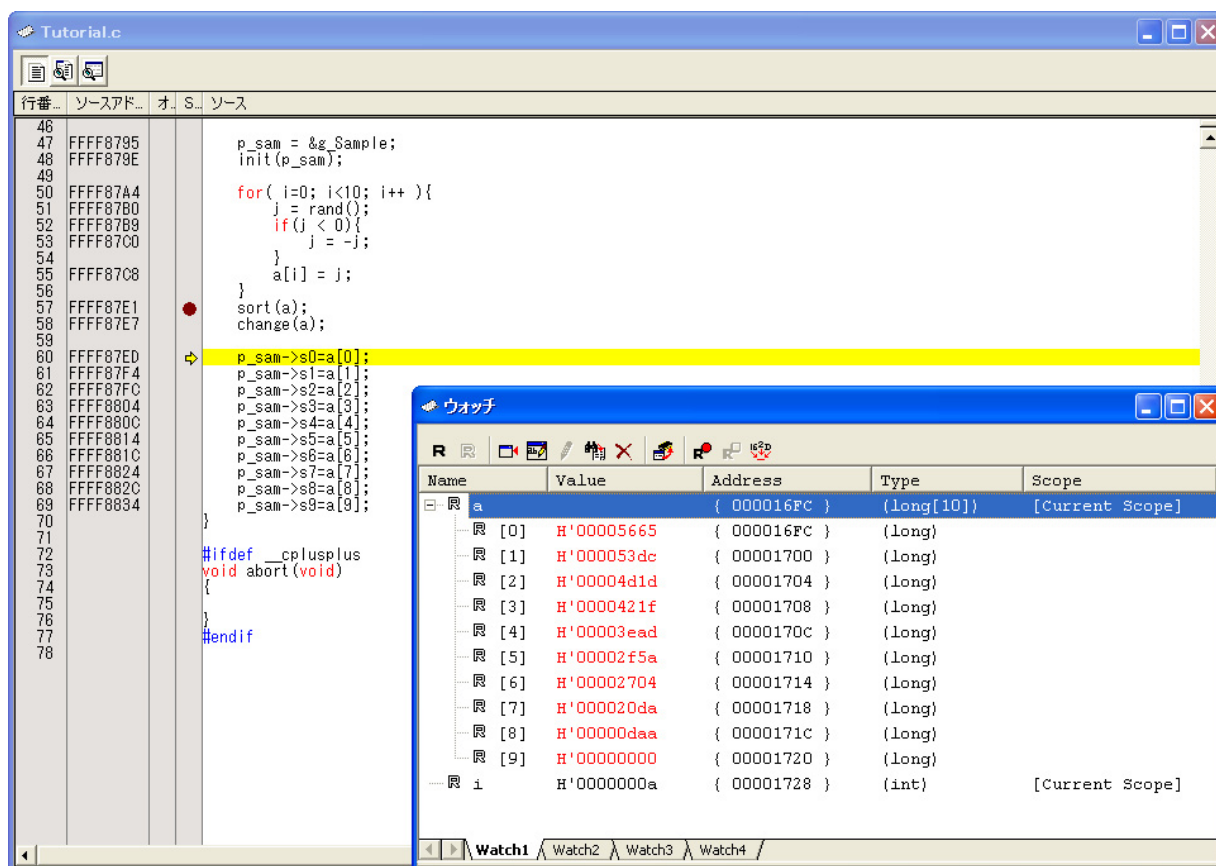


図5.25 [エディタ] ウィンドウ(ステップオーバ)

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

5.14 プログラムの強制ブレーク

High-performance Embedded Workshop は、プログラムを強制的にブレークすることができます。
ブレークポイントをすべて解除してください。

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



図5.26 実行ボタン

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



図5.27 停止ボタン

5.15 オンチップブレーク機能

マイコンの機能に応じて、オンチップブレーク機能を使用できます。オンチップブレーク機能は、プログラムが指定したアドレスを実行した場合（命令フェッチ）または、指定したメモリを読み込んだ、あるいは書き込んだとき（データアクセス）などに、プログラムの実行を停止させる機能です。

5.15.1 指定アドレスを実行したときにプログラム実行を停止

[エディタ] ウィンドウにおいて、命令フェッチイベントを簡単に設定できます。たとえば、`sort` 関数のコール箇所（命令フェッチイベント）を設定します。

`sort` 関数コールを含む行の[オンチップブレークポイント] カラムをダブルクリックしてください。

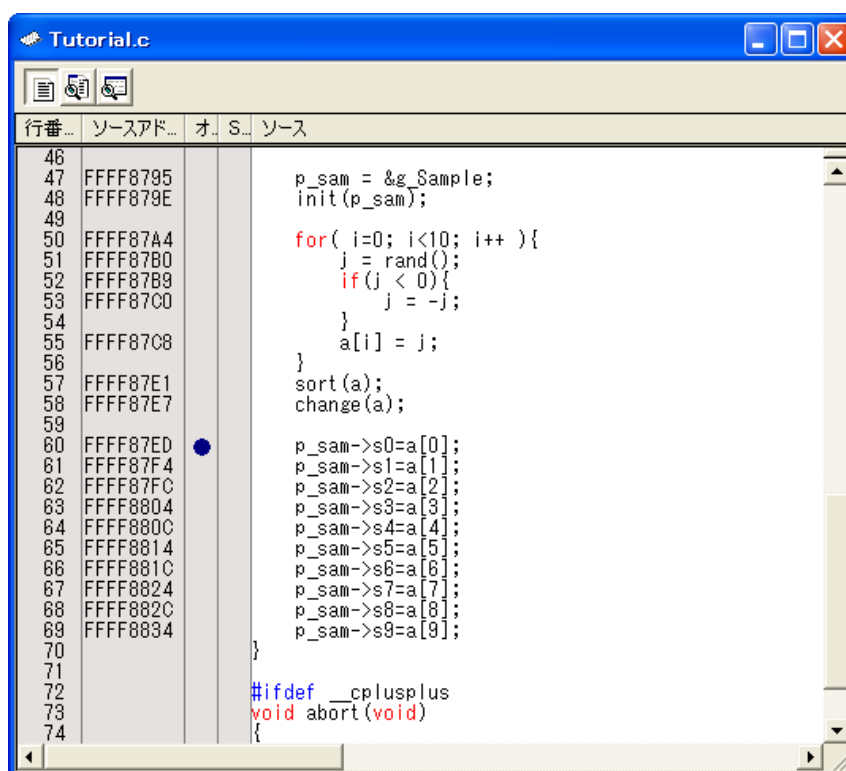


図5.28 [エディタ]ウィンドウ（オンチップブレークポイントの設定）

`sort` 関数を含む行に青色の印が表示されます。この表示により命令をフェッチしたときに実行を停止するオンチップブレークポイントが設定されたことを示します。

5.16 メモリにアクセスしたときにプログラム実行を停止

グローバル変数に対して値を読み書きした場合にプログラムを停止するには、以下のように設定します。

[表示 -> イベント -> オンチップブレイク条件設定] を選択して、[オンチップブレイク条件設定] ダイアログボックスを表示します。

[オンチップブレイク条件設定] ダイアログボックスの[OR] ページを表示します。値を読み書きしたときにプログラムを停止させたいグローバル変数を、[エディタ] ウィンドウで選択して、この[OR] ページ内にドラッグ&ドロップします。

その後、[適用] ボタンをクリックします。

プログラムを実行すると設定したグローバル変数に値が読み書きされたときに実行を停止します。



図5.29 [オンチップブレイク条件設定] ダイアログボックス

- 【注】
1. 設定できるグローバル変数は、変数のサイズが1バイトあるいは2バイトの変数のみです。
 2. ローカル変数は指定できません。

5.17 トレース機能

E1/E20 エミュレータは、マイコンの機能に応じて以下のトレース機能を使用できます。


- 内蔵トレース機能

マイコンに内蔵されているトレースバッファを使用して実現します。分岐情報やデータアクセス情報を表示できます。表示内容や取得可能なサイクル数はマイコンによって異なります。

- 外部トレース出力機能

E20エミュレータのみ対応しています。外部トレース出力が可能なマイコンを使用し、マイコンの外部トレース端子をE20エミュレータに接続している場合に有効です。表示内容や取得可能なサイクル数はマイコンによって異なります。

RX600 シリーズマイコン使用時の設定例を以下に示します。

[表示 -> コード -> トレース] を選択するか、[トレース] ツールバーボタン  をクリックしてください。[トレース] ウィンドウが表示されます。

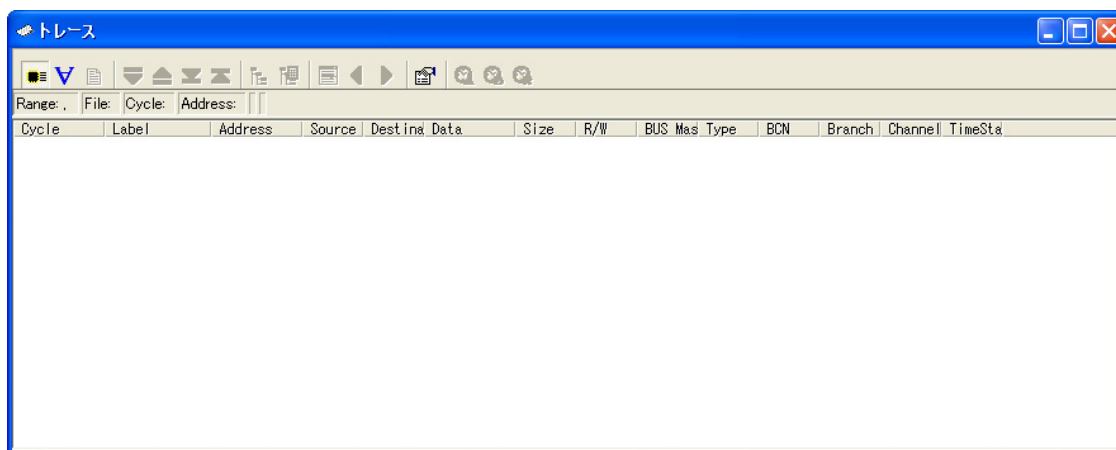


図5.30 [トレース] ウィンドウ

次頁からトレース機能の概要と設定方法について説明します。

5.17.1 フリートレースによるトレース情報の表示

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

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

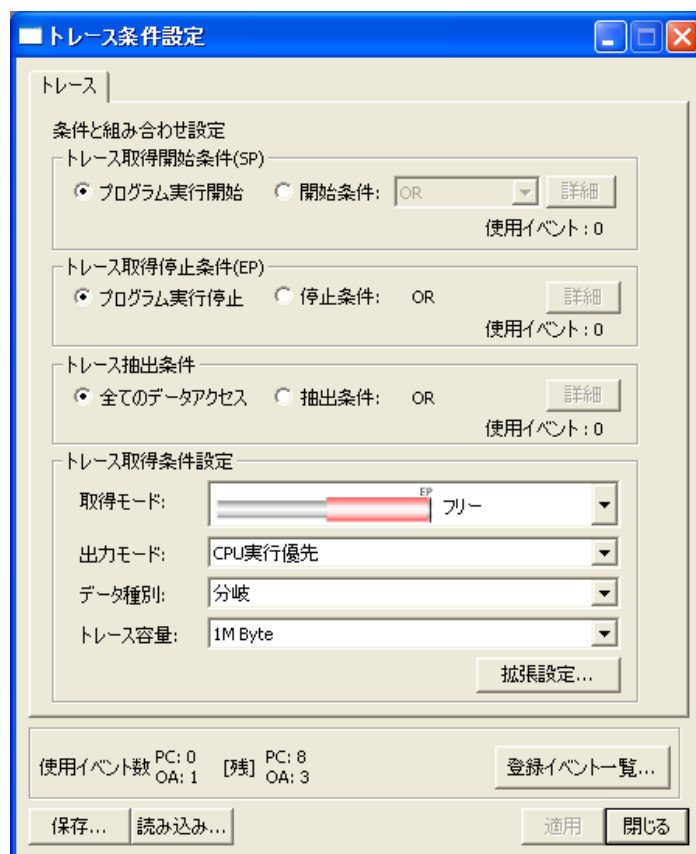


図5.31 [トレース条件設定] ダイアログボックス(フリートレース)

(2) tutorial関数内の「p_sam->s0=a[0];」と記述されている行にS/Wブレークを設定してください。

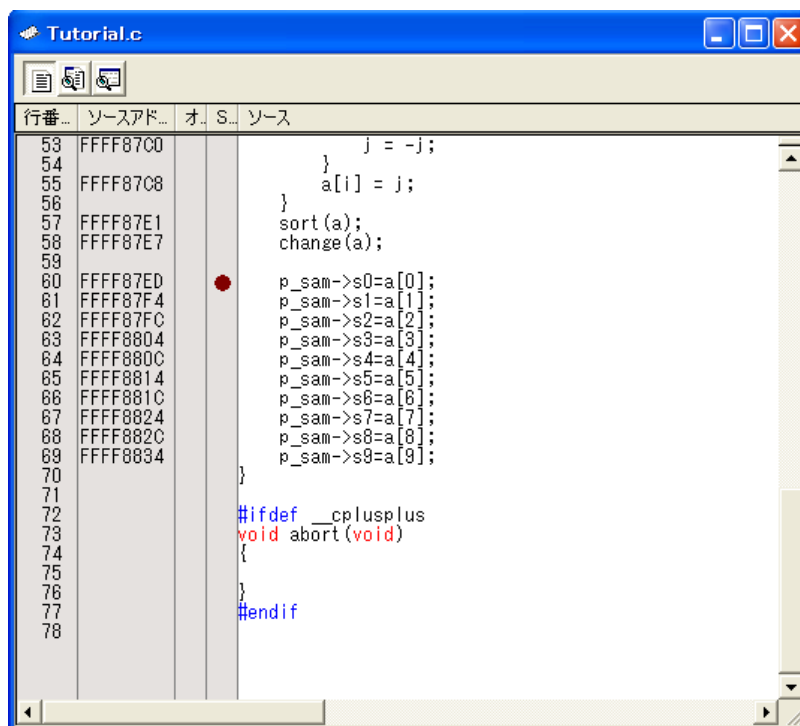


図5.32 [エディタ] ウィンドウ(tutorial 関数内への S/W ブレーク設定)

(3) [デバッグ] メニューから[リセット後実行] を選択してください。ブレークにより処理が停止し、[トレース] ウィンドウにブレークするまでのトレース情報が表示されます。

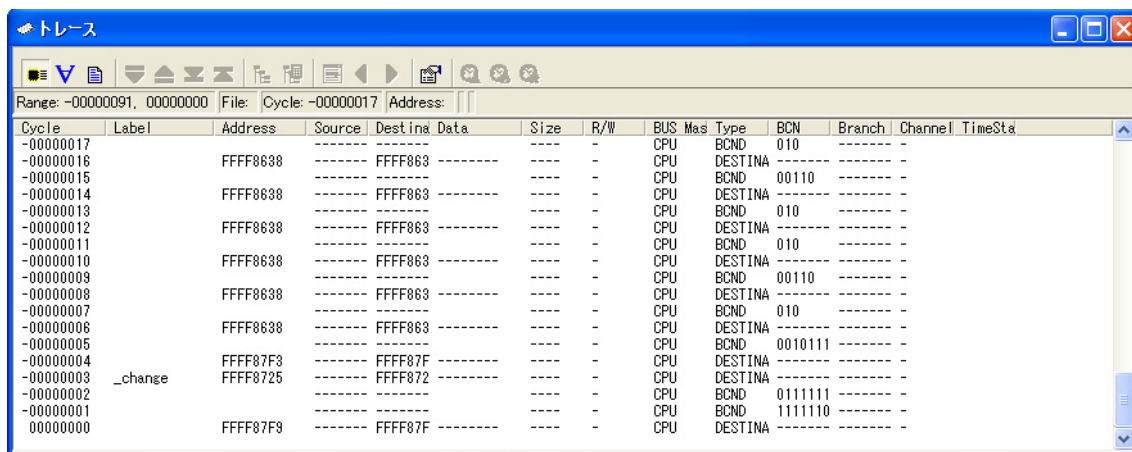


図5.33 [トレース] ウィンドウ(フリートレース)

- (4) バス表示、逆アセンブル表示、ソース表示の混合表示ができます。ポップアップメニューの[表示モード -> DIS] を選択すると、バスと逆アセンブルの混合表示ができます。

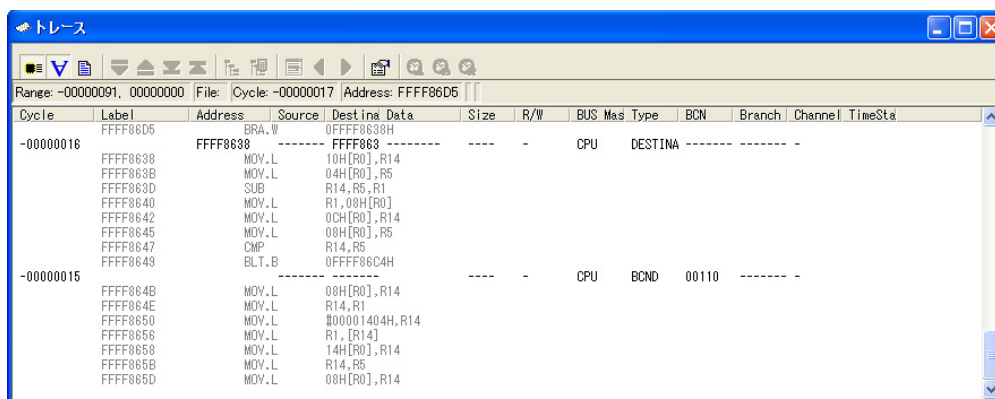


図5.34 [トレース] ウィンドウ(バスと逆アセンブルの混合表示)

- (5) さらにポップアップメニューの[表示モード -> SRC] を選択すると、バスと逆アセンブル、ソースの混合表示ができます。

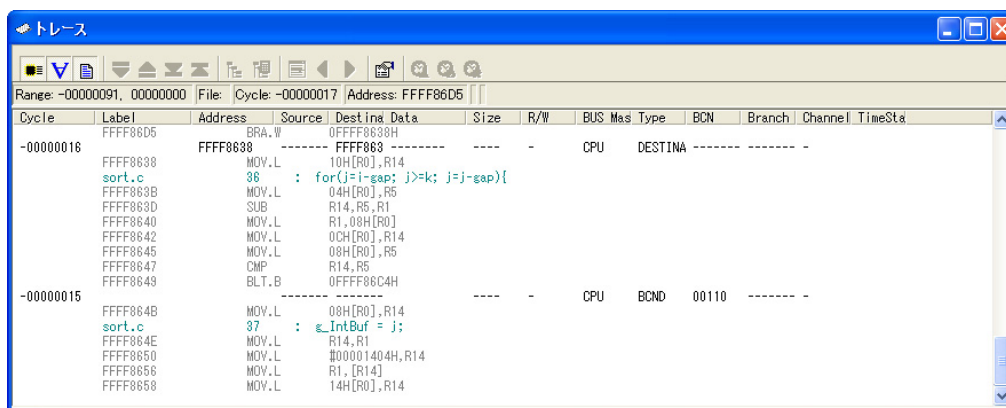
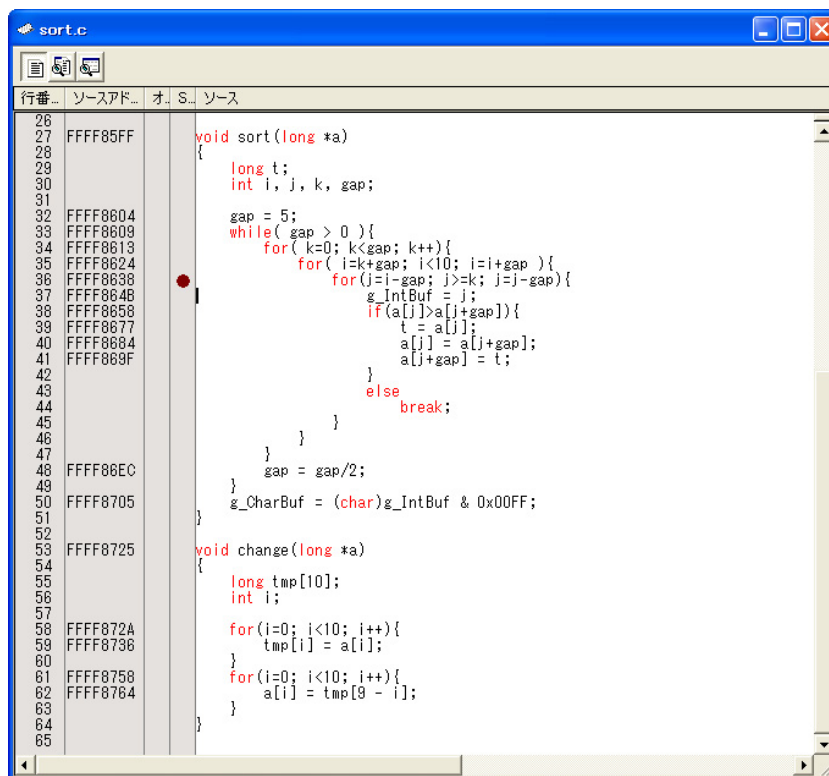


図5.35 [トレース] ウィンドウ(バスと逆アセンブル、ソースの混合表示)

5.18 スタックトレース機能

スタック情報を用いて、現在の PC がある関数がどの関数からコールされているかを表示できます。

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



The screenshot shows a code editor window titled 'sort.c'. The editor has three columns: '行番' (Line No.), 'ソースアド' (Source Addr), and 'ソース' (Source). The code is as follows:

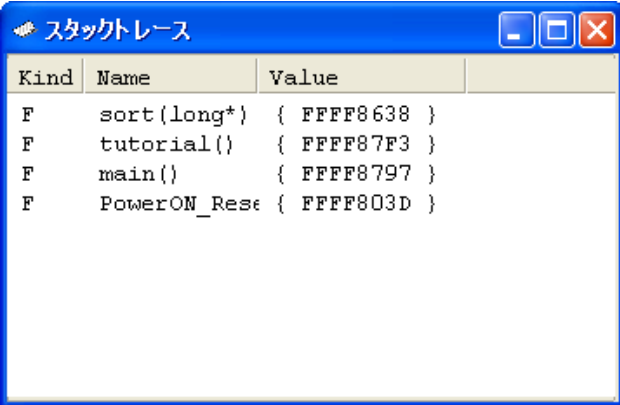
```
26 void sort(long *a)
27 {
28     long t;
29     int i, j, k, gap;
30     gap = 5;
31     while (gap > 0) {
32         for (k=0; k<gap; k++){
33             for (i=k+gap; i<10; i=i+gap) {
34                 for (j=i-gap; j>k; j=j-gap) {
35                     g_IntBuf = j;
36                     if (a[j]>a[j+gap]) {
37                         t = a[j];
38                         a[j] = a[j+gap];
39                         a[j+gap] = t;
40                     }
41                     else break;
42                 }
43             }
44             gap = gap/2;
45         }
46     }
47     g_CharBuf = (char)g_IntBuf & 0x00FF;
48 }
49
50 void change(long *a)
51 {
52     long tmp[10];
53     int i;
54     for (i=0; i<10; i++){
55         tmp[i] = a[i];
56     }
57     for (i=0; i<10; i++){
58         a[i] = tmp[9 - i];
59     }
60 }
61
62
63
64
65
```

A red dot is placed in the 'S/W ブレークポイント' column next to line 37, indicating that a breakpoint has been set at that location.

図5.36 [エディタ] ウィンドウ (S/W ブレークポイントの設定)

[デバッグ] メニューから [リセット後実行] を選択してください。

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



Kind	Name	Value
F	sort(long*)	{ FFFF8638 }
F	tutorial()	{ FFFF87F3 }
F	main()	{ FFFF8797 }
F	PowerON_Reset	{ FFFF803D }

図5.37 [スタックトレース] ウィンドウ

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

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

5.19 さてつぎは？

このチュートリアルでは、E1/E20 エミュレータのいくつかの主な特徴と、High-performance Embedded Workshop の使い方を紹介しました。

E1/E20 エミュレータで提供されるエミュレーション機能を使用することによって、高度なデバッグを行うことができます。それによって、ハードウェアとソフトウェアの問題が発生する条件を正確に分離し、識別すると、それらの問題点を効果的に調査することができます。

6. 使用上の注意事項

6.1 メモリ

6.1.1 I/Oレジスタ領域

IOレジスタ領域の内容をメモリウィンドウなどにより参照/設定する場合、アクセスサイズを対象マイコンのハードウェアマニュアルの指定に合わせる必要があります。このため、IOウィンドウを使用することをお勧めします。

6.1.2 内蔵フラッシュROM領域

- ダウンロード以外の操作（メモリウィンドウからの操作やラインアSEMBル）で内蔵フラッシュROM領域を編集した場合、編集内容は次の実行操作時にフラッシュROMに書き込まれます。
- 消去された状態のデータフラッシュROMの内容を参照すると、マイコンの仕様どおり不定値が表示されます。なお、デバッガからデータフラッシュROMの内容を書き換えた場合は256バイト単位で書き込みを行います。書き込まれた領域は不定値とはなりません。
- [内蔵フラッシュメモリの上書き] ページで登録していないブロックは、消去してからダウンロードが行われます。
- 複数のファイルをダウンロードする場合で、それぞれの領域がフラッシュの同一ブロックで重複している場合は、[内蔵フラッシュメモリの上書き] ページで、そのブロックを登録してください。
- E1/E20エミュレータを接続してデバッグに使用したマイコンは、エミュレーション時にフラッシュメモリの書き込みを繰り返しておりストレスがかかっています。デバッグに使用したマイコンはユーザの量産製品には使用しないでください。

6.1.3 内蔵フラッシュROMへのダウンロード

[ダウンロードモジュール] ダイアログボックスの一部の「アクセスサイズ」には対応していません。アクセスサイズは”1”,”2”または”4”としてください。

6.1.4 内蔵フラッシュROMの書き換え

- 内蔵フラッシュROM（プログラムROM領域）をユーザプログラムで書き換えた後に、メモリウィンドウ等で書き換えた内容を参照する場合は、[コンフィグレーションプロパティ] ダイアログボックスの[システム] ページで[内蔵プログラムROMを書き換えるプログラムをデバッグする] チェックボックスにチェックを入れてください。
チェックを入れていない場合は、書き換えた内容を参照することができません。
同様にデータフラッシュ領域では、[内蔵データフラッシュを書き換えるプログラムをデバッグする] チェックボックスにチェックを入れてください。
- ユーザプログラム実行中にユーザプログラムで書き換えた内蔵フラッシュROM（プログラム領域、データフラッシュ領域）の内容をメモリウィンドウ等で参照しないでください。
ユーザプログラム実行中に、書き換えた領域を参照した場合、正しく値が読み出せません。

6.1.5 FCU-RAM, FCUファーム領域

FCU-RAM 領域の内容をデバッガから書き換えしないでください。また、FCU ファーム領域はデバッガから書き換えできません。

6.1.6 デバッガが使用するワークRAM領域

[コンフィグレーションプロパティ] ダイアログボックスの[MCU] ページで指定したワーク RAM アドレスから指定されている使用バイト分は、内蔵フラッシュメモリ書き換え時にファームウェアが使用します。メモリ内容の退避・復帰が行われるためユーザプログラムでもこの領域を使用できますが、この領域を DMA または DTC 機能の転送元、および転送先には指定しないでください。

なお、ワーク RAM 領域は 内蔵 RAM 領域を指定し、ユーザプログラムでは内蔵 RAM を無効に設定しないようにしてください。

6.2 メモリウィンドウ

6.2.1 コピー、比較、検索機能

コピー、比較、検索機能において、データサイズに 8 バイトを指定しないでください。8 バイトを指定した場合、正しく動作しません。

また、コピー、比較、検索機能の範囲は最大 16MB です。

6.2.2 オプションメニュー

メモリウィンドウの機能のうち、メニュー項目でグレー表示の項目については動作しません。

6.3 実行操作

6.3.1 カーソル位置まで実行

- 「カーソル位置まで実行」の操作では、途中にあるブレークポイントで停止しません。

【RX600 シリーズマイコン】

- トレース設定条件およびパフォーマンス条件設定にて実行アドレスタイプのイベントが合計 8 点以上設定された場合、「カーソル位置まで実行」は使用できません。

【RX200 シリーズマイコン】

- トレース設定条件およびパフォーマンス条件設定にて実行アドレスタイプのイベントが合計 4 点以上設定された場合、「カーソル位置まで実行」は使用できません。

6.3.2 条件を指定して実行

「条件を指定して実行」の操作で指定したテンポラリ PC ブレークポイントは、最初の 1 点のみが有効となります。

6.3.3 ステップアウト実行

サブルーチン復帰命令以外の命令（ジャンプ命令）で呼び出し元関数に戻る関数については、正しくステップアウト実行できません。

6.4 リセット

6.4.1 エミュレータシステムによる操作とマイコンリセットの競合について

端子リセットやウォッチドッグタイマなどによるリセットとエミュレータシステムによるマイコンに対する操作（[メモリ] ウィンドウでメモリ参照など）が競合した場合、“タイムアウトエラーが発生しました。MCU がリセット状態です。システムリセットを発行しますか？”とエラーメッセージが表示されます。

[はい] ボタンをクリックした場合、エミュレータを初期化してユーザプログラムが停止します。システムリセット発行後は、トレース記録も初期化します。また、[いいえ] ボタンをクリックした場合、エミュレータは初期化されずユーザプログラムも停止しません。

なお、[はい] ボタンをクリックした後は、デバッグを継続して行うことが可能です。

6.4.2 トレース機能使用時のマイコンリセット

ユーザプログラム実行中に端子リセットやウォッチドッグタイマなどによるリセットが発生した場合、リセット前後のトレース記録が正しく行えなくなります。

6.4.3 リアルタイムRAMモニタ機能使用時のマイコンリセット

ユーザプログラム実行中に端子リセットやウォッチドッグタイマなどによるリセットが発生した場合、それ以降の RAM モニタの動作は保証できません。（正しい値を表示できない可能性があります。）

6.4.4 ユーザプログラム実行中のリセット

以下に示すどちらかの条件で端子リセットまたは内部リセットが発生した場合、マイコンの動作モードおよびエミュレータ通信インタフェースに依存した注意事項があります。端子リセット発生による注意事項を表 6.1 に、内部リセット発生による注意事項を表 6.2 に示します。

- ・ 内蔵 ROM 無効拡張モードまたはユーザブートモードでのユーザプログラム実行中
- ・ FINE 通信インタフェース使用でのユーザプログラム実行中

表6.1 端子リセット発生による注意事項

グループ	インタフェース	動作モード	ユーザプログラム実行中の端子リセット発生による注意事項
RX610, RX621, RX62N	JTAG	内蔵ROM無効拡張	リセットはエミュレータが解除します。このため、実際のマイコン単体動作とはタイミングが異なります。
RX630, RX631, RX63N, RX63T	JTAG	ユーザブート 内蔵ROM無効拡張	
RX630, RX631, RX63N, RX63T, RX210, RX21A, RX220	FINE	全モード	
RX210, RX21A, RX220	FINE	ユーザブート 内蔵ROM無効拡張	ユーザプログラム実行中に端子リセットによるリセットが発生した場合、パフォーマンス解析カウンタ値およびトレース取得データが初期化されます。

表6.2 内部リセット発生による注意事項

グループ	インタフェース	動作モード	ユーザプログラム実行中の内部リセット発生による注意事項
RX610, RX621, RX62N, RX630, RX631, RX63N, RX63T	JTAG	内蔵ROM無効拡張	リセット解除後、ユーザプログラムで内蔵ROM無効拡張モードに設定された後にデバッグ可能となります。
RX630, RX631, RX63N, RX63T	JTAG	ユーザブート	内部リセットが発生した場合、エミュレータからの制御ができなくなります。
RX630, RX631, RX63N, RX63T, RX210, RX21A, RX220	FINE	全モード	ウォッチドッグタイマなどによる内部リセットを発生させないでください。

6.5 IOウィンドウ

6.5.1 I/Oレジスタファイルのカスタマイズ

E1/E20 エミュレータデバッガでは[IO] ウィンドウから内蔵 I/O レジスタにアクセスできます。

また、マイコン仕様は変更になる場合がありますので、I/O レジスタファイルの各 I/O レジスタと、ターゲットマイコンのハードウェアマニュアル記載のアドレスに相違がある場合は、ハードウェアマニュアルの記載にしたがって修正してご使用ください。I/O レジスタは、I/O レジスタファイルのフォーマットにしたがい、カスタマイズすることが可能です。

6.5.2 ベリファイ

[IO] ウィンドウにおいては、入力値のベリファイ機能は無効です。

6.6 トレース機能

6.6.1 記録可能なアクセス

- CPU バスによるアクセスのみ記録可能です。DMAC や DTC によるアクセスは記録できません。

6.6.2 トレース情報

- トレース取得開始条件とトレース取得終了条件を両方指定すると、開始条件と終了条件が成立するたびにトレース情報が取得されます。この場合、トレースウィンドウで正しく逆アセンブリ表示できません。
- トレースウィンドウでの逆アセンブリ表示は、1つの分岐間に対して 1024 命令まで表示できます。分岐間命令が 1025 命令以上の場合、逆アセンブリ表示はできません。

【RX600 シリーズマイコン】

- “BRANCH”、“DESTINATION”および“BRANCH/DESTINATION”が出力される以前の“BCND:条件分岐”のトレース情報は表示されません。したがって、ユーザプログラム実行結果がすべて、“BCND:条件分岐”の場合、トレース情報は表示されません。

【RX200 シリーズマイコン】

- トレース取得条件設定のデータ種別が“データアクセス”、または“データアクセス+時間”の場合、必ずトレース抽出機能にてアドレス条件を設定してください。設定しない場合、データアクセスのトレース情報が取得されません。
データアクセスのトレース情報には、指定されたアドレスへのアクセスのみ記録します。なお、トレースウィンドウの Address カラムには、必ずトレース抽出機能で設定した開始アドレス値が表示されます。
- トレースウィンドウのタイムスタンプは、パフォーマンス測定用カウンタを用いて実現しています。その為、[パフォーマンス条件設定] ダイアログにてイベント組み合わせによるカウンタの開始 / 終了条件を設定した場合、期待したタイムスタンプが表示されません。

6.7 イベント機能

6.7.1 検出可能なアクセス

- CPUバスによるアクセスのみ検出可能です。DMAC や DTC によるアクセスは検出できません。

6.7.2 イベント組み合わせ

- イベントブレイク条件の組み合わせに AND またはシーケンシャルを指定した場合、トレース取得開始条件のイベント組み合わせは OR となります。また、トレース取得開始条件のイベント組み合わせに AND またはシーケンシャルを指定した場合、イベントブレイク条件の組み合わせは OR となります。

【RX600 シリーズマイコン】

- シーケンシャル条件でデータアクセスイベントを指定できるのは、リセットイベント、および、1 から 3 番目のイベントのみです。
- シーケンシャル条件で、アドレス範囲を指定したデータアクセスイベントを登録できるのは 1 番目のイベントのみです。
- シーケンシャル条件のリセットイベントには、通過回数条件を指定できません。

【RX200 シリーズマイコン】

- シーケンシャル条件でデータアクセスイベントを指定できるのは、1 と 2 番目のイベントのみです。

6.7.3 通過回数条件

【RX600 シリーズマイコン】

- 通過回数条件を指定できるイベントは、全体で 1 点のみです。

6.7.4 データアクセスイベントのアドレス範囲条件

【RX600 シリーズマイコン】

- アドレス範囲条件を指定できるイベントは全体で 1 点のみです。

6.7.5 イベント登録について

- 実行前 PC イベントリストにデータアクセスの条件を登録した場合、設定が無視されます。
- トレース抽出イベントリストに実行アドレスの条件を登録した場合、設定が無視されます。

【RX200 シリーズマイコン】

- [メモリ] ウィンドウのアドレス範囲をドラッグ&ドロップしてイベント登録した場合、アドレス範囲を条件としたデータアクセスイベントの設定ができないため、開始アドレスのみ設定されたイベントが無効な状態でイベントリストに追加されます。

[メモリ] ウィンドウからドラッグ&ドロップでイベント登録する場合は、1 バイト、2 バイトあるいは 4 バイトサイズの 1 データのみを選択してください。

6.7.6 WAIT命令に対するイベント

【RX630, RX631, RX63N, RX63T, RX210, RX21A, RX220 グループマイコン】

- WAIT命令またはその1命令前に設定した実行アドレスタイプのイベントでブレークが発生した場合、タイムアウトエラーが発生する可能性があります。ターゲットプログラムをWAIT命令またはその1命令前でブレークさせる場合は、オンチップブレークポイントを使用してください。

6.7.7 イベント要因について

- [アウトプット] ウィンドウ、[ステータス] ウィンドウなどに表示されるブレーク要因で、オンチップブレークポイント(OR)による停止の際に、"Event break at "で表示されるイベント要因 ("PC0"など) の末尾の数値はイベントに割り当てられたチャンネルを示します。イベントに割り当てられたチャンネルは[オンチップブレーク条件設定]ダイアログで参照できます。なお、上記の"Event break at "で表示されるイベント要因は、EVENT_SETコマンドで<event>に指定したイベント番号とは異なります。

6.8 ブレーク機能

6.8.1 ブレークポイント設定時の注意事項

- S/Wブレークポイントは、命令を置き換えることにより実現します。
内蔵ROMと内蔵RAM以外の領域には指定することができません。
- ステップ実行中は、S/Wブレークポイントおよびオンチップブレークポイント指定は無効です。
- S/Wブレークポイントおよび実行前PCブレークポイントで停止後、再度そのアドレスから実行を再開した場合、1度そのアドレスをシングルステップにより実行してから実行を継続するので、リアルタイム性はなくなります。

なお、このシングルステップ実行で成立したイベントは、その後の実行時にクリアされます。このため、トレースおよびパフォーマンスの開始イベントトリガがこのステップで設定された場合、再度イベントが成立するまでトレース取得もパフォーマンス測定も開始しません。また、ブレークおよびトレース開始組み合わせイベントも同様に、ステップで成立したイベントはAND(累積)/状態遷移の対象になりません。

- 低消費電力状態中はオンチップブレークの設定はできません。
- 低消費電力状態中に[エディタ] ウィンドウのオンチップブレークポイントカラムからオンチップブレークを設定した場合、以下のエラーメッセージが[Output] ウィンドウに表示されます。

スリープモードの場合：

「MCUのスリープモード中で内部クロックが停止状態です。」

全モジュールクロックストップモード、ソフトウェアスタンバイモード、ディープソフトウェアスタンバイモードの場合：

「MCUのスタンバイモード中で内部クロックが停止状態です。」

その後、オンチップブレークポイントカラムからオンチップブレークの設定ができなくなります。

この状態になった場合は、一度、[オンチップブレーク条件設定] ダイアログボックスをオープン、クローズしてください。

(設定は変更しなくても構いません。)

6.9 リアルタイムRAMモニタ機能

- CPUバスによるアクセスのみ監視可能です。DMACやDTCによるアクセスは監視できません。
- ユーザプログラム実行中に端子リセットやウォッチドッグタイマなどによるリセットが発生した場合、それ以降のRAMモニタの動作は保証できません。（正しい値を表示できない可能性があります。）
- リアルタイムRAMモニタ機能は、トレースデータを基にウィンドウの値を書き換えている為、ユーザプログラム実行中/停止中にかかわらず、[I/O] ウィンドウや[メモリ] ウィンドウにて値を書き換えても[RAMモニタ] ウィンドウには反映されません。
- トレース抽出機能により監視範囲を設定した場合、監視範囲外のデータ(グレー表示箇所)に対するウォッチ機能は使用できません。

6.9.1 ブロック境界のメモリ表示

- [RAMモニタ] ウィンドウにて境界を含んだ1ブロック、または連続する2ブロック以上の領域をメモリ表示する場合、必ず表示開始アドレスを表示サイズの倍数に合わせてください。アドレスとサイズを合わせずに表示した場合、メモリ表示内容がグレー表示になります。
- RAMモニタ領域設定にて登録したブロックの前ブロックが未登録(グレー表示箇所)の場合、ビッグエンディアンで以下のアクセスが発生しても登録ブロックでは正しくモニタできません。
登録ブロックの先頭アドレスを n 番地とした場合
 - (1) $n-1$ 番地に対する2バイトおよび4バイトアクセス
 - (2) $n-2$ 番地に対する4バイトアクセス
 - (3) $n-3$ 番地に対する4バイトアクセス

6.9.2 初期化抜け検出機能

- [初期化抜け検出] ダイアログボックスにて検出領域を登録した場合、[トレース条件設定] ダイアログボックスの出力モードが「トレース出力優先」に切り替わりますので、注意してください。
なお、登録した検出領域を全て削除すると、「CPU実行優先」に切り替わります。
また、検出領域を登録した後に手動で出力モードを変更した場合、変更した設定で動作します。
- 初期化抜け検出領域とRAMモニタ領域は共通の領域資源です。その為、RAMモニタで登録していないブロック領域を初期化抜け検出で登録した場合、RAMモニタ領域設定のリストにも対象ブロックが登録されます。

6.10 パフォーマンス測定機能

6.10.1 パフォーマンス測定中のリセットについて

パフォーマンス測定中にリセットが入った場合は、リセット期間中も計測が行われます。但し、計測用のクロック設定も初期化されるため正しい計測が行えない場合があります。

6.10.2 パフォーマンス測定中のネスト制限

【RX600 シリーズマイコン】

割り込み/例外発生回数と RTE/RTFI 命令の発行回数が同じで、それぞれ対応付けが取れる場合であることを前提としています。もし、割り込み/例外発生回数と RTE/RTFI 命令の発生回数が異なる場合、それぞれの対応付けが取れないため、正しく処理サイクルを計測することができません。

なお、ネストは 16 回分までしか保持できません。したがって、17 回以上の割り込み・例外でネストが発生した場合、正しく計測ができません。

6.10.3 [一度だけ計測] チェック時の注意事項

【RX600 シリーズマイコン】

- 以下の計測実施中にカウント開始イベントおよびカウント終了イベントが成立しても、一度も計測条件が成立しなかった場合、計測結果は表示されません。

- 有効命令発行回数
- 例外・割り込み受付回数
- 例外受付回数
- 割り込み受付回数
- イベントマッチ回数

(例)

- 計測区間中に一度も例外や割り込みが発生しなかった。
- 計測区間中に一度もイベントマッチが成立しなかった。
- パフォーマンス測定において、開始イベントが 2 回成立すると、終了イベントが成立していなくても計測を停止します。

6.11 ダウンロード機能

6.11.1 アクセスサイズについて

ダウンロードの際には、ダウンロードモジュールのアクセスサイズを指定することができます。

アクセスサイズを指定することにより、指定したアクセスサイズでメモリへのダウンロード（書き込み）を行います。

なお、指定可能なダウンロードモジュールのアクセスサイズは領域毎に異なりますので、ご注意ください。

内蔵 ROM/RAM 領域は、“1”、“2”または“4”のみ指定が可能（“8”は指定しないでください）ですが、外部フラッシュ ROM 領域のデータを含むダウンロードモジュールのアクセスサイズには“1”のみしか選択できません。

また、外部 RAM 領域へのアクセスは、起動時に[コンフィグレーションプロパティ] ダイアログボックスの外部領域で領域毎に指定したバス幅が優先されます。

アクセスサイズに“2”を指定する必要がある場合は、起動時の[コンフィグレーションプロパティ] ダイアログボックスで外部 RAM 領域に対するバス幅を“16bit”に設定し、外部フラッシュ ROM 領域のダウンロードモジュールを別のダウンロードモジュールに分け、外部フラッシュ ROM 領域を含まないダウンロードモジュールに“2”を指定してください。

なお、外部フラッシュ ROM 領域については、USD ファイルで指定されたアクセスサイズが使用されません。

詳しくは、下記 URL に掲載されている「External Flash Definition Editor」のマニュアルを参照してください。

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

例：以下のプログラムにおいて、外部 RAM のアクセスサイズを“2”に設定したい場合

内蔵RAM
外部RAM
外部ROM
内蔵ROM

以下のように(a)と(b)のモジュールに分けてダウンロードしてください。

- (a) アクセスサイズ”2”に設定したダウンロードモジュール（外部RAMへのアクセスサイズは起動時のバス幅”16bit”が優先）



- (b) アクセスサイズ”1”に設定したダウンロードモジュール



6.12 外部フラッシュメモリへのダウンロード

- セクタ数が4096以下のフラッシュメモリを登録可能です。
セクタ数が4096を越えるフラッシュメモリへの書き込みは保証できません。
- USDファイルで指定するスクリプトは、High-performance Embedded Workshop デバッガのコマンドラインから実行可能なmemory_fillコマンドに準拠しています。
ただし、一部のオプションは指定できませんのでご注意ください。
スクリプト中で使用できるmemory_fillコマンドのフォーマットは、次の通りです。

mf <start> <end> <data> [<mode>]

<start>	開始アドレス
<end>	終了アドレス
<data>	データ値
<mode>	BYTE (1バイト)
	WORD (2バイト)
	LONG (4バイト)
	省略 : BYTEと同じ

※USD ファイルで指定するスクリプトとは、以下を指します。

- 「ダウンロード前実行スクリプト」
 - 「ダウンロード後実行スクリプト」
 - 「外部 RAM 使用前実行スクリプト」
 - 「外部 RAM 使用后実行スクリプト」
- 外部アドレス空間へのS/Wブ레이크ポイントは設定できません。
 - メモリウィンドウから外部フラッシュメモリを書き換える事はできません。
 - ダウンロードモジュールのアクセスサイズは”1”を選択してください。
 - 外部フラッシュメモリへダウンロード時に外部RAMをワークRAMとして使用する場合は、CPUと同じエンディアンに設定してください。
 - ロックコマンドが使用できないデバイスをご使用の場合は、USDファイルの生成時にロックビット解除を指定してください。この指定により、不要なロックビット状態の確認処理を省略することができます。

6.13 実行時間

ステータスバー、および、ステータスウィンドウに表示される実行時間は、プログラム実行から停止の間の時間です。100us 未満の値は切り捨てられます。なお、ステップ実行、ステップオーバー実行、および、ステップアウト実行では正しい実行時間が表示されません。

6.14 Start/Stopファンクション

Start/Stopファンクション機能にてStartファンクションからユーザプログラムへ移行しない場合、およびStopファンクションが停止しない場合は、”START/STOP時の指定ルーチン実行処理でタイムアウトエラーが発生しました。システムリセットを発行しますか?”とエラーメッセージが表示されます。

[はい] ボタンをクリックした場合、エミュレータを初期化してユーザプログラムが停止します。システムリセット発行後は、トレース記録も初期化します。また、[いいえ] ボタンをクリックした場合、エミュレータを初期化せずにユーザプログラムが停止します。

なお、[はい] ボタンをクリックした後は、デバッグを継続して行うことが可能です。

6.15 ウォッチ機能

ユーザプログラム実行中にシンボル登録した場合、[ウォッチ] ウィンドウの[Value] カラムが”Not available now.”と表示され、未登録状態となります。シンボル登録する場合は、ユーザプログラム停止状態で登録してください。

6.16 デバッグコンソール機能

デバッグコンソール機能を使用する場合、ユーザプログラム中のcharput, charget関数内でシステムクロックを切り替えないでください。

システムクロックを切り替えるとエミュレータとMCUの送受信に影響を与え、データが正常に送受信できない恐れがあります。

6.17 FINEインタフェース

- FINEインタフェースによる外部トレース出力機能およびリアルタイムRAMモニタ機能はサポートしていません。
- FINEインタフェースによるホットプラグイン機能はサポートしていません。

【RX630, RX631, RX63N, RX63T グループマイコン】

- FINEインタフェースは、FINEC, MD/FINED端子を用いた2線式のみをサポートしています。1線式はサポートしていません。

【RX210, RX21A, RX220 グループマイコン】

- FINEインタフェースは、MD/FINED端子を用いた1線式をサポートしています。

6.18 オプション設定関連レジスタ

6.18.1 エンディアン選択レジスタ(MDEB, MDES)について

【RX630, RX631, RX63N, RX63T, RX210, RX21A, RX220 グループマイコン】

エンディアン選択レジスタ (MDEB, MDES)は、[起動設定] ダイアログボックスの[起動と通信] ページにて設定するエンディアン情報がデバッグする動作モードに対応したレジスタへ書き込まれます。

シングルチップモードでデバッグする場合、エンディアン選択レジスタ S (MDES)を[メモリ] ウィンドウ等で書き換えることはできません。ユーザブートモードでデバッグする場合、エンディアン選択レジスタ B (MDEB)を[メモリ] ウィンドウ等で書き換えることはできません。

6.18.2 オプション機能選択レジスタ 1(OFS1)の設定について

【RX630, RX631, RX63N, RX63T, RX210, RX21A, RX220 グループマイコン】

- ダウンロード以外の操作（メモリウィンドウからの操作やラインアセンブル）でオプション機能選択レジスタ1 (OFS1)を書き換える場合、設定を反映するためには以下の手順が必要です。

- ① OFS1を書き換える
- ② 実行操作(OFS1への書き込みが発生、詳細は6.1.2 を参照してください)
- ③ CPUのリセットを実施

ダウンロード操作の場合はダウンロード後にCPUのリセットを実施した時点で反映されます。

- オプション機能選択レジスタ1 (OFS1)では、電圧監視0リセットの有効/無効と電圧検出レベルを設定することができます。本レジスタはフラッシュメモリ上にあり、設定によってE1/E20エミュレータで制御できない可能性があります。

フラッシュROMを書き換える操作(ダウンロードやメモリウィンドウからの操作など)の時にオプション機能選択レジスタ1 (OFS1)の設定値が電圧監視0リセットを発生させる条件の場合、エラーが発生します。

オプション機能選択レジスタ1(OFS1)については、マイコンのハードウェアマニュアルを参照してください。

6.19 その他

6.19.1 フラッシュメモリ書き換え時のレジスタ値

プログラムダウンロード、S/W ブレークポイントの設定、または、メモリウィンドウからの値設定などによりフラッシュメモリの書き換えが発生した場合、フラッシュメモリ関連レジスタ値がデバッガにより書き換えられた状態となりますのでご注意ください。

6.19.2 DMAC、DTCに関して

ユーザプログラム停止の状態ではDMAC、DTC 要求がマイコン内蔵フラッシュROM（転送元）に対して発生しているとき、下記操作をしないでください。

- マイコン内蔵フラッシュROM領域へのS/Wブレークポイントの設定
- [メモリ]ウィンドウや[コマンドライン] ウィンドウからマイコン内蔵フラッシュROM領域への書き込み
- マイコン内蔵フラッシュROM領域へのダウンロード

6.19.3 高速クロック発振器(HOCO)に関して

【RX210, RX21A, RX220 グループマイコン】

- エミュレータは、デバイスとの通信を実現するためにデバイス内蔵の高速クロック発振器(以下、HOCO)を使用します。そのため、HOCOに関連するレジスタの設定に関係なくHOCOは常に発振状態になります。
- HOCOの周波数切り替えとメモリアクセス動作が競合した場合、そのメモリアクセス動作は保証できません。

6.19.4 ロックビット

(a) ダウンロード時のロックビット解除について

ロックビットを設定したフラッシュ ROM 領域に対してユーザプログラムをダウンロードする場合、E1/E20 エミュレータはダウンロードデータがあるブロックのロックビットを解除してからダウンロードします。

(b) ダウンロード完了後のロックビット復帰について

[コンフィグレーションプロパティ] ダイアログボックスの[内蔵プログラム ROM を書き換えるプログラムをデバッグする] チェックボックスの設定により動作が異なります。

【チェックボックスをチェックした場合】

プログラム ROM 領域に対してダウンロードデータがあるブロック、およびダウンロードデータがなく [内蔵フラッシュの上書き] にて上書き登録されていないブロックは、ダウンロード完了後にロックビットを再設定します。

【チェックボックスのチェックを外した場合】

ダウンロード完了後にロックビットを再設定せず解除したままとなります。

6.19.5 エミュレータの切断について

E1/E20 エミュレータの切断は、「2.7 エミュレータを切断する」に記載されたいずれかの方法で必ず切断してください。

強制的にデバッグを終了した場合、次回起動時に正しく起動できません。この状態になった場合は、E20 エミュレータは電源スイッチを切換え、E1 エミュレータは USB インタフェースケーブルを抜き差しして電源を入れ直してください。

6.19.6 動作周波数について

【RX630, RX631, RX63N, RX63T, RX210, RX21A, RX220 グループマイコン】

E1/E20 エミュレータにてデバッグ可能なマイコンの最低動作周波数は32.768kHzです。

6.19.7 USBブートプログラムが格納されたマイコンについて

【RX630, RX631, RX63N グループマイコン】

USBブートプログラムが格納されているマイコンを使ってデバッガでユーザブートモード起動する場合、フラッシュ開発ツールキット(Flash Development Toolkit)または Renesas Flash Programmer を用いて USBブートプログラムを消去してください。

USBブートプログラムが格納されたままデバッガでユーザブートモード起動すると、起動エラーが発生します。

6.19.8 クロック操作を許可する設定について

【RX630, RX631, RX63N, RX63T, RX210, RX21A, RX220 グループマイコン】

デバッガの[コンフィグレーションプロパティ]ダイアログでクロック操作を許可した場合、ターゲット MCU の FlashIF クロック FCLK が動作保証範囲外(LOCO やサブクロック動作中など)でデバッガによる内蔵フラッシュ ROM の書き換えを実施した時、E1/E20 エミュレータがクロックソースを切り替えます。内蔵フラッシュ ROM 書き換えが完了した後は、元のクロックソースに復帰します。

クロックソースを切り替えるため、内蔵フラッシュメモリ書き換え中の周辺クロック動作周波数が変更されますのでご注意ください。

クロック操作の許可設定は、プログラム実行やステップ実行などの実行系機能を実施した後の内蔵フラッシュ ROM を書き換える時に有効となります。なお、デバッガ起動直後や CPU リセット発行時はクロック操作の許可設定に関わらず FCLK が動作保証範囲外の場合は強制的にクロックソースを切り替えます。

6.19.9 MPU領域のアクセスについて

【MPU 実装マイコン】

MPU 領域の参照および設定はプログラム停止中のみ可能です。

プログラム実行中の MPU 領域アクセスは以下になります。

- 参照時のデータ表示は0x00になります。
- 設定時はエラーが発生します。

MPU 領域に対して以下の操作は実施できません。

- ラインアセンブル
- メモリ検索
- メモリ比較
- メモリ転送

付録A メニュー一覧

GUIメニューの一覧を表A.1 に示します。

表A.1 GUIメニュー一覧

メニュー	メニューオプション	ショート カットキー	ツールバー ボタン	備考	
表示	差分			[差分] ウィンドウを表示します	
	マップ			[Map Section Information] ウィンドウを表示します	
	コマンドライン	Ctrl+L		[コマンドライン] ウィンドウを表示します	
	TCL ツールキット	Ctrl+Shift+K		[Console] ウィンドウを表示します	
	ワークスペース	Alt+K		[Workspace] ウィンドウを表示します	
	アウトプット	Alt+O		[Output] ウィンドウを表示します	
	ステータスバー	Alt+A		ステータスバーの表示/非表示を切り替えます	
	逆アセンブリ	Ctrl+D		[逆アセンブリ] ウィンドウを表示します	
	CPU	レジスタ	Ctrl+R		[レジスタ] ウィンドウを表示します
		メモリ	Ctrl+M		[メモリ] ウィンドウを表示します
		IO	Ctrl+I		[IO] ウィンドウを表示します
		ステイタス	Ctrl+U		[ステイタス] ウィンドウを表示します
		RAM モニタ			[RAM モニタ] ウィンドウを表示します
		デバッグ コンソール			[デバッグコンソール] ウィンドウを表示します
イベント	オンチップ ブ레이크 条件設定			[オンチップブ레이크条件設定] ダイアログボックスを表示します	
	トレース 条件設定			[トレース条件設定] ダイアログボックスを表示します	
	パフォーマンス 条件設定			[パフォーマンス条件設定] ダイアログボックスを表示します	

表 A.1 GUI メニュー一覧(つづき)

メニュー	メニューオプション		ショート カットキー	ツールバー ボタン	備考
表示	シンボル	ラベル	Ctrl+Shift+A		[ラベル] ウィンドウを表示します
		ウォッチ	Ctrl+W		[ウォッチ] ウィンドウを表示します
		ローカル	Ctrl+Shift+W		[ローカル] ウィンドウを表示します
	グラフィック	画像	Ctrl+Shift+G		[画像プロパティ] ダイアログボックスを表示します
		波形	Ctrl+Shift+V		[波形プロパティ] ダイアログボックスを表示します
		グラフ			[グラフ] ウィンドウを表示します。
	コード	トレース	Ctrl+T		[トレース] ウィンドウを表示します
		スタック トレース	Ctrl+K		[スタックトレース] ウィンドウを表示します
	パフォーマンス	パフォーマンス 解析	Ctrl+Shift+ P		[パフォーマンス解析] ウィンドウを表示します
デバッグ	同期デバッグ...				同期デバッグの設定が可能な[同期デバッグ] ダイアログボックスを表示します
	デバッグセッション...				デバッグセッションの一覧表示、および追加/削除等が可能な[デバッグセッション] ダイアログボックスを表示します
	デバッグの設定...				デバッグ時の条件やダウンロードモジュール等の設定を行う[デバッグの設定] ダイアログボックスを表示します
	CPUのリセット				ターゲットマイコンをリセットし、PC をリセットベクタアドレスに設定します
	実行		F5		現在の PC からユーザプログラムを実行します
	リセット後実行		Shift+F5		ターゲットマイコンをリセットし、リセットベクタアドレスからユーザプログラムを実行します

表 A.1 GUI メニュー一覧(つづき)









メニュー	メニューオプション	ショート カットキー	ツールバー ボタン	備考
デバッグ	フリー実行			ブレークポイントを無視してユーザプログラムを実行します
	カーソル位置まで実行			現在の PC からテキストカーソルの位置までユーザプログラムを実行します
	カーソル位置に PC 値を設定			テキストカーソルの位置に PC を設定します
	条件を指定して実行...			実行時の PC や PC ブレークポイントの設定が可能な[条件を指定して実行] ダイアログボックスを表示します
	PC 位置を表示	Ctrl+Shift+Y		[エディタ] ウィンドウに PC 位置を表示します
	ステップイン	F11		ユーザプログラムの 1 ブロックを実行して停止します
	ステップオーバ	F10		ユーザプログラムの 1 ブロックを実行して停止しますが、サブルーチンを呼び出す場合は、サブルーチンには入りません
	ステップアウト	Shift+F11		現在の関数の終わりに到達するまでユーザプログラムを実行します
	ステップ...			ステップ動作の設定が可能な[プログラムステップ] ダイアログボックスを表示します
	ステップモード	自動		[エディタ] ウィンドウがアクティブの場合はソースライン一行だけをステップ実行します、[逆アセンブリ] ウィンドウがアクティブの場合はアセンブリ言語命令単位にステップ実行します
	アセンブリ			アセンブリ言語命令単位にステップ実行します
	ソース			ソースライン一行だけをステップ実行します
	プログラムの停止	Esc		ユーザプログラムの実行を停止します
	初期化			デバッグプラットフォームを切断し、再起動させようとしています
	アンロード			オブジェクトプログラムをアンロードします

表 A.1 GUI メニュー一覧(つづき)

メニュー	メニューオプション		ショート カットキー	ツール バー ボタン	備考
デバッグ	接続				デバッグプラットフォームを接続します
	接続解除				デバッグプラットフォームを切断します
	メモリの保存...				メモリ内容をファイル (.bin、.hex、.mot 形式) へ保存します
	メモリのベリファイ...				メモリ内容を保存したファイルをもとにベリファイします
	ダウンロード				オブジェクトプログラムをロードします
基本設定	基数	16 進数			数値の表示/入力時の基数のデフォルト設定を 16 進数とします
		10 進数			数値の表示/入力時の基数のデフォルト設定を 10 進数とします
		8 進数			数値の表示/入力時の基数のデフォルト設定を 8 進数とします
		2 進数			数値の表示/入力時の基数のデフォルト設定を 2 進数とします
	エミュレータ	デバイス セッティング...			ターゲットマイコンに関わる設定を行う[起動設定] ダイアログボックスを表示します
		システム...			エミュレータシステム全体に関わる設定を行う[コンフィグレーションプロパティ] ダイアログボックスを表示します
		Start/Stop ファンクション設定			[Start/Stop ファンクション設定] ダイアログボックスを表示します。

付録B High-performance Embedded Workshopの注意事項

ロードモジュール作成後のソースファイル位置移動に関する注意事項

ロードモジュール作成後にソースファイルを移動させた場合、作成したロードモジュールのデバッグ中にソースファイルを指定するための[ファイルを開く]ダイアログボックスが表示されることがあります。対応するソースファイルを選択し、[開く]ボタンを押してください。

ソースレベル実行機能

- ソースファイル

ロードモジュールに対応しないソースファイルをプログラムウィンドウに表示した場合、そのソースファイルがロードモジュールに対応するソースファイルと同名であるとアドレス表示しますが、そのプログラムウィンドウでは操作できません。

- ステップ

標準Cライブラリ等にも移行します。上位関数に戻るにはステップアウトを使用してください。

なお、メニュー[基本設定]->[オプション]の[デバッグ] ページで「デバッグ情報のないアドレスにステップインしない」をチェックすると、デバッグ情報のないライブラリに 移行しなくなります。

また、forおよびwhile文では、1回のステップでは次の行に進みません。進める場合はもう一度ステップしてください。

ファイルアクセス中の操作について

ロードモジュールのダウンロード中、[メモリ]ウィンドウでの比較、[保存]、[トレース]ウィンドウでのセーブなどの処理中に他の操作を行わないでください。ファイルアクセス処理が正しく実行されない場合があります。

ウォッチ機能

- 最適化時の局所変数

最適化オプションでコンパイルされたソースの局所変数表示は、生成されたオブジェクトコードによって、正しく表示できないことがあります。[逆アセンブリ]ウィンドウを表示し、生成されたオブジェクトコードを確認してください。

また、指定した局所変数の割付け領域がない場合があります。この場合、次のように表示します。

例) 変数名を asc とする。

```
asc Not available now.
```

- 変数名の指定

変数名でないシンボル名(関数名)等を指定した場合、内容は表示しません。

例) 関数名を main とする。

```
main Not available now.
```

コマンドラインインタフェース

- バッチファイル

バッチファイル実行中に、“Not currently available”が表示される場合は、sleepコマンドを挿入してください。sleepさせる時間は動作環境によって異なりますので、調整してください。

例) memory_fillで、“Not currently available”を表示する場合

```
sleep d'3000
memory_fill 0 ffff 0
```

- コマンドファイルでのファイル指定

コマンドファイルの指定方法によりカレントディレクトリが移動する場合があります。コマンドファイル内のファイル指定は、カレントディレクトリの移動に影響をうけないように絶対パスで記述することをお勧めします。

例) FILE_LOAD C:\¥Workspace¥Tutorial¥E1E20¥RX600¥Tutorial ¥Tutorial_LittleEndian
¥Debug_RXxxx_E1_E20_SYSTEM¥tutorial.abs

モトローラSタイプ形式のファイルのロード

High-performance Embedded Workshopでは、レコード末尾が"CRコード"(0Dh)のみのモトローラSタイプ形式ファイルはサポートしていません。モトローラSタイプ形式のファイルをロードする場合は、レコード末尾に"CRコードとLFコード"(0D0Ah)が付いている形式のものを使用してください。

プログラム実行中の[レジスタ]ウィンドウ動作に関する注意事項

プログラム実行中は、[レジスタ]ウィンドウからレジスタ値を変更できません。表示されますが、変更してもレジスタ内容は変更されません。

RUN-TIME 表示における注意事項

E1/E20エミュレータでは、ステータスバー、および[ステータス]ウィンドウにおいてユーザプログラムの実行時間を表示していますが、エミュレータ内部の32ビットカウンタを使用していますので、100µs未満の値は切り捨てられます。なお、ステップ実行、ステップオーバ実行、およびステップアウト実行では正しい実行時間が表示されません。

memory test 機能

[メモリ]メニューから[テスト...]を選択することによって使用するmemory test機能は、本製品ではサポートしていません。

フラッシュライターモード

- 連続してマイコンを複数個書き込む場合、ターゲットのパワーオフ/オンを必ず行ってください。
- フラッシュライターモードでは、ユーザプログラムのダウンロード以外のデバッグ機能を使用することはできません。

逆アセンブリ表示における注意事項

[エディタ]ウィンドウを逆アセンブリモードで表示したときに、定義されていないアセンブラ命令コードの場合は"???"が表示されます。

E1/E20エミュレータ
ユーザーズマニュアル別冊
(High-performance Embedded Workshop RXデバッグ編)

発行年月日 2012年9月1日 Rev.1.00

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



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

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

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<http://japan.renesas.com/contact/>

E1/E20エミュレータ
ユーザーズマニュアル別冊
(High-performance Embedded Workshop RXデバッグ編)