

# E1/E20エミュレータ ユーザーズマニュアル別冊(接続時の注意事項)

## 対象デバイス

R8Cファミリ / R8C/5xシリーズ

R8Cファミリ / R8C/3xT-Aシリーズ

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

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

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

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

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

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

## 目次

	ページ
1. E1/E20 エミュレータマニュアル構成 .....	7
2. E1/E20 エミュレータ仕様 .....	8
2.1 対応マイコン .....	8
2.2 エミュレータ仕様一覧 .....	8
2.3 対応ツールチェーンおよびサードパーティツール .....	11
3. ユーザシステムの設計 .....	13
3.1 E1/E20エミュレータとユーザシステムとの接続コネクタ .....	13
3.2 小型コネクタ変換アダプタ .....	16
3.3 システムグランド系の接続 .....	17
3.4 E1/E20エミュレータ接続用コネクタとマイコン間の推奨接続例 .....	18
3.5 E1/E20エミュレータ内インタフェース回路 .....	20
4. デバッグの準備をする .....	21
4.1 High-performance Embedded Workshopを起動する .....	21
4.2 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） .....	22
4.3 新規にワークスペースを作成する（ツールチェーン使用） .....	24
4.4 既存のワークスペースを開く .....	26
4.5 エミュレータを接続する .....	27
4.5.1 エミュレータを接続する .....	27
4.5.2 エミュレータを再接続する .....	27
4.6 エミュレータを切断する .....	27
4.6.1 エミュレータを切断する .....	27
4.7 High-performance Embedded Workshop を終了する .....	27
4.8 デバッグ設定を行う .....	28
4.8.1 ダウンロードモジュールを指定する .....	28
4.8.2 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 .....	29
4.9 E1/E20エミュレータデバッグの起動手順 .....	30
4.10 エミュレータデバッグ起動時の設定 .....	32
4.11 [起動設定]ダイアログボックス .....	33
4.11.1 デバイスタブ .....	34
4.11.2 通信タブ .....	36
4.12 接続中ダイアログの表示 .....	37
4.12.1 RXファミリ等他のエミュレータデバッグでE1/E20 エミュレータを使用していた場合 .....	38
4.12.2 E1/E20 エミュレータ内のエミュレータファームウェアバージョンが古い場合 .....	39
4.12.3 [起動設定]ダイアログボックスの電源供給設定の内容が正しくない場合 .....	40
4.12.4 IDコード領域に“FFFFFFFFFFFFFFh”以外のIDコードが書き込まれている場合 .....	41
4.12.5 MCUとの通信が出来ない場合 .....	42
4.13 [コンフィグレーションプロパティ]ダイアログボックス .....	43
4.13.1 システムタブ .....	43
4.13.2 MCUタブ .....	44
4.13.3 内蔵フラッシュメモリの上書きタブ .....	45
4.14 R8C E1/E20エミュレータデバッグの起動 .....	46

5. デバッグ機能.....	47
5.1 デバッグ機能一覧.....	47
5.2 ダウンロード機能.....	49
5.3 ソースファイルを開く.....	50
5.3.1 ソースコードを表示する.....	50
5.3.2 ソースファイルでカラムをオフにする.....	51
5.3.3 1つのソースファイルでカラムをオフにする.....	51
5.3.4 アセンブリ言語コードを表示する.....	52
5.3.5 アセンブリ言語コードを修正する.....	52
5.4 メモリアクセス機能.....	53
5.4.1 メモリリード/ライト機能.....	53
5.4.2 そのほかのメモリ操作機能.....	53
5.4.3 SFR領域アクセス時の注意事項.....	54
5.5 ブレーク機能概要.....	56
5.5.1 強制ブレーク機能.....	56
5.5.2 S/Wブレーク(ソフトウェアブレーク).....	56
5.5.3 オンチップブレーク機能.....	56
5.6 S/Wブレークポイントの使用.....	57
5.6.1 S/Wブレークポイントを追加する/削除する.....	58
5.6.2 S/Wブレークポイントを有効にする/無効にする.....	59
5.7 オンチップブレーク機能の概要.....	60
5.7.1 [オンチップイベント条件設定]ダイアログボックス.....	60
5.7.2 イベントタブ.....	61
5.7.3 実行前PCブレークタブ.....	63
5.7.4 [イベント設定]ダイアログボックス.....	64
5.7.5 [イベント設定]ダイアログボックス(イベントタイプ:データアクセス).....	65
5.7.6 イベント設定に関する注意事項.....	66
5.8 オンチップイベント(DAイベント)の追加.....	67
5.8.1 [オンチップイベント条件設定]ダイアログボックスからの追加/変更.....	67
5.8.2 他ウィンドウからドラッグ&ドロップ(追加のみ).....	68
5.8.3 [オンチップイベント条件設定]ダイアログボックスからの削除/有効/無効.....	69
5.8.4 コマンドライン.....	69
5.9 オンチップイベント(PCイベント)の設定.....	70
5.9.1 [オンチップイベント条件設定]ダイアログボックスの追加.....	70
5.9.2 [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.....	71
5.9.3 他ウィンドウからドラッグ&ドロップ(追加のみ).....	72
5.9.4 [オンチップイベント条件設定]ダイアログボックスからの削除/有効/無効.....	73
5.9.5 コマンドライン.....	73
5.10 イベントの登録.....	74
5.10.1 [登録イベント一覧]ダイアログボックス.....	74
5.10.2 イベントの登録.....	75
5.10.3 イベントを都度入力する/再使用する.....	77
5.10.4 イベントを適用させる.....	77
5.11 オンチップイベント条件設定内容を保存する/読み込む.....	78
5.11.1 [オンチップイベント条件設定]内容を保存する.....	78
5.11.2 [オンチップイベント条件設定]内容を読み込む.....	78
5.12 [登録イベント一覧]ダイアログボックスの設定内容を保存する/読み込む.....	78
5.12.1 [登録イベント一覧]ダイアログボックスの内容を保存する.....	78
5.12.2 [登録イベント一覧]ダイアログボックス内容を読み込む.....	78

5.13	トレース機能	79
5.13.1	トレース機能概要	79
5.13.2	トレース設定項目	80
5.13.3	トレースメニュー	81
5.13.4	分岐トレース	81
5.13.5	統計	82
5.13.6	トレース情報をファイルに保存する	82
5.14	ステータスバー機能	83
5.15	Start/Stopファンクション機能	84
5.15.1	[Start/Stopファンクション設定]ダイアログボックスを開く	84
5.15.2	実行するルーチンを指定する	84
5.15.3	Start/Stopファンクションの制限事項	85
5.15.4	指定ルーチンの記述に関する制限事項	85
5.16	簡易スタックオーバーフロー機能	86
5.17	オンラインヘルプ	86
6.	チュートリアル	87
6.1	はじめに	87
6.2	High-performance Embedded Workshopの起動	87
6.3	エミュレータの接続	87
6.4	チュートリアルプログラムのダウンロード	88
6.4.1	チュートリアルプログラムをダウンロードする	88
6.4.2	ソースプログラムを表示する	89
6.5	S/Wブレークポイントの設定	90
6.6	プログラムの実行	91
6.6.1	CPUのリセット	91
6.6.2	プログラムを実行する	91
6.7	ブレークポイントの確認	92
6.7.1	ブレークポイントを確認する	92
6.8	レジスタ内容の変更	93
6.9	シンボルの参照	94
6.10	メモリ内容の確認	95
6.11	変数の参照	96
6.12	ローカル変数の表示	98
6.13	プログラムのステップ実行	99
6.13.1	ステップインの実行	99
6.13.2	ステップアウトの実行	100
6.13.3	ステップオーバの実行	100
6.14	プログラムの強制ブレーク	101
6.15	オンチップブレーク機能	102
6.15.1	指定アドレスを実行したときにプログラム実行を停止(実行前PCブレーク)	102
6.15.2	メモリにアクセスしたときにプログラム実行を停止	103
6.16	トレース機能	104
6.16.1	トレース情報の表示	105
6.17	スタックトレース機能	107
6.18	さてつぎは？	108

7. E1/E20 エミュレータ使用上の注意事項.....	109
7.1 占有するマイコン資源.....	109
7.1.1 E1/E20 エミュレータ用プログラム占有領域.....	109
7.1.2 E1/E20 エミュレータ用占有端子.....	111
7.1.3 割り込み機能(使用不可).....	111
7.1.4 E1/E20 エミュレータ用使用スタック領域.....	111
7.1.5 E1/E20 エミュレータ使用SFR.....	111
7.1.6 オプション機能選択領域.....	112
7.1.7 E1/E20 エミュレータでのレジスタ初期化.....	113
7.1.8 RAM初期化.....	113
7.1.9 マイコンの予約領域.....	113
7.1.10 ユーザプログラム停止中のDTC.....	113
7.1.11 2.7V未満でのデバッグに関する注意事項.....	113
7.1.12 消費電力低減のデバッグ.....	113
7.2 リセット.....	114
7.3 内部ROM領域(フラッシュメモリ).....	116
7.3.1 内部ROM領域の変更.....	116
7.3.2 CPU書き換えモードのデバッグに関する注意事項.....	116
7.3.3 E1/E20 エミュレータからのフラッシュメモリ書き換え中の注意事項.....	117
7.3.4 ユーザプログラム実行中のフラッシュメモリ.....	117
7.3.5 デバッグに使用したマイコン.....	117
7.3.6 フラッシュメモリのIDコード.....	118
7.4 電源供給に関して.....	119
7.5 ユーザプログラム停止中の動作.....	119
7.6 ユーザプログラム実行中のメモリアクセス.....	119
7.7 プログラムの最終評価.....	119
7.8 デバッグ機能.....	120
7.8.1 ステップ実行機能.....	120
7.8.2 その他デバッグ機能.....	121
付録A メニュー一覧.....	122
付録B High-performance Embedded Workshopの注意事項.....	127

## 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E1/E20ユーザーズマニュアル別冊(本マニュアル)の2部で構成されています。

E1/E20エミュレータご使用にあたり、両方のユーザーズマニュアルを必ずお読みになってください。  
なお本書では、“L”アクティブの信号を表記するため信号名の末尾に“#”を付加しています(例: RESET#)。

### (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

E1/E20エミュレータユーザーズマニュアル別冊には、エミュレータデバッガの機能説明および操作方法、マイコンに依存する内容や注意事項が記載されています。

- R8C E1/E20エミュレータデバッガがサポートするデバッグ機能
- E1/E20エミュレータデバッガの各機能
- 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
- E1/E20エミュレータが使用するマイコンの資源
- E1/E20エミュレータ使用時の注意事項
- E1/E20エミュレータデバッガ起動時の設定 等
- E1/E20エミュレータデバッガの操作方法
- E1/E20エミュレータデバッガの起動からデバッグ作業までのチュートリアル 等

#### 【留意事項】

- FDT(別売)の仕様および対応MCU等に関しては、弊社Webサイト(<http://japan.renesas.com/tools>)のフラッシュ開発ツールキットページにてご確認ください。
- FDTはフラッシュ開発ツールキット(Flash Development Toolkit)を示します。
- 本マニュアルに記載の図は、一部R8C/3xシリーズMCUで動作させた時のものを使用しています。

#### 商標

Microsoft, MS-DOS, Visual SourceSafe, 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登録商標または商標です。

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。

#### 用語説明

- エミュレータファームウェア: E1/E20エミュレータ内部に格納されている制御プログラムを指します。  
エミュレータデバッガとの通信内容を解析して、  
エミュレータのハードウェアを制御します。  
エミュレータデバッガのバージョンアップ時等には、エミュレータデバッガからダウンロードすることができます。
- デバッグモニタ: ターゲットMCU内部に書き込んで、エミュレータファームウェアとの通信内容を解析して、ターゲットMCUを制御します。

## 2. E1/E20エミュレータ仕様

### 2.1 対応マイコン

本マニュアルでの対応マイコンを表 2.1に示します。

表 2.1 対応マイコン一覧

項目	内容
対応マイコン	R8CファミリR8C/5xシリーズ R8C/5xE : R8C/54E, 56Eグループ R8C/5xF : R8C/54F, 56Fグループ R8C/5xG : R8C/54G, 56Gグループ R8C/5xH : R8C/54H, 56Hグループ
	R8CファミリR8C/3xT-Aシリーズ R8C/36T-A, 38T-Aグループ
対応動作モード	シングルチップモード

### 2.2 エミュレータ仕様一覧

表 2.2にR8C E1/E20エミュレータデバッグで対応のエミュレータ仕様、表 2.3、表 2.4にR8C E1/E20エミュレータデバッグの動作環境を示します。表 2.5にR8C E1/E20エミュレータデバッグのMCU関連仕様を示します。

表 2.2 対応エミュレータ仕様

項目	内容	
対応エミュレータ	E1 (R0E000010KCE00) E20 (R0E000200KCT00)	
エミュレータ用電源	不要 (USBバスパワードのため、ホストマシンより供給)	
対応エミュレータデバッグ	R8C E1/E20エミュレータデバッグ	
使用環境条件	温度	動作時 : 10~35℃ 非動作時 : -10~50℃
	湿度	動作時 : 35~80%RH 結露なし 非動作時 : 35~80%RH 結露なし
	振動	動作時 : 最大2.45m/s <sup>2</sup> 非動作時 : 最大4.9m/s <sup>2</sup> 梱包輸送時 : 最大14.7m/s <sup>2</sup>
	周囲ガス	腐食性ガスのないこと



表 2.3 R8C E1/E20エミュレータデバッグ動作環境(Windows® XP)

項目	内容
ホストマシン	IBM PC/AT互換機
OS	Windows® XP(32ビット版)* *64ビット版のWindows® XPには対応していません。
CPU	Pentium 4 1.6GHz 以上を推奨
ホストマシンとの インタフェース	USB(USB2.0 フルスピード/ハイスピード)* * USB1.1対応のホストコンピュータにも接続可能 * USBインタフェースは、すべてのホストコンピュータ、USBデバイス、 USBハブの組合せでの動作を保障するものではありません。
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グ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グのインストールに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 2.4 R8C E1/E20エミュレータデバッグ動作環境(Windows Vista®, Windows® 7)

項目	内容
ホストマシン	IBM PC/AT互換機
OS	Windows® 7(32ビット版, 64ビット版) Windows Vista®(32ビット版) *64ビット版のWindows Vista®には対応していません。
CPU	Pentium 4 3.0GHz以上またはCore2 Duo 1GHz以上を推奨
ホストマシンとの インタフェース	USB(USB2.0 フルスピード/ハイスピード)* * USB1.1対応のホストコンピュータにも接続可能 * USBインタフェースは、すべてのホストコンピュータ、USBデバイス、 USBハブの組合せでの動作を保障するものではありません。
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グ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グのインストールに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 2.5 R8C E1/E20エミュレータデバッグの仕様一覧(MCU関連項目)

項目	内容
動作電圧	1.8~5.5[V] 【注1】 【注2】
動作周波数範囲	最大動作周波数 : 32MHz 【注2】 最低動作周波数 : 32.768kHz
ユーザシステムとのインタフェース	14ピンコネクタ* * E20エミュレータ接続時には、E20エミュレータ付属の「38ピン/14ピン変換アダプタ」が必要です。
MCUとの通信用端子	MODE端子
MCUとの通信方式	1線式クロック非同期形シリアル
占有するマイコンの資源	<ul style="list-style-type: none"> <li>・ 内部ROM 2Kバイト (一部製品のみ, 「7.1 占有するマイコン資源」(109ページ)参照)</li> <li>・ スタック 8バイト</li> <li>・ アドレス一致割り込み</li> <li>・ ベクタ領域(BRK命令, アドレス一致, シングルステップ, アドレスブレイク)</li> </ul>
電源供給機能	ユーザシステムに3.3Vまたは5.0V供給可能(最大200mA) 【注3】

【注1】 フラッシュ書き換えが発生する場合の動作電圧は、MCUのハードウェアマニュアルのプログラム/イレートの電圧範囲内にしてください。

詳細は、「7.1.11 2.7V未満でのデバッグに関する注意事項」(113ページ)を参照してください。

【注2】 詳細は使用されるMCUのハードウェアマニュアルをご参照ください。

【注3】 電源供給機能はE1エミュレータのみです。

量産工程でのフラッシュ書き込み機として使用する場合は、エミュレータからの電源供給機能は使用せずにマイコン仕様に合致した電源を別途ユーザシステムから供給してください。

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。

E1エミュレータ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、精度の保証が出来ません。

### 2.3 対応ツールチェーンおよびサードパーティツール

表 2.6に示すツールチェーンおよびサードパーティツールで作成したモジュールのデバッグが可能です。

表 2.6 対応ツールチェーンとサードパーティツール

ツールチェーン	M3T-NC30WA V. 5. 20 Release 01以降
サードパーティツール	TASKING M16C C/C++/EC++コンパイラ V. 2. 3r1以降【注1】 IAR EWM16C V. 2. 12以降

**【注1】 ELF/DWARF2フォーマットで作成したロードモジュールのデバッグに関する注意事項**

TASKING M16C C/C++/EC++コンパイラ V3. 0r1を使用してELF/DWARF2フォーマットのロードモジュールを作成した場合、ウォッチウィンドウで基底クラスのメンバ変数を表示する際に以下の注意事項があります。

〈注意事項〉

基底クラスを持つクラスオブジェクトを定義した場合、

- ケース1：クラスオブジェクトから基底クラスのメンバ変数を直接参照することができません(\*1)。  
=>クラスオブジェクトからの間接参照により基底クラスのメンバ変数を参照してください(\*2)(\*3)。
- ケース2：PC値が派生クラスのメンバ関数内にある場合、基底クラスのメンバ変数を直接参照することができません(\*4)。  
=>thisポインタからの間接参照により基底クラスのメンバ変数を参照してください(\*5)(\*6)。

図 2. 1に記述例、図 2. 2にウォッチウィンドウへの登録例を示します。

```

////////////////////////////////////
*.h
class BaseClass
{
public:
    int m_iBase;
public:
    BaseClass() {
        m_iBase = 0;
    }
    void BaseFunc(void);
};

class DerivedClass : public BaseClass
{
public:
    int m_iDerive;
public:
    DerivedClass() {
        m_iDerive = 0;
    }
    void DerivedFunc(void);
};

*.cpp
main()
{
    class DerivedClass ClassObj;
    ClassObj.DerivedFunc();
    return;
}

void BaseClass::BaseFunc(void)
{
    m_iBase = 0x1234;
}

void DerivedClass::DerivedFunc(void)
{
    BaseFunc();
    m_iDerive = 0x1234;
}
////////////////////////////////////

```

図 2.1 記述例

```

////////////////////////////////////
ケース1 : PC値がmain()関数内にある場合
(1)"ClassObj.m_iBase"                :参照不可(*1)
(2)"ClassObj.__b_BaseClass.m_iBase"   :参照可(*2)
(3)"ClassObj"
    - "__b_BaseClass"
    - "m_iBase"                       :参照可(*3)
    - "m_iDerive"
    - : 展開記号

ケース2 : PC値がDerivedClass::DerivedFunc()関数内にある場合
(1)"m_iBase"                          :参照不可(*4)
(2)"this->__b_BaseClass.m_iBase"       :参照可(*5)
(3)"__b_BaseClass.m_iBase"            :参照可(*5)
(4)"this"
    - "*"
    - "__b_BaseClass"
    - "m_iBase"                       :参照可(*6)
    - "m_iDerive"
(5)"__b_BaseClass"
    - "m_iBase"                       :参照可(*6)
////////////////////////////////////

```

図 2.2 ウォッチウィンドウへの登録例

### 3. ユーザシステムの設計

#### 3.1 E1/E20エミュレータとユーザシステムとの接続コネクタ

E1/E20エミュレータを接続するためには、ユーザシステム上にユーザインタフェースケーブル接続用のコネクタを実装する必要があります。

E1/E20エミュレータが推奨するE1/E20接続コネクタを表 3.1、14ピンコネクタ使用時のE1エミュレータとの接続方法を図 3.1、E20エミュレータでの38ピン/14ピン変換アダプタ使用時の接続方法を図 3.2に示します。

FDTにてプログラマとしてE1/E20エミュレータを使用する場合も接続仕様は同じです。

ユーザシステム設計の際には、「図 3.3 E1/E20用14ピン接続コネクタのピン配置」(15ページ)および「3.4 E1/E20エミュレータ接続用コネクタとマイコン間の推奨接続例」(18ページ)を参考にしてください。

また、ユーザシステム設計の際には、E1/E20エミュレータユーザーズマニュアルおよび関連デバイスのハードウェアマニュアルを必ずお読みになってください。

表 3.1 推奨コネクタ

	型名	メーカー	仕様
14ピンコネクタ	7614-6002	住友スリーエム株式会社	14ピンストレートタイプ(日本国内推奨)
	2514-6002	3M Limited	14ピンストレートタイプ (日本国外推奨)

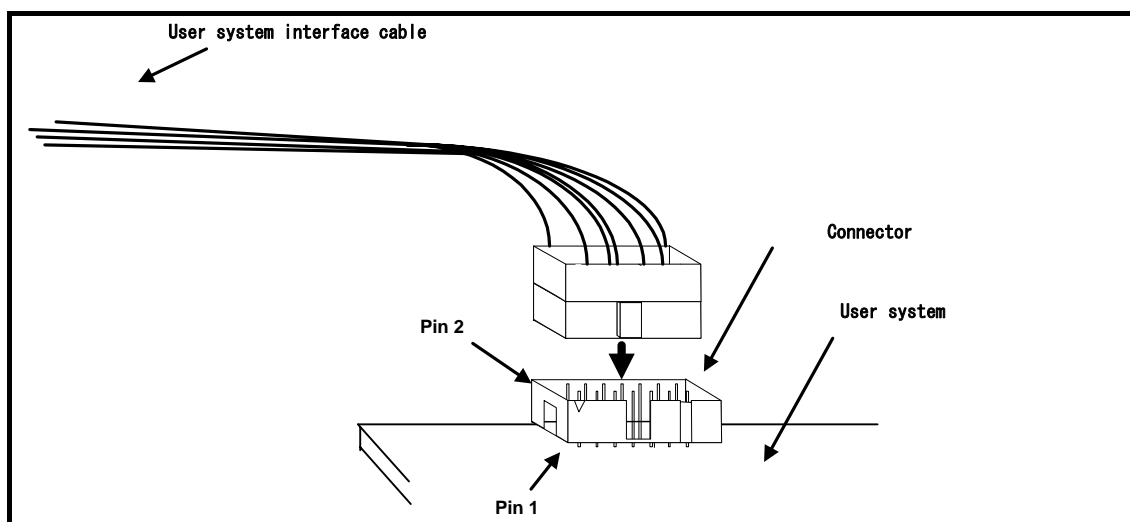


図 3.1 14ピンコネクタ使用時のユーザインタフェースケーブル接続方法

#### 【留意事項】

- 14ピン接続コネクタの周囲5 mm四方に部品高さが10 mmを超える部品を実装しないでください。
- 14ピン接続コネクタの2, 12, 14ピンはユーザシステム基板上でしっかりとGNDに接続してください。  
電気的なGNDとして使用する他、E1/E20エミュレータがユーザシステムコネクタの接続を監視するためにも使用しています。
- ユーザインタフェースケーブルをエミュレータ、ユーザシステムのコネクタから抜き差しする時は、必ずケーブル先端のコネクタカバーをつかんで抜き差しを行ってください。コネクタカバーをつかまずにケーブル部分をつかんで抜き差しを行うと、ケーブル断線の原因となります。
- 対応マイコンのトレースはMCU内蔵機能のみで実現しています。  
このため外部トレース出力機能はサポートしていませんので、E20エミュレータに付属の38ピン/14ピン変換基板でユーザシステム上の14ピンコネクタへ接続してください。

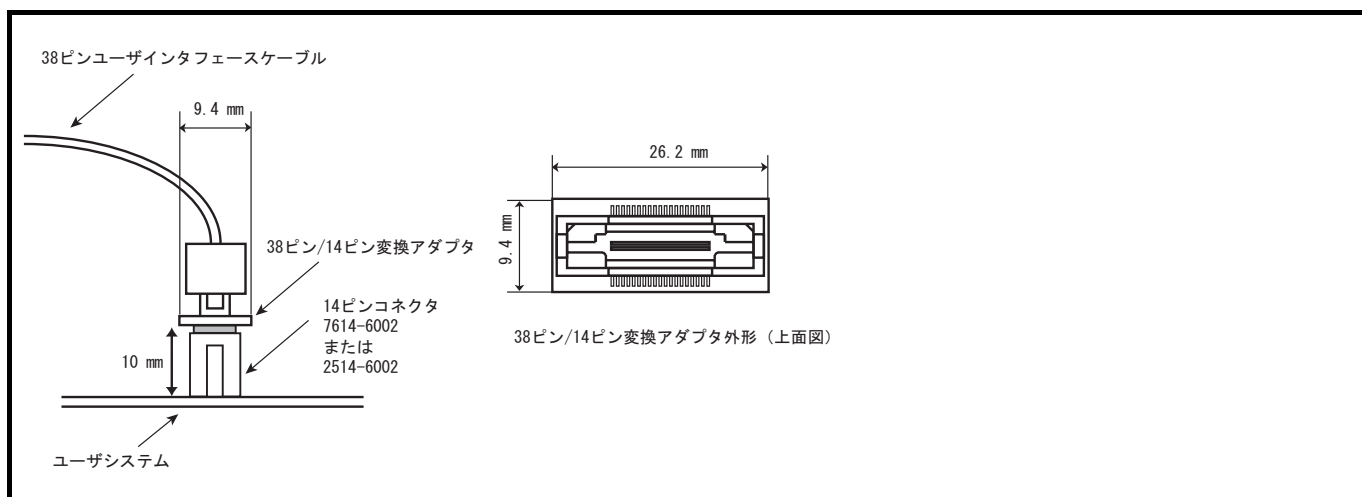


図 3.2 38ピン/14ピン変換アダプタ使用時のユーザインタフェースケーブル接続方法 (E20使用時)

【留意事項】

- 14ピン接続コネクタの周囲5 mm四方に部品高さが10 mmを超える部品を実装しないでください。
- 14ピン接続コネクタの2, 12, 14ピンはユーザシステム基板上でしっかりとGNDに接続してください。  
電気的なGNDとして使用する他、E1/E20エミュレータがユーザシステムコネクタの接続を監視するためにも使用しています。
- ユーザインタフェースケーブルをエミュレータ、ユーザシステムのコネクタから抜き差しする時は、必ずケーブル先端のコネクタカバーをつかんで抜き差しを行ってください。コネクタカバーをつかまずにケーブル部分をつかんで抜き差しを行うと、ケーブル断線の原因となります。

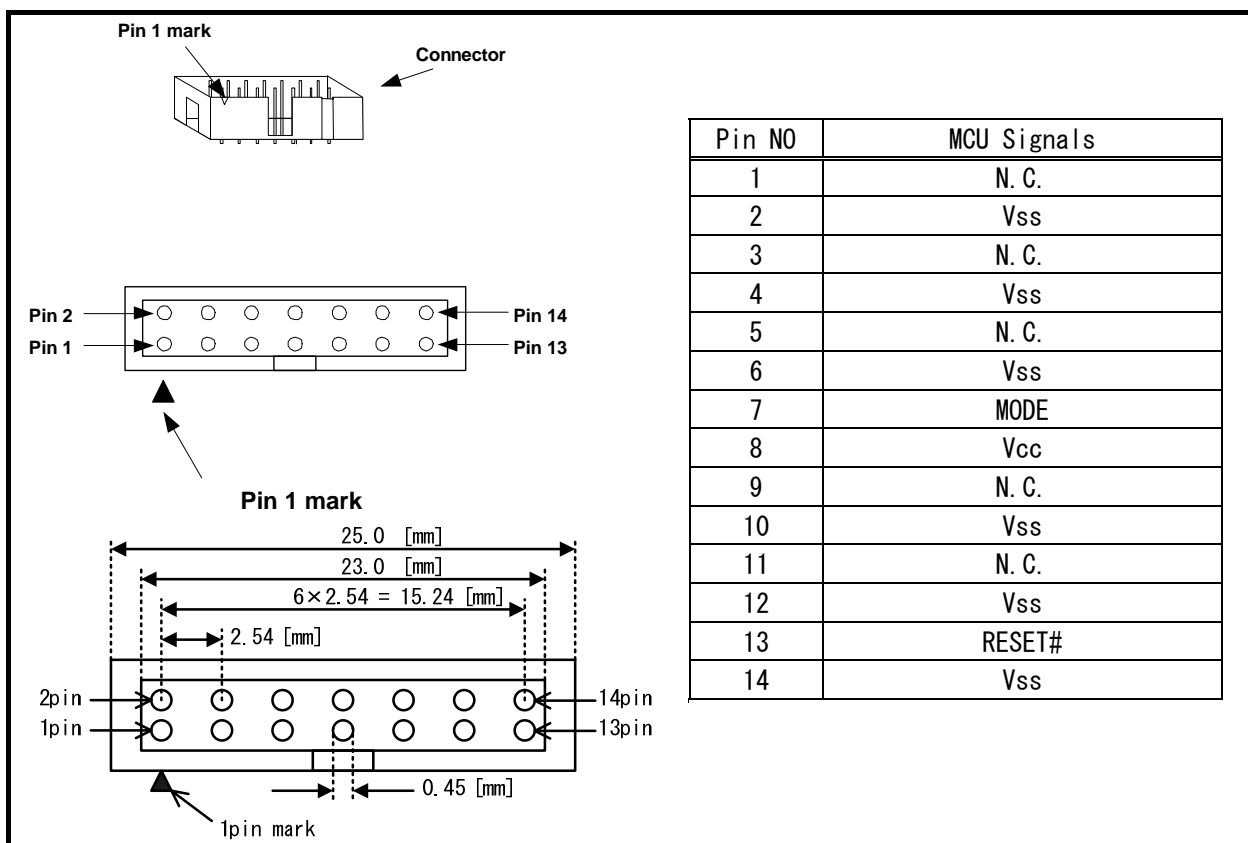


図 3.3 E1/E20用14ピン接続コネクタのピン配置

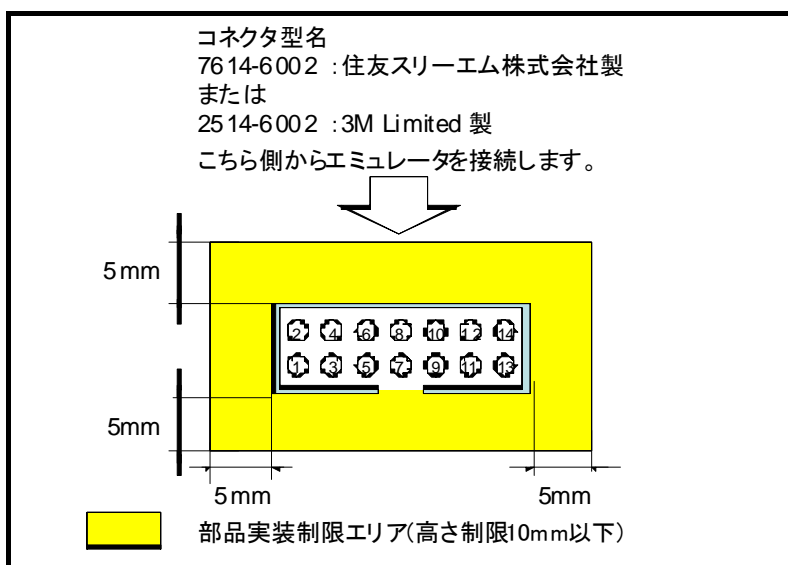


図 3.4 14ピン接続コネクタの実装制限エリア

## 【留意事項】

- 14番ピンはE1/E20エミュレータとユーザシステムの接続確認に使用されており、E1/E20エミュレータ内部では直接Vssに接続されていません。Vssとしては14番ピンのほかに必ず2, 4, 6, 10, 12番ピンも接続してください。
- ユーザシステムコネクタのピン配置には注意してください。
- N. C. は何も接続しないでください。
- エミュレータ接続コネクタ周辺に他の部品を実装する場合、高さ制限がありますので注意してください。

### 3.2 小型コネクタ変換アダプタ

別売のE1エミュレータ用小型コネクタ変換アダプタ (R0E000010CKZ11) をご使用になる場合、E1エミュレータの標準インタフェースコネクタとピン配置が異なります。

E1エミュレータ用小型コネクタ変換アダプタ使用時の14ピンコネクタのピン配置を表2. 3に示します。

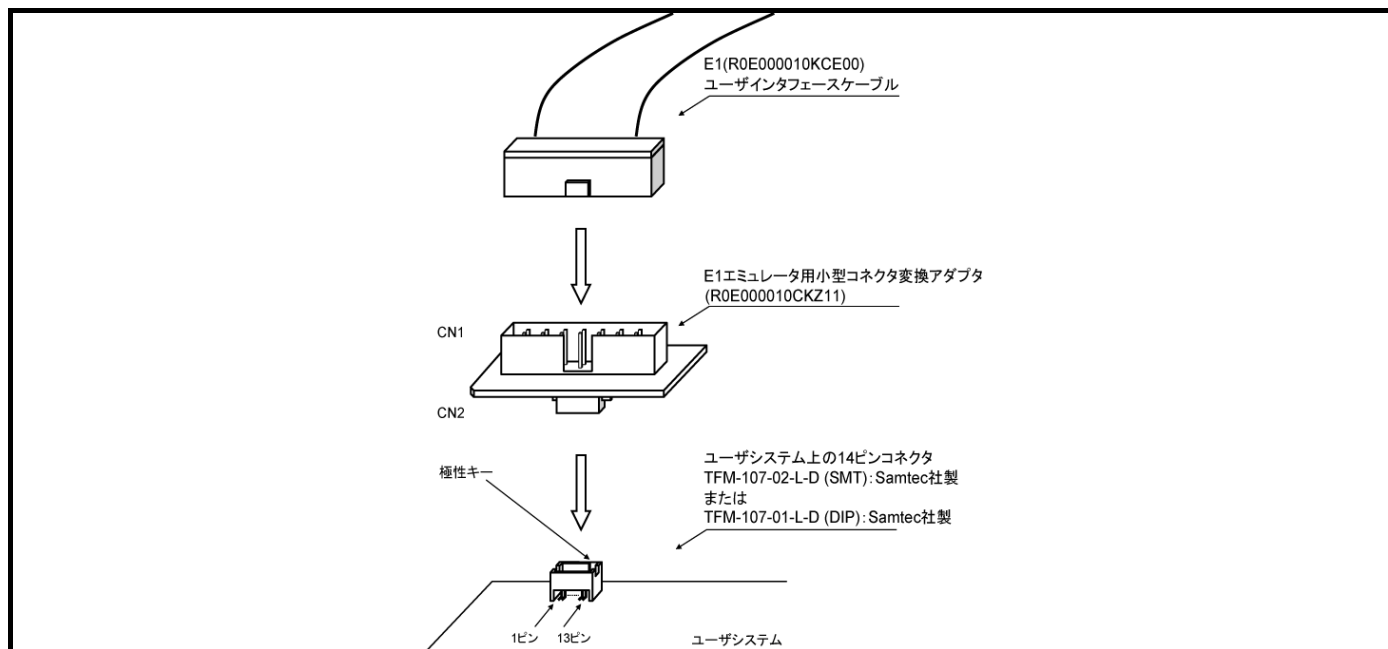


図 3.5 E1エミュレータ用小型コネクタ変換アダプタの使用方法

表 3.2 E1エミュレータ用小型コネクタ変換アダプタ使用時のコネクタピン配置

pin No	MCU Signals
1	Vss
2	MODE
3	Vcc
4	N. C
5	RESET#
6	Vss
7	Vss
8	N. C
9	N. C
10	N. C
11	Vss
12	Vss
13	N. C
14	Vss

#### 【留意事項】

- 14番ピンはE1/E20エミュレータとユーザシステムの接続確認に使用されており、E1/E20エミュレータ内部では直接Vssに接続されていません。Vssとしては14番ピンのほかに必ず1, 6, 7, 11, 12番ピンも接続してください。
- ユーザシステムコネクタのピン配置には注意してください。
- N. C. は何も接続しないでください。
- エミュレータ接続コネクタ周辺に他の部品を実装する場合、高さ制限がありますので注意してください。
- E1エミュレータ用小型コネクタ変換アダプタの仕様はR0E000010CKZ11のユーザーズマニュアルを参照ください。



### 3.3 システムグランド系の接続

エミュレータのシグナルグランドは、ユーザシステムのシグナルグランドに接続されます。

エミュレータ内部では、シグナルグランドとフレームグランドが接続されています。ユーザシステムでは、シグナルグランドとフレームグランドを接続せず、フレームグランドだけを接地してください(図 3.6参照)。

ユーザシステム内でフレームグランドとシグナルグランドを切り離すのが難しい場合、ホストマシンのDC電源入力 (ACアダプタ) のGND電位とユーザシステムのフレームグランドを同電位にしてください。ホストマシンとユーザシステムのGNDに電位差がある場合、インピーダンスが低いGNDラインに過電流が流れ、細いラインの焼損などの危険性があります。

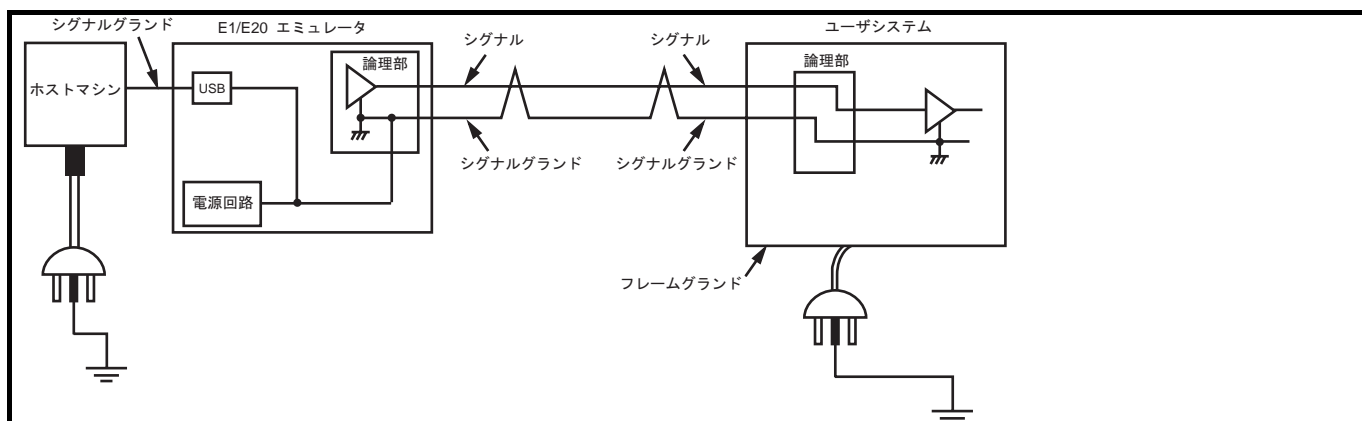


図 3.6 システムグランド系の接続

## 警告



システムグランドは必ずユーザシステム上で、フレームグランドとシグナルグランドを切り離してください。フレームグランドとシグナルグランドを接続した状態でエミュレータを接続するとグランド電位の差により発煙、発火、および感電の危険性があります。

## 3.4 E1/E20エミュレータ接続用コネクタとマイコン間の推奨接続例

図 3.7にE1/E20エミュレータ接続用コネクタとマイコン間の推奨接続例を示します。  
FDTにてプログラマとしてE1/E20エミュレータを使用する場合も接続仕様は同じです。

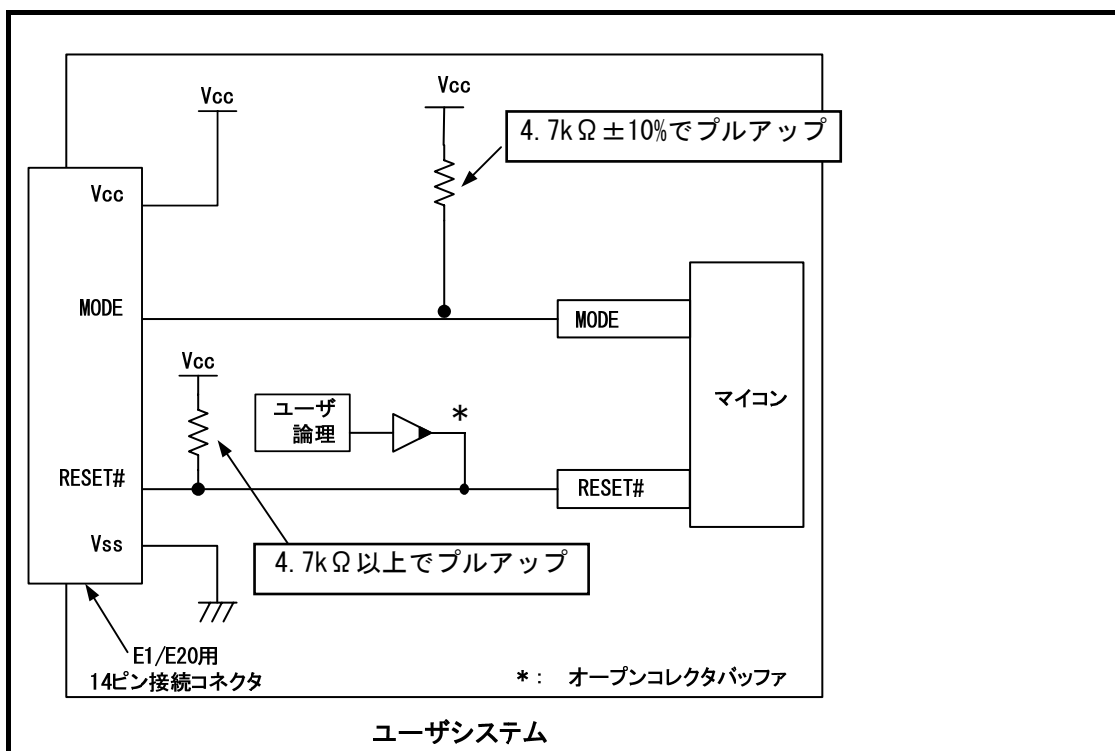


図 3.7 E1/E20エミュレータの接続例

## 【留意事項】

- プルアップに連抵抗を使用する場合、他の端子によるノイズの影響を受ける可能性がありますのでMODE端子の抵抗は他端子の抵抗と分けてください。
- エミュレータ接続コネクタとマイコン間のパターン長はできるだけ短く（推奨50mm以内）してください。また、基板上でエミュレータ接続コネクタとマイコン間以外への信号線の引き回しは行わないでください。
- E1/E20エミュレータを使用しない場合の端子処理については、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。

## (1) MODE端子

MODE端子は、E1/E20エミュレータが占有してマイコン制御および強制ブレークコントロールに使用します。本端子にはコンデンサ等を接続しないでください。

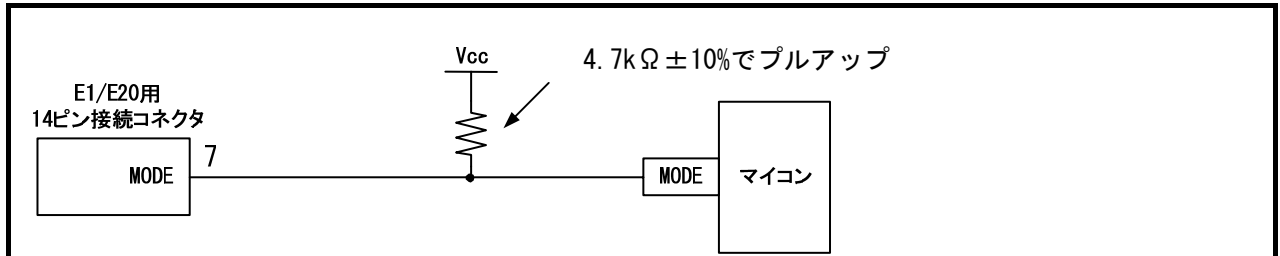


図 3.8 E1/E20エミュレータとMODE端子の接続

## (2) RESET#端子

RESET#端子はE1/E20エミュレータが、“L”出力および端子状態監視のため使用します。そのため、ユーザシステムのリセット回路はオープンコレクタ出力またはORによるリセット回路をご使用ください。プルアップ抵抗の推奨値は4.7kΩ以上です。

E1/E20エミュレータから“L”を出力することにより、マイコンをリセットしています。“H”を出力するタイプのリセットICでは、ユーザシステム上のリセット回路を“L”にすることができないため、正常にE1/E20エミュレータを動作させることができません。

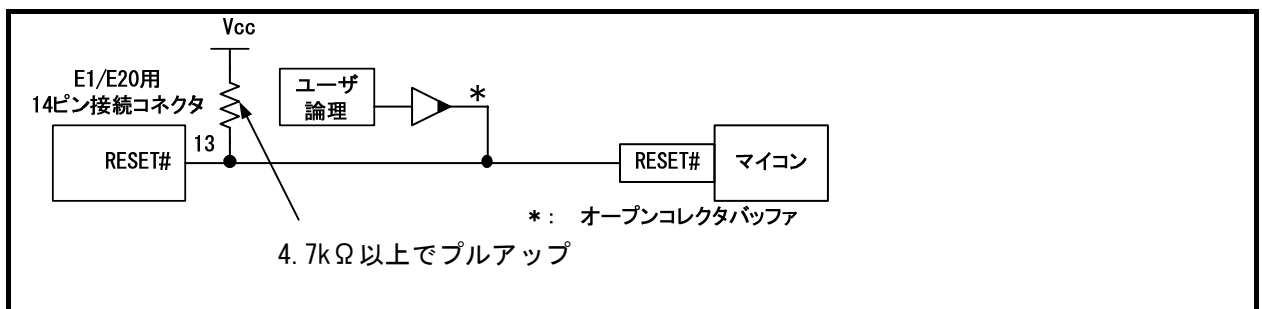


図 3.9 E1/E20エミュレータとRESET#端子の接続

## (3) その他端子

- VssはマイコンのVss端子に、VccはマイコンのVcc端子に接続してください。
- Vcc端子への入力可能電圧は、マイコンの保証範囲内です。
- 14番ピンはE1/E20エミュレータとユーザシステムの接続確認に、4, 6, 10番ピンは内部回路に接続されており、E1/E20エミュレータ内部では直接Vssに接続されていません。
- Vssとしては4, 6, 10, 14番ピンのほかに必ず2, 12番ピンも接続してください。
- N. C. は何も接続しないでください。

## ⚠ 警告



電源供給をする時、ユーザシステムの電源回路がショートしていないか確認をしてください。エミュレータ接続コネクタのピンの並びに問題がないかを確認した上で、E1/E20エミュレータを接続してください。誤って接続した場合、ホストマシン、E1/E20エミュレータとユーザシステムの発煙発火の可能性があります。

## 3.5 E1/E20エミュレータ内インタフェース回路

図 3.10にE1エミュレータ内インタフェース回路、図 3.11にE20エミュレータ内インタフェース回路を示します。プルアップ抵抗の値などを決定するときに参考にしてください。

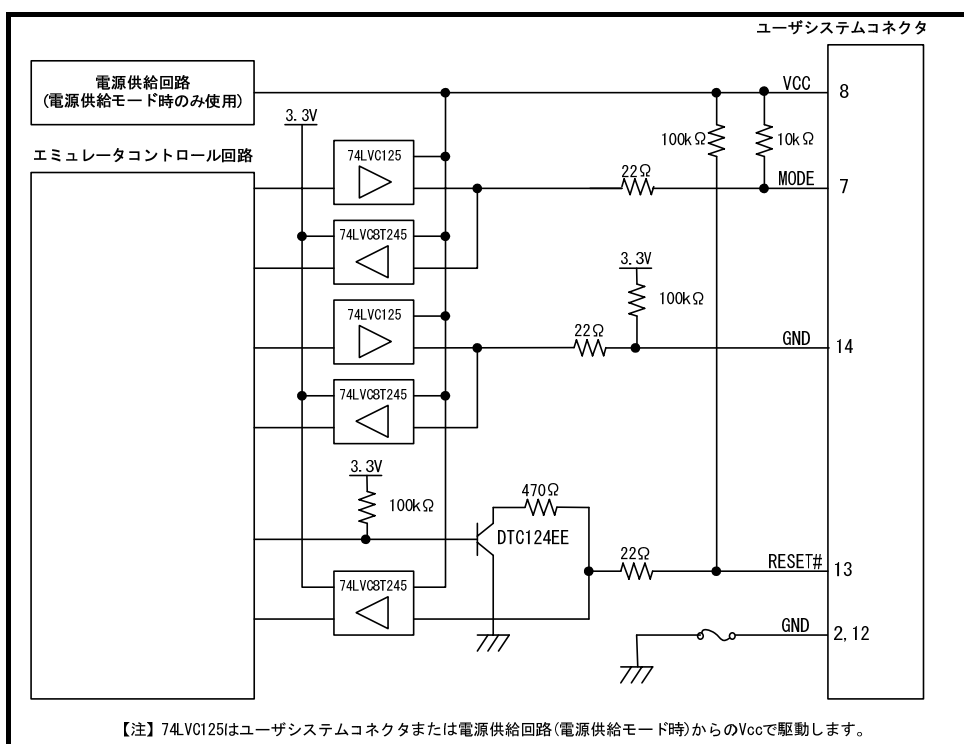


図 3.10 E1エミュレータ内インタフェース回路 (参考)

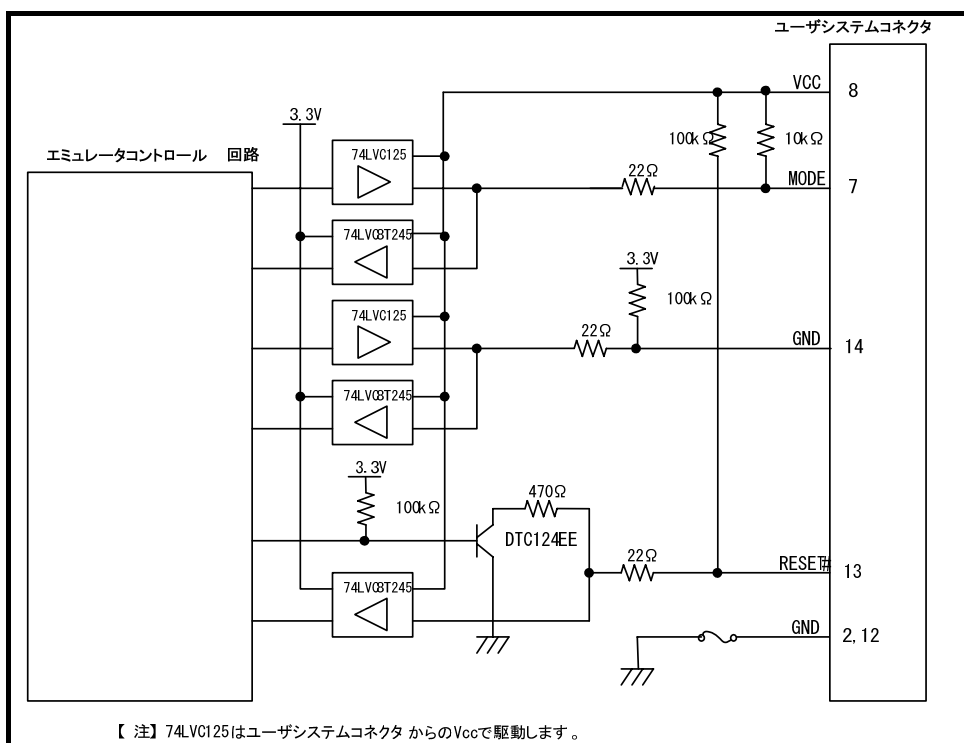


図 3.11 E20エミュレータ内インタフェース回路 (参考)

## 4. デバッグの準備をする

### 4.1 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。

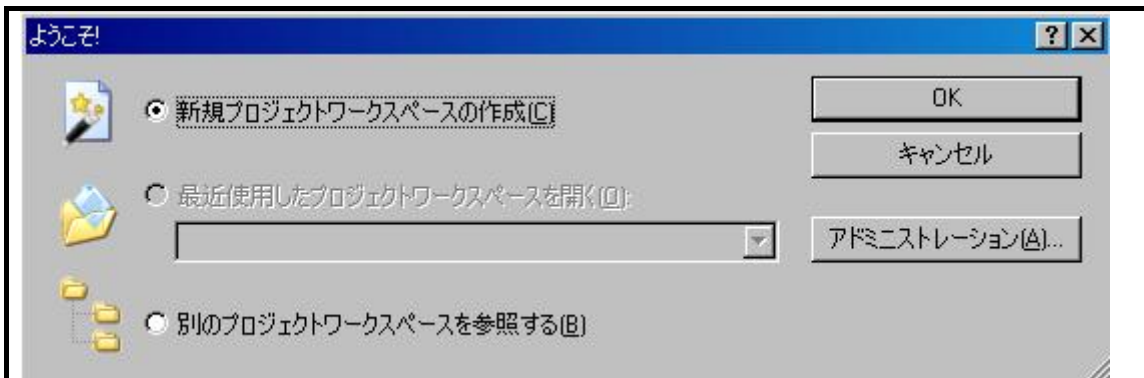


図 4.1 [ようこそ!] ダイアログボックス

- (3) 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。

## 4.2 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）

ツールチェイ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、ツールチェインは含まれていません。ツールチェインは 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。

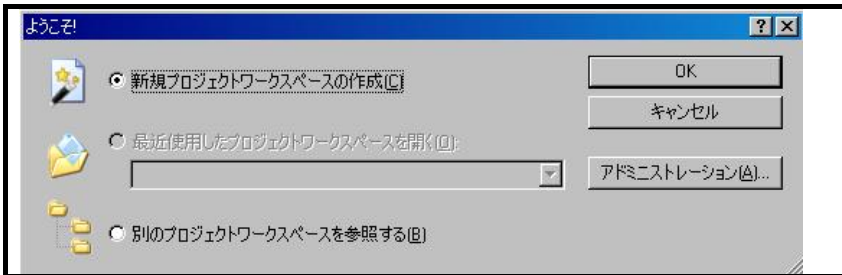


図 4.2 新規プロジェクトワークスペースの作成

- (2) Project Generatorを開始します。



図 4.3 [新規プロジェクトワークスペース] ダイアログボックス

- [ワークスペース名]: 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。

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。

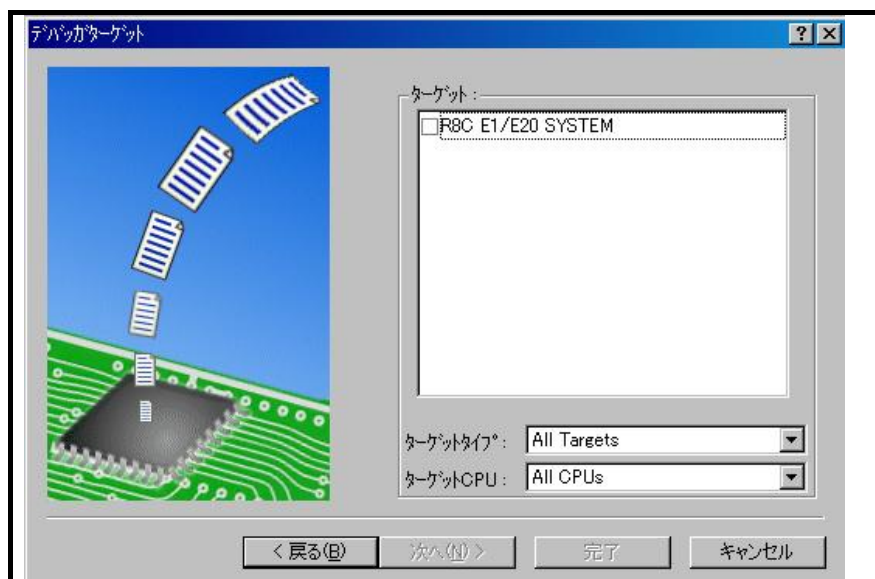


図 4.4 [デバッガターゲット] ダイアログボックス

(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。

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。

(5) [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。



図 4.5 [デバッガオプション] ダイアログボックス

### 4.3 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。

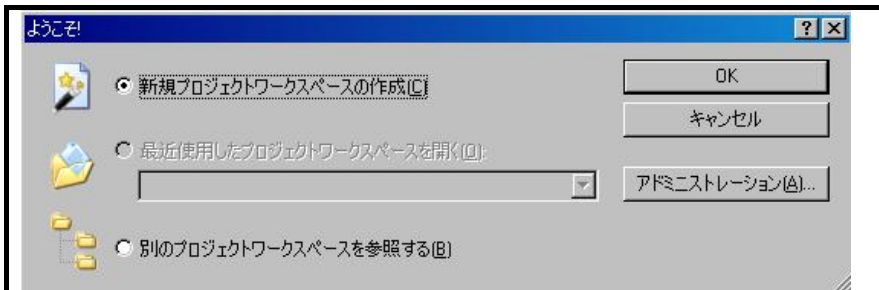


図 4.6 [ようこそ!] ダイアログボックス

- (2) Project Generatorを開始します。

下記項目の設定が完了したら [OK] ボタンを押してください。

- [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。

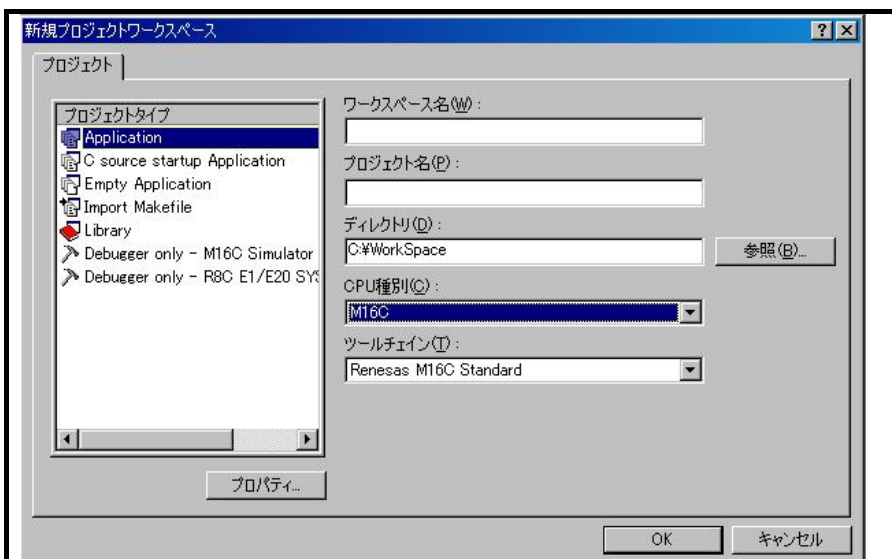


図 4.7 [新規プロジェクトワークスペース] ダイアログボックス



- (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。  
 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。

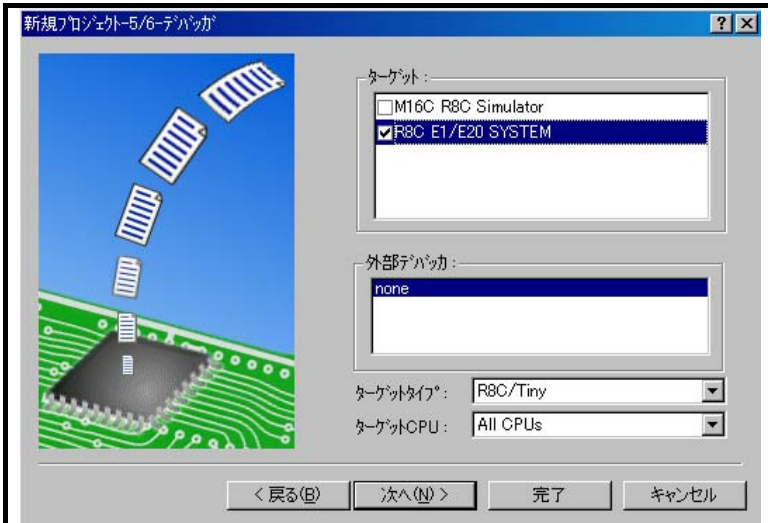


図 4.8 [新規プロジェクト-5/6-デバッガ] ダイアログボックス

- (5) コンフィグレーション名を設定します。  
 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。
- (6) [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。
- (7) 起動後、エミュレータを接続してください。



図 4.9 [新規プロジェクト-6/7-デバッガオプション] ダイアログボックス

#### 4.4 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。

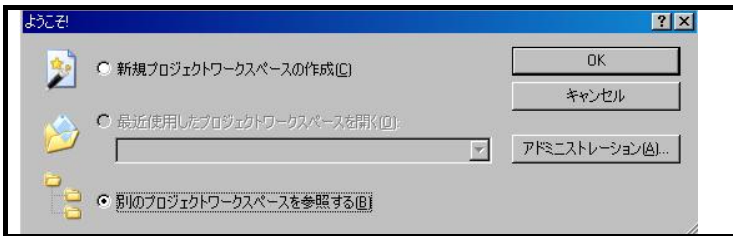


図 4.10 [ようこそ!] ダイアログボックス

- (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。  
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。

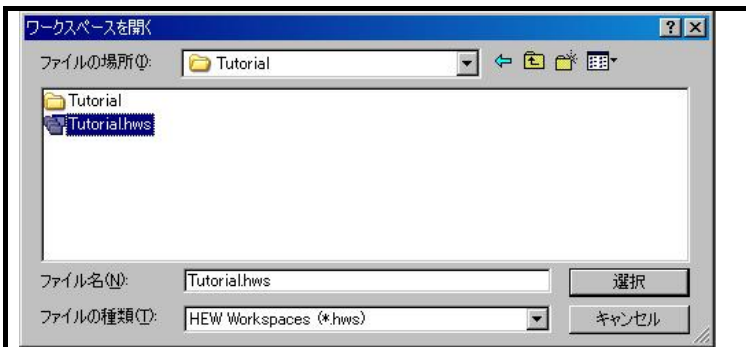


図 4.11 [ワークスペースを開く] ダイアログボックス

- (3) High-performance Embedded Workshop が起動され、指定したワークスペースの保存状態が復元されます。  
指定したワークスペースがエミュレータに接続された状態を保存していた場合には、エミュレータへの接続が自動で行われます。  
指定したワークスペースがエミュレータに接続されていない状態を保存していた場合に、エミュレータの接続を行う場合は、「4.5 エミュレータを接続する」(27ページ)を参照してください。

## 4.5 エミュレータを接続する

### 4.5.1 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。

### 4.5.2 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。

## 4.6 エミュレータを切断する

### 4.6.1 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。

## 4.7 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。

## 4.8 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。

### 4.8.1 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。

[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。

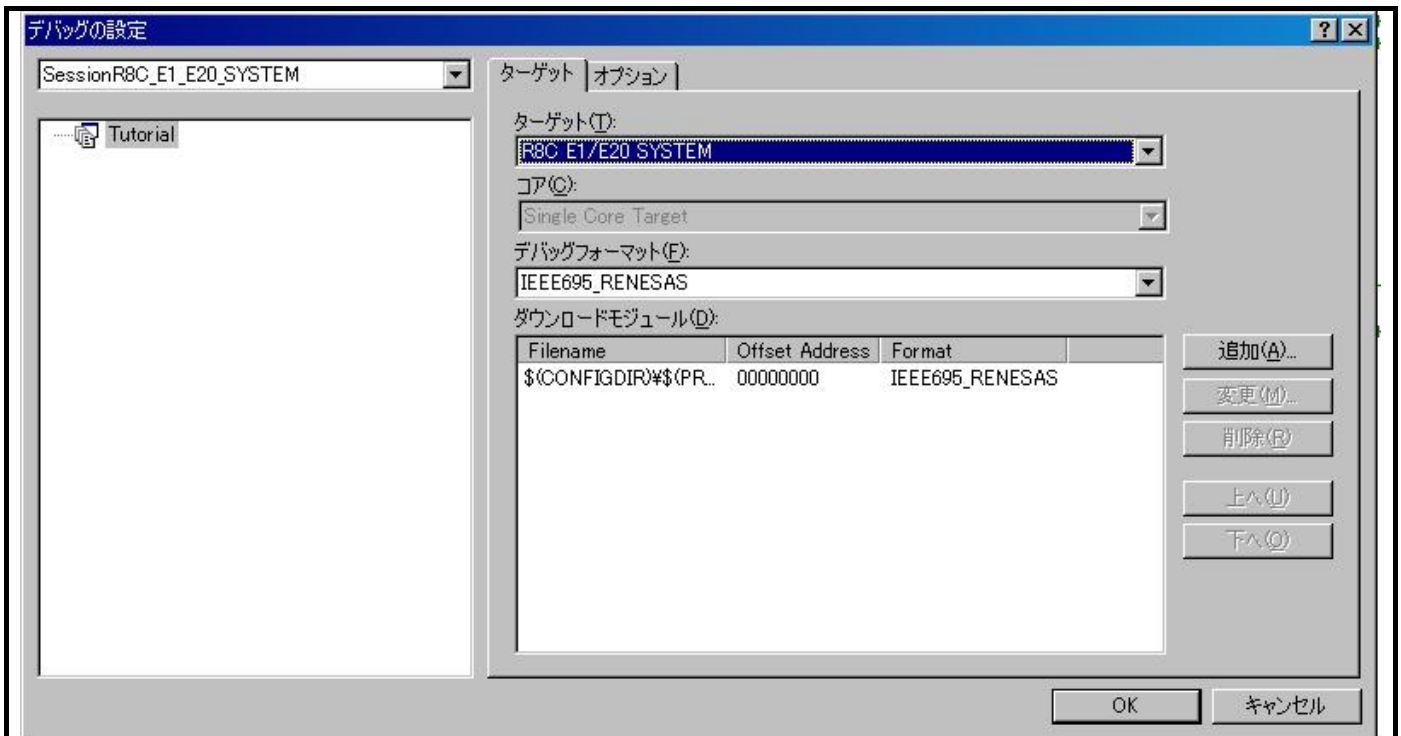


図 4.12 [デバッグの設定] ダイアログボックス

【注1】 この時点ではプログラムのダウンロードはされていません。  
ダウンロード方法については、「5.2 ダウンロード機能」(49ページ)を参照してください。

## 4.8.2 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] ページをクリックしてください。

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。

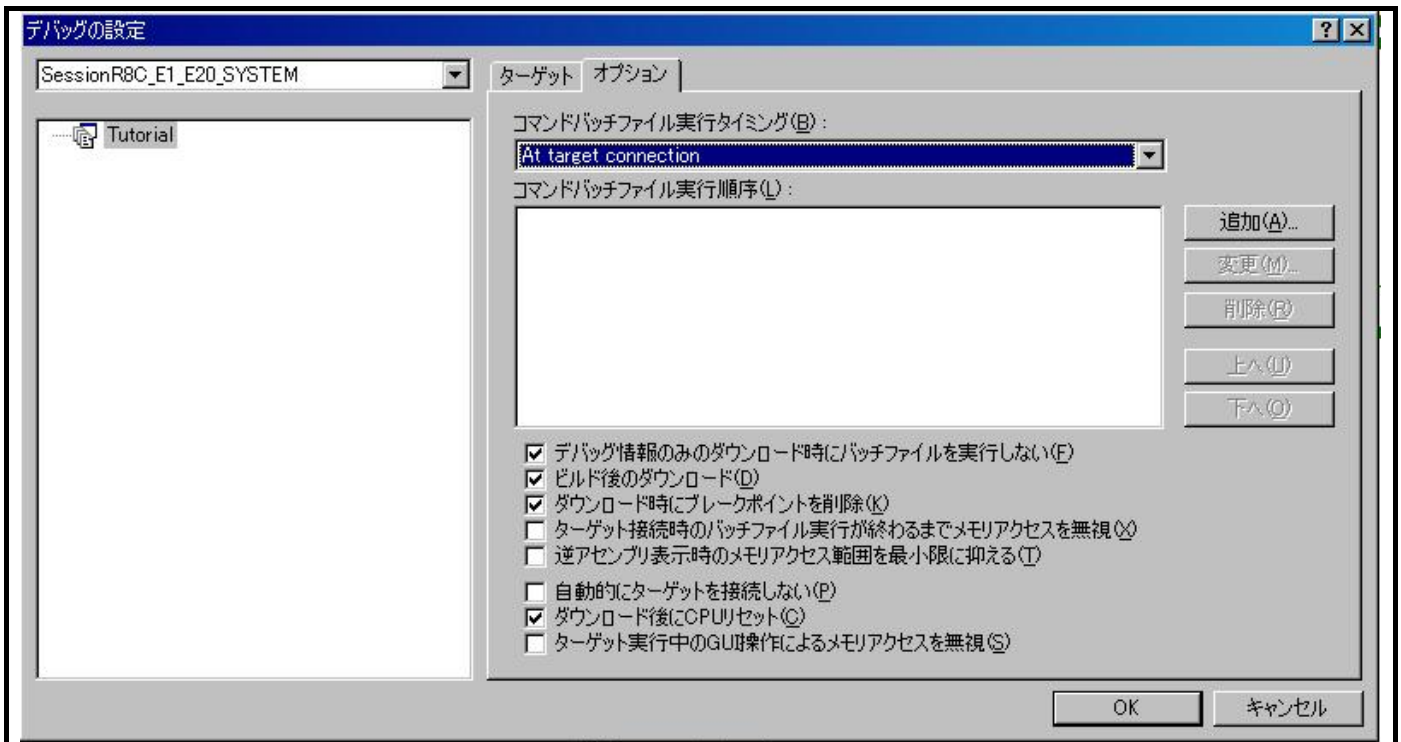


図 4.13 [デバッグの設定] ダイアログボックス

#### 4.9 E1/E20エミュレータ デバッガの起動手順

本項では、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。

(1) 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。

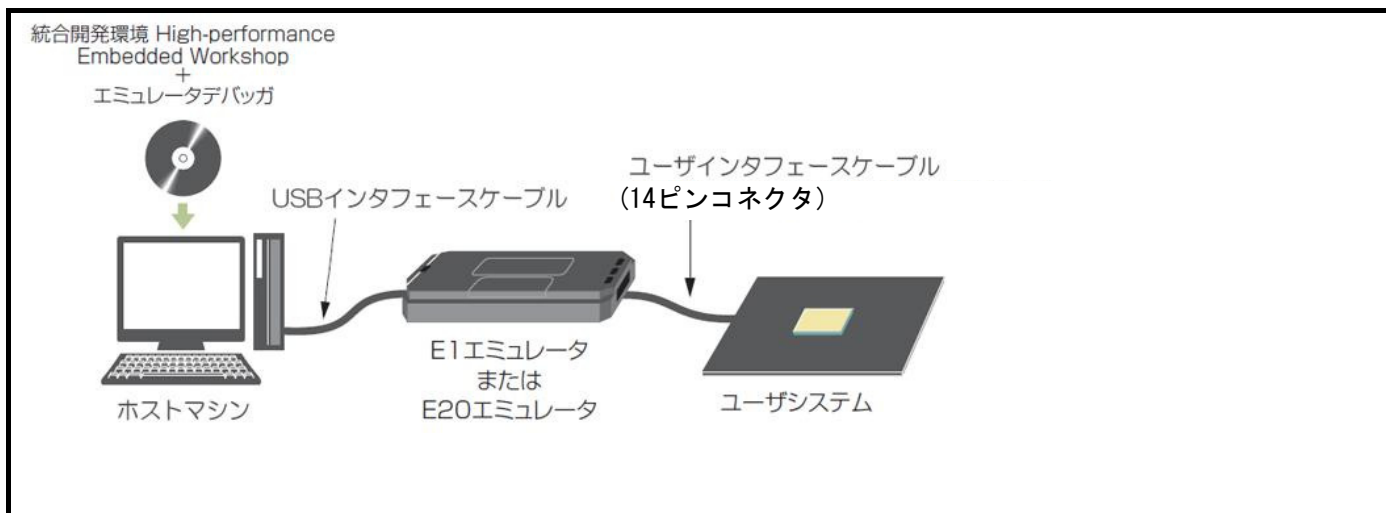


図 4.14 システムチェックの構成

(2) [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。

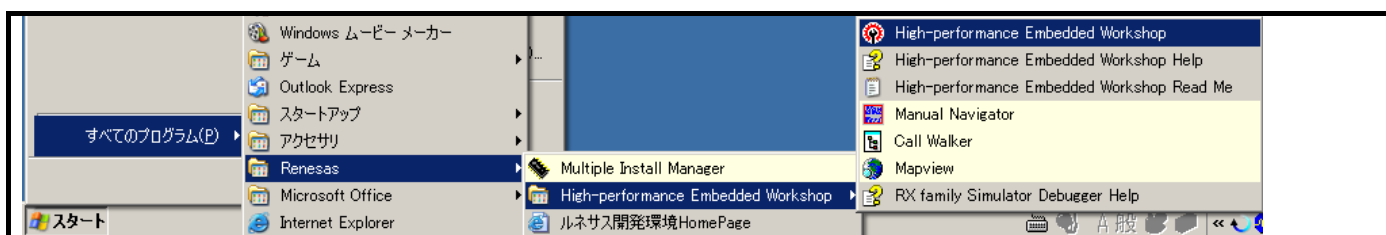


図 4.15 [スタート]メニュー

(3) [ようこそ!]ダイアログボックスが表示されます。

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。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。

<OSがインストールされているドライブ>:¥Workspace¥Tutorial¥E1E20¥R8C¥Tutorial

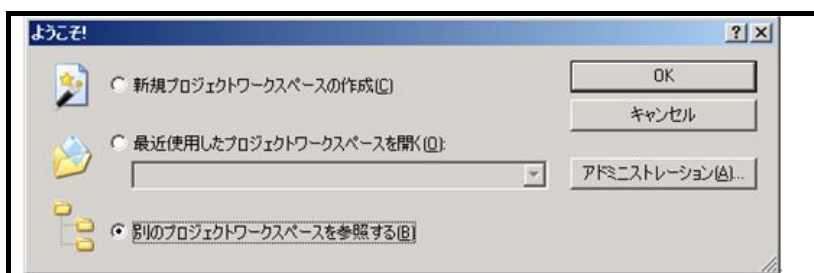


図 4.16 [ようこそ!]ダイアログボックス

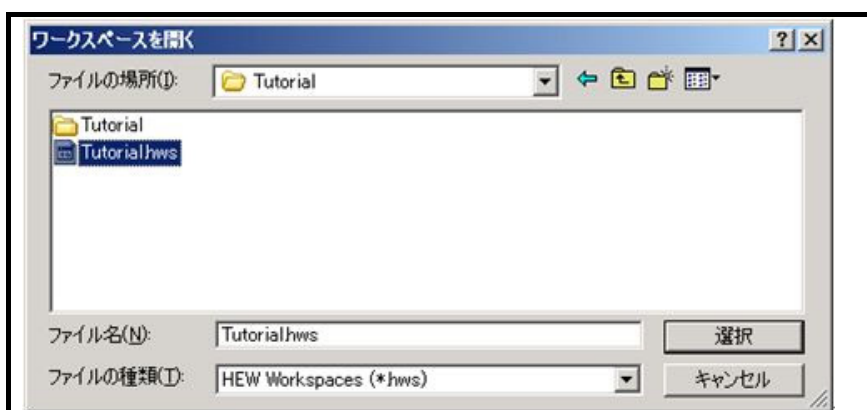


図 4.17 [ワークスペースを開く]ダイアログボックス

#### 4.10 エミュレータデバッグ起動時の設定

エミュレータデバッグ起動時には、以下の3つのダイアログボックスが表示されます。

(1) [起動設定]ダイアログボックス

ターゲットマイコンの選択、通信の確立を行います。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 -> 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、起動後に設定を変更してもすぐに反映されず、次回のエミュレータ接続時の初期値として設定されます。

(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

[起動設定]ダイアログボックスの後に表示されます。

エミュレータおよびデバッグ機能に関する設定を行います。

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。

(3) [接続中]ダイアログボックス

起動処理の進行状態を表示します。



#### 4.11 [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、デバッグ起動時に必要な項目を設定するためのダイアログボックスです。電源供給以外の設定は、次回起動時にも有効です。

新規プロジェクトワークスペース作成後、初めてデバッグを起動する場合は、[起動設定]ダイアログボックスをウィザード形式で表示します。

なおここで設定した内容は、エミュレータ起動後に変更することができません。

[起動設定]ダイアログボックスでの設定内容を変更する場合、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。



図 4.18 [起動設定]ダイアログボックス

ダイアログボックス下部の[次回起動時にこのダイアログボックスを表示しない。]チェックボックスをチェックすると、次回デバッグ起動時にこの[起動設定]ダイアログボックスをオープンしないようにすることができます。

なお、[次回起動時にこのダイアログボックスを表示しない。]チェックボックスが選択されているときは、E1エミュレータからユーザシステムへの電源供給は行いません。

[起動設定]ダイアログボックスは、以下のいずれかの方法で再表示できます。

- デバッグ起動後、メニュー[基本設定]→[エミュレータ]→[デバイスセッティング(D)...]を選択する。
- Ctrlキーを押しながらデバッグを起動する。

#### 【留意事項】

選択したMCUの種類によっては、未サポートのオプション部分はグレー表示となる場合があります。

## 4.11.1 デバイスタブ

デバイスタブでは、デバイスの選択、動作モードの指定、電源供給設定を行います。

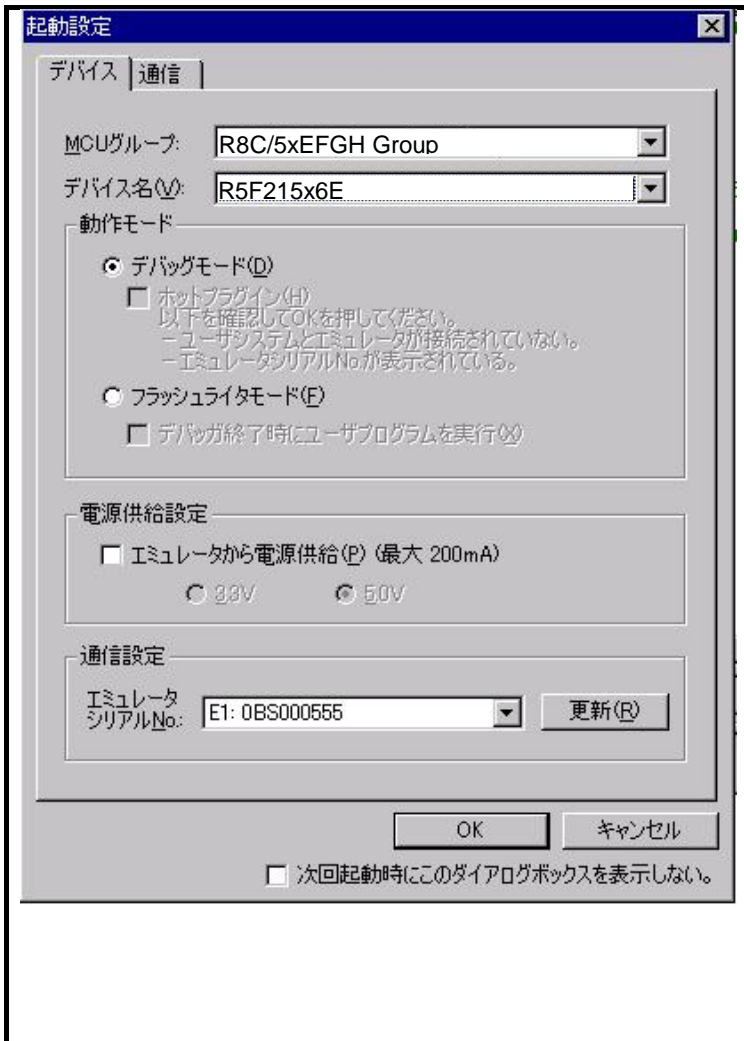


図 4.19 [起動設定] ダイアログボックス (デバイスタブ)

**MCUグループの選択**

ドロップダウンリストボックスから、ご使用になるマイコングループ名を選択してください。

**デバイス名 (V) の選択**

ドロップダウンリストボックスから、ご使用になるマイコン型名を選択してください。対象マイコンの一覧は「7.1 占有するマイコン資源」(109ページ)を参照ください。

**動作モードの指定**

ご使用になるモードを選択してください。詳細は「4.11.1 (1) 動作モードの指定」(35ページ)を参照ください。

**電源供給設定の指定**

ユーザシステムへの電源供給方法を指定してください。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(P) (最大200mA)] チェックボックスをチェックし、"3.3V" もしくは"5.0V" を選択してください。【注1】

**通信設定の指定**

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、ホストマシンにUSBで接続されているエミュレータ 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。【注2】

- 【注1】 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。また、選択したMCUの種類によっては選択可能な電源が1つの場合があります。マイコンの動作電源電圧範囲は使用されるMCUのハードウェアマニュアルをご参照ください。
- 【注2】 エミュレータシリアルNoが表示されない場合、ホストマシンがエミュレータを認識していません。(この場合電源供給設定も選択できません)  
[更新] ボタンのクリックでも表示されない場合は、ホストマシンにエミュレータが接続されていないか、もしくはUSBドライバがインストールされていない場合がありますのでご確認願います。

## (1) 動作モードの指定

表 4.1 動作モードの指定

モード指定	説明
デバッグモード (D) 【注1】	デバッグするためのモードです。 このモードでは、エミュレータデバッグ起動時にE1/E20エミュレータ用プログラム、E1/E20エミュレータが使用するベクタ領域も書き込まれます。 また、OFS、OFS2領域、IDコード領域をE1/E20エミュレータが書き変えて使用します。
フラッシュライターモード (F) 【注2】	ユーザプログラムのみを書き込むためのモードです。 (E1/E20エミュレータ用プログラムは書き込まれません)。 本モードで可能な機能は、プログラムのダウンロード、MCUとの接続/接続解除のみです。 これ以外はエラーとなります。  [デバッグ終了時にユーザプログラムを実行]チェックボックスをチェックした場合、エミュレータをユーザシステムに接続した状態でデバッグ終了と同時にユーザプログラムを実行します。本チェックボックスの設定は[フラッシュライターモード]を選択しているときのみ設定可能です。 チェックボックスがチェックされない場合は、エミュレータデバッグ終了時にMCUをリセットして終了します。

【注1】 このモードでは、ベクタ領域がE1/E20エミュレータ用アドレスが設定されているため、ユーザプログラムをダウンロードした後、E1/E20エミュレータを取り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(CPU単体で動作させることはできません)。  
CPU単体で動作させる場合は、「フラッシュライターモード」を使用してください。

本モードではユーザプログラムダウンロード後、ダウンロード対象ブロックのロックビットはユーザプログラムでの設定状態になります。  
また、本モードでは内部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。

【注2】 このモードではプログラムのデバッグはできません。

本モードではユーザプログラムダウンロード後、ダウンロード対象ブロックのロックビットはアンロック状態になります。  
なお、本モードではIDコードはユーザプログラムで設定した内容がマイコンの内蔵フラッシュメモリに書き込まれます。

## 4.11.2 通信タブ

E1/E20エミュレータとマイコン間の通信ボーレートを選択します。  
通常は、500000bps(デフォルト)を選択してください。【注1】

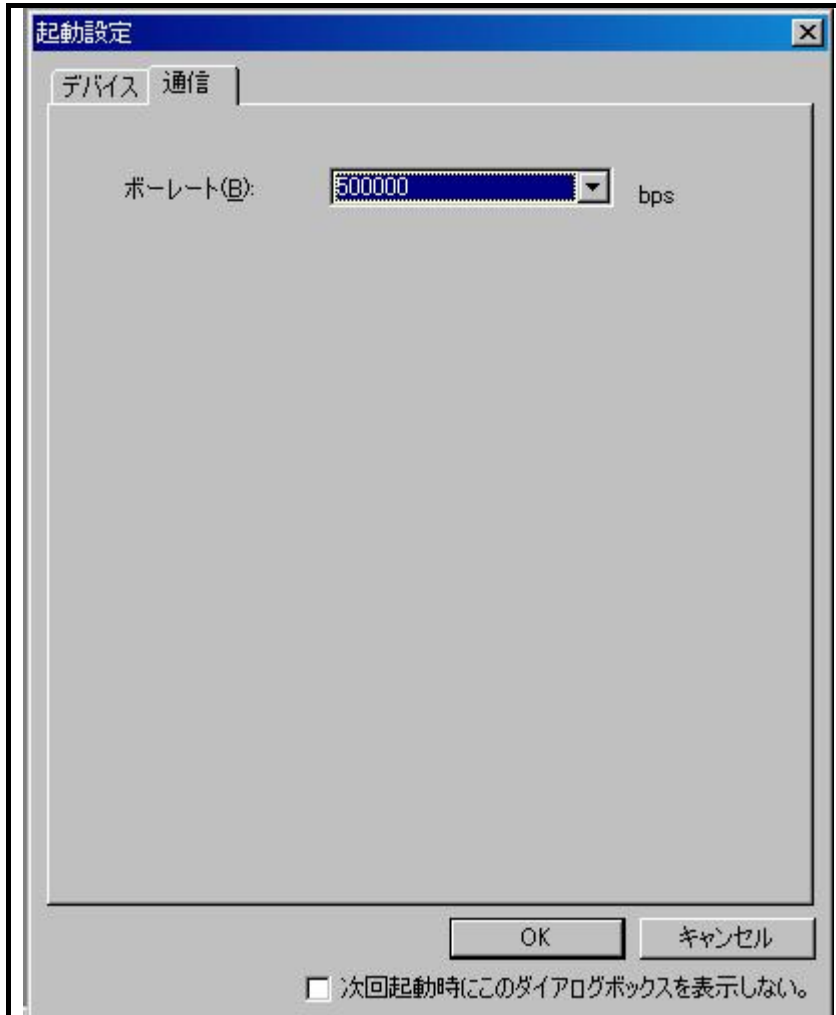


図 4.20 [起動設定]ダイアログボックス(通信タブ)

- 【注1】 ユーザシステム上でのMODE信号の配線長や配線方法によっては、選択した通信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。  
またここで設定した通信情報は、エミュレータデバッガ起動後に変更することができません。  
通信ボーレートの変更は、一度MCUとの接続を解除し、再接続させる必要があります。
- 【注2】 57600bps以下の通信ボーレートは、E1/E20エミュレータとの接続異常があった場合の確認用です。  
これら低い通信ボーレートでは、ターゲットマイコンのフラッシュメモリ書き込みに大変時間が掛かり、エミュレータデバッガの応答がないような状態に見ることがあります。  
また、1024バイト以上のメモリ参照/メモリフィル等では、通信時間がかかるためタイムアウトエラーが発生したりする場合があります。

#### 4.12 接続中ダイアログの表示

[起動設定]ダイアログボックスでOKボタンをクリックすることで、[接続中...]ダイアログボックスが表示されMCUと初期通信を開始します。

次に、以下に示す5つのケースを除き、4.13 項の[コンフィグレーションプロパティ]ダイアログボックスが表示されます。

表 4.2 接続ダイアログの表示

No	内容	表示のウィンドウ
1	RXファミリ等他のエミュレータデバッガでE1/E20エミュレータを使用していた場合	4.12.1 項参照
2	E1/E20エミュレータ内のエミュレータファームウェアバージョンが古い場合	4.12.2 項参照
3	[起動設定]ダイアログボックスの電源供給設定の内容が正しくない場合【注1】	4.12.3 項参照
4	IDコード領域に“FFFFFFFFFFFFh”以外のIDコードが書き込まれている場合	4.12.4 項参照
5	MCUとの通信が出来ない場合	4.12.5 項参照

起動処理の進行状態は、[接続中...]ダイアログボックスで確認することができます。

[接続中...]ダイアログボックスは、起動処理の開始から終了まで表示し続けます。

[デバイスセッティング]、[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ている間は、本ダイアログボックスを操作することはできません。

ここで表示される内容は、不具合レポートに記録されます。不具合レポートの内容は、[ヘルプ -> テクニカルサポート -> 不具合レポートの作成...]から確認することができます。



図 4.21 [接続中...]ダイアログボックス

【注1】 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。

## 4.12.1 RXファミリ等他のエミュレータデバッガでE1/E20エミュレータを使用していた場合

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。

## ① エミュレータファームウェアの書き換え可否確認

[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。

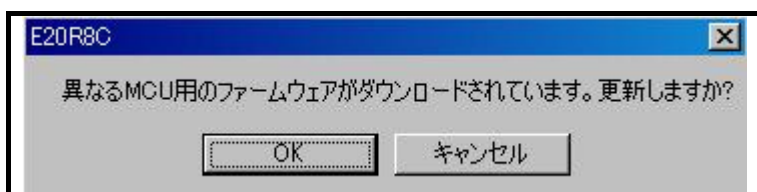


図 4.22 [ファームウェアの書き換え]ダイアログボックス

## ② エミュレータファームウェアの書き換え

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]プログレス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。【注1】



図 4.23 [ダウンロード]プログレスバー

---

【注1】 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。

---

## 4.12.2 E1/E20エミュレータ内のエミュレータファームウェアバージョンが古い場合

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。

## ③ エミュレータファームウェアの書き換え可否確認

[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。

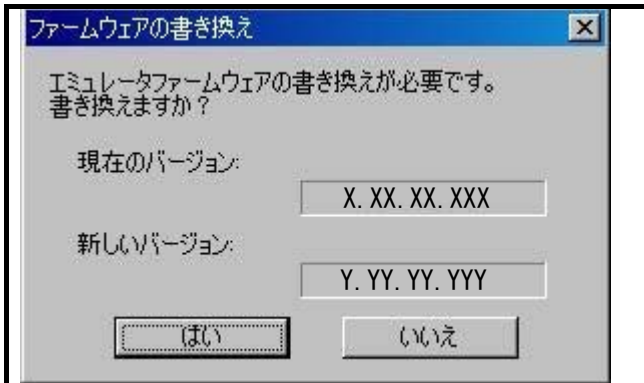


図 4.24 [ファームウェアの書き換え]ダイアログボックス

## ④ エミュレータファームウェアの書き換え

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]プログレス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。【注1】



図 4.25 [ダウンロード]プログレスバー

---

【注1】 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。

---

## 4.12.3 [起動設定]ダイアログボックスの電源供給設定の内容が正しくない場合

[起動設定]ダイアログボックスの電源供給設定とユーザシステムの電源供給状態が一致しない場合、下記のウィンドウが表示されます。

なおこれらのウィンドウは、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。

- 「エミュレータから電源供給」にチェック有りで、ユーザシステムに既に電源供給されている場合
- 「エミュレータから電源供給」にチェック無しで、ユーザシステムに既に電源供給されている場合

- (1) 「エミュレータから電源供給」にチェック有りで、ユーザシステムに既に電源供給されている場合  
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[エミュレータから電源供給]の項目にチェックを入れた場合は、図 4.26のダイアログボックスが表示されます。

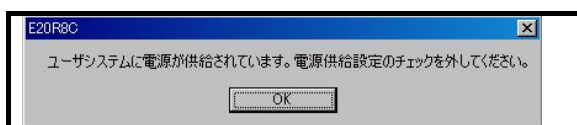


図 4.26 ユーザシステム側の電源供給確認ダイアログボックス

[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 4.27 [電源]ダイアログボックス

- (2) 「エミュレータから電源供給」にチェック無しで、ユーザシステムに電源供給されていない場合  
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[エミュレータから電源供給]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。

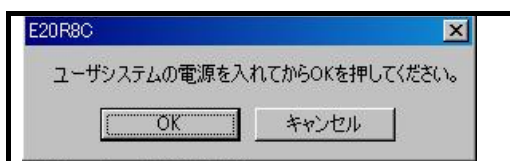


図 4.28 電源供給確認ダイアログボックス

[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 4.29 [電源]ダイアログボックス



## 4.12.4 IDコード領域に“FFFFFFFFFFFFh”以外のIDコードが書き込まれている場合

(1) ターゲットマイコンのIDコード領域に“FFFFFFFFFFFFh”以外のIDコードが書き込まれている場合、IDコード判定のための下記のウィンドウが表示されます。

(2) 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。

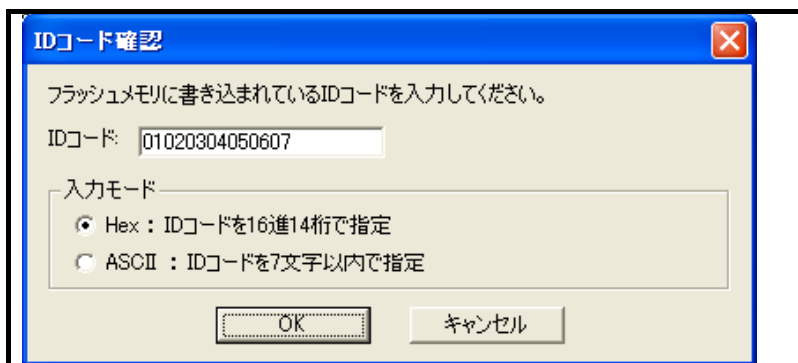


図 4.30 [IDコード確認] ダイアログボックス

【注1】 IDコード領域の詳細については下記を参照ください。

- (1) マイコンのハードウェアマニュアル
- (2) 「7.3.6 フラッシュメモリのIDコード」(118ページ)

## 4.12.5 MCUとの通信が出来ない場合

[起動設定]ダイアログボックスでOKボタンをクリックすることで、MCUと初期通信を開始します。  
この時、MCUと通信できない場合のエラー内容と原因の例を表 4.3に示します。

表 4.3 MCUと初期通信時のエラー内容と原因

エラー内容	原因	対策
ユーザシステム上のMODE端子が“L”固定のためコマンドが実行できません。 「MODE端子の状態を確認してください。」	MODE端子が“L”固定のため、E1/E20エミュレータがMCUと通信できません。	MODE端子はプルアップしてください (図 3.8参照)。
ユーザシステム上のリセット端子が“L”状態のためコマンド処理を実行できません。 「リセット端子の状態を確認してください。」	RESET端子が“L”固定のため、E1/E20エミュレータがMCUと通信できません。	RESET端子の回路をご確認願います (図 3.9参照)。
ターゲットMCUと選択したデバイスが一致しません。デバッグ起動時に指定したデバイス名を再確認ください。	[起動設定]ダイアログボックスでのMCU 型名設定が、実際にE1/E20エミュレータ と接続されているMCUと違います。	[起動設定]ダイアログボックスでのMCU 型名を実際にE1/E20エミュレータに接続 しているMCUに指定し直してください。
MCUへの端子処理等の影響でMCUが起動 出来ませんでした	E1/E20エミュレータとMCU間の結線を間違えている。 あるいは結線が推奨回路どおりになっていない。 推奨回路の結線と異なる場合、正しく通信ができない場合があります。	E1/E20エミュレータとMCU間の接続を再 確認願います (図 3.7参照)。
	マイコン動作電圧が規定外のものとなっている。	MCUのハードウェアマニュアルを参照 ください。
	マイコンがICソケットを使用して実装されているが、接触不良を起こしている。	ユーザシステム上のICソケットが対応 MCUのパッケージと一致しているか、また ねじ止め等で接触不良になっていないか 再確認ください。
	E1/E20エミュレータの電源端子が全て接続されていない。	「3.4 (3) その他端子」(19ページ)参照
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。	ユーザシステムの電源電圧を ご確認ください。

### 4.13 [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。[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。

設定した内容は、次回起動時にも有効です。このタブでOKボタンをクリックすることで設定が完了し、エミュレータデバッグが起動します。

なお、[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、デバッグ起動後、メニュー[基本設定]→[エミュレータ]→[システム(S)...]を選択することで再表示できます。

#### 【留意事項】

選択したMCUの種類によっては、未サポート機能のためグレー表示もしくはオプション自体が表示されない場合があります。

#### 4.13.1 システムタブ

#### デバッグモニタ配置領域の選択

どの領域にデバッグモニタを配置するか設定してください。設定はユーザプログラムで使用しない領域を設定してください。

- デバッグモニタ配置領域  
「データフラッシュ領域」と「ユーザフラッシュ領域」のいずれの領域にデバッグモニタを配置するか設定します。ユーザシステムで使用しない2Kバイトを指定してください。  
【注1】 【注2】

#### デバッグ機能の選択

デバッグ機能を選択します。

- 内蔵フラッシュメモリの書き換えを伴うプログラムのデバッグ(F)  
MCU内蔵フラッシュメモリ（プログラムROM領域）の書き換えを伴うプログラムをデバッグ（CPU書き換えモードのデバッグ）する際に指定します。内蔵フラッシュメモリ（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。  
【注3】 【注4】
- ユーザプログラム停止中は全タイマのカウントを停止する(T)  
ユーザプログラム停止中にウォッチドックタイマ以外の全タイマのカウントを停止するかどうかを選択します。チェックボックスにチェックした場合、ユーザプログラムが停止した時点で全タイマカウントが停止します。

図 4.31 [コンフィグレーションプロパティ]ダイアログボックス(システムタブ)

【注1】 選択したMCUの種類によっては、「データフラッシュ領域の選択ができない場合」および「設定が不要のためデバッグモニタ配置領域の内容がグレー表示となる場合」があります。

【注2】 デバッグモニタ配置領域に「データフラッシュ領域」を割り当てた場合は、250000bps以下の通信ボーレートで使用してください。但し、この場合バスサイクルが増えるため遅くなります。

【注3】 ユーザプログラム停止中のメモリアクセスは常にE1/E20エミュレータ内のキャッシュに対して実施します。実際のフラッシュメモリへアクセスは、ユーザプログラム再開前およびユーザプログラム停止直後に実施します。このオプション選択時には、ユーザプログラム停止直後にキャッシュの内容を更新しますが、オプション未選択時には、キャッシュの内容は更新されません。この場合、表示とMCU内部ROMの内容が一致しませんので”CPU書き換えモード”でユーザプログラム実行中に内蔵フラッシュメモリを書き変える場合、必ずこのオプションをチェックください。

【注4】 「内蔵フラッシュメモリの書き換えを伴うプログラムのデバッグ(F)」のオプションチェック時、S/Wブレークは設定できません。また、[メモリ]ウィンドウ等では内部ROMを書き変えないでください。

## 4.13.2 MCUタブ

**動作モード(M)の選択**

MCUの動作モードを選択します。  
本MCUでは設定の必要はありません。

図 4.32 [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タブ)

## 4.13.3 内蔵フラッシュメモリの上書きタブ

フラッシュメモリのブロックごとの上書き設定を行います。エミュレータデバッガ起動後も本[コンフィグレーションプロパティ]を開き、設定を変更することができます。

選択しているデバイスに合わせた、フラッシュメモリのブロック一覧が自動表示されます。

なお本タブの内容は”フラッシュライターモード”の場合も適用されます。

## (1) チェックボックスにチェック済のブロック

- ① 選択されたブロックはいったんデータを読み込みます。
- ② 全ブロックをイレースします。
- ③ ダウンロードコマンド等でユーザプログラムダウンロード時に、①のデータをマージ(デバイス上のデータはそのまま)します。

## (2) チェックボックスに未チェックのブロック

- ① 全ブロックをイレースします。
- ② ダウンロードコマンド等でユーザプログラムをダウンロードします。  
ダウンロードするプログラムにデータの無い領域は、フラッシュメモリの初期値となります。



図 4.33 [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タブ)



## 5. デバッグ機能

### 5.1 デバッグ機能一覧

動作モードが「フラッシュライターモード」選択時のサポートする機能の一覧を表 5.1、動作モードが「デバッグモード」選択時の機能一覧を表 5.2に示す。

表 5.1 R8C E1/E20エミュレータデバッガ機能一覧(フラッシュライターモード選択時)

項目	対応	内容
プログラムダウンロード	○	対象:内部ROM(ブロック単位での設定可能)
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
コマンドライン機能	○	コマンド入力をサポート 入力順に羅列したコマンドのファイルでバッチ処理可能 プログラムのダウンロードコマンドのみ使用可能
その他機能	×	

○:サポート、△:代替機能サポート、×:未サポート

表 5.2 R8C E1/E20エミュレータデバッグ機能一覧(デバッグモード選択時)

項目	対応	内容
プログラムダウンロード	○	対象:内部RAMおよび内部ROM(ブロック単位での設定可能)
リセット機能	○	ユーザプログラム停止時エミュレータデバッグからのリセット
メモリアクセス機能		
設定/参照/フィル/コピー/比較/検索	○	対象: SFRレジスタ、内部RAM、内部ROM
リアルタイムRAMモニタ	△	ユーザプログラム実行中のメモリ内容の自動更新機能あり
1行アセンブル	○	
逆アセンブル表示	○	
CPUレジスタアクセス	○	CPUレジスタの参照/設定
SFRレジスタアクセス	○	I/Oウィンドウからの参照/設定
ユーザプログラム実行機能		
実行機能	○	実行/フリー実行/カーソル位置まで実行/リセット後実行
ステップ機能	○	(1) シングルステップ(1ステップ:1命令) (2) ソースレベルステップ機能(1ステップ: ソース1行) (3) ステップオーバ機能(サブルーチン内はブレイクしない) (4) ソースレベルステップオーバ機能 (サブルーチン内はブレイクしない) (5) ステップアウト機能 (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)
ブレイク機能	○	S/Wブレイクポイント(最大256点)
	○	強制ブレイク
	○	実行前PCブレイク(アドレス一致ブレイク)8点
	○	データアクセスブレイク2点 Event A: アドレス/データのマスク付き比較、 アクセス条件(R, W, R/W) 設定可 Event B: アドレスマスク付き比較、アクセス条件(R, W, R/W) 設定可
	○	トレースフルブレイク
トレース機能	○	内部トレース
	×	外部トレース
	○	4分岐(分岐元先PC)、または指定データサイクル最大8サイクル
	×	検索/フィルタ
パフォーマンス測定機能	×	マイコン内蔵カウンタで指定区間の経過サイクル等測定機能
デバッグコンソール機能	×	ユーザプログラムからの標準入出力をサポート
スタックトレース機能	○	PC(プログラムカウンタ)がある関数がどの関数からコールされているかを表示
Start/Stop ファンクション	○	ユーザプログラムの実行開始直前/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
コマンドライン機能	○	コマンド入力をサポート 入力順に羅列したコマンドのファイルでバッチ処理可能
実行時間計測機能	○	ユーザプログラム実行中の時間を表示します。
ホットプラグイン	×	
カバレッジ機能	×	
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

○:サポート、△:代替機能サポート、×:未サポート



## 5.2 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。

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。

- 【注1】 プログラムをダウンロードする場合、ロードモジュールとして High-performance Embedded Workspace に登録する必要があります。
- 【注2】 アクセスサイズ指定「1バイト」以外の場合、端数分データが正しくダウンロードできません。  
以下のダウンロード機能での[アクセスサイズ]指定は、ダウンロード時のアクセスサイズは1(バイト)固定で使ってください。  
①[ダウンロードモジュールの追加]メニューで開くダイアログ  
②メモリウィンドウの[読み込み]メニューから開くダイアログ
- 【注3】 ベクタ領域およびIDコード領域およびOFS、OFS2領域はエミュレータが値を書き換えているため、チェックサムが一致しませんのでご注意ください(デバッグモード選択時)。  
また、リセットベクタアドレスについてはメモリウィンドウ等ではダウンロードした値が表示されますが、これはエミュレータ上のキャッシュ値であり実際のフラッシュメモリにはエミュレータでの指定値が設定されていますのでご注意ください。

### 5.3 ソースファイルを開く

#### 5.3.1 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。

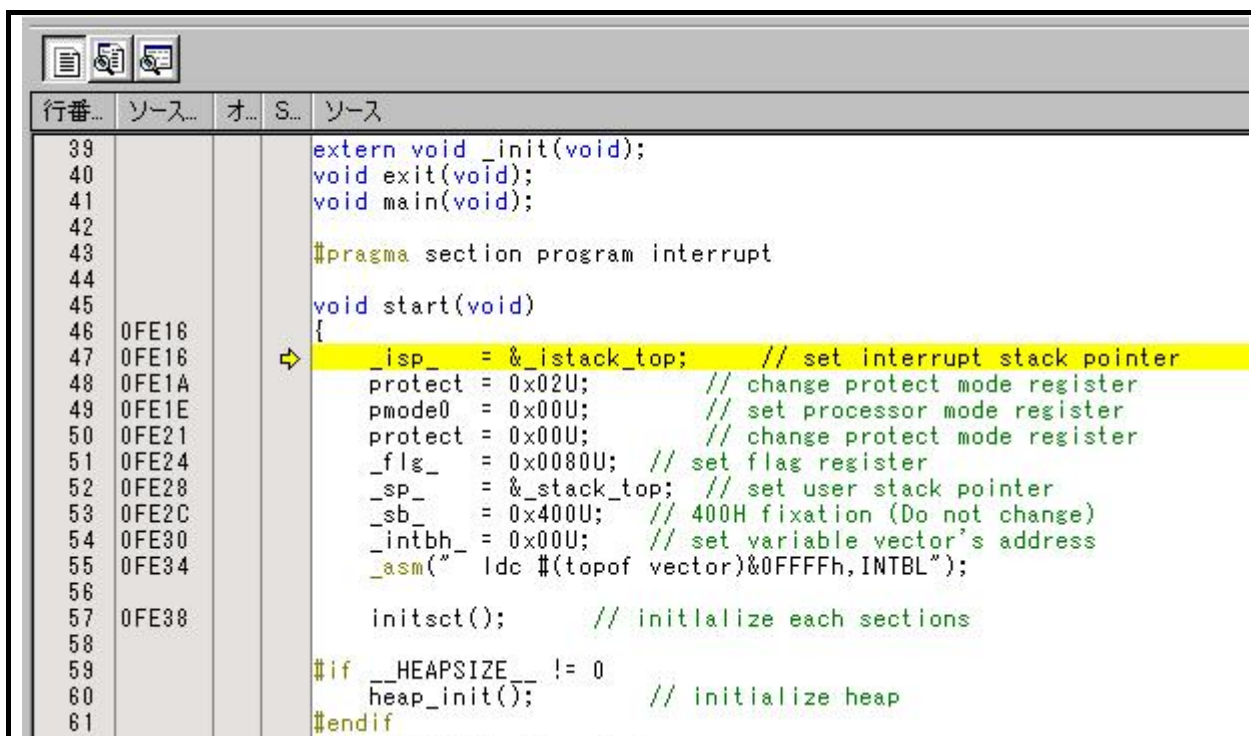


図 5.1 [エディタ]ウィンドウ

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	(オンチップブレークポイントカラム)	オンチップブレークポイント情報
3列目	(S/Wブレークポイントカラム)	PC、ブックマーク、ブレークポイント情報

#### ソースアドレスカラム

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

#### オンチップブレークポイントカラム

オンチップブレークポイントカラムには下記を表示します。

- 実行前PCブレー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。

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

#### S/Wブレークポイントカラム

S/Wブレークポイントカラムには下記を表示します。

- ブックマークを設定している
- S/W ブレークを設定している
- PC位置

### 5.3.2 ソースファイルでカラムをオフにする

- (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。

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。

- (4) [OK] ボタンをクリックして、新しいカラム設定を有効にしてください。




図 5.2 [エディタ全体のカラム状態]ダイアログボックス

### 5.3.3 1つのソースファイルでカラムをオフにする


- (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。

## 5.3.4 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。

ディスアセンブルニーモニック(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。

行番...	オ..	S..	逆アセ..	オブジェクトコー...	ラベル	混合
47			0FE16	EB401605	_start	<code>_isp_ = &amp;_istack_top; // set interrupt stack pointer</code>
48	→		0FE1A	C7020A00		<code>LDC #0516H,ISP</code>
49			0FE1E	B70400		<code>protect = 0x02U; // change protect mode register</code>
50			0FE21	B70A00		<code>MOV.B #02H,000AH</code>
51			0FE24	EB308000		<code>pmode0 = 0x00U; // set processor mode register</code>
52			0FE28	EB509604		<code>MOV.B #0,0004H</code>
53			0FE2C	EB600004		<code>protect = 0x00U; // change protect mode register</code>
54			0FE30	EB200000		<code>MOV.B #0,000AH</code>
55			0FE34	EB10DCFE		<code>_flg_ = 0x0080U; // set flag register</code>
						<code>LDC #0080H,FLG</code>
						<code>_sp_ = &amp;_stack_top; // set user stack pointer</code>
						<code>LDC #0496H,SP</code>
						<code>_sb_ = 0x400U; // 400H fixation (Do not change)</code>
						<code>LDC #0400H,SB</code>
						<code>_intbh_ = 0x00U; // set variable vector's address</code>
						<code>LDC #0000H,INTBH</code>
						<code>_asm(" ldc #((topof vector)&amp;0FFFFh,INTBL");</code>
						<code>LDC #FEDCH,INTBL</code>

図 5.3 [逆アセンブリ]ウィンドウ

## 5.3.5 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。

- ① [アセンブル]ダイアログボックスが開きます。  
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。
- ② 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。
- ③ "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。
- ④ [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。

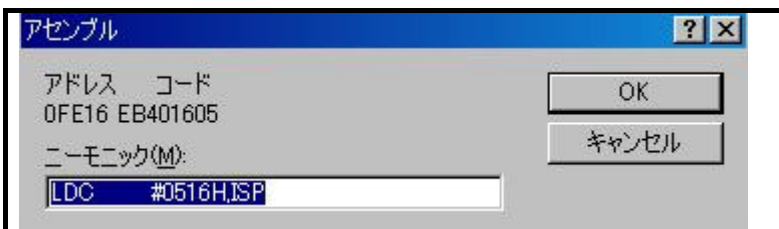


図 5.4 [アセンブル]ダイアログボックス


[注1] アセンブリ言語表示は、実際のメモリ上の機械語コードからディスアセンブルします。メモリの内容を修正すると、ダイアログボックス(および[逆アセンブリ]ウィンドウ)には新しいアセンブリ言語コードを表示します。しかし、[エディタ]ウィンドウの表示内容は変更しません。これはソースファイルにアセンブラを含む場合も同じです。

## 5.4 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。

### 5.4.1 メモリリード/ライト機能【注1】

メモリの内容は[メモリ]ウィンドウで確認することができます。

[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表示させる開始アドレスを入力すると[メモリ]ウィンドウが表示されます。

#### (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。

詳細は、「5.2 ダウンロード機能」(49ページ)を参照ください。

#### (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) メモリデータの自動更新機能【注2】

メモリウィンドウ等でMCUのメモリ内容とアクセス状況を定期的に参照する機能です。

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。

MCU内蔵のデバッグ機能により、ユーザプログラムの停止時間を最小に抑えてメモリ内容をアクセスする機能です。前回と値の変化があった箇所は赤色で表示します。更新間隔は変更することが可能です。

なお表示の更新間隔は、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

### 5.4.2 そのほかのメモリ操作機能【注3】 【注4】

その他、以下の機能があります。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。

- メモリフィル機能
- メモリコピー機能
- メモリ比較機能
- メモリ検索機能
- ラベル名、変数名とその内容を表示する機能

【注1】 [メモリ]ウィンドウは、表示しているアドレス範囲のみ参照します。内部ROM領域以外は、エミュレータ内にキャッシュを持っていないため常にターゲットマイコンにアクセスします。

また、[メモリ]ウィンドウから値をライトした場合は、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アドレス範囲を全てアクセスします。

[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。

【注2】 ユーザプログラム実行中のS/Wリセットやハードウェアリセットなどによるリセットが発生した場合、その時点のメモリアクセスは正しく設定/変更ができません。

【注3】 コピー/比較/検索機能で、データサイズ4バイトおよび8バイトを指定しないでください。

データサイズ4バイトおよび8バイト指定時は正しく動作しません。

また、コピー/比較/検索機能の最大範囲は16MBです。

コピー/比較/検索機能は、内部RAM領域および内部ROM領域のみ有効です。

【注4】 デバッグモニタ領域に対してはこれらの機能が正常に動作しません。

## 5.4.3 SFR領域アクセス時の注意事項

SFR領域の内容は[メモリ]ウィンドウまたは[I/O]ウィンドウから、参照および設定することができますが、以下のことを注意してご使用ください。

## (1) 特殊レジスタへアクセスする場合1

ユーザプログラム停止中、以下の特殊レジスタは、正常アクセスできない場合があります。【注1】

- アクセス禁止アドレス
- アクセス順序が、上位バイト→下位バイトに指定されているレジスタ
- 特定命令でのみアクセス可能なレジスタ
- バス幅指定が[メモリ]ウィンドウの設定バス幅と一致していないレジスタ
- レジスタアクセスに条件(foco-fがCPUより早いクロックでアクセスが必要等)があるレジスタ等

## (2) 特殊レジスタへアクセスする場合2

ユーザプログラム実行中、以下の特殊レジスタは、正常アクセスできない場合があります。【注1】

また、表 5.3に示すレジスタは、アクセスによって動作に違いがあるのでご注意ください。

- アクセス禁止アドレス
- アクセス順序が、上位バイト→下位バイトに指定されているレジスタ
- 特定命令でのみアクセス可能なレジスタ等

表 5.3 特殊レジスタアクセス時の動作

分類	エミュレータでの動作	備考
設定手順が必要なもの ① “0” 設定後→② “1” 設定 or ① “1” 設定後→② “0” 設定	MCUは①と②の間の、エミュレータからのアクセスも手順の1つと取ります。	① “0” 書き込み ② エミュレータからアクセス ③ “1” 書き込み (②の動作があるため設定不可)
クリア手順が必要なもの “1” 参照後→ “0” 設定	エミュレータからの “0” 設定によって強制クリアされます。	
通信系	送信レジスタ (設定後送信開始)	違いはありません。
	受信レジスタ (参照後受信バッファクリア)	エミュレータからの参照によって受信バッファはクリアされません。

【注1】 SFR領域へのアクセスは、ターゲットマイコンのハードウェアマニュアルに従ってください。

## (2) [I0]ウィンドウでアクセス可能なSFRレジスタ【注1】【注2】

[I0]ウィンドウでデフォルト表示されるI0ファイルは、[起動設定]ダイアログボックスで選択したMCUグループに依存して下記(最大ピン数のI0ファイル)の通りになります。

表 5.4 [I0]ウィンドウでのデフォルト指定されるI0ファイル

種類	[起動設定]ダイアログボックス表示	I0ファイルのデフォルト
R8C/5xE	R8C/5xEFGH Group	R8C/56E(未添付)
R8C/5xF	R8C/5xEFGH Group	R8C/56F(未添付)
R8C/5xG	R8C/5xEFGH Group	R8C/56G(未添付)
R8C/5xH	R8C/5xEFGH Group	R8C/56H(未添付)
R8C/3xT-A	R8C/3xT-A Group	R8C/38T-A (未添付)

【注1】 I0ファイルはデバッグインストール時にインストールされた、ターゲットマイコン用のファイルに変更することが可能です。  
また、I0ファイル(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。

<High-performance Embedded Workshopフォルダ>

¥Tools¥Renesas¥DebugComp¥Platform¥E1E20¥E20R8C¥I0Files

【注2】 R8C E1/E20エミュレータデバッグに添付されているI0ファイルは、デバッグ開発時点で公開中のSFRヘッダファイルを元に作成されています。

(本対応MCUのSFRヘッダファイルはデバッグ開発時点で開発中のため、本製品には未添付です)

このため対象となるSFRヘッダファイルが誤記等で更新された場合、I0ファイルと一致しない場合があります。I0ファイルは上記の通り編集することができます。

## 5.5 ブレーク機能概要

R8C E1/E20エミュレータデバッガは、強制ブレーク機能とS/Wブレーク機能、そしてオンチップブレーク機能の3種類のブレーク機能が使用可能です。ブレーク機能は単体もしくは同時に設定することが可能です。

本マニュアルでの対応マイコンでサポートするブレーク機能一覧を表 5.5に示す。

### 5.5.1 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。

### 5.5.2 S/Wブレーク(ソフトウェアブレーク)

指定したアドレスをデバッグ専用の命令 (BRK命令) に置き換えてブレークする機能です。【注1】

S/Wブレークポイント設定時、指定アドレスのオペコードをデバッグ専用命令に書き換えるためにMCU内蔵資源 (フラッシュメモリおよびRAM) 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同様にメモリへのライトが発生します)。

S/Wブレークポイント機能の詳細は「5.6 S/Wブレークポイントの使用」(57ページ)を参照してください。

### 5.5.3 オンチップブレーク機能

本マニュアルのサポートMCUでは、下記の実行前PGブレーク機能とデータアクセスブレーク機能、トレースフルブレーク機能の3種類を使用することができます。

オンチップブレーク機能の詳細は「5.7 オンチップブレーク機能の概要」(60ページ)を参照してください。

- 実行前PGブレーク機能(アドレス一致ブレーク)  
指定したアドレスの命令を実行する直前でブレークする機能です。MCUのアドレス一致割り込みを使用して実現しています。本機能でのイベントを「PGイベント」と呼びます。
- データアクセスブレーク  
指定イベントの成立時にブレークする機能です。2点のデータアクセスイベントを組み合わせで指定できます。本機能でのイベントを「DAイベント」と呼びます。
- トレースフルブレーク  
トレースバッファが一杯になったときにブレークする機能です。

表 5.5 ブレーク機能一覧

ブレークタイプ		設定可能 (Max)	登録可能 (Max)	ブレーク条件	フラッシュ 書き換え	ユーザプログラム 実行中の設定	
強制ブレーク				なし	なし	指定可能	
S/Wブレーク【注2】		256点		指定アドレス	あり	指定不可【注3】	
オンチップ ブレーク	実行前PGブレーク【注2】 (アドレス一致ブレーク)	8点	32点	指定アドレス	なし	指定可能	
	データアクセス ブレーク	EventA	1点	8点	指定アドレス & 指定データ	なし	指定不可
		EventB	1点	8点	指定アドレス	なし	指定不可
トレースフルブレーク				トレース バッファ満杯	なし	指定不可	

【注1】 S/Wブレークポイントに置き換えたアドレスは、[メモリ]ウィンドウで表示させても“BRK”命令(00h)は表示されません。

【注2】 S/Wブレークポイントおよび実行前PGブレークポイントで停止後、再度そのアドレスからユーザプログラムを実行した場合、1度そのアドレスをシングルステップ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。

【注3】 内部RAMへのS/Wブレークポイントの設定/削除は可能です。



## 5.6 S/Wブレークポイントの使用

作成したプログラムを実行して S/Wブレークポイントを設定したアドレスに達すると、[アウトプット]ウィンドウの[Debug]シートに“Software Break”というメッセージを表示し、実行を停止します。

その時、[エディタ]または[逆アセンブリ]ウィンドウを更新し、停止位置をS/Wブレークポイントカラムに矢印[→]で表示します。

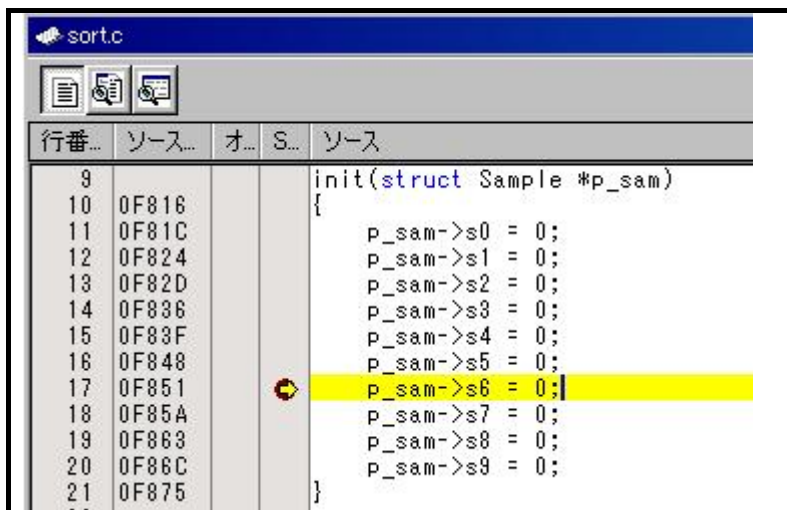


図 5.5 S/Wブレークでの停止

- 【注1】 ブレーク発生時には、S/Wブレークポイントを設定した行または命令を実行する直前で停止します。そのS/Wブレークポイントで停止した後にGo またはStep を選択した場合、矢印で表示した行から実行します。
- 【注2】 複数のS/Wブレークポイントを指定した場合、いずれかのブレークポイント到達でブレークします。
- 【注3】 S/Wブレークは、MCU内部RAM領域および内部ROM領域以外の領域には指定できません。
- 【注4】 [エディタ]ウィンドウでプログラム変更し、MCUに対してそのプログラムをダウンロードする場合、変更内容によっては設定されているS/Wブレークの設定アドレスが正常に補正されない場合があります。このため、設定したS/Wブレークの設定アドレスがずれる場合があります。ユーザプログラムダウンロード後は、S/Wブレークの設定アドレスを再確認してください。
- 【注5】 S/Wブレーク設定アドレスからユーザプログラム実行する場合、該当アドレス(ユーザプログラム開始アドレス)をステップ実行した上で、ユーザプログラムが実行されます。このためS/Wブレークが設定されていない場合と比べ、実行時間や割込み等の動作が異なります。
- 【注6】 S/Wブレークポイント設定後、設定解除せずにターゲットマイコンとの接続を解除し、再接続した場合、次のケースではS/Wブレークポイントが設定されたままになっています。
  - ・起動時の[コンフィグレーション]ダイアログボックスで、S/Wブレークポイントを設定したフラッシュブロックにチェック(データを残す設定)した場合。
- 【注7】 MCU内部RAMにS/Wブレークポイントが設定されている場合、MCU内部RAMに設定されたS/Wブレークポイントアドレスの内容を[メモリ]ウィンドウ等で書き換えしないでください。

## 5.6.1 S/Wブレークポイントを追加する／削除する

以下のいずれかの方法で、S/Wブレークポイントを追加／削除してください。

- [エディタ]ウィンドウまたは [逆アセンブリ]ウィンドウから
- [ブレークポイント]ダイアログボックスから（削除のみ）
- コマンドラインから

## (1) [エディタ]または[逆アセンブリ]ウィンドウからS/Wブレークポイントを追加／削除する

- ① S/Wブレークポイントを設定する位置の[エディタ]または[逆アセンブリ]ウィンドウが開いていることを確認します。
- ② プログラムを停止したい行で、S/Wブレークポイントカラムをダブルクリックします。  
または、下記操作でも設定することができます。
  - ポップアップメニューの[ブレークポイントの挿入/削除]を選択する、または“F9”キーを押します。
- ③ S/Wブレークポイントが設定されると、[エディタ]または[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に赤丸[●]を表示します。  
もう一度ダブルクリックするとブレークポイントは削除されます。

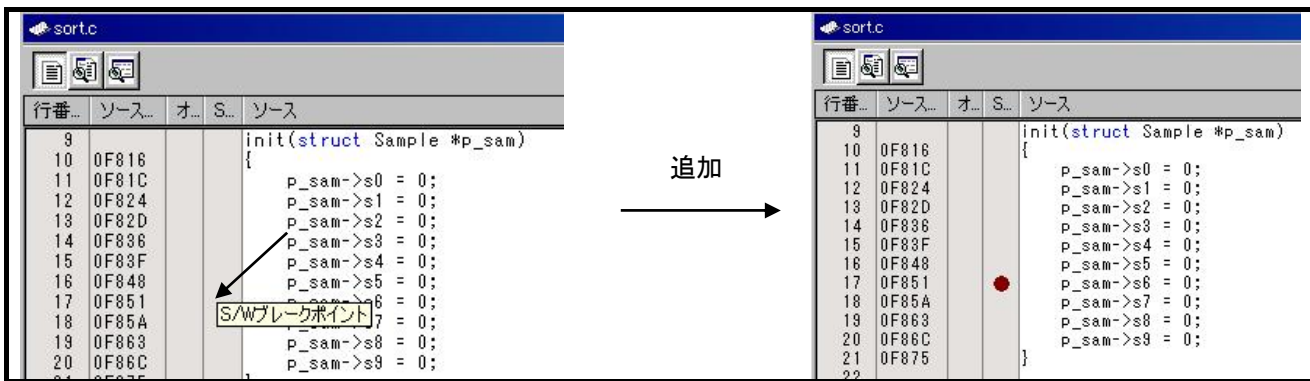


図 5.6 [エディタ]ウィンドウからS/Wブレークポイントの追加/削除

## (2) [ブレークポイント]ダイアログボックスから

[編集 -> ソースブレークポイント...]を選択し、[ブレークポイント]ダイアログボックスを表示します。このダイアログボックスでは、現在設定しているブレークポイントの有効/無効の切り替え、および削除をすることができます。

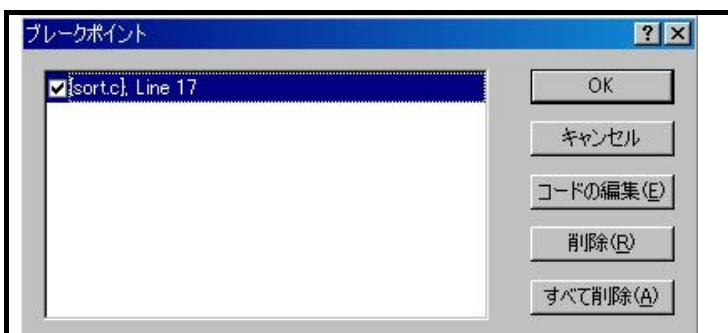


図 5.7 [ブレークポイント]ダイアログボックスからのブレークポイント有効/無効切替

## (3) コマンドラインから

ヘルプファイルを参照ください。

## 5.6.2 S/Wブレークポイントを有効にする／無効にする

以下のいずれかの方法で、S/Wブレークポイントを有効または無効にしてください。

- [エディタ]ウィンドウまたは[逆アセンブリ]ウィンドウから
- [ブレークポイント]ダイアログボックスから
- コマンドラインから

## (1) [エディタ]または[逆アセンブリ]ウィンドウから

- ① S/Wブレークポイントがある行にカーソルを置いて、ポップアップメニューから[ブレークポイントの有効化／無効化]を選択するか、“Ctrl + F9”キーを押します。
- ② S/Wブレークポイントの有効／無効が切り替わります。

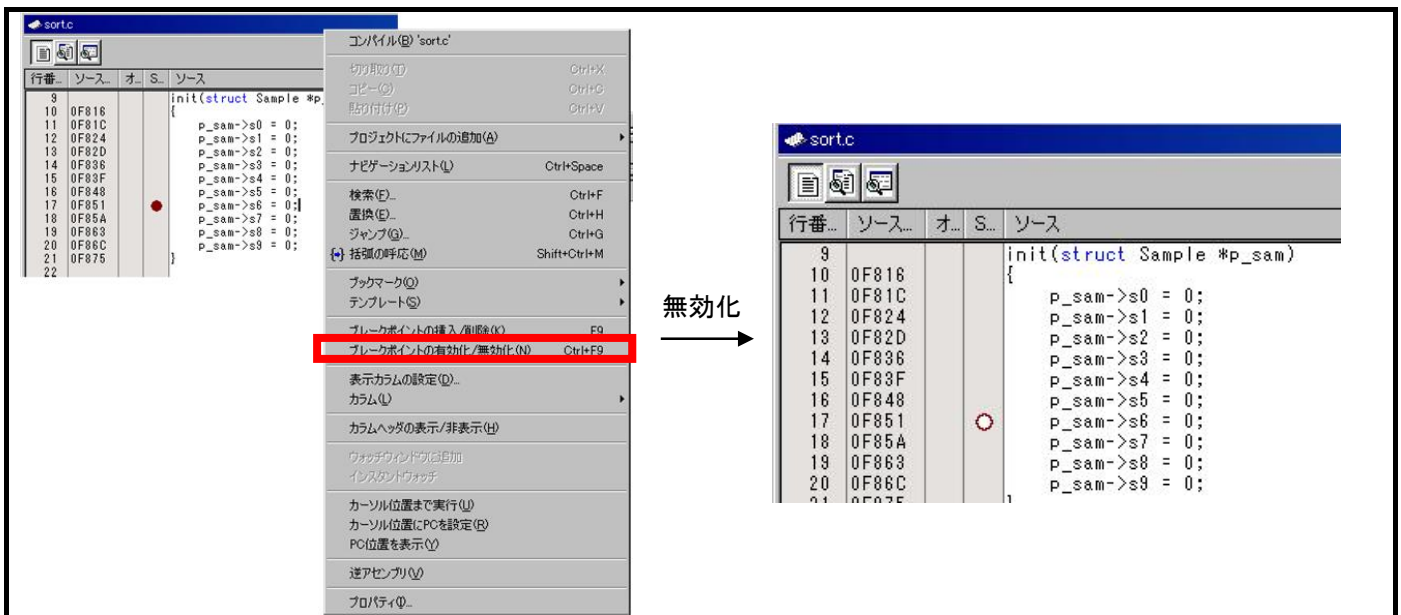


図 5.8 [エディタ]ウィンドウとポップアップメニュー

## (2) [ブレークポイント]ダイアログボックスから

[編集 -> ソースブレークポイント...]を選択し、[ブレークポイント]ダイアログボックスを表示します。このダイアログボックスでは、現在設定しているブレークポイントの有効/無効の切り替え、および削除をすることができます。

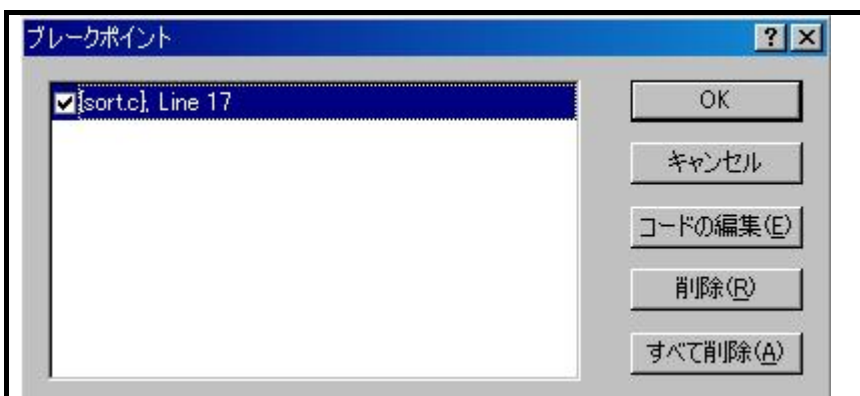


図 5.9 [ブレークポイント]ダイアログボックスからのブレークポイント有効/無効切替

## (3) コマンドラインから

ヘルプファイルを参照ください。

## 5.7 オンチップブレーク機能の概要

### 5.7.1 [オンチップイベント条件設定]ダイアログボックス

オンチップイベントのブレーク条件、トレース条件を設定するためのウィンドウです。

[表示 → イベント → オンチップイベント条件設定]を選択して、[オンチップイベント条件設定]ダイアログボックスを表示します。



図 5.10 [オンチップイベント条件設定]ダイアログボックス

#### ① 条件設定タブ

本ダイアログボックスでは、下記2種類のイベントについて追加、削除、有効、無効が可能です。

タブ	内容	備考
イベント	Event A, Bに関する設定および トレース条件設定	ウィンドウオープン時デフォルト表示
実行前PCブレーク	PCイベントを用いた実行前PCブレークの設定	

#### ② 登録イベント一覧

登録済みイベントの一覧が表示されます。詳細は「5.10 イベントの登録」(74ページ)参照

#### ③ 保存...

設定済みの本ダイアログボックスの内容をファイルに保存します。

#### ④ 読み込み...

保存済みの[オンチップイベント条件設定]ダイアログボックスの設定内容をファイルから読み込みます。正常に読み込んだ場合、本ダイアログボックスの設定内容を破棄してファイルの内容に置き換えます。

#### ⑤ 適用

適用ボタンのクリックで、本ダイアログボックスで設定した内容がE1/E20エミュレータに設定されます。  
[オンチップイベント条件設定]ダイアログボックスにて、1箇所でも設定内容が変更された場合もしくは「読み込み」にてファイルから設定内容を読み込んだ場合、タイトルバーのウィンドウ名の横に"\*"が表記されます。「適用」ボタンをクリック後、この"\*"表記は消えます。

#### ⑥ 閉じる

本ダイアログボックスを閉じます。本ダイアログボックスで設定変更したが、「適用」ボタンがクリックされていない場合は、確認のためのメッセージボックスが表示されます。このメッセージボックスの選択で下記動作となります。

- 「はい」選択時： 本ダイアログボックスでの設定内容適用の上、本ダイアログボックスを閉じます。
- 「いいえ」選択時： 本ダイアログボックスでの設定内容を破棄して、本ダイアログボックスを閉じます。

5.7.2 イベントタブ

本イベントタブでは、EventA、EventBの追加/削除/有効/無効の設定、ブレーク条件設定、トレース条件設定を行います。図 5.11に[オンチップイベント条件設定]ダイアログボックス、表 5.6にEventA、EventBの設定内容、表 5.7にブレーク条件の設定内容、表 5.8にトレース条件の設定内容を記載します。



図 5.11 [オンチップイベント条件設定]ダイアログボックス(イベントタブ)

表 5.6 EventA, EventBの設定

項目	内容		
表示内容	登録済みのイベントをダブルクリック時、 [イベント設定]ダイアログボックスが表示され、イベント内容の参照/設定が可能です。 関数名などのテキストをリストにドロップすることで新規登録可能です。		
	項目	内容	備考
	チェック欄	チェックボックスの チェックで有効化	有効可能なイベントはEventA, EventBで各1個 2個目のチェック時、1個目が無効、2個目が有効化
	イベント	EVxxと表示	EventA, EventBでそれぞれ最大8個まで登録可能
	タイプ	DA(データアクセスイベント)が表示	
	設定内容	イベントの設定内容が表示	
	チャンネル	EventAの場合、" 0" が表示 EventBの場合、" 1" が表示	
	コメント	各イベントのコメントが表示	
「追加」 ボタン	[イベント設定]ダイアログボックスが表示され、イベント内容の追加が可能です。 新規追加されたイベントは有効化されます。 イベント数の登録が8点の場合は、本ボタンはグレー表示され新規登録できません。		
「削除」 ボタン	リスト上で選択されたイベントを全てリストから削除します。 ([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)		
「有効」 ボタン	リスト上で選択されたイベントを有効化します。 2点以上のイベントが選択されている場合は、本ボタンはグレー表示されます。		
「無効」 ボタン	リスト上で選択されたイベントを無効化します。		

表 5.7 ブレーク条件の設定

項目	内容
条件	ブレーク条件を選択します。 None : イベントによるブレークなし(デフォルト) Event A : EventAが成立したらブレーク Event A or B : Event A, Bのいずれかが成立したらブレーク Event A and B : Event A, Bの両方が成立したらブレーク Event B->A : イベントがB->A の順番で成立したらブレーク
トレースフルブレーク	チェックするとトレースバッファが一杯になったときにブレークします。 デフォルトは未チェック(トレースフルブレーク無効)です。 イベントによるブレーク条件と同時に設定できます。

表 5.8 トレース条件の設定

オプション	説明
種別	トレース種別を下記から選択します。 Branch Trace(デフォルト) : 分岐トレース ユーザプログラム実行中に発生した、 分岐処理の分岐元、分岐先アドレスをトレース Data Access : データアクセス ユーザプログラム実行中に発生したイベントの データアクセス情報をトレース
開始	トレース計測の開始条件を下記から選択します。 Go(デフォルト) : ユーザプログラム実行開始時に計測開始 Event A : イベントAが成立したら計測開始 Event A or B : イベントA, Bのいずれかが成立したら計測開始 Event A and B : イベントA, Bの両方が成立したら計測開始 Event B->A : イベントがB->A の順番で成立したら計測開始
終了	トレース計測の終了条件を下記から選択します。 Break(デフォルト) : ユーザプログラム実行停止時に計測終了 Trace FULL : トレースデータが一杯になったら計測終了 Event A : イベントAが成立したら計測終了 Event A or B : イベントA, Bのいずれかが成立したら計測終了 Event A and B : イベントA, Bの両方が成立したら計測終了 Event B->A : イベントがB->A の順番で成立したら計測終了
記録	トレース種別が「Data Access」時に記録するイベントを下記から選択します。 Event A(デフォルト) : イベントAの条件に該当するデータアクセスのみ記録する Event A or B : イベントA, Bいずれかの条件成立時のデータアクセスを記録

- 【注1】 トレース開始条件に関する注意事項  
トレース開始条件にイベント("Go"以外)を設定した場合、イベント成立時のデータはトレースデータに記録されず、次に成立したイベントのデータが記録されます。
- 【注2】 トレース終了条件に関する注意事項  
トレース開始条件と終了条件が同時に発生した場合、トレース終了条件が無効になります。  
トレース終了条件に"Break"以外を設定した場合、トレース終了条件が成立した後もユーザプログラムが停止するまでは、トレースウィンドウの表示内容は更新されません。
- 【注3】 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リセット等リセットが発生した場合、リセットまでのイベント成立、トレース記録はクリアされます。
- 【注4】 トレース情報はCPUバスによるアクセスのみ記録可能です。DTCIによるアクセスは記録できません。

## 5.7.3 実行前PCブレークタブ

本イベントタブでは、実行前PCブレーク設定を行います。図 5.12に[オンチップイベント条件設定]ダイアログボックスの実行前PCブレークタブ、表 5.9にその設定内容を記載します。



図 5.12 [オンチップイベント条件設定]ダイアログボックス(実行前PCブレークタブ)

表 5.9 [オンチップイベント条件設定]ダイアログボックス(実行前PCブレークタブ)

項目	内容																					
表示内容	登録済みのイベントをダブルクリック時、 [イベント設定]ダイアログボックスが表示され、イベント内容の更新が可能です。 関数名などのテキストをリストにドロップすることで新規登録可能です。																					
	<table border="1"> <thead> <tr> <th>項目</th> <th>内容</th> <th>備考</th> </tr> </thead> <tbody> <tr> <td>チェック欄</td> <td>チェックボックスのチェックで有効化</td> <td>有効可能なイベントは最大8個</td> </tr> <tr> <td>イベント</td> <td>EVxxと表示</td> <td>最大32個まで登録可能</td> </tr> <tr> <td>タイプ</td> <td>PC(アドレス一致イベント)が表示</td> <td></td> </tr> <tr> <td>設定内容</td> <td>イベントの設定内容が表示</td> <td></td> </tr> <tr> <td>チャンネル</td> <td>0~7までのいずれかが表示【注1】</td> <td></td> </tr> <tr> <td>コメント</td> <td>各イベントのコメントが表示</td> <td></td> </tr> </tbody> </table>	項目	内容	備考	チェック欄	チェックボックスのチェックで有効化	有効可能なイベントは最大8個	イベント	EVxxと表示	最大32個まで登録可能	タイプ	PC(アドレス一致イベント)が表示		設定内容	イベントの設定内容が表示		チャンネル	0~7までのいずれかが表示【注1】		コメント	各イベントのコメントが表示	
項目	内容	備考																				
チェック欄	チェックボックスのチェックで有効化	有効可能なイベントは最大8個																				
イベント	EVxxと表示	最大32個まで登録可能																				
タイプ	PC(アドレス一致イベント)が表示																					
設定内容	イベントの設定内容が表示																					
チャンネル	0~7までのいずれかが表示【注1】																					
コメント	各イベントのコメントが表示																					
「追加」ボタン	[イベント設定]ダイアログボックスが表示され、イベント内容の追加が可能です。【注2】 イベント数の登録が32点の場合は、本ボタンはグレー表示され新規登録できません。																					
「削除」ボタン	リスト上で選択されたイベントを全てリストから削除します。 ([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)																					
「有効」ボタン	リスト上で選択されたイベントを全て有効化します。																					
「無効」ボタン	リスト上で選択されたイベントを無効化します。																					

【注1】 「適用」ボタンをクリックするまでは、「#」と表示されています。

【注2】 イベント数の登録が7点までは、新規追加されたイベントが有効化されます。  
イベント数の登録が8点を超えると、「PCイベントの最大数を超えました」のメッセージが表示され無効状態で登録されます。

#### 5.7.4 [イベント設定]ダイアログボックス

PCイベント(アドレス一致ブレイク)、DAイベント(EventA, Bブレイク)の内容を設定するためのウィンドウです。  
[オンチップイベント条件設定]ダイアログボックスの[イベント]タブおよび[実行前PCブレイク]タブからイベント設定する場合および[登録イベント一覧]ダイアログボックスでイベント登録時に表示されます。

##### ① イベントタイプ

「アドレス一致」(PCイベント用)または「データアクセス」(DAイベント用)の選択が可能です。  
選択変更により設定項目が切り替わります。

[オンチップイベント条件設定]ダイアログボックスの[イベント]タブから本ダイアログボックスを開いた場合は、「データアクセス」が選択され、[実行前PCブレイク]タブから本ダイアログボックスを開いた場合は、「アドレス一致」が選択されます。

[登録イベント一覧]ダイアログボックスでイベント登録時はいずれの場合でも登録可能です。【注1】

##### ② 条件設定タブ

本ダイアログボックスでは、[条件]タブでイベントの条件、[コメント]タブでイベントのコメント設定が可能です。

- (1) [イベント設定]ダイアログボックス(イベントタイプ:アドレス一致)  
イベントタイプで「アドレス一致」選択時の表示を図 5.13に示す。

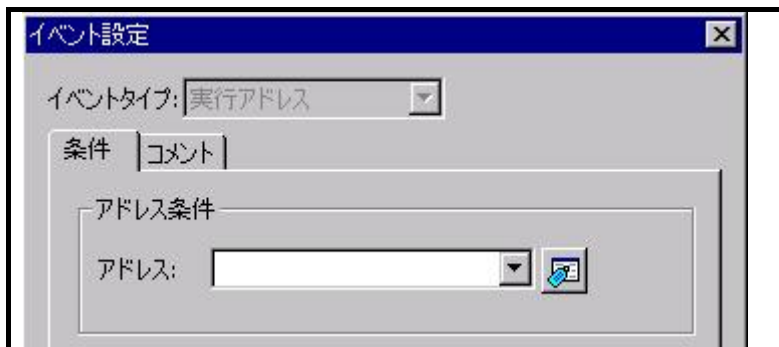


図 5.13 [イベント設定]ダイアログボックス(アドレス一致選択時)

【注1】 イベント登録可能な最大数を超える場合は登録できません。



## 5.7.5 [イベント設定]ダイアログボックス(イベントタイプ:データアクセス)

イベントタイプで「データアクセス」選択時の表示を図 5.14、表示内容を表 5.10に示す。



図 5.14 [イベント設定]ダイアログボックス(データアクセス選択時)

表 5.10 [イベント設定]ダイアログボックスの表示内容

項目	説明
<b>アドレス条件</b>	
開始アドレス	データアクセス検出アドレスを指定します。
マスク	<ul style="list-style-type: none"> <li>・チェックすることでアドレスマスク指定が可能 デフォルトはチェックなし(マスクなし)</li> <li>・アドレスマスクはビット数の指定で行います。 指定アドレスの下位ビットを指定されたビット数だけマスクします。 指定可能は"1"~"15"</li> </ul>
<b>データ条件</b>	
リード/ライト	<p>アクセス種別が下記より選択できます。</p> <p>リード/ライト: リードまたはライト</p> <p>リード: リードのみ</p> <p>ライト: ライトのみ</p>
アクセスサイズ	<p>アクセスサイズがBYTE、WORD、無指定から選択します。</p> <p>指定のアクセスサイズに一致しないデータアクセスのイベントは成立しません。</p> <p>また、アクセスサイズにWORD選択時、開始アドレスは偶数アドレスの指定が必要です。</p>
比較条件【注1】	<p>チェックすることでデータ比較指定が可能</p> <p>デフォルトはチェックなし(データ比較なし) 【注2】</p>
比較値	<p>比較するデータ値を指定します。</p> <p>アクセスサイズ(BYTE指定時): 00~FFh</p> <p>アクセスサイズ(WORD指定時): 0000~FFFFh</p>
マスク	<ul style="list-style-type: none"> <li>・チェックすることでデータマスク指定が可能 デフォルトはチェックなし(マスクなし)</li> <li>・マスクの値が空欄の場合は、「マスクなし」となります。</li> </ul>

【注1】 EventAの設定時のみ条件設定が可能です(入力は可能ですが、EventBへの登録はできません)。

イベント登録可能な最大数を超える場合は登録できません。

また、EventBの登録時にはデータ条件が入った設定は登録できません。

【注2】 本チェックボックスにチェックが入っている状態で"比較値"ボックスに値が入っていない場合、OKボタンをクリックしても本ダイアログは閉じません。

## 5.7.6 イベント設定に関する注意事項

Event AまたはEvent Bを設定する場合、アドレス、アクセスサイズおよびアクセス種別は表 5.11に従って設定してください。

表 5.11 イベント設定可否一覧表

イベント設定条件	イベント設定可否	[イベント設定]ダイアログボックスの設定例
偶数アドレスへのバイトリード	○	アドレス : 400h アクセスサイズ : BYTE アクセス種別 : READまたはR/W
偶数アドレスへのバイトライト	○	アドレス : 400h アクセスサイズ : BYTE アクセス種別 : WRITEまたはR/W
偶数アドレスへのワードリード	○	アドレス : 400h アクセスサイズ : WORD アクセス種別 : READまたはR/W
偶数アドレスへのワードライト	○	アドレス : 400h アクセスサイズ : WORD アクセス種別 : WRITEまたはR/W
奇数アドレスへのバイトリード	○	アドレス : 401h アクセスサイズ : BYTE アクセス種別 : READまたはR/W
奇数アドレスへのバイトライト	○	アドレス : 401h アクセスサイズ : BYTE アクセス種別 : WRITEまたはR/W
奇数アドレスへのワードリード	○	アドレス : 401h アクセスサイズ : BYTE【注1】 アクセス種別 : READまたはR/W
奇数アドレスへのワードライト	○	アドレス : 401h アクセスサイズ : BYTE【注1】 アクセス種別 : WRITEまたはR/W

【注1】 奇数アドレスへのワードアクセスの検出時には、アクセスサイズは“BYTE”を指定してください。  
なお、本条件において、比較可能なデータは、下位1バイト分となります。

## 【注2】 Event A設定に関する注意事項

Event Aのイベント設定では、アドレスのマスク指定とデータのマスク指定を同時に設定しないでください。同時に設定した場合、イベントが成立しません。

## 【注3】 イベント設定に関する注意事項

イベントに以下に示すアドレスを設定しないでください。

これらのアドレスを設定した場合、不正なブレークが発生することがあります。

- ・ 割り込みベクタテーブル内のアドレス
- ・ 割り込みベクタテーブルに設定されているアドレス(割り込みルーチン先頭アドレス)
- ・ 分岐命令の分岐先アドレス

上記の割り込みベクタテーブルには、固定ベクタテーブル、可変ベクタテーブルとも該当します。

## 5.8 オンチップイベント (DAイベント) の追加

以下のいずれかでイベントの設定が可能です。

- [オンチップイベント条件設定]ダイアログボックス
- 他ウィンドウからドラッグ&ドロップ (追加のみ)
- コマンドライン

### 5.8.1 [オンチップイベント条件設定]ダイアログボックスからの追加/変更

- (1) [表示 → イベント → オンチップイベント条件設定]を選択して、[オンチップイベント条件設定]ダイアログボックスを表示します。また、[イベント]タブに切り変えます。
- (2) イベントの追加は、[イベント]タブ内のEventA欄もしくはEventBの[追加]ボタンをクリックし、[イベント設定]ダイアログボックスを表示させます。  
イベントの変更および参照は、既存のイベントをダブルクリックして、[イベント設定]ダイアログボックスを表示させます。
- (3) [イベント設定]ダイアログボックスでイベント条件を設定または参照し、[OK]ボタンをクリックします。
- (4) [オンチップイベント条件設定]ダイアログボックスに、(3)で設定した条件が追加されるので、[適用]ボタンをクリックした後、[OK]ボタンをクリックします。



図 5.15 オンチップイベント (DAイベント) 設定例



図 5.16 [イベント設定]ダイアログボックス (DAイベント) 設定例

## 5.8.2 他ウィンドウからドラッグ&amp;ドロップ(追加のみ)

## (1) [エディタ]ウィンドウの変数名、関数名をドラッグ&amp;ドロップする場合

[オンチップイベント条件設定]ダイアログボックス内のイベント設定欄内に、変数名をドラッグ&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。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。

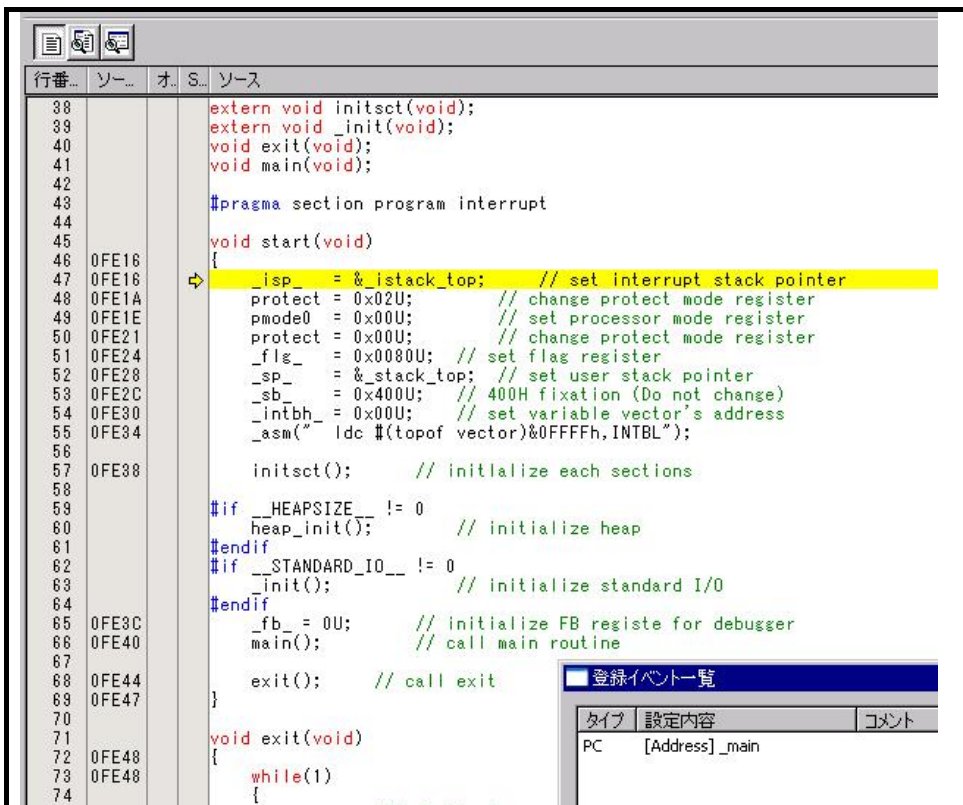


図 5.17 ドラッグ&ドロップによるイベント追加

## (2) [メモリ]ウィンドウのアドレス範囲をドラッグ&amp;ドロップする場合

[オンチップイベント条件設定]ダイアログボックス内のイベント設定欄内に、[メモリ]ウィンドウのメモリ内容を選択して、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を条件としてイベントを設定できます。

### 5.8.3 [オンチップイベント条件設定]ダイアログボックスからの削除/有効/無効

- (1) [表示 → イベント → オンチップイベント条件設定]を選択して、[オンチップイベント条件設定]ダイアログボックスを表示します。また、[イベント]タブに切り変えます。
- (2) [イベント]タブ内のEventA欄もしくはEventB欄について下記のように設定する。

イベントの削除	削除対象のイベントを選択して[削除]ボタンをクリックする。 複数のイベントを削除する場合は、削除したいイベント行をShiftキーまたはCtrlキーを押しながら選択して[削除]ボタンをクリックする。 ([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)
イベントの有効	有効化するイベントを選択して[有効]ボタンをクリックする。
イベントの無効	無効化するイベントを選択して[無効]ボタンをクリックする。

- (3) [適用]ボタンをクリックした後、[OK]ボタンをクリックします。

### 5.8.4 コマンドライン

オンラインヘルプを参照ください。

## 5.9 オンチップイベント(PCイベント)の設定

以下のいずれかでイベントの設定が可能です。

- [オンチップイベント条件設定]ダイアログボックス
- [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
- 他ウィンドウからドラッグ&ドロップ(追加のみ)
- コマンドライン

### 5.9.1 [オンチップイベント条件設定]ダイアログボックスの追加

- (1) [表示 → イベント → オンチップイベント条件設定]を選択して、[オンチップイベント条件設定]ダイアログボックスを表示します。また、[実行前PCブレーク]タブに切り変えます。
- (2) イベントの追加は、[追加]ボタンをクリックし、[イベント設定]ダイアログボックスを表示させます。イベントの変更および参照は、既存のイベントをダブルクリックして、[イベント設定]ダイアログボックスを表示させます。
- (3) [イベント設定]ダイアログボックスでイベント条件を設定または参照し、[OK]ボタンをクリックします。
- (4) [オンチップイベント条件設定]ダイアログボックスに、(3)で設定した条件が追加されるので、[適用]ボタンをクリックした後、[OK]ボタンをクリックします。



図 5.18 オンチップイベント(PC イベント)設定例

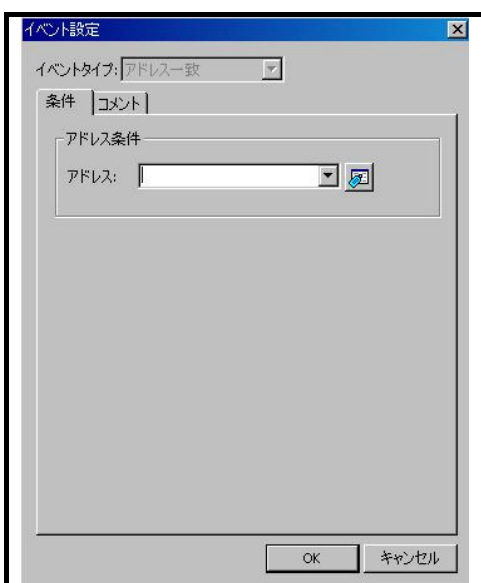


図 5.19 [イベント設定]ダイアログボックス(PCイベント)設定例

## 5.9.2 [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

## (1) ブレーク条件の追加/削除

[エディタ]ウィンドウにて、実行前PCブレークが設定されていない行に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と、実行前PCブレークが有効状態(青●)で設定することができます。

また、実行前PCブレークが有効状態(青●)もしくは無効状態(白○)の行に、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と、実行前PCブレークが削除されます。

なお、[エディタ]ウィンドウからの実行前PCブレーク条件設定はユーザプログラム実行中でも可能です。

## (2) ブレーク条件の有効/無効設定

有効/無効はこのウィンドウ上から設定できません。[オンチップイベント条件設定]ダイアログボックスもしくは「コマンドライン」からの設定が必要です。

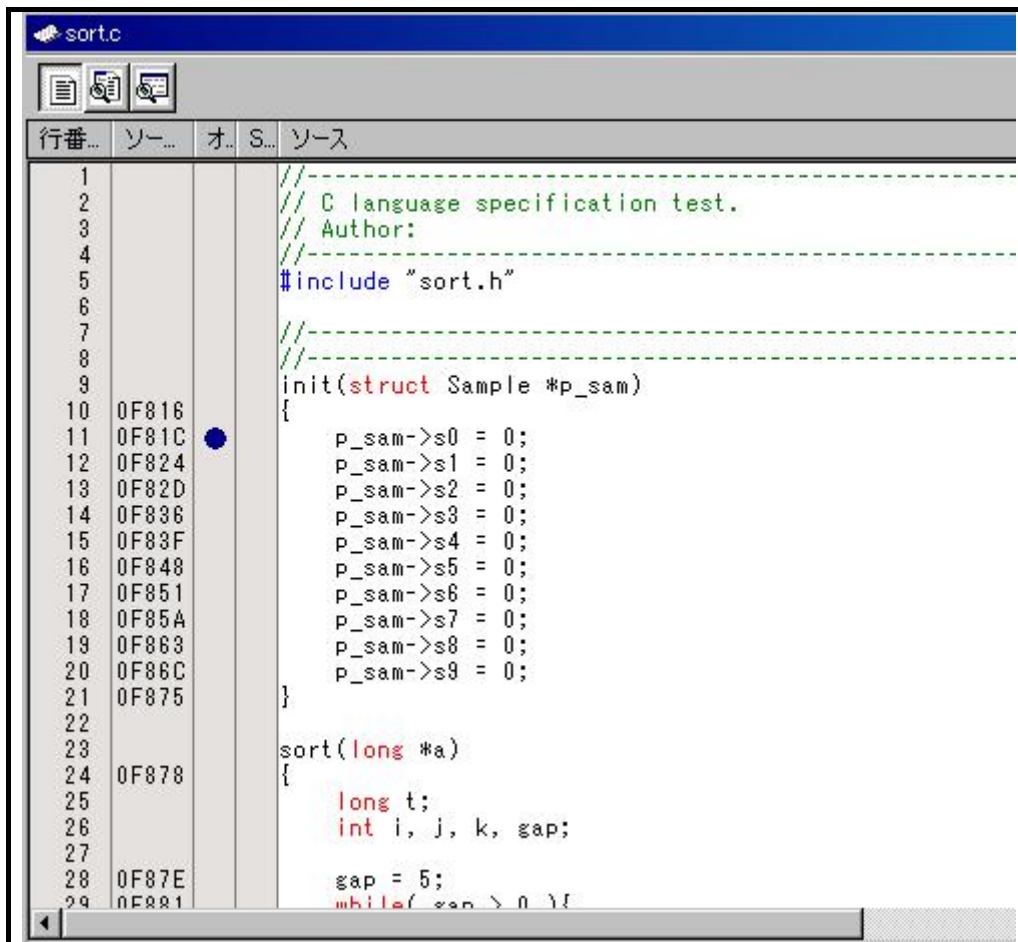


図 5.20 [エディタ] ウィンドウ

- 【注1】 [オンチップイベント条件設定]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、設定内容が有効になっていない(タイトルバーの最後に「\*」が付いている)場合、[エディタ]ウィンドウでの実行前PCブレークポイントは設定できません。
- 【注2】 [オンチップイベント条件設定]ダイアログボックスで設定変更後に[有効]ボタンがクリックされた場合や、コマンドラインからのオンチップイベント設定された場合、[エディタ]ウィンドウ上のカラム表示も更新されます。また、[エディタ]ウィンドウでの実行前PCブレーク条件設定内容も、[オンチップイベント条件設定]ダイアログボックスやコマンドラインでの実行前PCブレーク条件の設定内容に反映されます。

## 5.9.3 他ウィンドウからドラッグ&amp;ドロップ(追加のみ)

## (1) [エディタ]ウィンドウの変数名、関数名をドラッグ&amp;ドロップする場合

[オンチップイベント条件設定]ダイアログボックス内のイベント設定欄内に、変数名をドラッグ&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。

[オンチップイベント条件設定]ダイアログボックス内のイベント設定欄内に、関数名をドラッグ&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。

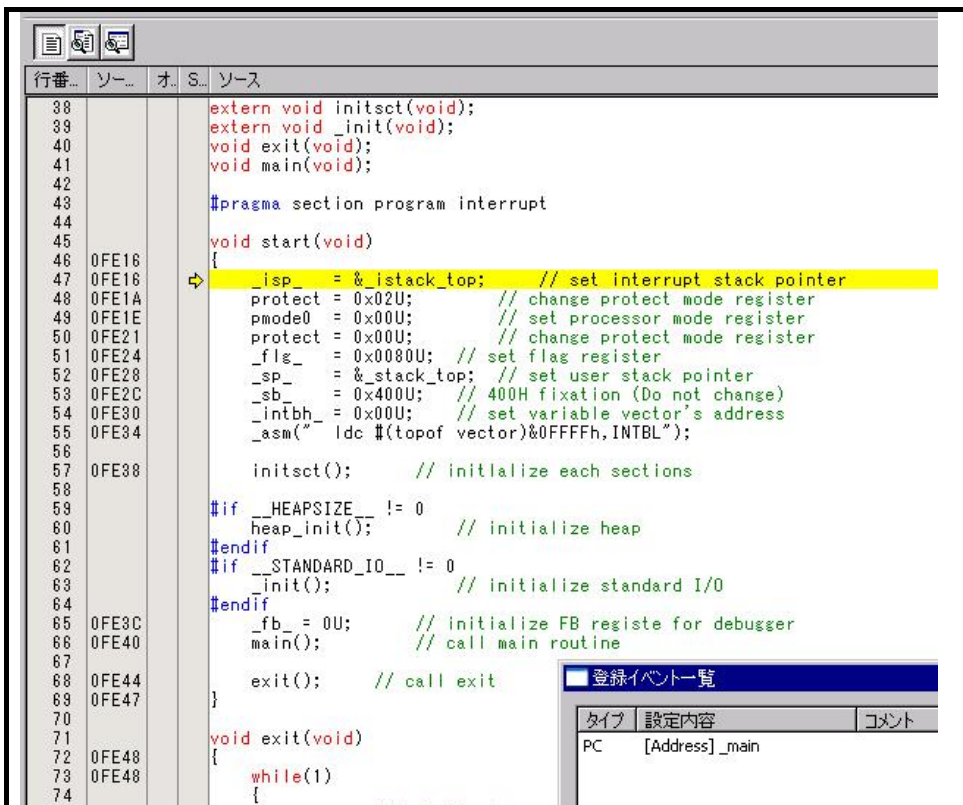


図 5.21 ドラッグ&ドロップによるイベント追加

## (2) [ラベル]ウィンドウのラベルをドラッグ&amp;ドロップする場合

[オンチップイベント条件設定]ダイアログボックス内のイベント設定欄内に、[ラベル]ウィンドウのラベルを選択して、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が設定できます。



#### 5.9.4 [オンチップイベント条件設定]ダイアログボックスからの削除/有効/無効

(1) [表示 → イベント → オンチップイベント条件設定]を選択して、[オンチップイベント条件設定]ダイアログボックスを表示します。また、[実行前PCブ레이크]タブに切り変えます。

(2) 下記のように設定する。

イベントの削除	削除対象のイベントを選択して[削除]ボタンをクリックする。 複数のイベントを削除する場合は、削除したいイベント行をShiftキーまたはCtrlキーを押しながら選択して[削除]ボタンをクリックする。 ([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)
イベントの有効	有効化するイベントを選択して[有効]ボタンをクリックする。
イベントの無効	無効化するイベントを選択して[無効]ボタンをクリックする。

(3) [適用]ボタンをクリックした後、[OK]ボタンをクリックします。

#### 5.9.5 コマンドライン

オンラインヘルプを参照ください。

## 5.10 イベントの登録

## 5.10.1 [登録イベント一覧]ダイアログボックス

[登録イベント一覧]ダイアログボックスは、[オンチップイベント条件設定]ダイアログボックス(図 5.22)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。

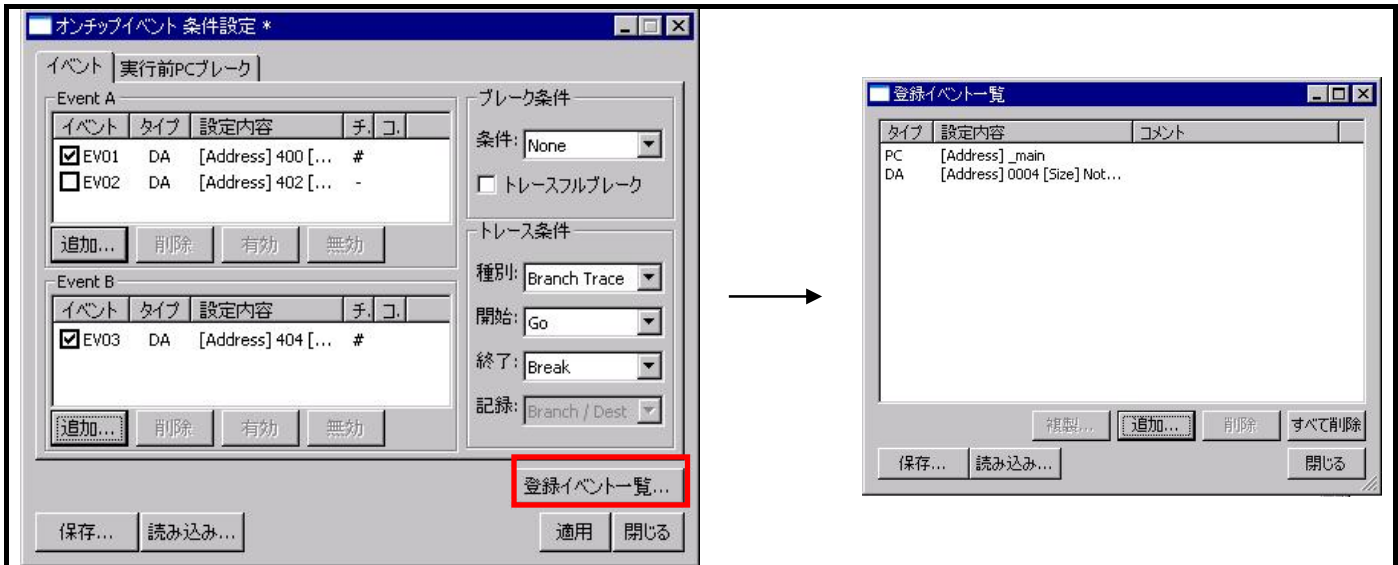


図 5.22 [登録イベント一覧] ダイアログボックスの表示方法

## (1) 複製

コピーしたいイベントを選択して[複製]ボタンをクリックしてください。選択したイベントがコピーされます。

## (2) 追加

[イベント設定]ダイアログボックスが表示され、イベント内容の追加が可能です。

## (3) 削除

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。選択したイベントをリストから削除します。([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)

## (4) すべて削除

全てのイベントを削除します。

## (5) 保存...

設定済みの本ダイアログボックスの内容をファイルに保存します。

## (6) 読み込み...

保存済みのイベント情報をファイルから読み込みます。正常に読み込んだ場合、本ダイアログボックスの設定内容を破棄してファイルの内容に置き換えます。

## (7) 閉じる

本ダイアログボックスを閉じます。

表 5.12 登録イベントダイアログの表示内容

項目	内容	備考
タイプ	DA(データアクセスイベント)もしくはPC(アドレス一致イベント)が表示	
設定内容	イベントの設定内容が表示	
コメント	各イベントのコメントが表示	

## 5.10.2 イベントの登録

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。

- [オンチップイベント条件設定]ダイアログボックスからイベントを登録
- ドラッグ&ドロップでイベントを登録
- [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

## (1) [オンチップイベント条件設定]ダイアログボックスからイベントを登録

[オンチップイベント条件設定]ダイアログボックスで[コメント]タブを表示して、[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た上で、[OK]ボタンをクリックします。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。

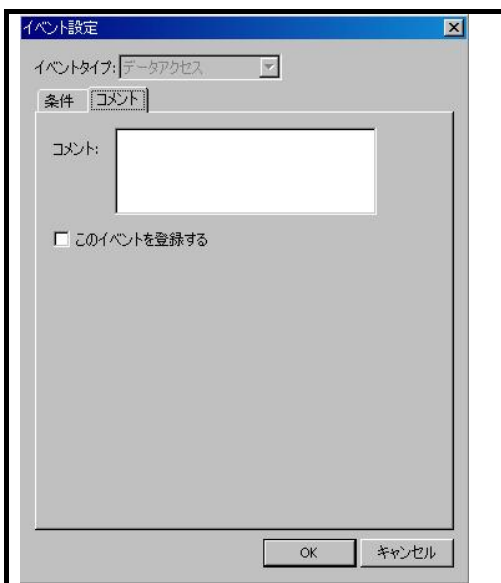


図 5.23 [イベント設定]ダイアログボックスからのイベント登録

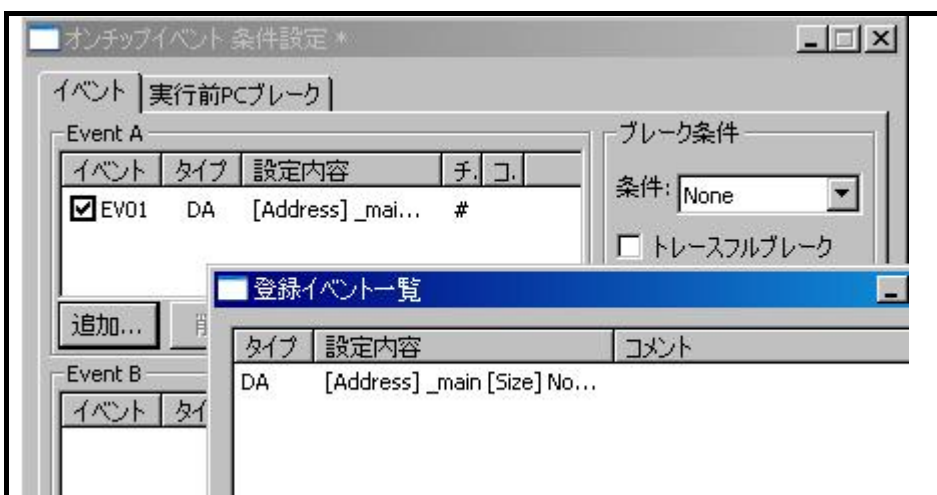


図 5.24 [オンチップイベント条件設定]ダイアログボックスと[登録イベント一覧]ダイアログボックス

## (2) ドラッグ&amp;ドロップでイベントを登録

作成したイベントを[登録イベント一覧]ダイアログボックスにドラッグ&ドロップして登録します。

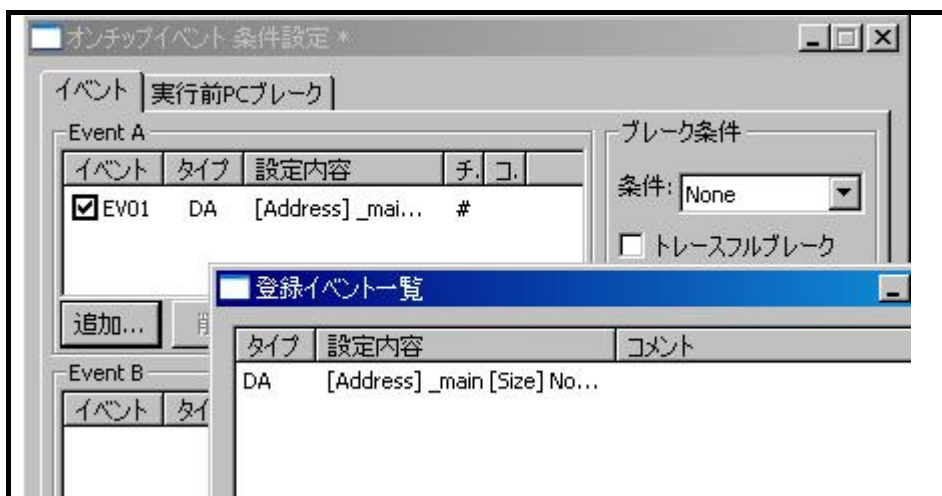


図 5.25 [オンチップイベント条件設定]ダイアログボックスと[登録イベント一覧]ダイアログボックス

## (3) [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

[登録イベント一覧]ダイアログボックスの[追加...]ボタンをクリックして、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。

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。



図 5.26 [登録イベント一覧]ダイアログボックス

## 5.10.3 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。

表 5.13 イベントの設定方法種類

イベント設定方法	説明
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。

## (1) 複数機能へのドラッグ&amp;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。

## (2) [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。

## 5.10.4 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。

[オンチップイベント条件設定]ダイアログボックス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 5.27 設定内容の適用

## 5.11 オンチップイベント条件設定内容を保存する／読み込む

[オンチップイベント条件設定]ダイアログボックスの内容は保存と読み込みが可能です。

### 5.11.1 [オンチップイベント条件設定]内容を保存する

(1) [オンチップイベント条件設定]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。

[保存]ダイアログボックスが開きます。

(2) 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 .rev です。省略した場合は、拡張子 .rev が付加されます。

### 5.11.2 [オンチップイベント条件設定]内容を読み込む

(1) [オンチップイベント条件設定]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。

[読み込み]ダイアログボックスが開きます。

(2) 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[オンチップイベント条件設定]内容は破棄して、読み込んだ設定で再設定します。

(3) [オンチップイベント条件設定]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[オンチップイベント条件設定]内容が確定します。

## 5.12 [登録イベント一覧]ダイアログボックスの設定内容を保存する／読み込む

[登録イベント一覧]ダイアログボックスの内容は保存と読み込みが可能です。

### 5.12.1 [登録イベント一覧]ダイアログボックスの内容を保存する

(1) [登録イベント一覧]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。

[保存]ダイアログボックスが開きます。

(2) 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 .lev です。省略した場合は、拡張子 .lev が付加されます。

### 5.12.2 [登録イベント一覧]ダイアログボックス内容を読み込む

(1) [登録イベント一覧]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。

[読み込み]ダイアログボックスが開きます。

(2) 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[登録イベント一覧]ダイアログボックス内容は破棄して、読み込んだ設定で再設定します。

(3) [登録イベント一覧]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[登録イベント一覧]ダイアログボックスの内容が確定します。

## 5.13 トレース機能

### 5.13.1 トレース機能概要

トレースとは、ユーザプログラム実行中の分岐情報やデータアクセスであるバス情報を取得する機能です。

ユーザプログラム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。

本マニュアルでの対応マイコンは、デバイス内蔵の表 5.14に示すトレース機能を使用できます。

表 5.14 対応MCUのトレース機能

トレース種類	内容	
分岐トレース	分岐数	最大4分岐
	分岐情報	分岐元/分岐先がセットで取得
	内容	分岐元、および、分岐先のアドレスとニーモニック、ソース行を表示
データトレース	データ数	最大8アクセス分
	内容	データアクセスイベントが成立したときのデータアクセスを表示

トレース結果は、[表示 -> コード -> トレース]を選択して、[トレース]ウィンドウを表示します。

図 5.28にトレース結果例、表 5.15に表示する項目を示します。

PTR	IP	Type	Address	Data	Instruction	Source	Label
-000007	0007	BRANCH	0000FD56		JEQ 0FD64H		
-000006	0006	DESTINATION	0000FD64		DIVU.W R1		
-000005	0005	BRANCH	0000FD68		JMP.B 0FD90H		
-000004	0004	DESTINATION	0000FD90		POPM R1,R3,A0,A1		
-000003	0003	BRANCH	0000FD92		RTS		
-000002	0002	DESTINATION	0000FAFD		ADD.B #4H,SP		
-000001	0001	BRANCH	0000FB08		MOV.W 0406H,R0		
+000000	0000	DESTINATION	0000806C		*** EML ***		

図 5.28 トレース結果例

表 5.15 トレース表示

項目	内容
PTR	トレースバッファ内ポインタ番号を表示します。トレース停止位置を0として昇順に表示します。
IP	命令ポインタを表示します(PTRの絶対値)。 データトレース時は何も表示しません。
Type	トレース情報の種別を表示します。 分岐トレース時はBRANCH/DESTINATIONを、データトレース時はREAD/WRITEを表示します。
Address	分岐トレース時は分岐元/分岐先命令のアドレスを表示します。 データトレース時は成立したイベントに設定したアドレスあるいはアドレス範囲を表示します。
Data	データトレース時はアクセスされた値を表示します。分岐トレース時は何も表示しません。
Instruction	分岐トレース時は該当するアドレスのニーモニックを表示します。 データトレース時は何も表示しません。 Instructionに”*** EML ***”と表示されることがあります。 これはブレーク等の制御のためにエミュレータ使用領域にアクセスしたことを示し、異常ではありません。
Source	Instructionに対応するソース行情報がある場合は、対応するソース行を表示します。 データトレース時は何も表示しません。
Label	Instructionのアドレスに対応するラベルがある場合は、対応するラベルを表示します。 データトレース時は何も表示しません。

## 5.13.2 トレース設定項目

トレース条件は、[オンチップイベント条件設定]ダイアログボックスで設定する。トレース条件は表 5.16に示す。

表 5.16 トレース条件

オプション	説明
種別	トレース種別を下記から選択します。 Branch Trace(デフォルト) : 分岐トレース ユーザプログラム実行中に発生した、 分岐処理の分岐元、分岐先アドレスをトレース Data Access : データアクセス ユーザプログラム実行中に発生したイベントの データアクセス情報をトレース
開始	トレース計測の開始条件を下記から選択します。 Go(デフォルト) : ユーザプログラム実行開始時に計測開始 Event A : イベントAが成立したら計測開始 Event A or B : イベントA, Bのいずれかが成立したら計測開始 Event A and B : イベントA, Bの両方が成立したら計測開始 Event B→A : イベントがB→A の順番で成立したら計測開始
終了	トレース計測の終了条件を下記から選択します。 Break(デフォルト) : ユーザプログラム実行停止時に計測終了 Trace FULL : トレースデータが一杯になったら計測終了 Event A : イベントAが成立したら計測終了 Event A or B : イベントA, Bのいずれかが成立したら計測終了 Event A and B : イベントA, Bの両方が成立したら計測終了 Event B→A : イベントがB→A の順番で成立したら計測終了
記録	トレース種別が「Data Access」時に記録するイベントを下記から選択します。 Event A(デフォルト) : イベントAの条件に該当するデータアクセスのみ記録する Event A or B : イベントA, Bいずれかの条件成立時のデータアクセスを記録する

- 【注1】 トレース開始条件に関する注意事項  
トレース開始条件にイベント("Go"以外)を設定した場合、イベント成立時のデータはトレースデータに記録されず、次に成立したイベントのデータが記録されます。
- 【注2】 トレース終了条件に関する注意事項  
トレース開始条件と終了条件が同時に発生した場合、トレース終了条件が無効になります。  
トレース終了条件に"Break"以外を設定した場合、トレース終了条件が成立した後もユーザプログラムが停止するまでは、トレースウィンドウの表示内容は更新されません。  
トレース結果は、対応するMCUシリーズによって違いがでることがあります。
- 【注3】 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リセット等リセットが発生した場合、リセットまでのトレース記録はクリアされます。
- 【注4】 トレース情報はCPUバスによるアクセスのみ記録可能です。DTCIによるアクセスは記録できません。
- 【注5】 本対応MCUでは、トレースデータがR8C/3xと比べ最終トレースデータ("IP=0000")まで取得可能です。



5.13.3 トレースメニュー

[トレース]ウィンドウでのツールバーメニューを表 5.17に示します。

表 5.17 [トレース]ウィンドウでのツールバーメニュー

メニュー名	説明
設定	トレース情報の取得条件を設定します。 (本対応マイコンでは本機能は無効です)
クリア	全てのトレースレコードをクリアします。
ソースを表示	選択している行の該当アドレスに対応したソースプログラムを表示します。
停止	トレース情報の取得を停止します。 (本対応マイコンでは本機能は無効です)
取得再開	トレース情報の取得を再開します。 (本対応マイコンでは本機能は無効です)
分岐トレース	分岐トレース時のソース補完可否を設定します。 (データトレース設定時にはグレー表示されます)
統計	統計情報の解析を実行します。
関数コール	関数コール箇所の表示を行います。

5.13.4 分岐トレース

[トレース]ウィンドウの設定が分岐トレースの場合、分岐先と分岐元間の実行状態についてソース情報を元に補完することができます。ツールバーメニューの「分岐トレース」ボタンのクリックで図 5.29に示すウィンドウが表示されるので、「許可」を選択することで[トレース]ウィンドウにソース補完された情報が表示されます。

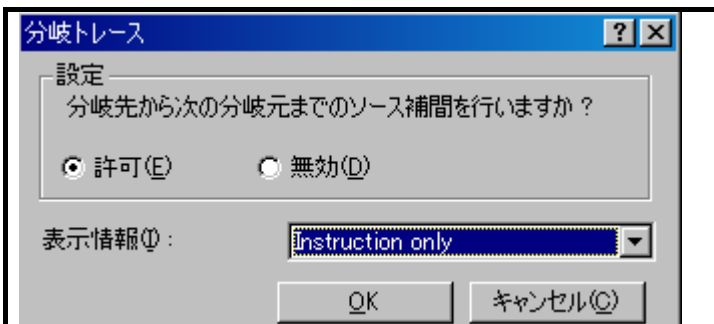


図 5.29 分岐トレースの補完設定

PTR	IP	Type	Address	Data	Instruction	Source	Label
-000007	0007	BRANCH	0000FB7D		EXITD		
-000006	0006	DESTINATION	0000FAB2		CMP.W ...		
			0000FAB4		JNE ...		
			0000FAB6		MOV.W ...		
-000005	0005	BRANCH	0000FAB8		EXITD		
-000004	0004	DESTINATION	0000F9A9		ADD.B ...		
			0000F9AB		MOV.W ...		
			0000F9AE		MOV.W ...	init(p_sam);	
-000003	0003	BRANCH	0000F9B1		JSR.A ...		
-000002	0002	DESTINATION	0000F81C		ENTER ...	{	\$init
			0000F819		MOV.W ...		
-000001	0001	BRANCH	0000F81C		MOV.W ...	p_sam->su ...	
+000000	0000	DESTINATION	0000803B		*** EML ***		

図 5.30 ソース補完された分岐トレース例

## 5.13.5 統計

トレース結果情報から、指定の条件を満たすトレース情報の個数とトレースバッファポインタ番号を表示することができます。ツールバーメニューの「統計」ボタンのクリックで図 5.31に示すウィンドウが表示されるので、「項目」を選択し、「新規」+「解析」ボタンをクリックすることでトレース情報の個数とトレースバッファポインタ番号が表示されます。

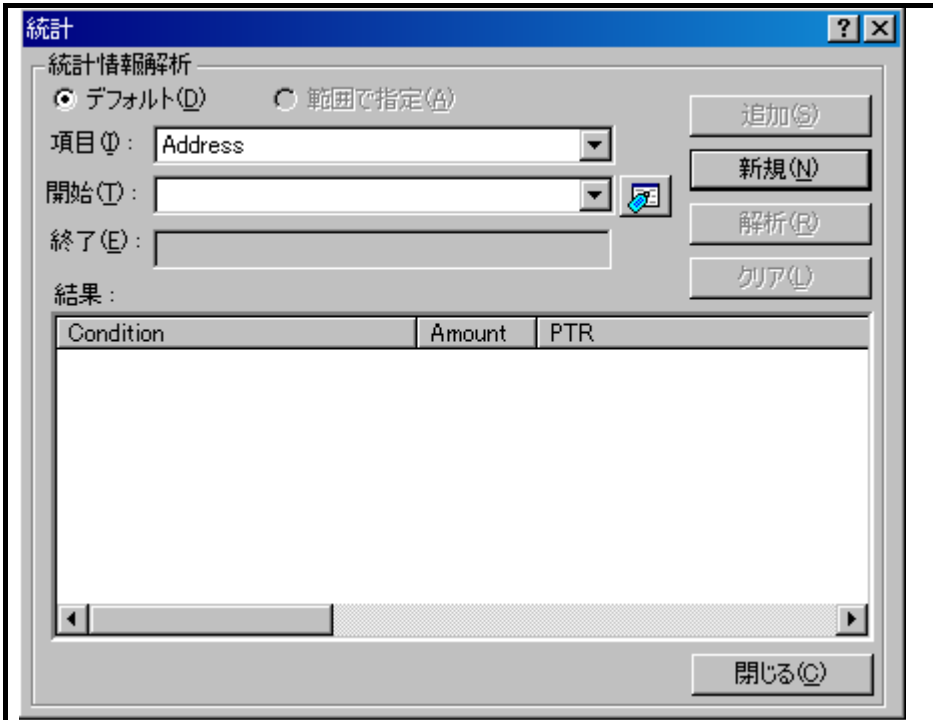


図 5.31 統計メニュー例

## (1)関数コール

トレース結果情報から、関数コール行のみを表示することができます。ツールバーメニューの「関数コール」ボタンのクリックでウィンドウが表示されるので、「許可」を選択することでトレースウィンドウには関数コール行の情報のみが表示されます。

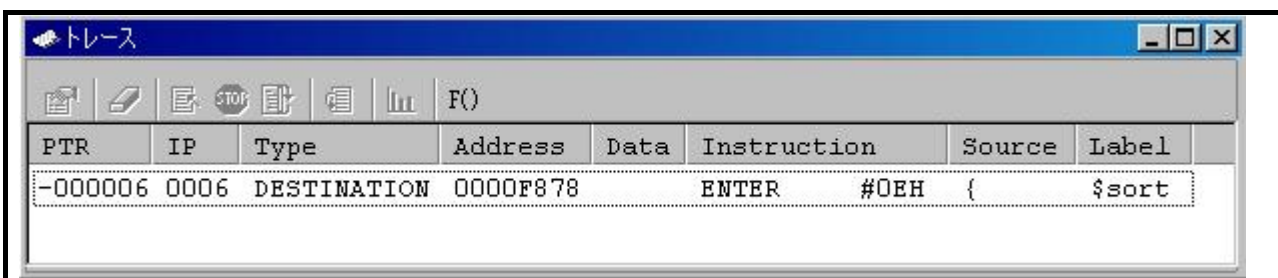


図 5.32 関数コールメニュー例

## 5.13.6 トレース情報をファイルに保存する

トレース情報をファイルに保存するには、ポップアップメニューから「保存」を選択します。[トレース] ウィンドウに表示しているトレース情報をテキスト形式で保存します。

## 5.14 ステータスバー機能

E1/E20エミュレータ接続時、デバッグプラットフォームの現在の状態を知るには[ステータス]バーを表示します。[ステータス]ウィンドウを開くには、[表示→ステータス]を選択します。

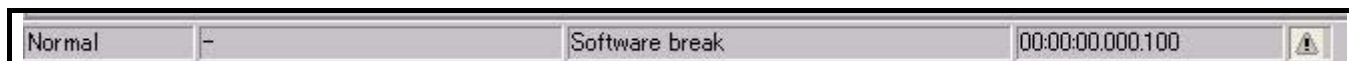


図 5.33 [ステータス]バー例

項目	内容	表示内容	備考
PC	プログラム実行中の状態	Normal	ユーザプログラム停止状態
		Running	ユーザプログラム実行状態
		Stop	ターゲットMCUがStop状態
		Wait	ターゲットMCUがWait状態【注1】
Task ID	OS使用モジュールがロード時、カレントタスクID番号表示	Task ID: xx	XX:カレントタスクID番号
Break Condition	ユーザプログラムの停止要因	Trace memory overflow break	トレースフルブレーク
		Stepping completed	ステップ実行
		Software break	S/Wブレーク
		Data access break at DAX, ...	データアクセスブレーク
		Data access break (Sequential)	データアクセスブレーク (シーケンシャル)
		Data access break (AND)	データアクセスブレーク (AND成立)
		Address match break	アドレス一致ブレーク
		User break	強制ブレーク
		Go to cursor break	カーソルまで実行
	Unknown break cause	要因不明	
Execution Time	RUN→STOP間の実行時間	hh:mm:ss.xxx.uuu	時:分:秒:ミリ秒:マイクロ秒

【注1】 CM30 ビットを“1”にしてウェイトモードに移行する場合、表示内容が“Wait”ではなく“Stop”と表示されます。

### 5.15 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。

#### 5.15.1 [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。

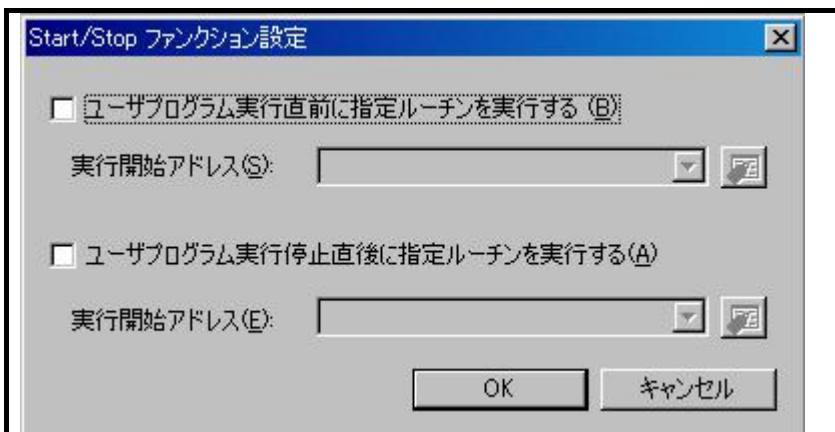


図 5.34 [Start/Stopファンクション設定]ダイアログボックス

#### 5.15.2 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。

## 5.15.3 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。

- (1) 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。
- 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
  - 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
- (2) 指定ルーチンで使用するCPU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 5.18 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	Uフラグ 指定ルーチン終了時に必ずフラグの値を0に設定してください。
Iフラグ	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。

- (3) 指定ルーチン実行時は、以下のデバッグ機能は無効です。
- トレース機能
  - ブレーク関連
  - 指定ルーチン内のイベント設定
- (4) 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。
- (5) 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、マイコンの状態は下記のようになります。

表 5.19 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
CPU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。
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。
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。

## 5.15.4 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、必ずユーザ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。

### 5.16 簡易スタックオーバーフロー機能

ソフトウェア開発時にスタック設計が不十分な場合、プログラムの暴走や誤動作を発生させてしまう可能性があります。

R8C E1/E20エミュレータデバッガは、スタックがレジスタ退避時にRAMの範囲を超えた場合に、スタックオーバーフローとして、「スタックがRAM領域の範囲を超えました。」と[アウトプット]ウィンドウに表示します。

**【注1】**

このエラーが発生した場合、エミュレータデバッガからリセット実行してください。

- 【注1】** スタックオーバーフローは、レジスタ退避時(ISPもしくはUSPのアドレスが400h以下の場合)のみ検出します。レジスタ復帰時にRAM領域の範囲を超えた場合は検出できません。  
スタックオーバーフローの正常検出は、スタックがMCU内部RAM領域の先頭アドレスである、400hから4バイト分(003FC~003FFh相当)までです。

### 5.17 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。

## 6. チュートリアル

### 6.1 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関数ではソートするランダムデータを生成します。

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

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

チュートリアルプログラムは、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】 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。
- 【注2】 チュートリアルプログラムは、tutorial.cファイルで提供しています。コンパイルされたロードモジュールは、Tutorial.x30ファイルとしてIEEE695フォーマットで提供しています。
- 【注3】 R8C E1/E20エミュレータデバッガに添付のチュートリアルプログラムは、R8C/3xシリーズ用です。本対応マイコンとはメモリマップが異なるため、このチュートリアルプログラムを使用するためにはSFRレジスタの設定変更および再コンパイルが必要ですのでご注意ください。

### 6.2 High-performance Embedded Workshopの起動

「4.9 E1/E20エミュレータデバッガの起動手順」(30ページ)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¥R8C¥Tutorial

ファイルは以下を指定してください。

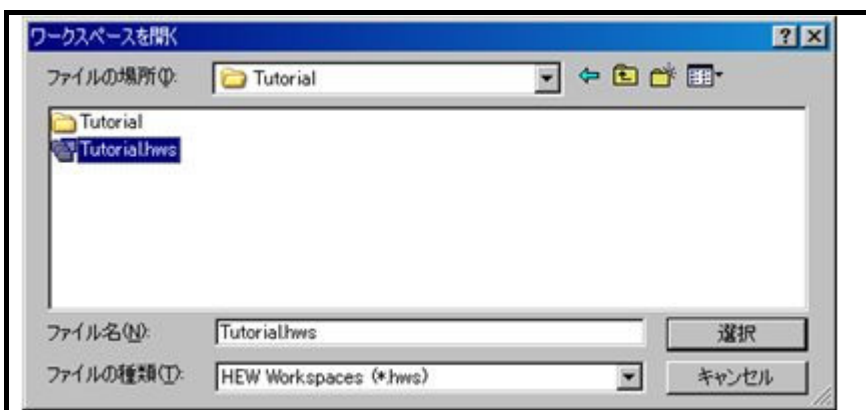


図 6.1 [ワークスペースを開く]ダイアログボックス

### 6.3 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。

## 6.4 チュートリアルプログラムのダウンロード

### 6.4.1 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。画面の表示などは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.x30]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 6.2 チュートリアルプログラムのダウンロード画面



## 6.4.2 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。

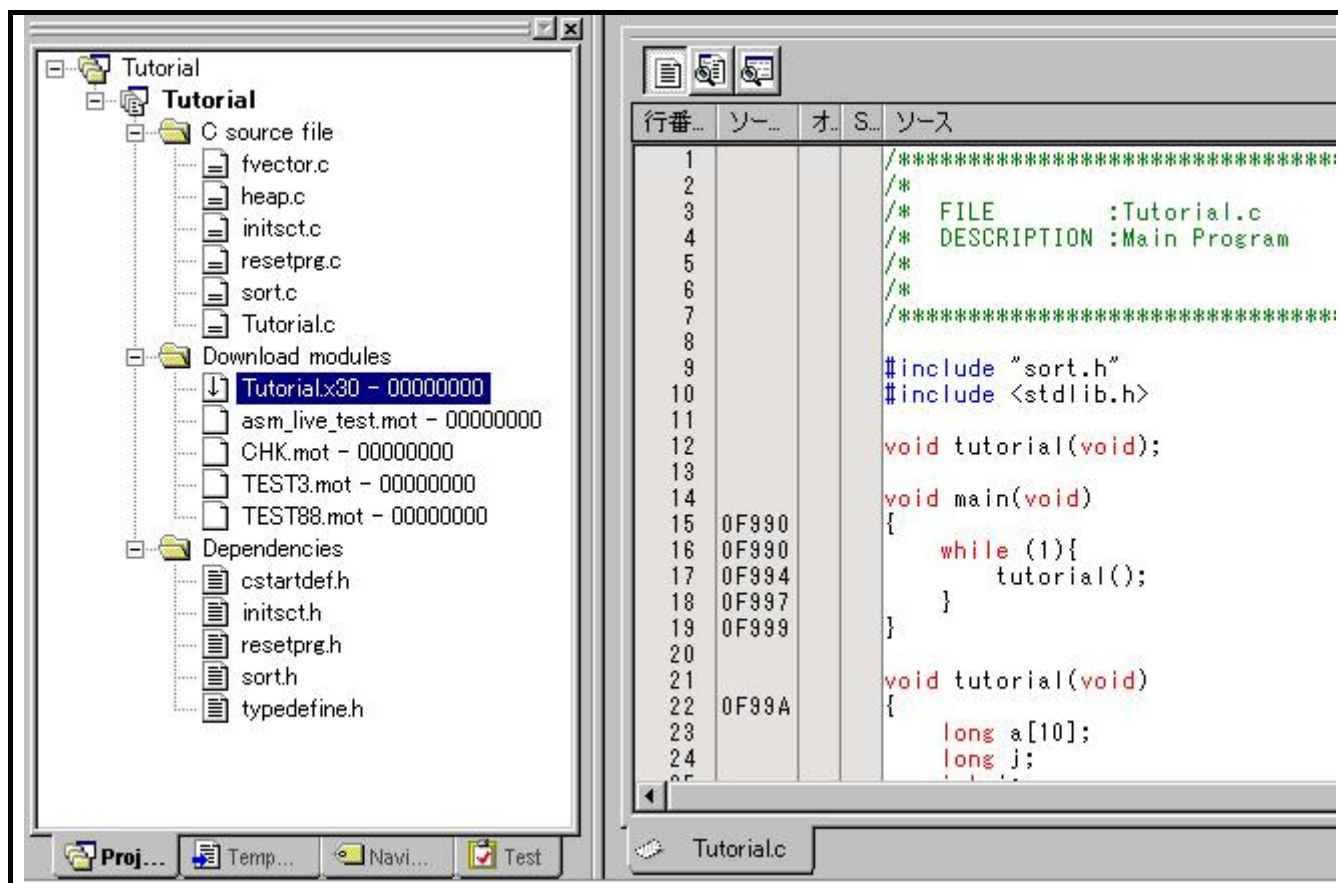


図 6.3 [エディタ]ウィンドウ (ソースプログラムの表示)

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。

## 6.5 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（S/W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。

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。

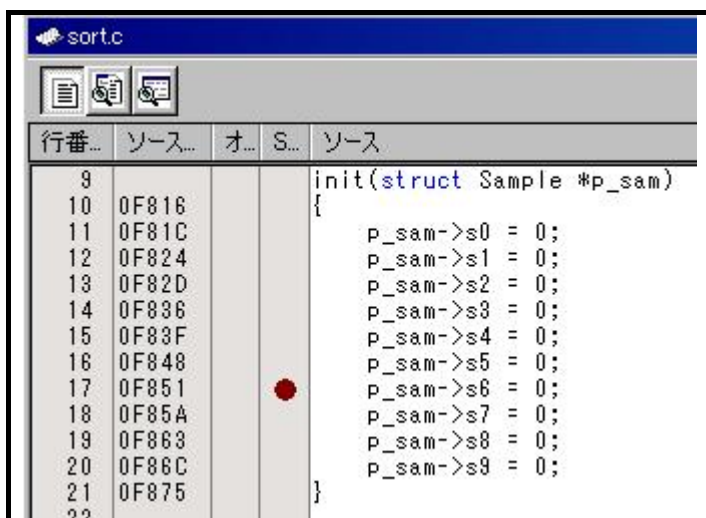



図 6.4 [エディタ]ウィンドウ (S/Wブレークポイントの設定)


## 6.6 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。

### 6.6.1 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。

### 6.6.2 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。実行を開始すると、ステータスバーに「Running」と表示します。

プログラムは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。

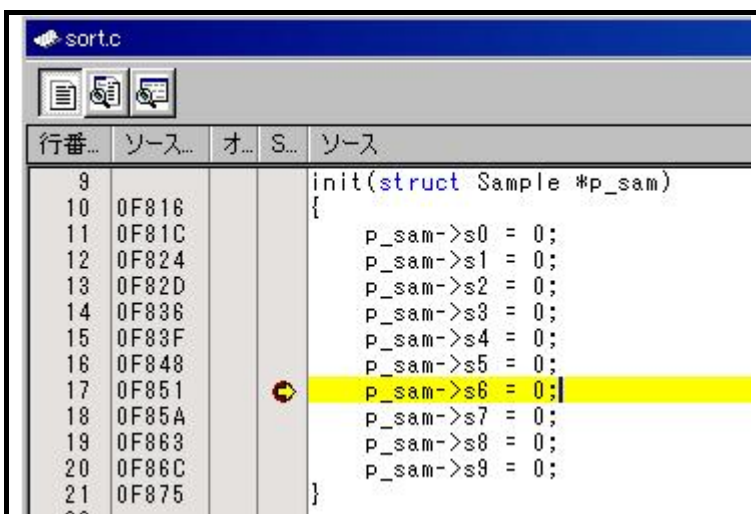



図 6.5 [エディタ]ウィンドウ (ブレーク状態)

#### 【留意事項】

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

<OSインストールドライブ>:\¥Workspace¥Tutorial¥E1E20¥R8C¥Tutorial¥Source

[ステータス]ウィンドウで最後に発生したブレークの要因が確認できます。

[表示→CPU→ステータス]を選択するか、[ステータスの表示]ツールバーボタンをクリックしてください。[ステータス]ウィンドウが表示されますので、[Platform]シートを開いて確認してください。

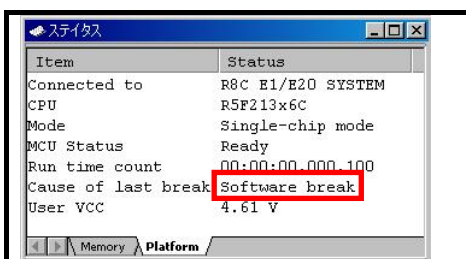


図 6.6 [ステータス]ウィンドウ

## 6.7 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。

### 6.7.1 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。

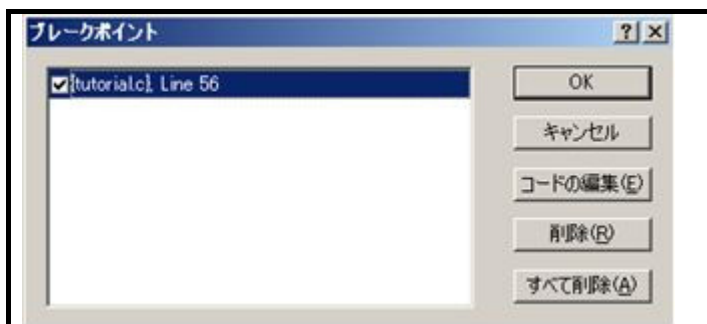



図 6.7 [ブレークポイント]ダイアログボックス

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1】 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。

## 6.8 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。

### (1) レジスタバンクの切替

デフォルトは「レジスタバンク0」が表示されます。[レジスタ]ウィンドウ上で右クリックし、ポップアップメニューを開いてバンク0とバンク1を切り替えることができます。

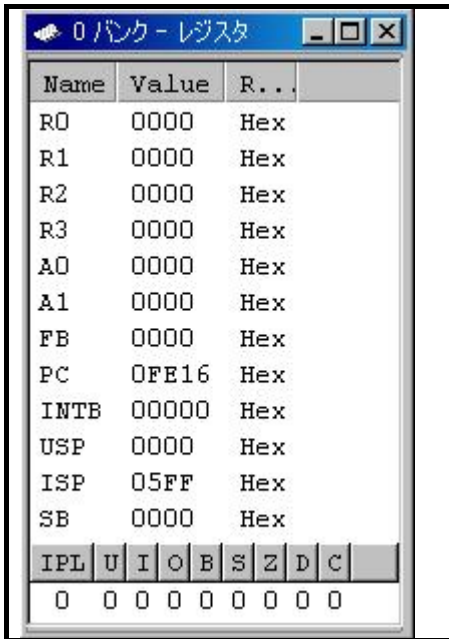


図 6.8 [レジスタ]ウィンドウ

### (2) レジスタ内容の変更

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。




図 6.9 [レジスタ値設定]ダイアログボックス (PC)

**[注1]** CPUレジスタは[レジスタ]ウィンドウおよびスクリプトコマンドでも設定可能ですが、設定値はユーザプログラム実行直前に更新されます。

## 6.9 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。

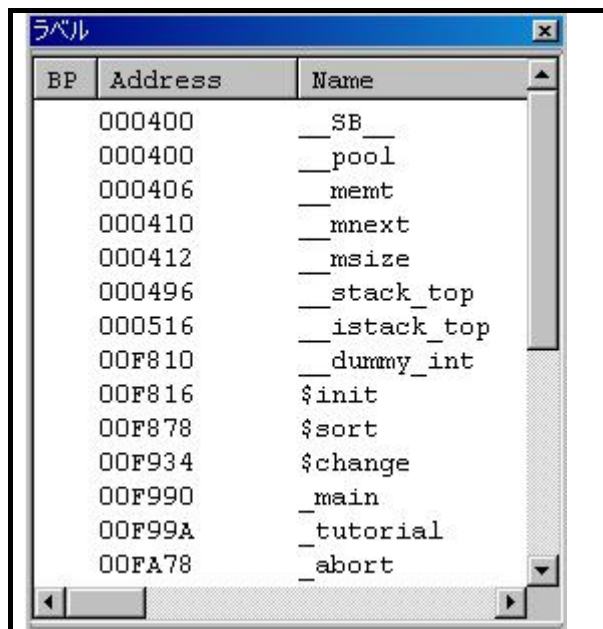



図 6.10 [ラベル]ウィンドウ

6.10 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、ワードサイズで\_mainに対応するメモリ内容を確認します。

[表示 → 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 6.11 [表示開始アドレス]ダイアログボックス

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

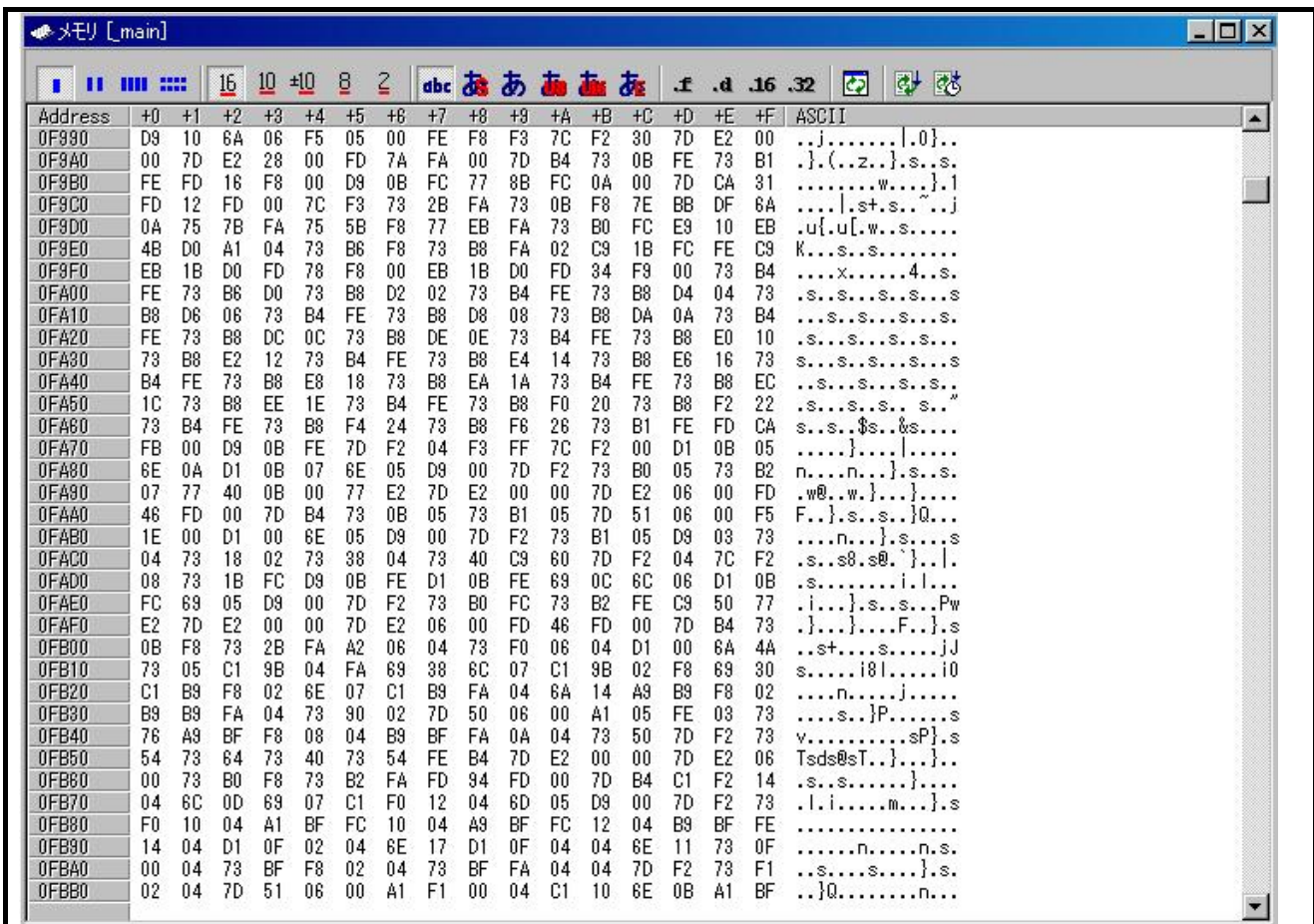


図 6.12 [メモリ]ウィンドウ

### 6.11 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。

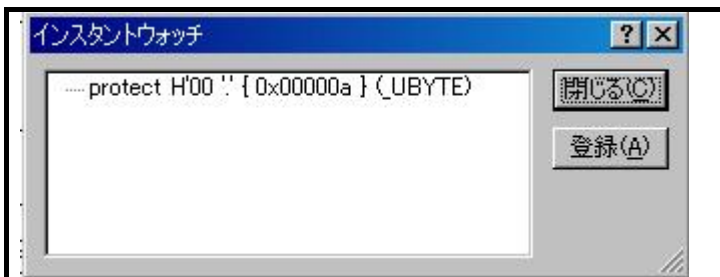


図 6.13 [インスタントウォッチ]ダイアログボックス

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

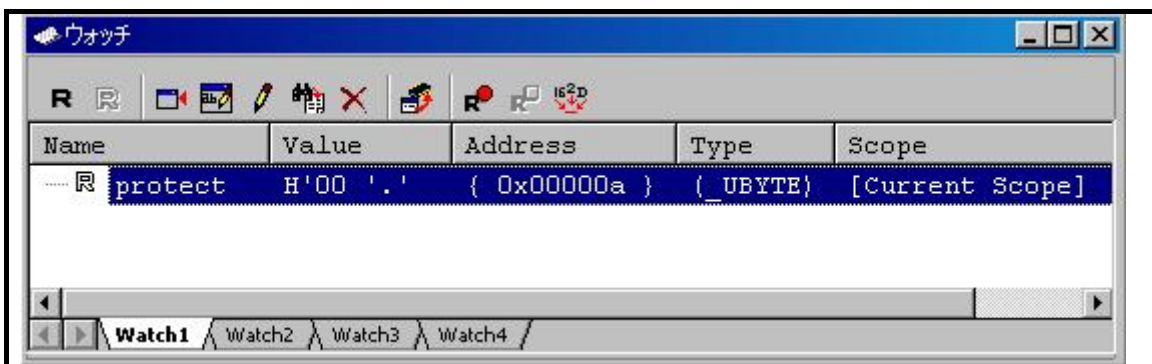


図 6.14 [ウォッチ]ウィンドウ (配列の表示)

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。

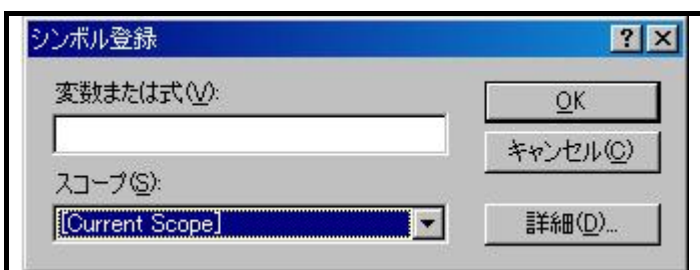


図 6.15 [シンボル登録]ダイアログボックス



[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。

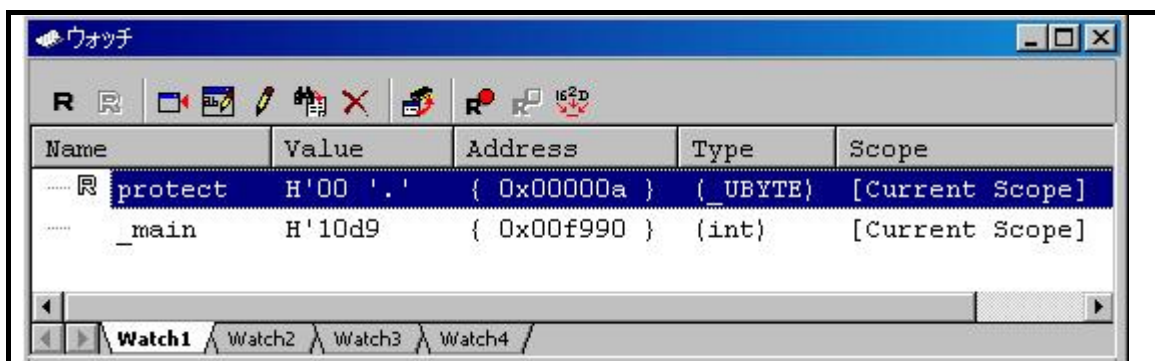


図 6.16 [ウォッチ]ウィンドウ (変数の表示)

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

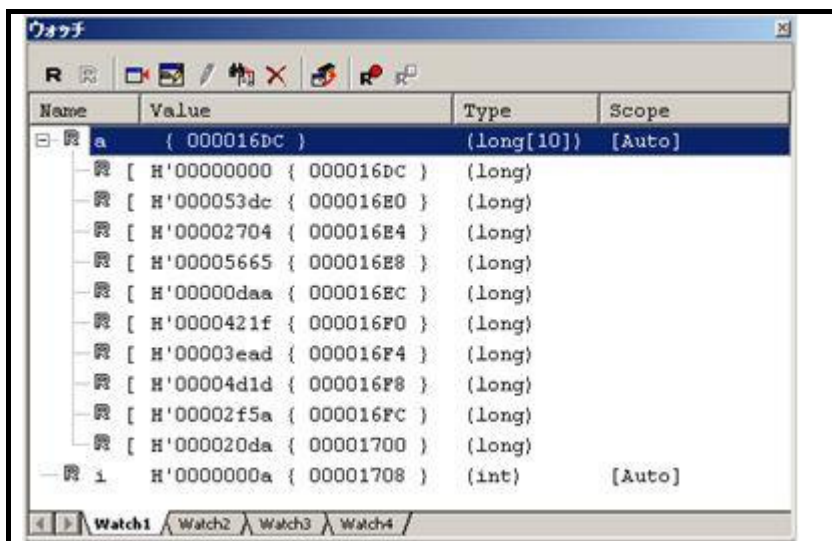



図 6.17 [ウォッチ]ウィンドウ (配列要素の表示)

## 6.12 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。

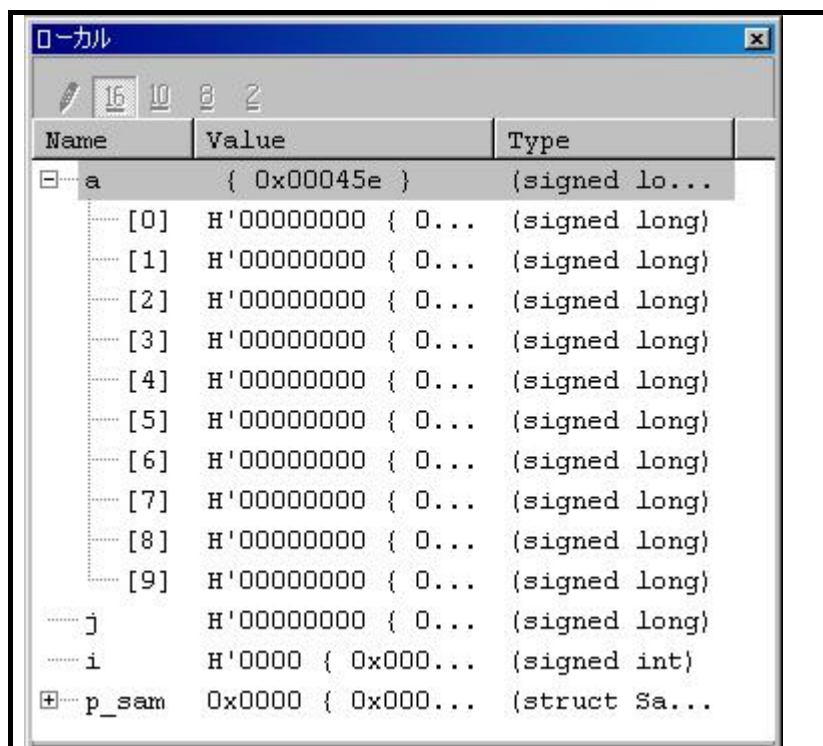


図 6.18 [ローカル]ウィンドウ

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

sort関数実行前と実行後のクラスインスタンス"a"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。

### 6.13 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 6.1 ステップオプション

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

#### 6.13.1 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。

[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。

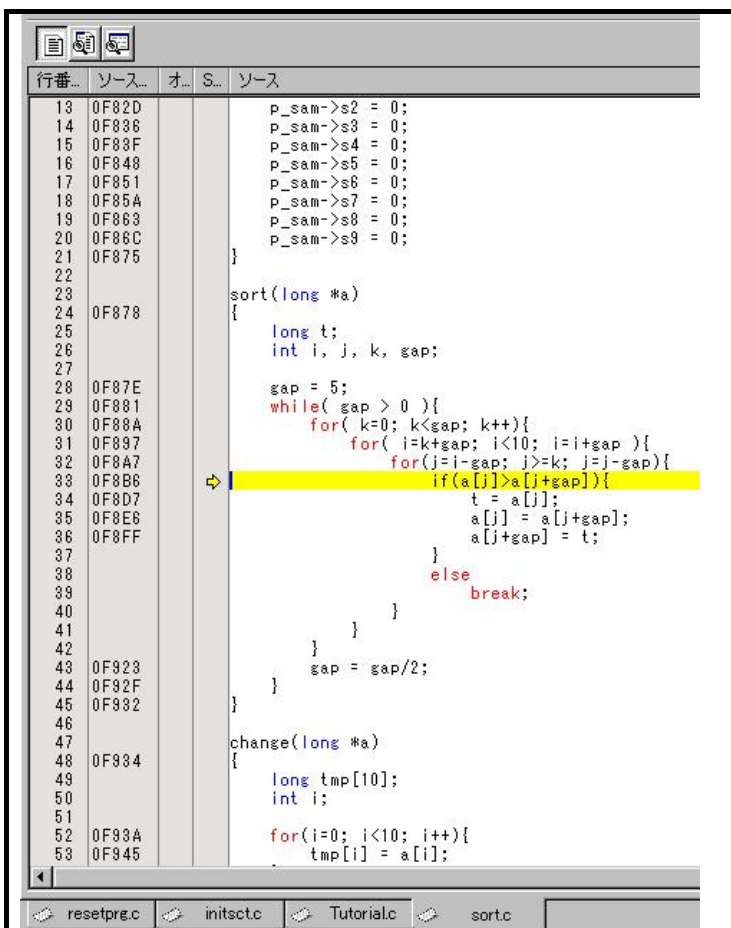



図 6.19 [エディタ]ウィンドウ (ステップイン)

## 6.13.2 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。

[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。

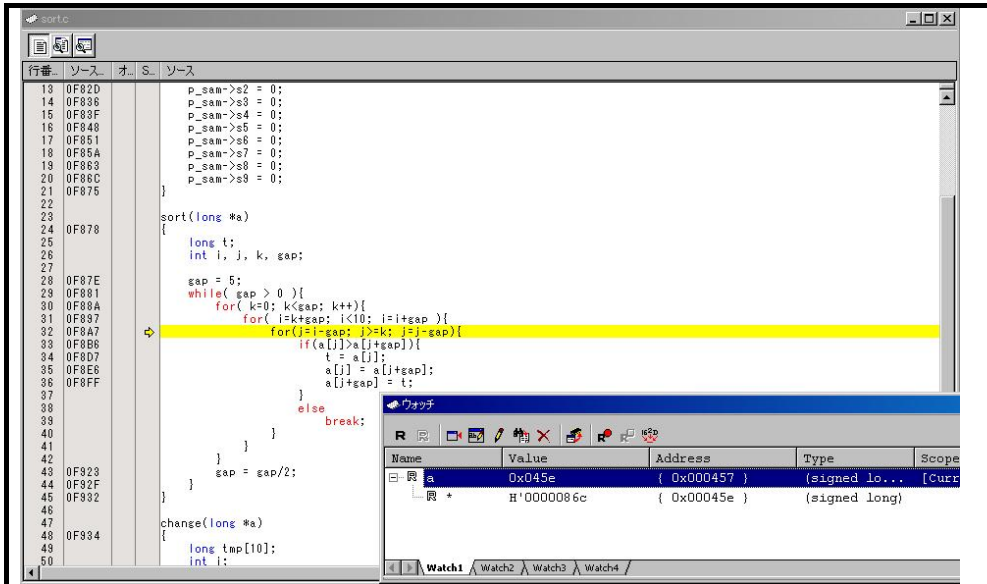



図 6.20 [エディタ]ウィンドウ (ステップアウト)

## 6.13.3 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。  
[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。

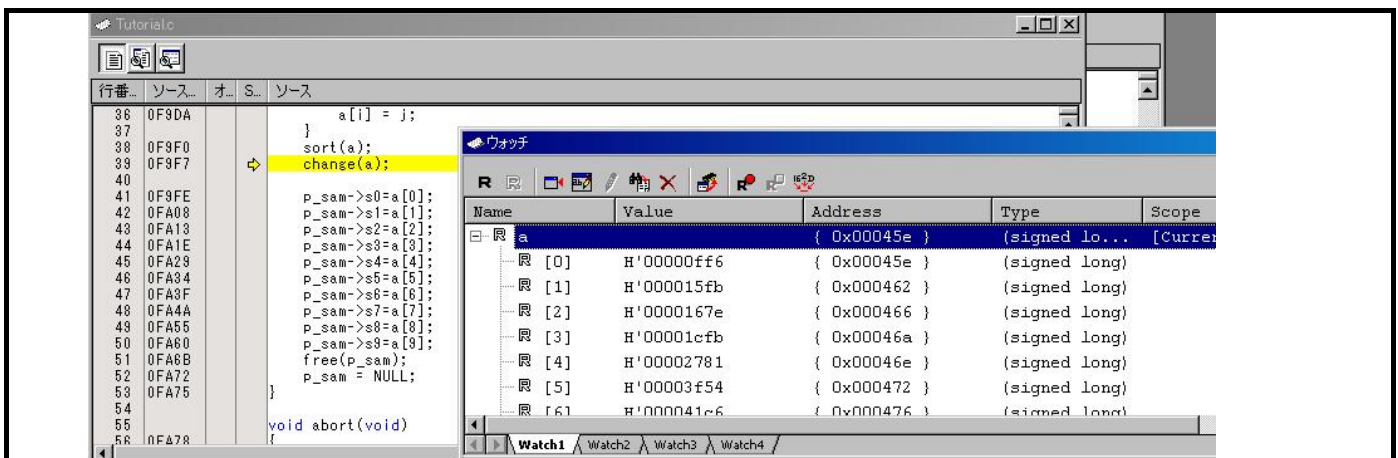




図 6.21 [エディタ]ウィンドウ (ステップオーバ)

## 6.14 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。

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。

## 6.15 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。

### 6.15.1 指定アドレスを実行したときにプログラム実行を停止(実行前PCブレーク)

[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関数のコール箇所に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。sort関数を含む行に ● 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。

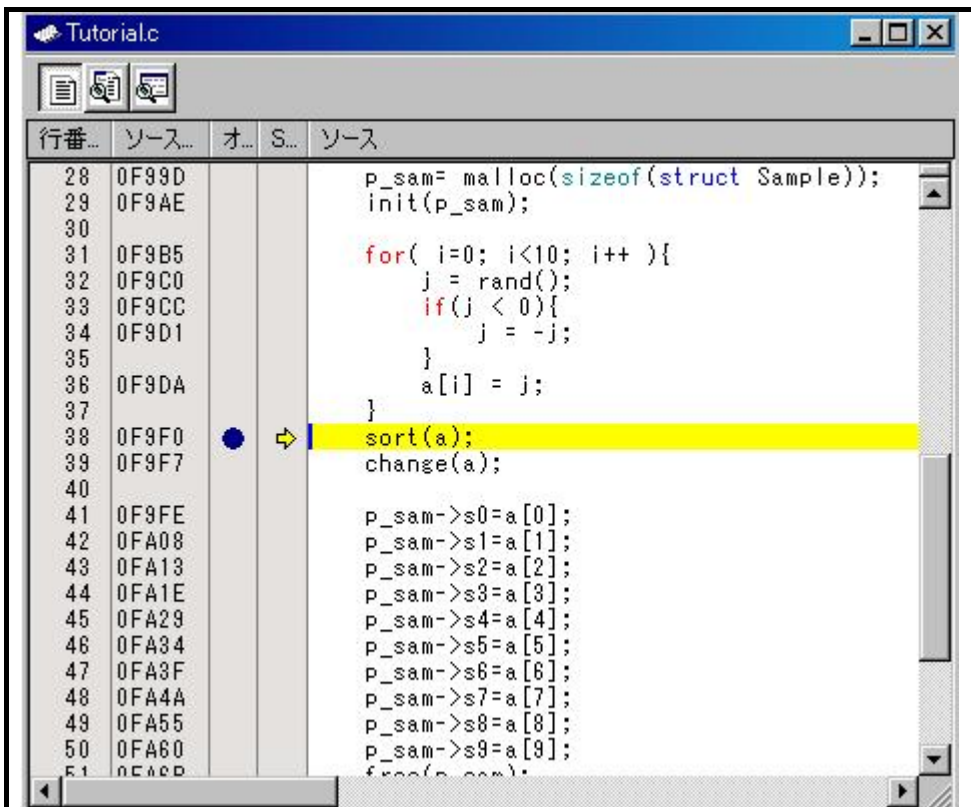


図 6.22 [エディタ]ウィンドウ (オンチップブレークポイントの設定)

## 6.15.2 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 -> オンチップイベント条件設定]を選択して、[オンチップイベント条件設定]ダイアログボックスを表示します。

[オンチップイベント条件設定]ダイアログボックスの[イベント]タブを表示します。

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[イベント]タブ内のEventAの欄にドラッグ&ドロップします。次に、ブレーク条件の条件に「EventA」を選択し、[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。

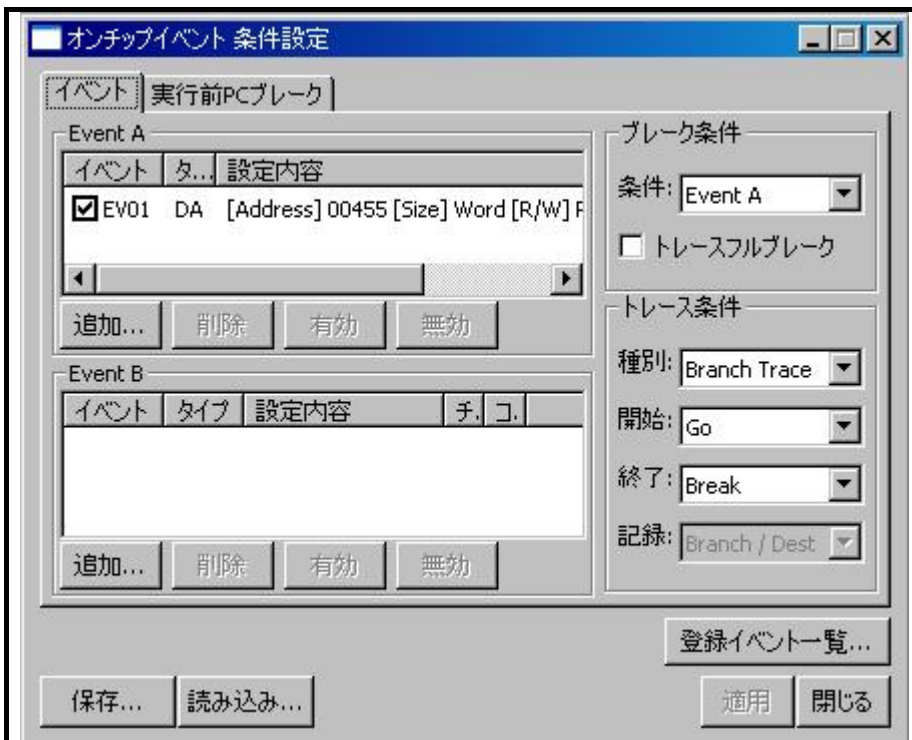



図 6.23 [オンチップイベント条件設定]ダイアログボックス

【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。ローカル変数は指定できません。

## 6.16 トレース機能

E1/E20 エミュレータは、デバイス内蔵のトレース機能を使用できます。  
R8C E1/E20エミュレータデバッガは分岐情報もしくは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る場合があります。

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。

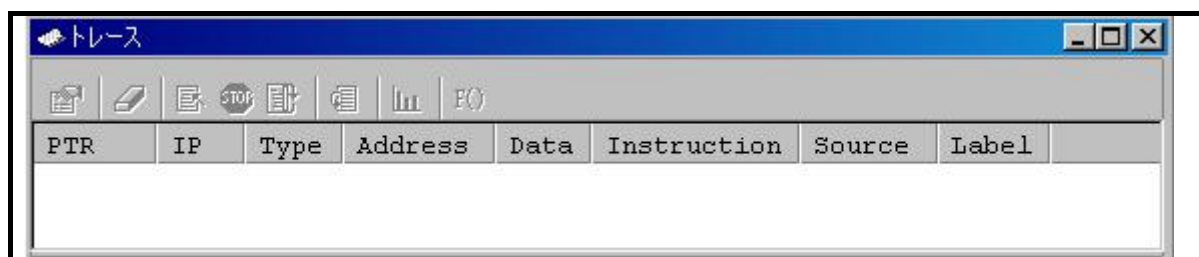


図 6.24 [オンチップイベント条件]ウィンドウ

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。




## 6.16.1 トレース情報の表示

トレース情報は、トレース開始条件発動時点からトレース終了条件発動時点までのトレース情報を取得します。ここでは下記の条件で説明します。

トレース開始条件： ユーザプログラム実行開始  
トレース終了条件： ユーザプログラム実行停止

(1) [表示 -> イベント -> オンチップイベント条件設定]を選択するか、[オンチップイベント条件設定]ボタン

 をクリックしてください。

(2) トレース条件の種別が「Branch Trace」、開始が「Go」、終了が「Break」になっていることを確認して、[閉じる]ボタンをクリックしてください。

なお、[オンチップイベント条件設定]ダイアログボックスで、設定内容変更した場合は[適用]ボタンをクリックした上で、[閉じる]ボタンをクリックしてください。



図 6.25 [トレース条件設定]ダイアログボックス

(3) 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。

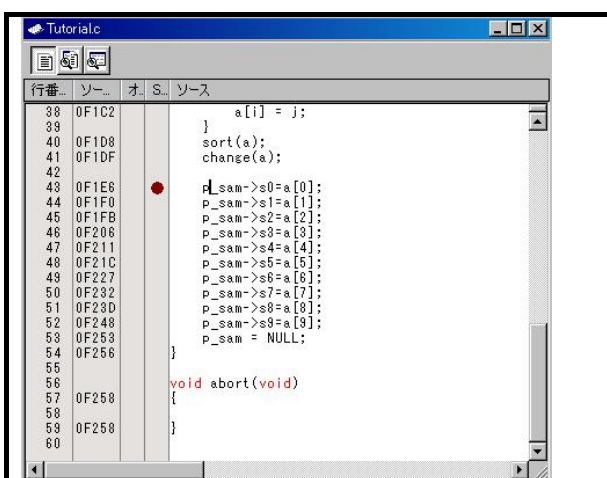


図 6.26 [エディタ]ウィンドウ (tutorial関数内へのS/Wブレーク設定)

(4) [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。

PTR	IP	Type	Address	Data	Instruction	Source	Label
-000007	0007	BRANCH	0000F180		JMP.B 0F159H		
-000006	0006	DESTINATION	0000F159		CMP.W #000AH,-4H[FB]		
-000005	0005	BRANCH	0000F15E		JGE 0F182H		
-000004	0004	DESTINATION	0000F182		EXITD	}	
-000003	0003	BRANCH	0000F182		EXITD	}	
-000002	0002	DESTINATION	0000F1E6		MOV.W -2H[FB],A0	p_sam->s0=a[0];	
-000001	0001	BRANCH	0000F1E6		MOV.W -2H[FB],A0	p_sam->s0=a[0];	
+000000	0000	DESTINATION	0000803B		*** EML ***		

図 6.27 [トレース]ウィンドウ

## 6.17 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。【注1】  
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。[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。

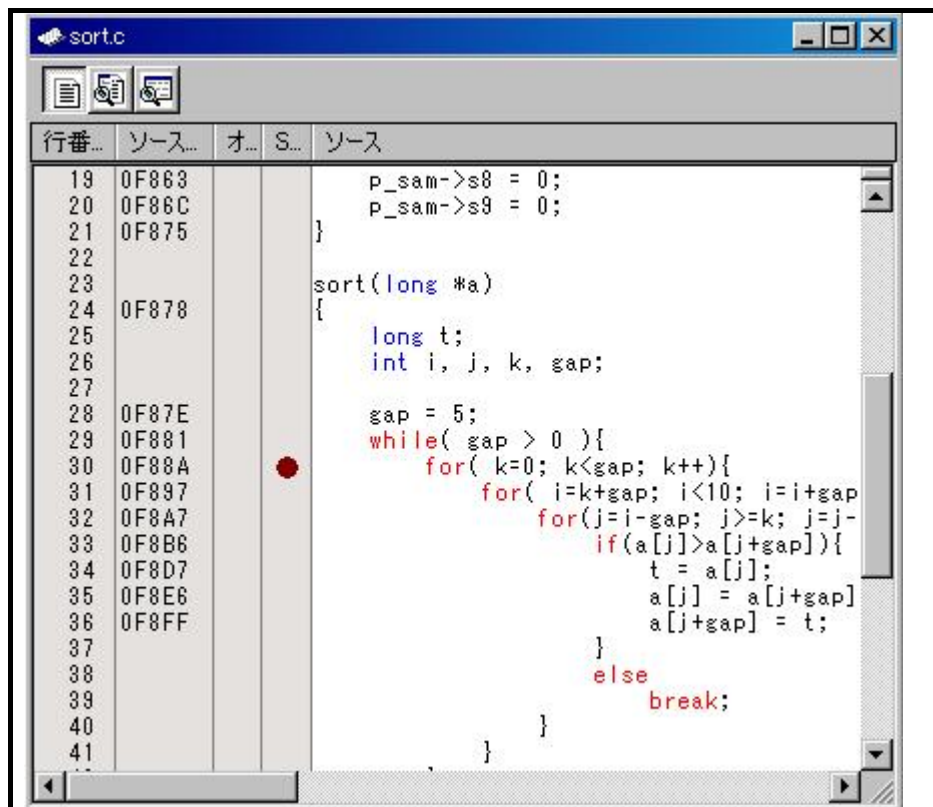


図 6.28 [エディタ]ウィンドウ (S/Wブレークポイントの設定)

【注1】 本機能は、IEEE695形式のデバッグ情報を持ったロードモジュールをロードした場合のみ使用できません。

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。

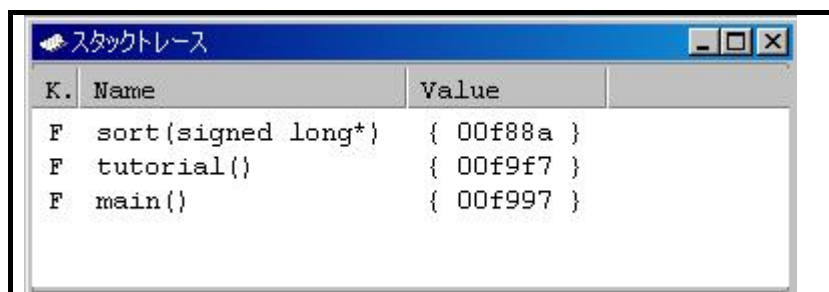


図 6.29 [スタックトレース]ウィンドウ

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。

### 6.18 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。

## 7. E1/E20エミュレータ使用上の注意事項

### 7.1 占有するマイコン資源

#### 7.1.1 E1/E20エミュレータ用プログラム占有領域

E1/E20エミュレータ用プログラムが占有する領域を、表 7.1～表 7.5に示します。

E1/E20エミュレータ用プログラム占有領域の内容は変更しないでください。変更した場合、E1/E20エミュレータでの制御ができなくなりますので、デバッグを切断後に再度接続してください。

表 7.1 E1/E20エミュレータ用プログラム占有領域 (R8C/5xE)

Group	型名	[起動設定]ダイアログボックス表示		ROM容量		E1/E20エミュレータプログラム占有領域		
		MCUグループ	デバイス名	Program ROM	Data Flash	ベクタ領域	ROM領域	
R8C/54E	R5F21546E	R8C/5xEFGH Group	R5F215x6E	32KB	1KB × 4 Block	FFE4h~FFE7h, FFE8h~FFEBh, FFEC~FFEFh, FFF4h~FFF7h, FFFCh~FFFEh	-	
	R5F21547E		R5F215x7E	48KB				
	R5F21548E		R5F215x8E	64KB				
	R5F2154AE		R5F215xAE	96KB				
	R5F2154CE		R5F215xCE	128KB				
R8C/56E	R5F21566E		R5F215x6E	32KB			-	ROM領域の2KB【注1】
	R5F21567E		R5F215x7E	48KB				
	R5F21568E		R5F215x8E	64KB				
	R5F2156AE		R5F215xAE	96KB				
	R5F2156CE		R5F215xCE	128KB				

表 7.2 E1/E20エミュレータ用プログラム占有領域 (R8C/5xF)

Group	型名	[起動設定]ダイアログボックス表示		ROM容量		E1/E20エミュレータプログラム占有領域		
		MCUグループ	デバイス名	Program ROM	Data Flash	ベクタ領域	ROM領域	
R8C/54F	R5F21546F	R8C/5xEFGH Group	R5F215x6F	32KB	-	FFE4h~FFE7h, FFE8h~FFEBh, FFEC~FFEFh, FFF4h~FFF7h, FFFCh~FFFEh	-	
	R5F21547F		R5F215x7F	48KB				
	R5F21548F		R5F215x8F	64KB				
	R5F2154AF		R5F215xAF	96KB				
	R5F2154CF		R5F215xCF	128KB				
R8C/56F	R5F21566F		R5F215x6F	32KB			-	ROM領域の2KB【注1】
	R5F21567F		R5F215x7F	48KB				
	R5F21568F		R5F215x8F	64KB				
	R5F2156AF		R5F215xAF	96KB				
	R5F2156CF		R5F215xCF	128KB				

【注1】 デバッグ起動時に[コンフィグレーションプロパティ]ダイアログが表示されますので、ユーザシステムで使用しない領域を指定してください(詳細は「4. 13. 1 システムタブ」(43ページ)参照)。

表 7.3 E1/E20エミュレータ用プログラム占有領域 (R8C/5xG)

Group	型名	[起動設定]ダイアログボックス表示		ROM容量		E1/E20エミュレータプログラム占有領域			
		MCUグループ	デバイス名	Program ROM	Data Flash	ベクタ領域	ROM領域		
R8C/54G	R5F21546G	R8C/5xEFGH Group	R5F215x6G	32KB	1KB × 4 Block	FFE4h~FFE7h, FFE8h~FFEBh, FFECCh~FFEFh, FFF4h~FFF7h, FFFCh~FFFEh	-		
	R5F21547G		R5F215x7G	48KB					
	R5F21548G		R5F215x8G	64KB					
	R5F2154AG		R5F215xAG	96KB					
	R5F2154CG		R5F215xCG	128KB				ROM領域の2KB【注1】	
R8C/56G	R5F21566G		R5F215x6G	32KB					-
	R5F21567G		R5F215x7G	48KB					
	R5F21568G		R5F215x8G	64KB					
	R5F2156AG		R5F215xAG	96KB					
	R5F2156CG		R5F215xCG	128KB					

表 7.4 E1/E20エミュレータ用プログラム占有領域 (R8C/5xH)

Group	型名	[起動設定]ダイアログボックス表示		ROM容量		E1/E20エミュレータプログラム占有領域			
		MCUグループ	デバイス名	Program ROM	Data Flash	ベクタ領域	ROM領域		
R8C/54H	R5F21546H	R8C/5xEFGH Group	R5F215x6H	32KB	-	FFE4h~FFE7h, FFE8h~FFEBh, FFECCh~FFEFh, FFF4h~FFF7h, FFFCh~FFFEh	-		
	R5F21547H		R5F215x7H	48KB					
	R5F21548H		R5F215x8H	64KB					
	R5F2154AH		R5F215xAH	96KB					
	R5F2154CH		R5F215xCH	128KB				ROM領域の2KB【注1】	
R8C/56H	R5F21566H		R5F215x6H	32KB					-
	R5F21567H		R5F215x7H	48KB					
	R5F21568H		R5F215x8H	64KB					
	R5F2156AH		R5F215xAH	96KB					
	R5F2156CH		R5F215xCH	128KB					

表 7.5 E1/E20エミュレータ用プログラム占有領域 (R8C/3xT-A)

Group	型名	[起動設定]ダイアログボックス表示		ROM容量		E1/E20エミュレータプログラム占有領域			
		MCUグループ	デバイス名	Program ROM	Data Flash	ベクタ領域	ROM領域		
R8C/36T-A	R5F21366S	R8C/3xT-A Group	R5F213x6S	32KB	1KB × 4 Block	FFE4h~FFE7h, FFE8h~FFEBh, FFECCh~FFEFh, FFF4h~FFF7h, FFFCh~FFFEh	-		
	R5F21367S		R5F213x7S	48KB					
	R5F21368S		R5F213x8S	64KB					
	R5F2136AS		R5F213xAS	96KB					
	R5F2136CS		R5F213xCS	128KB				ROM領域の2KB【注1】	
R8C/38T-A	R5F21386S		R5F213x6S	32KB					-
	R5F21387S		R5F213x7S	48KB					
	R5F21388S		R5F213x8S	64KB					
	R5F2138AS		R5F213xAS	96KB					
	R5F2138CS		R5F213xCS	128KB					

【注1】 デバッガ起動時に[コンフィグレーションプロパティ]ダイアログが表示されますので、ユーザシステムで使用しない領域を指定してください(詳細は「4. 13. 1 システムタブ」(43ページ)参照)。

## 7.1.2 E1/E20エミュレータ用占有端子

E1/E20エミュレータとの接続時、以下の端子を使用してマイコンを制御します。

- RESET#端子、MODE端子

## 7.1.3 割り込み機能(使用不可)

BRK命令割り込み、アドレス一致割り込み、シングルステップ割り込みおよびアドレスブレイク割り込みは、E1/E20エミュレータ用プログラムで使用します。したがって、ユーザプログラムでこれらの割り込みを使用しないでください。また、E1/E20エミュレータはこれら割り込みのベクタ値をエミュレータが使用する値に書き換えますので、ユーザプログラム中に割り込みベクタ値が書かれていてもかまいません。また、これらベクタアドレスはメモリウィンドウから書換えることができません。

## 7.1.4 E1/E20エミュレータ用使用スタック領域

E1/E20エミュレータでは、ユーザプログラム停止時にスタックポインタ (ISP) を最大8バイト分使用します。したがって、スタックには8バイト分の余裕を確保してください。

## 7.1.5 E1/E20エミュレータ使用SFR

表 7.6に示すSFRはユーザプログラムでご使用になれますが、E1/E20エミュレータ用プログラムでも使用します。

- メモリウィンドウなど、ユーザプログラム以外で値を変更しないでください。  
またこれらのSFRの内容を参照した場合、E1/E20エミュレータ用プログラムで設定した値を読み出します。

表 7.6 E1/E20エミュレータ用プログラムが使用するSFR (1)

番地	レジスタ	シンボル	該当ビット
0007h	プロテクトレジスタ	PRCR	ビット0
0012h	高速オンチップオシレータ制御レジスタ0	FRA0	ビット0【注1】
0280h	DTC起動制御レジスタ	DTCTL	ビット0

表 7.7に示すSFRは、E1/E20エミュレータ用プログラムで使用しユーザプログラムではご使用になれません。これらのSFRは値を変更しないでください。変更した場合、E1/E20エミュレータでの制御ができなくなります。

表 7.7 E1/E20エミュレータ用プログラムが使用するSFR (2)

番地	レジスタ	シンボル	該当ビット
0260h~0261h	アドレス一致割り込みアドレス0Lレジスタ	AIADROL	全ビット
0262h	アドレス一致割り込みアドレス0Hレジスタ	AIADROH	全ビット
0263h	アドレス一致割り込み許可0レジスタ	AIENO	全ビット
0264h~0265h	アドレス一致割り込みアドレス1Lレジスタ	AIADR1L	全ビット
0266h	アドレス一致割り込みアドレス1Hレジスタ	AIADR1H	全ビット
0267h	アドレス一致割り込み許可1レジスタ	AIEN1	全ビット

【注1】 E1/E20エミュレータでのデバッグ時、高速オンチップオシレータ許可ビットの設定 (FRA00:高速オンチップオシレータ停止) は可能ですが高速オンチップオシレータは停止しません。

高速オンチップオシレータを停止させての消費電力低減等の確認は、ユーザプログラムのみをMCUに書き込み、E1/E20エミュレータを外した状態の最終形態で確認ください。

なお、デバッグ起動時の[起動設定]ダイアログボックスの動作モードで「フラッシュライトモード」を選択し、ユーザプログラムのみをMCUに書き込み、デバッグ終了後と同時にユーザプログラムを実行させることで同等の確認が可能です。

なお高速オンチップオシレータの周波数変更や周波数調整用のレジスタは、必ずユーザプログラムで変更してください。メモリウィンドウ等で変更した場合E1/E20エミュレータが制御不能となります。

## 7.1.6 オプション機能選択領域

E1/E20エミュレータではオプション機能選択レジスタ (OFS:0FFFFh)のビット0、ビット7をいずれも1bに設定します。このアドレスはメモリウィンドウ等で書き換え可能で、書き換えた値も参照可能ですが、対象ビット(ビット0)に関しては変更値無効です。

- b0:ウォッチドッグタイマ起動選択ビット                    1:リセット後、ウォッチドッグタイマは停止状態
- b7:リセット後カウントソース保護モード選択ビット   1:リセット後、カウントソース保護モード無効

また、オプション機能選択レジスタ2 (OFS2:0FFDBh)についても、E1/E20エミュレータが下位4ビットを1111bに設定します。

- b1, b0: ウォッチドッグタイマアンダーフロー周期設定ビット   11 : 3FFFh
- b3, b2: ウォッチドッグタイマリフレッシュ受付周期設定ビット   11 : 100%

【注1】 ユーザプログラム停止中、E1/E20エミュレータ用プログラムがウォッチドッグタイマのリフレッシュを行います。ウォッチドッグタイマを使用したユーザプログラムでは、ユーザプログラム停止中にE1/E20エミュレータ用プログラムによりウォッチドッグタイマのリフレッシュが発生し、実際の動作タイミングと異なるので注意してください。

ウォッチドッグタイマのリフレッシュは、ユーザプログラム実行中は実施していませんのでご注意ください。

【注2】 E1/E20エミュレータでのデバッグ時、カウントソース保護モードは使用できません。



## 7.1.7 E1/E20エミュレータでのレジスタ初期化

E1/E20エミュレータは、起動時にCPUレジスタを初期化していますので注意してください。表 7.8にE1/E20エミュレータで初期化しているレジスタ名と設定値を示します。

表 7.8 E1/E20エミュレータでのレジスタ初期値

状態	レジスタ名	初期値
E1/E20エミュレータ 起動時	PC	ベクタアドレステーブル中のリセットベクタ値
	R0~R3 (バンク0, 1)	0000h
	A0, A1 (バンク0, 1)	0000h
	FB (バンク0, 1)	0000h
	INTB	00000h
	USP	0000h
	ISP	05FFh (MCU仕様との差異)
	SB	0000h
	FLG	0000h

## 7.1.8 RAM初期化

E1/E20エミュレータは、起動時にMCU内部RAM領域の一部(00400h~004FFh)を"00h"で初期化していますので注意してください。

## 7.1.9 マイコンの予約領域

MCUのハードウェアマニュアルに定義されていないアドレスは、予約領域です。予約領域の内容は変更しないでください。変更した場合、E1/E20エミュレータでの制御ができなくなります。

- この領域は、メモリウィンドウで参照時、"不定値"が表示されます。
- この領域は、メモリウィンドウの検索、比較、コピー機能が正常に動作しません。

## 7.1.10 ユーザプログラム停止中のDTC

ユーザプログラム停止時はDTCのデータ転送を禁止にしています。

但し、DTCのデータ転送要求自体は停止しませんので、ユーザプログラム停止中に発生したDTC転送要求はユーザプログラム再開時に実行されます。

## 7.1.11 2.7V未満でのデバッグに関する注意事項

対象マイコンの仕様はフラッシュメモリへの書き込み/消去時の最低電圧は2.7[V]です。

起動時のMCU接続を含む以下の操作はフラッシュの書き換えを伴いますので、マイコンへの供給電圧が2.7V未満では使用できません。

- ユーザプログラムのダウンロード
- S/Wブレークの設定および解除(アドレス一致、オンチップブレークは設定/解除可能です)
- メモリウィンドウ等でのフラッシュメモリ領域の値の変更

## 7.1.12 消費電力低減のデバッグ

低消費電流リードモードのデバッグをする際は、低消費電流リードモード解除後に実行される処理部にブレークポイントを設定するなどして、ブレークポイントで止まるまで画面の操作をしないでください。

またフラッシュメモリ停止のデバッグをする際は、フラッシュメモリ停止解除後に実行される処理部にブレークポイントを設定するなどして、ブレークポイントで止まるまで画面の操作をしないでください。

## 7.2 リセット

### (1) リセット機能

パワーオンリセットおよび電源監視0リセットは使用できません。実行した場合、E1/E20エミュレータが制御不能となります。

また、リセット端子が“L”状態のままユーザプログラムを停止させないでください。タイムアウトエラーが発生します。

### (2) リセットベクタアドレス

E1/E20エミュレータでのデバッグ時、リセットベクタアドレスはE1/E20エミュレータ用プログラムが使用しています。ユーザプログラム停止中は、ダウンロードしたプログラムもしくはメモリウィンドウでの設定値が参照可能ですが、ユーザプログラム実行中は、E1/E20エミュレータ用プログラムが設定した値が表示されます。なお、ユーザプログラム停止中にリセットベクタアドレスの値を設定した場合、E1/E20エミュレータ内のキャッシュのみ更新されます。

### (3) リセット後の動作

ユーザプログラム実行中にマイコンをリセットした場合、表 7.9に示すようにいったんE1/E20エミュレータ用プログラムに移行した後、ユーザプログラムが再開します。

このためリセット後、ユーザプログラム開始までのタイムラグが発生します。なお表に示す通り、端子リセットとその他のリセットでは動作が異なります。【注1】

表 7.9 ユーザプログラム実行中のリセット再開動作

リセットの名称	リセット発生時の動作
端子リセット	(1) E1/E20エミュレータで端子リセット検出
	(2) E1/E20エミュレータは、MCUと再接続する。
	(3) MCUがリセットされ、E1/E20エミュレータ用プログラムが動作する。
	(4) E1/E20エミュレータ用プログラムは、予めユーザプログラムのリセットベクタアドレスを保持しており、このアドレスへジャンプする。
ウォッチドッグタイマリセット またはS/Wリセット【注2】	(1) MCUがリセットされ、E1/E20エミュレータ用プログラムが動作する。
	(2) E1/E20エミュレータ用プログラムは、予めユーザプログラムのリセットベクタアドレスを保持しており、このアドレスへジャンプする。

【注1】 アドレス一致割り込みはユーザプログラム実行中に設定/削除が可能ですが、ユーザプログラム実行中のアドレス一致ブレイクの設定はリセットによって初期化されます。

【注2】 ユーザプログラム実行中のウォッチドッグタイマリセットまたはS/Wリセットでは、SFRレジスタが初期化されるためアドレス一致ブレイク (ch0~3) は無効 (ブレイクしない) となります。アドレス一致ブレイクのch番号は、[オンチップイベント条件設定] ダイアログボックスの [実行前PCブレイク] タブで、確認することが可能です。

### (4) ウォッチドッグタイマリセット

ウォッチドッグタイマは、ユーザプログラム停止中もカウント停止しないためE1/E20エミュレータ用プログラム内でリフレッシュしています。このためユーザプログラムブレイク直後にアンダーフロー動作が発生する等、リフレッシュ直前にウォッチドッグタイマリセットが発生した場合は、E1/E20エミュレータが制御不能となる場合があります。

## (5) リセットに関する制限事項

ユーザプログラム停止中に端子リセットしないでください。E1/E20エミュレータでの制御が出来なくなる場合があります。なお、ユーザプログラム停止中に端子リセットが入った場合、CPUレジスタの内容は初期化されず、ユーザプログラム停止時直後の値もしくはCPUレジスタウィンドウでの設定値が優先されます。

## (6) リセットの競合

「端子リセットやウォッチドッグタイマリセットなどによるリセット」と「E1/E20エミュレータからのマイコンに対する操作(メモリウィンドウでのメモリアクセス等)が競合した場合、E1/E20エミュレータが制御不能となる場合があります。この場合、タイムアウトエラー等のエラーが表示されます。

またこの時、システムリセットを促すメッセージボックスが表示され、「はい」をクリックするとエミュレータを初期化しユーザプログラムは停止します。「いいえ」をクリックするとエミュレータは初期化せずユーザプログラムも停止しません。なお、「はい」または「いいえ」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。

**【注1】** また、メモリウィンドウやウォッチウィンドウの自動更新機能を有効にした状態では、マイコンをハードウェアリセットしないでください。MCUとの通信が中断され、E1/E20エミュレータが制御不能となる場合があります。

### 7.3 内部ROM領域(フラッシュメモリ)

#### 7.3.1 内部ROM領域の変更

エミュレータデバッガからダウンロード以外で、内部ROMの内容を変更する場合、変更内容はE1/E20エミュレータ内のキャッシュのみ更新し、マイコンへはユーザプログラム実行直前に反映されます。

#### 7.3.2 CPU書き換えモードのデバッグに関する注意事項 【注1】

##### (1) CPU書き換えモードでの書き換え不可の領域

下記領域を含む内部ROM領域に対して、CPU書き換えをしないでください。書き換えた場合、E1/E20エミュレータでの制御ができなくなります。

- 固定割込みベクタ領域
- E1/E20エミュレータ用プログラム(デバッグモニタ)を含む領域

##### (2) CPU書き換えモード中の動作

- CPU書き換えモードのデバッグでは、CPU書き換えモード有効状態およびイレーズサスペンド状態の間ユーザプログラムを停止させないでください。また、CPU書き換えモード有効状態およびイレーズサスペンド状態になる命令をステップ実行させないでください。  
停止させた場合、E1/E20エミュレータでの制御ができなくなる場合があります。  
また、予めウォッチウィンドウでは自動更新を無効化にし、メモリウィンドウでは表示固定にするなど、ユーザプログラム実行中のメモリアクセスが発生しないようにしてください。
- CPU書き換えモードで内部ROMを書き換え後の内部ROM領域は、CPU書き換えモード無効化後にユーザプログラムを停止させ、メモリウィンドウなどで参照してください。  
また、データフラッシュ領域に対してCPU書き換えを行い、かつイレーズ処理がサスペンドされる可能性がある場合、S/Wブレークを使用しないでください。
- プログラム領域のフラッシュメモリを書き換える場合は、エミュレータデバッガ起動時に表示される[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、[内蔵フラッシュフラッシュメモリの書換えを伴うデバッグ(F)]の設定にチェックしてください。【注2】

【注1】 CPU書き換えモードのデバッグでメモリ内容の参照や変更機能は使用可能ですが、下記の場合はこれらの機能を使用しないでください。

- ・ 連続書き込みが必要なレジスタへの書き込み命令実行中  
(例:FMR13ビット)

命令の途中でメモリ内容の参照変更処理が入るため、MCUが連続的な書き込みと判定されません。

【注2】 [内蔵フラッシュフラッシュメモリの書換えを伴うデバッグ(F)]の設定にチェックが入っている状態では、[メモリ]ウィンドウ等で内部ROM領域を書き換えしないでください。

また、この状態ではS/Wブレークは設定できません。

### 7.3.3 E1/E20エミュレータからのフラッシュメモリ書き換え中の注意事項

E1/E20エミュレータで内部ROM(フラッシュメモリ)書き換え中に、マイコンへのリセットおよびデバッグ操作をしないでください。

フラッシュメモリ書き換えは、High-performance Embedded Workshopのアウトプットウィンドウ上で、"Flash memory writing ... "から"Flash memory writing ... OK"の間に実施されます。

フラッシュメモリ書き換え中のマイコンへのリセットおよびデバッグ操作は、ユーザプログラムまたはE1/E20エミュレータ用プログラムが破壊される可能性があります。

フラッシュメモリ書き換えが発生するケースは以下の通りです。

- ユーザプログラムダウンロード時
- 内部ROM領域にS/Wブレークを設定し、ユーザプログラム実行操作した後
- 内部ROM領域に設定したS/Wブレークを解除し、ユーザプログラム実行操作した後
- [メモリ]ウィンドウもしくはコマンドライン等で内部ROM領域の値を書き換え、ユーザプログラム実行操作した後

### 7.3.4 ユーザプログラム実行中のフラッシュメモリ

ユーザプログラム実行中に、ユーザプログラム以外(メモリウィンドウ等)で内部ROM領域の変更はできません。

### 7.3.5 デバッグに使用したマイコン

デバッグ中はE1/E20エミュレータによりフラッシュメモリの書き換えを頻繁に行います。したがって、デバッグで使用したマイコンを製品に使用しないでください。

また、デバッグ中のマイコンにはE1/E20エミュレータ用プログラムが書き込まれますので、デバッグで使用したマイコンのフラッシュメモリの内容を保存し、製品用ROMデータとして使用しないでください。

## 7.3.6 フラッシュメモリのIDコード

フラッシュメモリを第三者に読み出されないようにするためのマイコンの機能です。

マイコンのIDコード領域に書き込まれたIDコード(表 7.10) とエミュレータデバッガ起動時に表示される[IDコード確認]ダイアログ(図 7.1)で入力したIDコードが一致しなければデバッガを起動することができません。ただし、フラッシュメモリに書かれているIDコードがFFh, FFh, FFh, FFh, FFh, FFh, FFhの場合は、IDコードが設定されていないと見なし、自動的に認証が行われますので、[IDコード確認]ダイアログは表示されません。IDコード領域へ書き込む値はモードにより下記の通り異なります。

- [フラッシュライターモード] 【注1】 : ユーザプログラムの内容
- [デバッグモード] 【注2】 : FFh, FFh, FFh, FFh, FFh, FFh, FFh  
(ダウンロードするユーザプログラム内容に関係無)

表 7.10 IDコード格納領域

番地	内容
FFDFh	IDコード 1バイト目
FFE3h	IDコード 2バイト目
FFEBh	IDコード 3バイト目
FFEFh	IDコード 4バイト目
FFF3h	IDコード 5バイト目
FFF7h	IDコード 6バイト目
FFFBh	IDコード 7バイト目

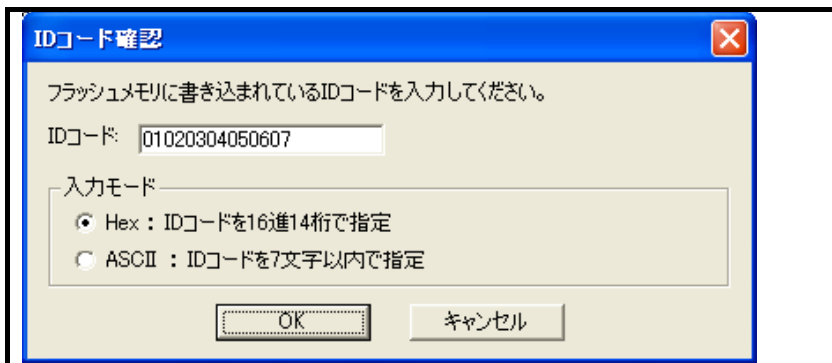


図 7.1 [ IDコード確認 ] ダイアログボックス

## 【注1】 [フラッシュライターモード]での注意事項

- lmc30の -IDオプションでIDコードを設定している場合は、MOTファイルまたはHEXファイルをダウンロードしてください。X30ファイルをダウンロードした場合、IDコードは反映されません。
- X30ファイルをダウンロードする場合は、アセンブラ指示命令“.BYTE”などを使用してIDコードを記述してください。
- また、アセンブラ指示命令“.ID”で指定したIDコードが出力されるファイルは、アセンブラのバージョンによって異なります。詳細は、ご使用のアセンブラのユーザーズマニュアルをご覧ください。

【注2】 マイコンのIDコード領域に書き込まれたIDコードとエミュレータデバッガ起動時に表示される[IDコード確認]ダイアログで入力したIDコードが一致した場合、E1/E20エミュレータはIDコード領域にFFh, FFh, FFh, FFh, FFh, FFh, FFhを書き込みます。このため次回起動時には、[IDコード確認]ダイアログは表示されません。

## 7.4 電源供給に関して

### (1) 消費電流

E1エミュレータからユーザシステムに電源を供給しない場合、E1エミュレータはユーザシステム電源を数mA～十数mA消費します。これは、ユーザシステム電源電圧に通信信号レベルを合わせるため、ユーザシステム電源で74LVC125、74LVC8T245を駆動しているためです。【注1】

### (2) E1エミュレータ電源供給機能に関する注意事項【注1】

E1エミュレータを使用して量産工程などでの信頼性を要求する書き込みを行う場合は、E1エミュレータ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。E1エミュレータ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。

【注1】 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。

## 7.5 ユーザプログラム停止中の動作

### (1) ユーザプログラム停止中の動作クロック

ユーザプログラム停止中のCPUクロックは、マイコン内蔵高速オンチップオシレータに変更しています。

なお、周辺機能は、ユーザプログラムで設定したクロックで動作します。

またこの時の高速オンチップオシレータの周波数は、エミュレータデバッグ起動時の動作電源電圧の電圧を元に設定しています。【注2】

### (2) ユーザプログラム停止中の周辺I/O

ユーザプログラム停止中、E1/E20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。しかし、周辺I/Oは動作し続けているため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。

例えば、タイマを動作させた後にブレークでユーザプログラムを停止させたとき、タイマはカウントし続けますが、タイマ割り込みは受け付けられません。

但し、「4.13.1 システムタブ」(43ページ)で「ユーザプログラム停止中の全タイマのカウントを停止する」にチェックした場合は除きます。

【注2】 周波数は2.7Vを境に切り変えています。このため2.7V辺りの電源電圧でご使用されている場合はE1/E20エミュレータで検出する電圧のレベルによって周波数が変わる可能性がありますのでご注意ください。

なお、E1/E20エミュレータが検出した電圧値は、エミュレータデバッグ起動時の接続中ダイアログもしくは[ステータス]ウィンドウに表示されます。

## 7.6 ユーザプログラム実行中のメモリアクセス

E1/E20エミュレータとMCUは、高速オンチップオシレータの周波数を通信用クロックとして使用します。このため、高速オンチップオシレータの周波数変更直後の、[メモリ]ウィンドウ等でのメモリアクセスは正しくアクセスできない場合があります。

## 7.7 プログラムの最終評価

量産前には必ず、E1/E20エミュレータが接続されていない状態で最終評価を必ず実施してください。

## 7.8 デバッグ機能

### 7.8.1 ステップ実行機能

#### (1) ステップ実行中の制限事項

ステップ実行中は、S/Wブレークおよびオンチップブレークは無効です。

#### (2) ステップアウト実行

サブルーチン復帰命令以外の命令 (JMP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できません。

#### (3) メモリ自動更新機能に関する注意事項

メモリウィンドウやウォッチウィンドウの自動更新機能を有効にした状態で、ステップアウト実行や連続ステップ実行を行わないでください。メモリ更新のため時間を要し、動作が遅くなります。

#### (4) 例外的なステップ

##### ① ソフトウェア割り込み命令

ソフトウェア割り込みを発生させる命令 (未定義命令、オーバフロー命令、BRK命令、INT命令) の内部処理を連続してステップ実行はできません (図 7.2参照)。

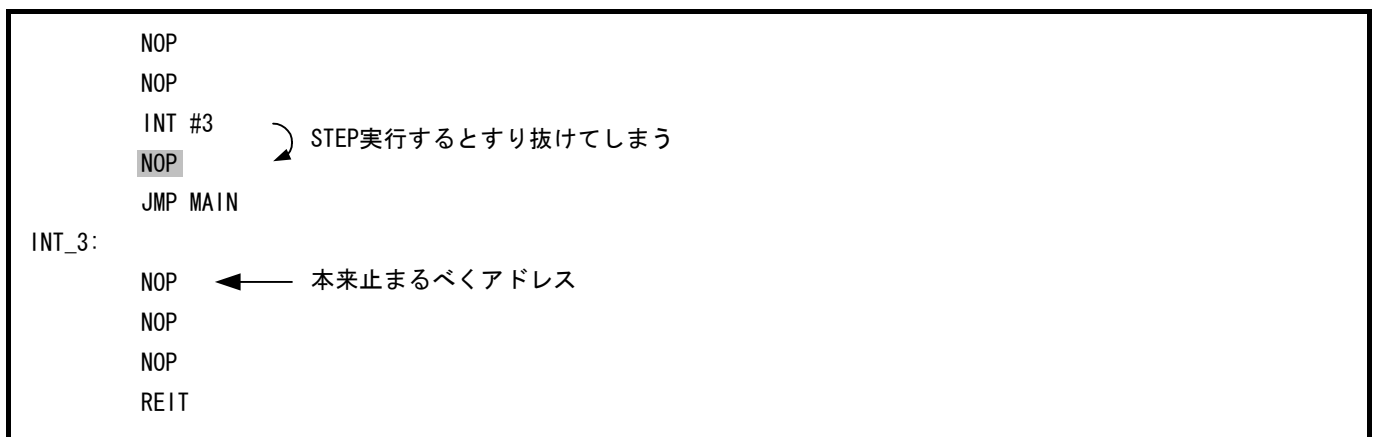


図 7.2 ソフトウェア割り込みの例

##### ② INT命令

INT命令を用いたプログラムのデバッグは、INT命令内部処理にS/Wブレークを設定し、GOコマンドと共に使用してください (図 7.3参照)。

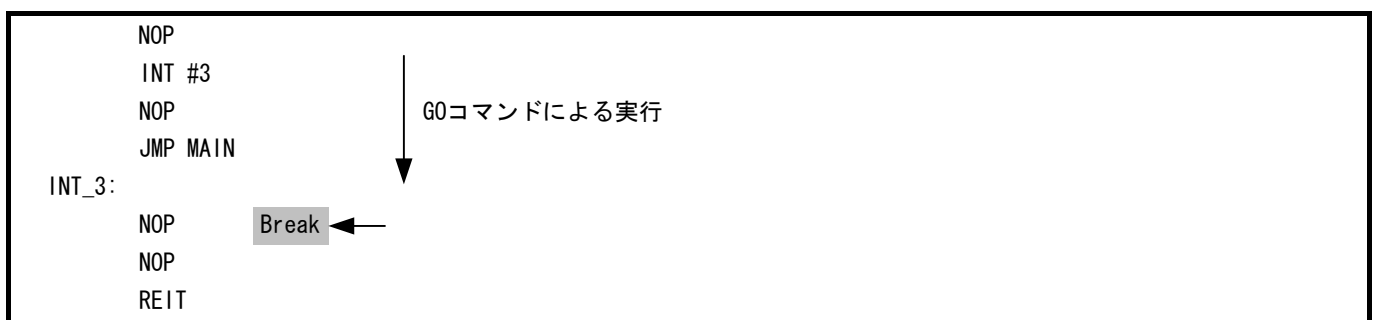


図 7.3 INT命令の場合

##### ③ その他フラグ操作命令

下記命令のシングルステップは、E1/E20エミュレータ内でのフラグ操作のみ実施するためMCU動作を伴いません。このため本命命令実行時にはStart/Stop機能は動作しませんのでご注意ください。

```

LDC src, FLG
STC FLG, dest
LDINTB src

```



### 7.8.2 その他デバッグ機能

#### (1) 「カーソル位置まで実行」機能

「カーソル位置まで実行」機能はアドレス一致ブレークを使用して実現しています。

また、「カーソル位置まで実行」を行った際、それまでに設定しているS/Wブレーク、オンチップブレークは一時無効になります。

#### (2) ストップモード、ウェイトモードのデバッグ

ストップモード、ウェイトモードのデバッグをする際は、ストップモード、ウェイトモード解除後に実行される処理部にブレークポイントを設定するなどして、ブレークポイントで止まるまで画面の操作をしないでください。また、ストップモードおよびウェイトモードになる命令をステップ実行させないでください。

本対応MCUでは、ストップモード、ウェイトモード中に内部RAMおよびSFR領域の参照・変更はユーザプログラムが停止することなく可能です。

但し、ストップモード、ウェイトモード中に内部ROM領域のメモリ内容の参照・変更を行った場合、ストップモード、ウェイトモードを解除し、MCUへメモリアクセスした後、ストップモード、ウェイトモードへ移行した次の命令から再実行します。メモリアクセスの途中でストップモード、ウェイトモードに移行した場合、メモリ内容の参照・変更は正常に動作しない場合があります。

このため、プログラムを実行する前に予めメモリウィンドウ、ウォッチウィンドウの自動更新無効化にし、メモリウィンドウでは表示固定に設定、プログラム実行中にはリフレッシュはしない等、ユーザプログラム実行中のメモリアクセスが発生しないようにしてください。

なお、ストップモード、ウェイトモード中に、プログラムを強制停止させた場合、ストップモード、ウェイトモードは解除されます。

また、ストップモード解除後のクロック設定はMCU仕様で8分周モードとなるため、ストップモードからの復帰時クロックが低速オンチップオシレータのように、CPUクロックが8分周モードで32.768kHz以下となるケースではR8C E1/E20エミュレータデバッグの最低動作周波数以下のため通信エラーとなります。

#### (3) ウォッチドッグタイマ回路

ユーザシステムのリセット回路にウォッチドッグタイマ機能がある場合は、エミュレータ使用時はウォッチドッグタイマ機能を禁止してください。

## 付録A メニュー一覧

表A.0.1 表示メニュー

メニュー	メニューオプション	サポート	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	○	[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	○	[Console]ウィンドウを表示します	
	ワークスペース	○	[Workspace]ウィンドウを表示します	
	アウトプット	○	[Output]ウィンドウを表示します	
	ステータスバー	○	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	○	[逆アセンブリ]ウィンドウを表示します	
	CPU	レジスタ	○	[レジスタ]ウィンドウを表示します
		メモリ	○	[メモリ]ウィンドウを表示します
		I/O	○	[I/O]ウィンドウを表示します
		ステータス	○	[ステータス]ウィンドウを表示します
	シンボル	ラベル	○	[ラベル]ウィンドウを表示します
		ウォッチ	○	[ウォッチ]ウィンドウを表示します
		ローカル	○	[ローカル]ウィンドウを表示します
	イベント	オンチップイベント条件設定	○	[オンチップイベント条件設定]ダイアログボックスを表示します
	グラフィック	画像	○	[画像プロパティ]ダイアログボックスを表示します
		波形	○	[波形プロパティ]ダイアログボックスを表示します
	コード	トレース	○	[トレース]ウィンドウを表示します
		スタックトレース	○	[スタックトレース]ウィンドウを表示します
RTOS	OSオブジェクト	○	[OSオブジェクト]ウィンドウを表示します	

表A.0.2 デバッグメニュー(1/2)

メニュー	メニューオプション	サポート	備考	
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	
	GPU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	○	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	○	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	
	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	○	[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	○	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	○	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	○	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	
RTOSデバッグ				
	カーソル位置まで実行	○	RTOSプログラムをロード時だけ有効	
	ステップイン	○		
	ステップオーバ	○		
	ステップアウト	○		

表A.0.3 デバッグメニュー(2/2)

メニュー	メニューオプション	サポート	備考
	プログラムの停止	○	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 とします
	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
	オーバーレイの構成	×	
	ダウンロード	○	ダウンロード オブジェクトプログラムをロードしま す
	アンロード	○	オブジェクトプログラムをアンロードします

表A.0.4 基本設定メニュー

メニュー	メニューオプション	サポート	備考	
基本設定	カスタマイズ	○		
	オプション	○		
	表示の形式	○		
	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
		StartStopファンクション設定	○	
RTOS	OS定義ファイル選択	○	RTOSプログラムロード時有効	

表A.0.5 メモリウィンドウのオプションメニュー

ポップアップメニューオプション		サポート	備考	
設定		○		
フィル		○		
コピー		○		
比較		○		
テスト		×		
テキストファイルに保存		○		
検索		○		
次を検索		○		
開始アドレス		○		
スクロール範囲		○		
レジスタ		×		
スタック追従		×		
スタートアップシンボル		○		
最新の情報に更新		○		
表示固定		○		
自動更新		○		
更新間隔		○		
データ長	1byte	○		
	2byte	○		
	4byte	○		
	8byte	○		
基数	16進数表示	○		
	10進数表示	○		
	符号付10進数表示	○		
	8進数表示	○		
	2進数表示	○		
表示コード	ASCII	○		
	SJIS	○		
	JIS	○		
	UNICODE	UTF-8	○	
		UTF-16	○	
	EUC	○		
	Float	Float	○	
		Double	○	
	Complex	Float Complex	○	
		Double Complex	○	
		Float Imaginary	○	
		Double Imaginary	○	
	Fixed	16bit Fixed	○	
		32bit Fixed	○	
24 bit Accum		○		
40 bit Accum		○		
レイアウト	ラベル	○		
	レジスタ	○		
	コード	○		
カラム		○		
カバレッジ	有効	×		
保存		○		
読み込み		○		
分割		○		

表A.0.6 トレースウィンドウのオプションメニュー

ポップアップメニューオプション	サポート	備考
クリア	○	トレース結果をクリア。
保存...	○	トレース結果をファイルに保存。
ソースファイル表示	○	任意行を選択の上、本ボタンをクリックすることで、エディタウィンドウに選択行のソースファイルを表示。
ソーストリム	○	Source欄のソース表示の左の空白を取り除いて表示
分岐トレース...	○	分岐先から次の分岐元までのソース補間を行う。 トレースデータが「分岐（分岐元/分岐先）」の時に有効。
統計...	○	条件指定で条件に合致したデータの個数を解析する機能。
関数コール...	○	関数実行履歴表示。 トレースデータが「分岐（分岐元/分岐先）」の時に有効。

## 付録B High-performance Embedded Workshopの注意事項

- (1) ロードモジュール作成後のソースファイル位置移動に関する注意事項  
ロードモジュール作成後にソースファイルを移動させた場合、作成したロードモジュールのデバッグ中にソースファイルを指定するための[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。
- (2) ソースレベル実行機能(ソースファイル)  
ロードモジュールに対応しないソースファイルをプログラムウィンドウに表示した場合、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。
- (3) ソースレベル実行機能(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。
- (4) 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。
- (5) 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.
- (6) ウォッチ機能(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.
- (7) 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
- (8) コマンドラインインタフェース(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\R8C\Tutorial\Debug\Tutorial.x30
- (9) 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。
- (10) 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。

## (11) 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[ステイタス]ウィンドウにおいてユーザプログラムの実行時間を表示していますが、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、100 $\mu$ s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。

## (12) 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。

## (13) 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。

## (14) 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。

## (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.”と表示されます。このシンボルが登録された状態でユーザプログラム実行中に全シンボル自動更新有効化を設定しないでください。

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.01	2011. 10. 21	-	初版発行



---

E1/E20エミュレータ  
ユーザーズマニュアル(別冊)接続時の注意事項  
対象デバイス: R8C/5x, R8C/3xT-Aシリーズ

発行年月日      2011年10月21日      Rev. 1.01

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

編集              株式会社ルネサス ソリューションズ  
ツール開発第二部

---



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/inquiry>

E1/E20エミュレータ  
ユーザーズマニュアル(別冊)