User's Manual

CS+ V6.00.00

統合開発環境

ユーザーズマニュアル RX デバッグ・ツール編

対象デバイス

RX ファミリ

本資料に記載の全ての情報は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。

ルネサス エレクトロニクス
www.renesas.com

Rev.1.00　2017.06

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を、全部または一部を問わず、改造、改変、複製、その他の不適切に使用しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

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

   高品質水準：輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

   当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することはできません。たとえ、意図しない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を、(1)核兵器、化学兵器、生物兵器等の大量破壊兵器およびこれらを運搬することができるミサイル（無人航空機を含みます。）の開発、設計、製造、使用もしくは貯蔵等の目的、(2)通常兵器の開発、設計、製造または使用の目的、または(3)その他の国際的な平和および安全の維持の妨げとなる目的で、自ら使用せず、かつ、第三者に使用、販売、譲渡、輸出、賃貸もしくは使用許諾しないでください。

   当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
10. お客様の転売、貸与等により、本書（本ご注意書きを含みます。）記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は一切その責任を負わず、お客様にかかる使用に基づく当社への請求につき当社を免責いただきます。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
12. 本資料に記載された情報または当社製品に関し、ご不明点がある場合には、当社営業にお問い合わせください。

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。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を、全部または一部を問わず、改造、改変、複製、その他の不適切に使用しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

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

   高品質水準：輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

   当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することはできません。たとえ、意図しない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を、(1)核兵器、化学兵器、生物兵器等の大量破壊兵器およびこれらを運搬することができるミサイル（無人航空機を含みます。）の開発、設計、製造、使用もしくは貯蔵等の目的、(2)通常兵器の開発、設計、製造または使用の目的、または(3)その他の国際的な平和および安全の維持の妨げとなる目的で、自ら使用せず、かつ、第三者に使用、販売、譲渡、輸出、賃貸もしくは使用許諾しないでください。

   当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
10. お客様の転売、貸与等により、本書（本ご注意書きを含みます。）記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は一切その責任を負わず、お客様にかかる使用に基づく当社への請求につき当社を免責いただきます。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
12. 本資料に記載された情報または当社製品に関し、ご不明点がある場合には、当社営業にお問い合わせください。

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。

このマニュアルの使い方

このマニュアルは，RX ファミリ用アプリケーション・システムを開発する際の統合開発環境である CS+ について説明します。

CS+ は，RX ファミリの統合開発環境（ソフトウエア開発における，設計，実装，デバッグなどの各開発フェーズに必要なツールをプラットフォームである IDE に統合）です。統合することで，さまざまなツールを使い分ける必要がなく，本製品のみを使用して開発のすべてを行うことができます。

対象者

このマニュアルは，CS+ を使用してアプリケーション・システムを開発するユーザを対象としています。

目的

このマニュアルは，CS+ の持つソフトウエア機能をユーザに理解していただき，これらのデバイスを使用するシステムのハードウエア，ソフトウエア開発の参照用資料として役立つことを目的としています。

構成

このマニュアルは，大きく分けて次の内容で構成しています。

1. 概　　説
2. 機　　能
A. ウインドウ・リファレンス
B. 入出力機能

読み方

このマニュアルを読むにあたっては，電気，論理回路，マイクロコンピュータに関する一般知識が必要となります。

凡例	データ表記の重み	： 左が上位桁，右が下位桁
	アクティブ・ロウの表記	： XXX （端子，信号名称に上線）
	注	： 本文中についた注の説明
	注意	： 気をつけて読んでいただきたい内容
	備考	： 本文中の補足説明
	数の表記	： 10 進数 ... XXXX 16 進数 ... 0xXXXX

すべての商標および登録商標は、それぞれの所有者に帰属します。

目次

1. 概　説 10
1.1 概　要 10
1.2 特　長 10
2. 機　能 11
2.1 概　要 11
2.2 デバッグを始める前の準備 13
2.2.1 ホスト・マシンとの接続を確認する 13
2.2.1.1 【E1】の場合 13
2.2.1.2 【E20】の場合 14
2.2.1.3 【EZ Emulator】の場合 14
2.2.1.4 【シミュレータ】の場合 15
2.3 デバッグ・ツールの動作環境設定 16
2.3.1 使用するデバッグ・ツールを選択する 16
2.3.2 【E1】の場合 17
2.3.2.1 ［接続用設定］タブ 18
2.3.2.2 ［デバッグ・ツール設定］タブ 22
2.3.2.3 ［ダウンロード・ファイル設定］タブ 28
2.3.2.4 ［フック処理設定］タブ 28
2.3.3 【E20】の場合 29
2.3.3.1 ［接続用設定］タブ 30
2.3.3.2 ［デバッグ・ツール設定］タブ 34
2.3.3.3 ［ダウンロード・ファイル設定］タブ 41
2.3.3.4 ［フック処理設定］タブ 42
2.3.4 【EZ Emulator】の場合 43
2.3.4.1 ［接続用設定］タブ 43
2.3.4.2 ［デバッグ・ツール設定］タブ 46
2.3.4.3 ［ダウンロード・ファイル設定］タブ 51
2.3.4.4 ［フック処理設定］タブ 52
2.3.5 【シミュレータ】の場合 53
2.3.5.1 ［接続用設定］タブ 53
2.3.5.2 ［デバッグ・ツール設定］タブ 54
2.3.5.3 ［ダウンロード・ファイル設定］タブ 58
2.3.5.4 ［フック処理設定］タブ 58
2.4 デバッグ・ツールとの接続／切断 59
2.4.1 CS+ にデバッグ・ツールを接続する 59
2.4.2 CS+ からデバッグ・ツールを切断する 62
2.4.3 CS+ にデバッグ・ツールをホットプラグイン接続する【E1(JTAG)】【E20(JTAG)】 62

2.5 ダウンロード／アップロード 64
2.5.1 ダウンロードを実行する 64
2.5.2 応用的なダウンロード方法. 66
2.5.2.1 ロード・モジュール・ファイルのダウンロード条件を変更する 67
2.5.2.2 ダウンロード・ファイル（*.hex/*.mot/*.bin）を追加する 69
2.5.2.3 複数のロード・モジュール・ファイルをダウンロードする. 70
2.5.2.4 ヘキサ・フォーマット／モトローラ S フォーマット／バイナリ・データ・フォーマットのファイルでソース・レベル・デバッグを行う72
2.5.2.5 外部フラッシュ・メモリへダウンロードする【E1】【E20】. 72
2.5.3 アップロードを実行する 75
2.6 プログラムの表示と変更 76
2.6.1 ソース・ファイルを表示する. 76
2.6.2 逆アセンブル結果を表示する. 77
2.6.2.1 表示モードを変更する. 77
2.6.2.2 表示形式を変更する 78
2.6.2.3 指定アドレスへ移動する. 78
2.6.2.4 シンボル定義箇所へ移動する 79
2.6.2.5 逆アセンブル結果の表示内容を保存する 79
2.6.3 他の処理と平行してビルドを実行する 80
2.6.4 ライン・アセンブルを行う. 81
2.6.4.1 命令を編集する 81
2.6.4.2 命令コードを編集する. 82
2.7 PIC/PID 機能の使用. 83
2.7.1 PIC/PID 機能を使用したロード・モジュールのアドレスの配置を変更する 83
2.8 オーバーレイ・セクションの設定. 85
2.8.1 優先セクションを指定する. 85
2.9 プログラムの実行 88
2.9.1 マイクロコントローラ（CPU）をリセットする 88
2.9.2 プログラムを実行する 88
2.9.2.1 マイクロコントローラ（CPU）をリセットしてから実行する. 89
2.9.2.2 現在のアドレスから実行する 89
2.9.2.3 PC 値を変更してから実行する 90
2.9.3 プログラムをステップ実行する 90
2.9.3.1 関数内にステップ・インする（ステップ・イン実行）. 90
2.9.3.2 関数をステップ・オーバーする（ステップ・オーバー実行）. 91
2.9.3.3 関数内でリターンが完了するまで実行する（リターン・アウト実行）. 91
2.9.4 指定ルーチンを実行する【E1】【E20】【EZ Emulator】. 91
2.10 プログラムの停止（ブレーク）. 94
2.10.1 プログラムの実行を手動で停止する 94
2.10.2 任意の場所で停止する（ブレークポイント）. 94
2.10.2.1 使用するブレークポイントの種別／動作を設定する【E1】【E20】【EZ Emulator】. 94
2.10.2.2 ブレークポイントを設定する 95
2.10.2.3 ハードウエア・ブレークポイントを編集する 96

2.10.2.4 ブレークポイントを削除する 96
2.10.3 任意の場所で停止する（ブレーク・イベント）【E1】【E20】【EZ Emulator】. 96
2.10.3.1 ブレーク・イベント（実行系）を設定する 96
2.10.3.2 ブレーク・イベント（実行系）を編集する 97
2.10.3.3 ブレーク・イベント（実行系）を削除する 97
2.10.4 変数 /I/O レジスタへのアクセスで停止する. 98
2.10.4.1 変数 /I/O レジスタへのブレーク・イベント（アクセス系）を設定する 98
2.10.4.2 変数 /I/O レジスタへのブレーク・イベント（アクセス系）を編集する 102
2.10.4.3 変数 /I/O レジスタへのブレーク・イベント（アクセス系）を削除する 102
2.10.5 複数のブレーク・イベントを組み合わせる（組み合わせブレーク）【E1】【E20】【EZ Emulator】. . . 102
2.10.6 その他のブレーク要因 103
2.11 メモリ，レジスタ，変数の表示／変更 105
2.11.1 メモリを表示／変更する 105
2.11.1.1 表示位置を指定する 105
2.11.1.2 値の表示形式を変更する. 106
2.11.1.3 メモリの内容を変更する. 107
2.11.1.4 プログラム実行中にメモリの内容を表示／変更する 108
2.11.1.5 メモリの内容を検索する. 111
2.11.1.6 メモリの内容を一括して変更（初期化）する 112
2.11.1.7 メモリの表示内容を保存する 113
2.11.2 CPU レジスタを表示／変更する 115
2.11.2.1 値の表示形式を変更する. 115
2.11.2.2 CPU レジスタの内容を変更する. 116
2.11.2.3 CPU レジスタの表示内容を保存する 116
2.11.3 I/O レジスタを表示／変更する. 116
2.11.3.1 I/O レジスタを検索する 117
2.11.3.2 I/O レジスタを整理する 117
2.11.3.3 値の表示形式を変更する. 117
2.11.3.4 I/O レジスタの内容を変更する 118
2.11.3.5 プログラム実行中に I/O レジスタの内容を表示／変更する 118
2.11.3.6 I/O レジスタの表示内容を保存する 118
2.11.4 グローバル変数／スタティック変数を表示／変更する 118
2.11.5 ローカル変数を表示／変更する 118
2.11.5.1 値の表示形式を変更する. 119
2.11.5.2 ローカル変数の内容を変更する 120
2.11.5.3 ローカル変数の表示内容を保存する. 120
2.11.6 ウォッチ式を表示／変更する. 121
2.11.6.1 ウォッチ式を登録する. 121
2.11.6.2 登録したウォッチ式を整理する 122
2.11.6.3 登録したウォッチ式を編集する 123
2.11.6.4 ウォッチ式を削除する. 123
2.11.6.5 値の表示形式を変更する. 123

2.11.6.6 ウォッチ式の内容を変更する 124
2.11.6.7 プログラム実行中にウォッチ式の内容を表示／変更する 124
2.11.6.8 ウォッチ式をエクスポート／インポートする 124
2.11.6.9 ウォッチ式の表示内容を保存する 126
2.12 スタックからの関数呼び出し情報の表示 127
2.12.1 コール・スタック情報を表示する 127
2.12.1.1 値の表示形式を変更する 127
2.12.1.2 ソース行へジャンプする 128
2.12.1.3 ローカル変数を表示する 128
2.12.1.4 コール・スタック情報の表示内容を保存する 128
2.13 実行履歴の収集 129
2.13.1 トレース動作の設定をする 129
2.13.1.1 【E1】の場合 129
2.13.1.2 【E20】の場合 131
2.13.1.3 【EZ Emulator】の場合 134
2.13.1.4 【シミュレータ】の場合 135
2.13.2 実行停止までの実行履歴を収集する 135
2.13.3 任意区間の実行履歴を収集する 136
2.13.3.1 トレース開始イベント／トレース終了イベントを設定する 136
2.13.3.2 複数のイベントを組み合わせる【E1】【E20】【EZ Emulator】 139
2.13.3.3 プログラムを実行する 139
2.13.3.4 トレース開始イベント／トレース終了イベントを編集する 139
2.13.3.5 トレース開始イベント／トレース終了イベントを削除する 139
2.13.4 条件を満たしたときのみの実行履歴を収集する 140
2.13.5 実行履歴の収集を停止／再開する【E20】【シミュレータ】 141
2.13.5.1 実行履歴の収集を一時的に停止する場合 141
2.13.5.2 実行履歴の収集を再開する場合 141
2.13.6 実行履歴を表示する 142
2.13.6.1 表示モードを変更する 142
2.13.6.2 値の表示形式を変更する 143
2.13.6.3 他のパネルと連動させる 143
2.13.7 トレース・メモリをクリアする 143
2.13.8 トレース・データを検索する 144
2.13.8.1 命令レベルで検索する 144
2.13.8.2 ソース・レベルで検索する 146
2.13.9 実行履歴の表示内容を保存する 148
2.14 実行時間の計測 150
2.14.1 タイマ計測動作の設定をする【E1】【E20】【EZ Emulator】 150
2.14.2 実行開始から停止までの実行時間を計測する 150
2.14.2.1 ステータス・バーでの確認 151
2.14.2.2 イベント パネルでの確認 151
2.14.3 任意区間の実行時間を計測する 152

2.14.3.1 タイマ開始イベント／タイマ終了イベントを設定する 152
2.14.3.2 プログラムを実行する. 155
2.14.3.3 タイマ計測イベントを編集する【E1】【E20】【EZ Emulator】. 157
2.14.3.4 タイマ開始イベント／タイマ終了イベントを編集する 158
2.14.3.5 タイマ開始イベント／タイマ終了イベントを削除する 158
2.14.4 測定可能時間の範囲 158
2.15 カバレッジの測定【シミュレータ】【E20【RX71M, RX64M グループ】】. 159
2.15.1 カバレッジ測定の設定をする 159
2.15.2 カバレッジ測定結果を表示する 160
2.16 プログラム内へのアクションの設定 163
2.16.1 printf を挿入する 163
2.16.1.1 Printf イベントを設定する 163
2.16.1.2 プログラムを実行する. 164
2.16.1.3 出力結果を確認する 164
2.16.1.4 Printf イベントを編集する 164
2.16.2 割り込みイベントを挿入する【シミュレータ】. 165
2.16.2.1 割り込みイベントを設定する 165
2.16.2.2 プログラムを実行する. 166
2.16.2.3 割り込みイベントを編集する 166
2.17 イベントの管理 167
2.17.1 設定状態（有効／無効）を変更する 169
2.17.2 特定のイベント種別のみ表示する 170
2.17.3 イベントのアドレスにジャンプする 170
2.17.4 イベントの詳細設定を編集する 170
2.17.4.1 実行系イベントを編集する 170
2.17.4.2 アクセス系イベントを編集する 171
2.17.4.3 イベントの組み合わせ条件を編集する【E1】【E20】【EZ Emulator】. 174
2.17.5 イベントを削除する 176
2.17.6 イベントにコメントを入力する 177
2.17.7 イベント設定に関する留意事項 177
2.17.7.1 有効イベント数の制限. 177
2.17.7.2 実行中に設定／削除可能なイベント種別 179
2.17.7.3 その他の注意事項 179
2.18 フック処理を設定する. 181
2.19 デバッグ・コンソールの使用 184
2.20 入力値について 188
2.20.1 入力規約 188
2.20.2 シンボル名の入力補完機能. 191
2.20.3 入力不備箇所に対するアイコン表示 192
2.21 E1/E20/EZ エミュレータとマイクロコントローラの相違点 193
2.22 その他の使用上の注意事項【E1】【E20】【EZ Emulator】. 195
2.22.1 内蔵フラッシュ ROM 領域 195

2.22.2 FINE インタフェース 195
2.22.3 エンディアン選択レジスタ（MDEB, MDES）について【RX71M, RX651, RX65N, RX64M, RX62x, RX630, RX631, RX63N, RX63T, RX634, RX210, RX21A, RX220, RX23x, RX24T, RX110, RX111, RX113, RX130】195
2.22.4 オプション機能選択レジスタ 1（OFS1）の設定について【RX62x, RX630, RX631, RX63N, RX63T, RX634, RX210, RX21A, RX220, RX23x, RX24T, RX130】195
2.22.5 オプション機能選択レジスタ 1（OFS1）の設定について【RX110, RX111, RX113】. 196
2.22.6 オプション機能選択レジスタ 1（OFS1）の設定について【RX71M, RX651, RX65N, RX64M】. 196
2.22.7 フラッシュメモリ書き換え時のレジスタ値 196
2.22.8 DMAC，DTC に関して 196
2.22.9 高速クロック発振器（HOCO）に関して【RX71M, RX64M, RX210, RX21A, RX220】. 196
2.22.10 ロックビット【ロックビット実装マイクロコントローラ】. 197
2.22.11 エリアプロテクション【エリアプロテクション実装マイクロコントローラ】. 197
2.22.12 動作周波数について 197
2.22.13 スタートアッププログラム保護機能について【RX23T, RX24T, RX110, RX111, RX113, RX130】. . . . 197
2.22.14 MPU 領域のアクセスについて【MPU 実装マイクロコントローラ】. 198
2.22.15 リセットについて 198
2.22.16 内蔵 ROM 無効拡張モード時のメモリアクセスについて 199
2.22.17 イベントの組み合わせ条件について【RX71M, RX64M, RX651, RX65N】. 199
2.22.18 カバレッジ計測機能について【E20【RX71M, RX64M, RX651, RX65N】】. 200
2.22.19 トレース機能について 200
2.22.20 Trusted Memory 機能について 200
2.22.21 FAW レジスタの FSPR ビットについて【RX651, RX65N】. 200
2.22.22 ROM キャッシュ機能について【RX651, RX65N, RX24T】. 200
2.22.23 コードフラッシュメモリのデュアルバンク機能について【RX651, RX65N】. 200
2.22.24 デュアルモード時のデバッグについて【RX651, RX65N】. 200
2.22.25 低消費電力モード中のメモリアクセスについて 201

A. ウインドウ・リファレンス 202
A.1 説　明. 202

B. 入出力機能 391
B.1 標準入出力／ファイル入出力 391

改訂記録 . . . . . C - 1

1. 概　説

CS+ は，RH850 ファミリ，RX ファミリ，V850 ファミリ，RL78 ファミリ，78K0R マイクロコントローラ，78K0 マイクロコントローラ用の統合開発環境プラットフォームです。
CS+ では，設計／コーディング／ビルド／デバッグ／フラッシュ・プログラミングなど，プログラムの開発における一連の作業を行うことができます。
本マニュアルは，こうした一連のプログラムの開発工程のうち，デバッグ工程について説明します。

注意　E2 エミュレータ Lite（略称：E2 Lite）を使用する場合は，本マニュアル中で「E1」と記載のある部分を「E2 Lite」と読み替えてください。

この章では，CS+ が提供するデバッグ機能の概要について説明します。

1.1 概　要

CS+ が提供するデバッグ機能を使用することにより，RX ファミリ用に開発されたプログラムを，効率良くデバッグすることができます。

1.2 特　長

次に，CS+ が提供するデバッグ機能の特長を示します。

- 各種デバッグ・ツールとの接続
  オンチップ・デバッギング・エミュレータ（E1/E20/EZ Emulator），およびシミュレータと組み合わせて使用することにより，より快適な開発環境を実現できます。

- C/C++ ソース・テキストと逆アセンブル・テキストの混合表示
  1 つのパネル上で，C/C++ ソース・テキストと逆アセンブル・テキストを混合表示することができます。

- ソース・レベル・デバッグと命令レベル・デバッグ
  C/C++ ソース・プログラムに対して，ソース・レベル・デバッグ，または命令レベル・デバッグを行うことができます。

- データフラッシュ・メモリへの書き込み
  選択しているマイクロコントローラがデータフラッシュ・メモリ内蔵品の場合では，通常のメモリ操作と同様のアクセス方法により，データフラッシュ・メモリの内容を表示／変更することができます（シミュレータを除く）。

- リアルタイム表示更新機能
  プログラムの実行が停止した際に，表示情報を自動的に更新するだけでなく，プログラムが実行中の状態であっても，リアルタイムにメモリ／レジスタ／変数の値を表示更新することができます。

- デバッグ環境の保存／復元
  ブレークポイントやイベントの設定情報，ファイルのダウンロード情報，パネルの表示状態／位置などのデバッグ環境を保存することができます。

2. 機　　能

この章では，CS+ を使用したデバッグの手順，およびデバッグに関する主な機能について説明します。

2.1 概　　要

CS+ を使用した，プログラムの基本的なデバッグ手順は次のとおりです。

(1) CS+ を起動する
Windows® の［スタート］メニューから CS+ を起動します。

備考　“CS+ を起動する ” についての詳細は，「CS+ 統合開発環境 ユーザーズマニュアル 起動編」を参照してください。

(2) プロジェクトを設定する
プロジェクトの新規作成，または既存のプロジェクトの読み込みを行います。

備考　“ プロジェクトを設定する ” についての詳細は，「CS+ 統合開発環境 ユーザーズマニュアル 起動編」を参照してください。

(3) ロード・モジュールを作成する
アクティブ・プロジェクトの設定，および使用するビルド・ツールの設定を行ったのち，ビルドを実行することにより，ロード・モジュールを作成します。

備考　CC-RX を使用して “ ロード・モジュールを作成する ” 場合についての詳細は，「CS+ 統合開発環境 ユーザーズマニュアル CC-RX ビルド・ツール操作編」を参照してください。

(4) ホスト・マシンとの接続を確認する
ホスト・マシンに，使用するデバッグ・ツール（E1/E20/EZ Emulator/ シミュレータ）を接続します。

(5) 使用するデバッグ・ツールを選択する
プロジェクトで使用するデバッグ・ツールを選択します。

備考　プロジェクト内で使用するマイクロコントローラにより，選択可能なデバッグ・ツールは異なります。

(6) デバッグ・ツールの動作環境設定を行う
(5) において選択したデバッグ・ツールのための動作環境を設定します。

- 【E1】の場合
- 【E20】の場合
- 【EZ Emulator】の場合
- 【シミュレータ】の場合

(7) CS+ にデバッグ・ツールを接続する
CS+ とデバッグ・ツールの通信を開始します。

(8) ダウンロードを実行する
(3) において作成したロード・モジュールを，デバッグ・ツールへダウンロードします。

(9) ソース・ファイルを表示する
ダウンロードしたロード・モジュールの内容（ソース・ファイル）をエディタ パネル，または逆アセンブル パネルで表示します。

(10) プログラムを実行する
目的に応じた実行方法により，プログラムを実行します。
なお，実行したプログラムを任意の箇所で停止する場合は，あらかじめブレークポイント／ブレーク・イベント[注]を設定しておきます（「2.10.2 任意の場所で停止する（ブレークポイント）」／「2.10.3 任意の場所で停止する（ブレーク・イベント）【E1】【E20】【EZ Emulator】」参照）。

注　使用するデバッグ・ツールにイベントを設定することにより実現する機能です。イベントを設定する際には，「2.17.7 イベント設定に関する留意事項」を参照してください。

(11) プログラムの実行を手動で停止する
実行したプログラムを停止します。
ただし，(10) においてブレークポイント／ブレーク・イベントを設定している場合では，その条件が満たされると同時にプログラムの実行は自動的に停止します。

(12) プログラムの実行結果を確認する
プログラムを実行することにより取得した各種情報を確認します。

- メモリ，レジスタ，変数の表示／変更
- スタックからの関数呼び出し情報の表示
- 実行履歴の収集注
- 実行時間の計測注
- カバレッジの測定【シミュレータ】【E20【RX71M, RX64M グループ】】

注　使用するデバッグ・ツールにイベントを設定することにより実現する機能です。イベントを設定する際には，「2.17.7 イベント設定に関する留意事項」を参照してください。

以後，必要に応じて (9) ～ (12) を繰り返すことによりデバッグ作業を進めます。
なお，この際に，プログラムに変更を加えた場合は，(3) および (8) の操作も繰り返す必要があります。

備考 1.　上記のほか，次の機能を利用して，プログラムの実行結果の確認を行うことができます。

- プログラム内へのアクションの設定
- フック処理を設定する
- デバッグ・コンソールの使用

備考 2.　取得した各種情報をファイルに保存することができます。

- 逆アセンブル結果の表示内容を保存する
- メモリの表示内容を保存する
- CPU レジスタの表示内容を保存する
- I/O レジスタの表示内容を保存する
- ローカル変数の表示内容を保存する
- ウォッチ式の表示内容を保存する
- コール・スタック情報の表示内容を保存する
- 実行履歴の表示内容を保存する

(13) アップロードを実行する
必要に応じ，プログラム（メモリ内容）を任意のファイル形式（ヘキサ・フォーマット／バイナリ・データ・フォーマットなど）で保存します。

(14) CS+ からデバッグ・ツールを切断する
CS+ とデバッグ・ツールとの通信を終了します。

(15) プロジェクト・ファイルを保存する
プロジェクトの設定情報をプロジェクト・ファイルに保存します。

備考　“ プロジェクト・ファイルを保存する ” についての詳細は，「CS+ 統合開発環境 ユーザーズマニュアル 起動編」を参照してください。

2.2 デバッグを始める前の準備

この節では，作成したプログラムのデバッグを開始するための準備について説明します。

2.2.1 ホスト・マシンとの接続を確認する

使用するデバッグ・ツールごとに，ホスト・マシンとの接続例を示します。
なお，プロジェクトで選択しているマイクロコントローラと接続可能なデバッグ・ツールの関係は次の表のとおりです。

備考　表内における “(Serial)” または “(JTAG)” は，E1/E20 を FINE 通信方式，または JTAG 通信方式で使用する場合を意味します。

表 2.1　マイクロコントローラの種類と接続可能なデバッグ・ツールの関係

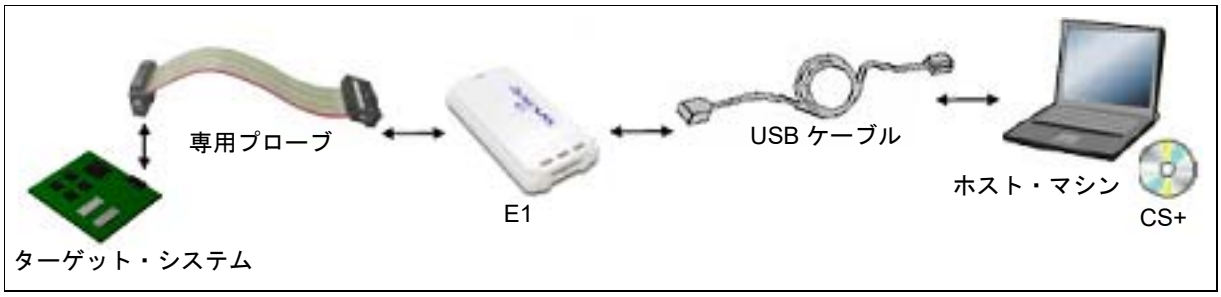
マイクロコントローラ	デバッグ・ツール					
	E1(Serial)	E1(JTAG)	E20(Serial)	E20(JTAG)	EZ Emulator	シミュレータ
RX610, RX621, RX62N, RX62T, RX62G グループ	–	○	–	○	–	○
RX630, RX631, RX63N グループ	○	○	○	○	–	○
RX63T グループ	○	○	○	○	○	○
RX110, RX111, RX113, RX210, RX220 グループ	○	–	○	–	○	○
RX21A グループ	○	–	○	–	–	○
RX71M, RX64M, RX65N, RX651, RX65N-2M, RX651-2M グループ	○	○	○	○	–	○

2.2.1.1 【E1】の場合
2.2.1.2 【E20】の場合
2.2.1.3 【EZ Emulator】の場合
2.2.1.4 【シミュレータ】の場合

2.2.1.1 【E1】の場合

ホスト・マシン，E1，および必要に応じてターゲット・ボードを接続します。
ターゲット・システムとの通信方式として，FINE 通信【E1(Serial)】，または JTAG 通信【E1(JTAG)】を選択することができます（「2.3.1 使用するデバッグ・ツールを選択する」参照）。
接続方法についての詳細は，E1 のユーザーズ・マニュアルを参照してください。

図 2.1　ホスト・マシンとデバッグ・ツールとの接続例【E1】



2.2.1.2 【E20】の場合

ホスト・マシン，E20，および必要に応じてターゲット・ボードを接続します。
ターゲット・システムとの通信方式として，FINE 通信【E20(Serial)】，または JTAG 通信【E20 (JTAG)】を選択することができます（「2.3.1 使用するデバッグ・ツールを選択する」参照）。
接続方法についての詳細は，E20 のユーザーズ・マニュアルを参照してください。

図 2.2 ホスト・マシンとデバッグ・ツールとの接続例【E20(JTAG)】

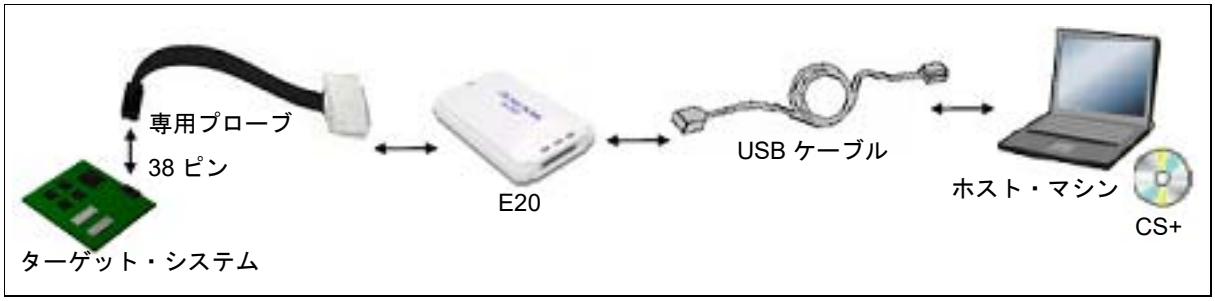
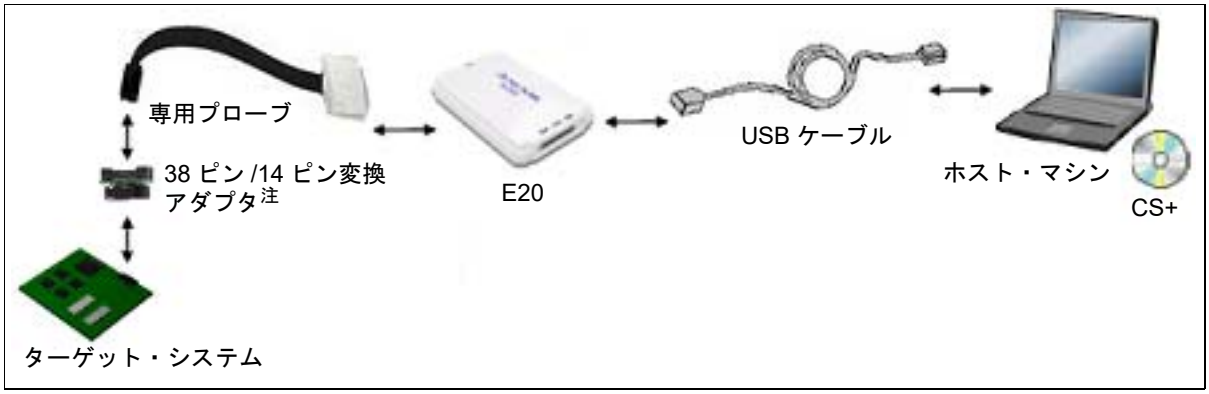


図 2.3 ホスト・マシンとデバッグ・ツールとの接続例【E20(Serial)】

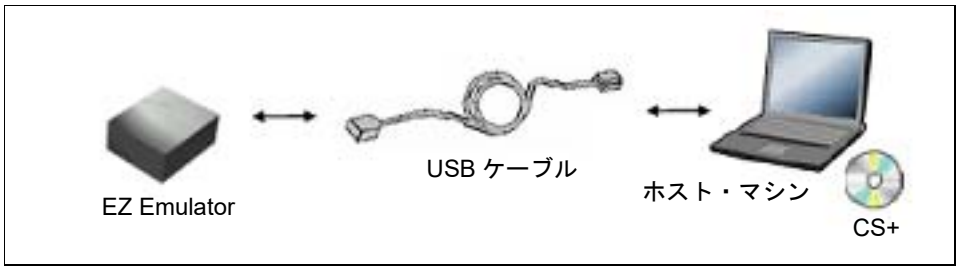


注 ターゲット・システムとの通信方式として，FINE 通信方式【E20(Serial)】を選択した場合，E20 エミュレータに付属の 38 ピン /14 ピン変換アダプタを使用してください。
38 ピン /14 ピン変換アダプタを用いた接続方法についての詳細は，E20 のユーザーズ・マニュアルを参照してください。

2.2.1.3 【EZ Emulator】の場合

ホスト・マシン，EZ Emulator，および必要に応じてターゲット・ボードを接続します。
接続方法についての詳細は，EZ Emulator のユーザーズ・マニュアルを参照してください。

図 2.4 ホスト・マシンとデバッグ・ツールとの接続例【EZ Emulator】



2.2.1.4 【シミュレータ】の場合

ホスト・マシンのみでデバッグ作業を行うことができます（エミュレータなどの接続は不要）。

図 2.5　　ホスト・マシンとデバッグ・ツールとの接続例【シミュレータ】

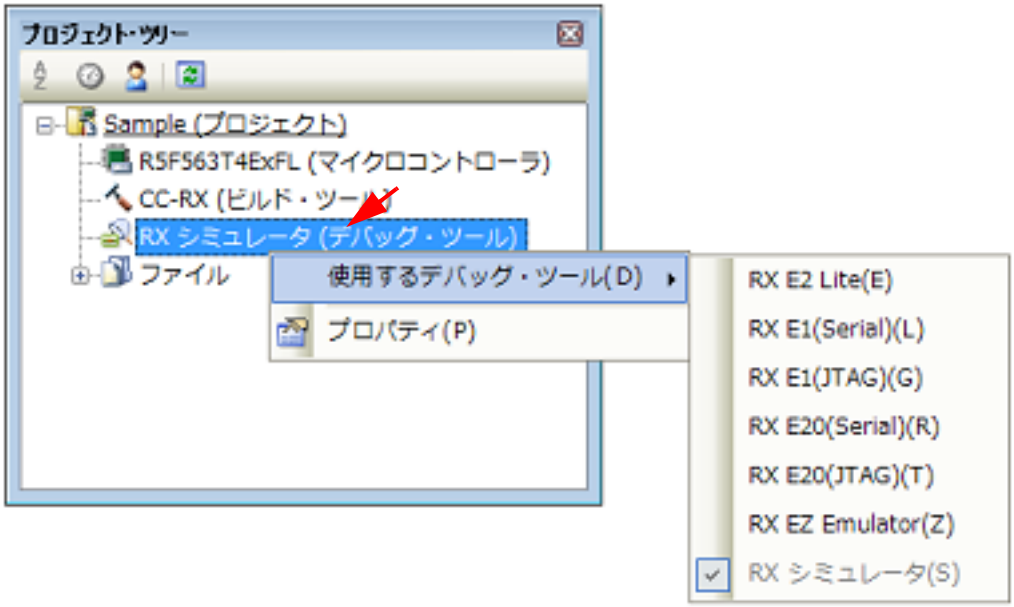
2.3 デバッグ・ツールの動作環境設定

この節では，各デバッグ・ツールの動作環境の設定方法について説明します。

2.3.1 使用するデバッグ・ツールを選択する

デバッグ・ツールの動作環境設定は，使用するデバッグ・ツールに対応したプロパティ パネルで行います。
そのため，まず，プロジェクト内で使用するデバッグ・ツールを選択します（使用するデバッグ・ツールはプロジェクトごとに選択可）。
使用するデバッグ・ツールの選択／切り替えは，プロジェクト・ツリー パネル上の［*品種名　デバッグ・ツール名*(デバッグ・ツール)］ノードを右クリックすることで表示されるコンテキスト・メニューにより行ってください。

図 2.6　　使用するデバッグ・ツールの選択／切り替え



注意　　表示されるコンテキスト・メニューは，プロジェクトで選択しているマイクロコントローラの種類により異なります（「表 2.1 マイクロコントローラの種類と接続可能なデバッグ・ツールの関係」参照）。

備考 1.　　【E1】【E20】
E1/E20 とターゲット・システムとの通信方式として，FINE 通信を使用する場合は［*品種名* E1(Serial)］／［*品種名* E20(Serial)］を，JTAG 通信を使用する場合は［*品種名* E1(JTAG)］／［*品種名* E20(JTAG)］を選択してください。

備考 2.　　【EZ Emulator】
EZ Emulator とターゲット・システムの通信方式は，FINE 通信のみとなります。

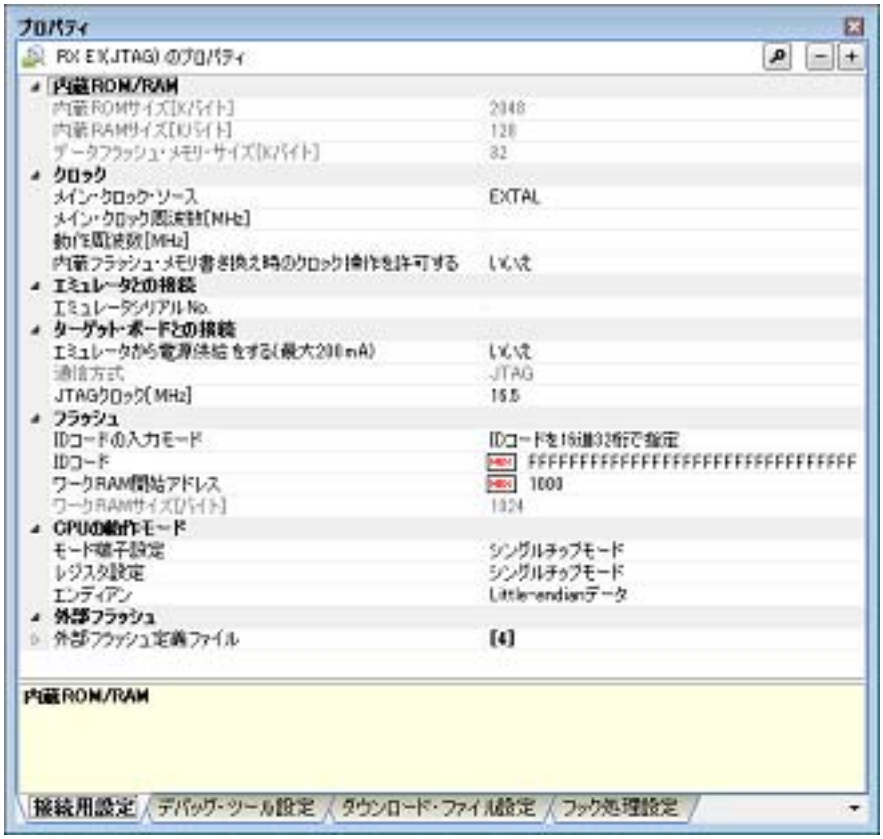
すでにプロパティ パネルがオープンしている場合，再び［*品種名　デバッグ・ツール名*(デバッグ・ツール)］ノードをクリックすると，選択したデバッグ・ツールのプロパティ パネルに表示が切り替わります。
プロパティ パネルがオープンしていない場合では，同ノードをダブルクリックすることで，該当するプロパティ パネルがオープンします。

2.3.2 【E1】の場合

E1 を使用する場合の動作環境の設定を次のプロパティ パネルで行います。
なお，ターゲット・システムとの通信方式（FINE 通信【E1(Serial)】/JTAG 通信【E1(JTAG)】）により，プロパティ パネルの内容は異なります。

図 2.7　　動作環境設定【E1(Serial)】（プロパティ パネル）

図 2.8　　動作環境設定【E1(JTAG)】（プロパティ パネル）



プロパティ パネル上の該当するタブを選択し，次の設定を順次行ってください。

2.3.2.1　［接続用設定］タブ
2.3.2.2　［デバッグ・ツール設定］タブ
2.3.2.3　［ダウンロード・ファイル設定］タブ
2.3.2.4　［フック処理設定］タブ

2.3.2.1　［接続用設定］タブ

［接続用設定］タブでは，次に示すカテゴリごとに，デバッグ・ツールとの接続に関する設定を行います。

(1)　［内蔵 ROM/RAM］
(2)　［クロック］
(3)　［エミュレータとの接続］
(4)　［ターゲット・ボードとの接続］
(5)　［フラッシュ］
(6)　［CPU の動作モード］
(7)　［外部フラッシュ］

(1)　［内蔵 ROM/RAM］
このカテゴリでは，内蔵 ROM/RAM に関する設定を表示します。

図 2.9　　［内蔵 ROM/RAM］カテゴリ

- ［内蔵 ROM サイズ [K バイト]］
  エミュレーションする内蔵 ROM サイズを表示します（単位：K バイト）。
  このプロパティ値を変更することはできません。

- ［内蔵 RAM サイズ [バイト]］
  エミュレーションする内蔵 RAM サイズを表示します（単位：バイト）。
  このプロパティ値を変更することはできません。

- ［データフラッシュ・メモリ・サイズ [K バイト]］
  選択しているマイクロコントローラのデータフラッシュ・メモリ領域のサイズを表示します（単位：K バイト）。
  選択しているマイクロコントローラがデータフラッシュ・メモリ非搭載品の場合は，［0］を表示します。
  このプロパティ値を変更することはできません。

(2)　［クロック］

このカテゴリでは，クロックに関する設定を行います。

注意　E1 と接続中にこのカテゴリ内のプロパティを変更することはできません。

図 2.10　［クロック］カテゴリ【HOCO】

クロック	
メイン・クロック・ソース	HOCO
動作周波数[MHz]	25.0000
内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する	いいえ

図 2.11　［クロック］カテゴリ【EXTAL】

クロック	
メイン・クロック・ソース	EXTAL
メイン・クロック周波数[MHz]	12.5000
動作周波数[MHz]	25.0000
内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する	いいえ

- ［メイン・クロック・ソース］
  メイン・クロックを EXTAL 周波数と内蔵 HOCO から指定します。
  内蔵 HOCO を持たないマイクロコントローラの場合は，[EXTAL] を表示します。

- ［メイン・クロック周波数 [MHz]］
  メイン・クロックの周波数（逓信前）を指定します。
  EXTAL 周波数値として 0.0001 ～ 99.9999（単位：MHz）の範囲で数値を直接入力します。
  入力値の少数第 5 位以下は切り捨てられ，指定可能範囲外の入力値は，0.0001(0 以下だった時) または 99.9999(100 以上だった時) に丸め込みます。
  このプロパティは，［メイン・クロック・ソース］プロパティで [EXTAL] を指定した場合のみ表示されます。

- ［動作周波数 [MHz]］
  動作周波数（ICLK）を指定します。
  0.0001 ～ 999.9999（単位：MHz）の範囲で数値を直接入力します。
  入力値の少数第 5 位以下は切り捨てられ，指定可能範囲外の入力値は，0.0001(0 以下だった時) または 999.9999(1000 以上だった時) に丸め込みます。

- ［内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する］
  内蔵フラッシュ・メモリ書き換え時に，デバッガによるクロック操作を許可するか否かを指定します。

注意　【E1【RX630, RX631, RX63N, RX63T, RX210, RX21A, RX220, RX110, RX111, RX113】】
[はい] を指定した場合，マイクロコントローラの FlashIF クロック（FCLK）が動作保証範囲外（LOCO やサブクロック動作中など）でデバッガによる内蔵フラッシュ ROM の書き換えを実施した時，E1 がクロック・ソースを切り替えます。なお，書き換え完了後は，元のクロック・ソースに復帰します。
クロック操作を許可した場合，クロック・ソースを切り替えるため，内蔵フラッシュ・メモリ書き換え中の周辺クロック動作周波数が変更されます。
クロック操作の許可設定は，プログラム実行やステップ実行などの実行系機能を実施した後の内蔵フラッシュ ROM を書き換える時に有効となります。なお，デバッグ・ツールの接続直後や CPU リセット発行時は，クロック操作の許可設定に関わらず FCLK が動作保証範囲外の場合は，強制的にクロック・ソースを切り替えます。

(3) ［エミュレータとの接続］
このカテゴリでは，E1 とホスト・マシンとの接続に関する設定を行います。

注意　E1 と接続中にこのカテゴリ内のプロパティを変更することはできません。

図 2.12　[エミュレータとの接続] カテゴリ

◢ エミュレータとの接続	
エミュレータシリアル No.	

- ［エミュレータシリアル No.］
接続されている E1 のシリアル番号をドロップダウン・リストに全て表示します。
接続する E1 のシリアル番号を指定してください。
なお，ドロップダウン・リストはドロップダウン時に毎回更新します。

(4) ［ターゲット・ボードとの接続］
このカテゴリでは，E1 とターゲット・ボードとの接続に関する設定を行います。

図 2.13　［ターゲット・ボードとの接続］カテゴリ【E1(Serial)】

◢ ターゲット・ボードとの接続	
エミュレータから電源供給をする(最大200mA)	はい
供給電圧	3.3V
通信方式	FINE
FINEボーレート[bps]	2000000

図 2.14　［ターゲット・ボードとの接続］カテゴリ【E1(JTAG)】

◢ ターゲット・ボードとの接続	
エミュレータから電源供給をする(最大200mA)	はい
供給電圧	3.3V
通信方式	JTAG
JTAGクロック[MHz]	16.5

- ［エミュレータから電源供給をする (最大 200mA)］
E1 からターゲット・システムへ電源を供給するか否かを指定します。
電源を供給する場合は［はい］を選択してください。

注意　E1 と接続中にこのプロパティを変更することはできません。

- ［供給電圧］
E1 からターゲット・システムへ供給する電圧をドロップダウン・リストより指定します。
このプロパティは，［エミュレータから電源供給をする (最大 200mA)］プロパティにおいて［はい］を指定した場合のみ表示されます。
なお，指定可能な電圧値はマイクロコントローラにより異なります。

注意　【E1】
E1 と接続中にこのプロパティを変更することはできません。
【E2 Lite】
このプロパティは変更できません。

- ［通信方式］
E1 がターゲット・システム上のマイクロコントローラと通信する際の通信方式を表示します。
プロジェクト・ツリー パネルにて使用するデバッグ・ツールを選択する際に ”RX E1(Serial)” を選択した場合，このプロパティには [FINE] が表示されます。また，”RX E1(JTAG)” を選択した場合，[JTAG] が表示されます。
E1 ではこのプロパティ値を変更することはできません。”RX E2 Lite” を選択した場合，このプロパティ値を変更可能かどうかは選択しているマイクロコントローラの種類と通信方式によって異なります。マイクロコントローラの種類と通信方式の関係については，「2.2.1 ホスト・マシンとの接続を確認する」を参照してください。
なお，デバッグ・ツールの選択についての詳細は，「2.3.1 使用するデバッグ・ツールを選択する」を参照してください。

- ［JTAG クロック [MHz]］
E1 がターゲット・システム上のマイクロコントローラと通信する際の通信速度 (JTAG クロック) をドロップダウン・リストより指定します。
このプロパティは，［通信方式］プロパティにおいて [JTAG] の場合のみ表示されます。
ドロップダウン・リストには，次の通信速度が表示されます。

- 16.5(デフォルト), 12.38, 6.188, 3.094, 1.547【E1】
- 6.00(デフォルト), 3.00, 1.50 【E2 Lite】

注意 1. E1 と接続中にこのカテゴリ内のプロパティを変更することはできません。

注意 2. ターゲット・システム上での JTAG 信号の配線長や配線方法によっては，選択した JTAG クロックで通信できない場合があります。この JTAG クロックを下げることによって，通信が出来るようになる場合があります。

- ［FINE ボーレート [bps]］
E1 がターゲット・システム上のマイクロコントローラと通信する際の通信速度 (FINE ボーレート) をドロップダウン・リストより指定します。
このプロパティは，［通信方式］プロパティにおいて [FINE] の場合のみ表示されます。
ドロップダウン・リストには，次の通信速度が表示されます。

- 2000000(デフォルト), 750000, 500000, 250000 【E1】
- 1500000(デフォルト), 750000, 500000, 250000 【E2 Lite】

注意 1. E1 と接続中にこのカテゴリ内のプロパティを変更することはできません。

注意 2. ターゲット・システム上での FINE 信号の配線長や配線方法によっては，選択した FINE ボーレートで通信できない場合があります。この FINE 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。

(5) ［フラッシュ］
このカテゴリでは，フラッシュ書き換えに関する設定を行います。

注意 E1 と接続中にこのカテゴリ内のプロパティを変更することはできません。

図 2.15 ［フラッシュ］カテゴリ

- ［ID コードの入力モード］
ID コードの入力モードを指定します。

注意 E1 と接続中にこのプロパティを変更することはできません。

- ［ID コード］
ID コードプロテクトを解除するための ID コードを指定します。
［ID コードの入力モード］プロパティにおいて，[ID コードを 16 進 32 桁で指定] を指定した場合，ID コードは 32 桁の 16 進数で入力してください。また，[ID コードを 16 文字以内で指定 (ASCII)] を指定した場合，ID コードは 16 文字以内の ASCII 文字列で入力してください。

注意 1. E1 と接続中にこのプロパティを変更することはできません。

注意 2. ID コードを 32 桁の 16 進数で入力する場合は，32 ビットのデータ単位に並べて入力してください。
フラッシュプログラマでは ID コードの並び順が CS+ と異なる場合があります。ご使用のフラッシュプログラマの ID コード入力の仕様を確認してください。

注意 3. ID コードを ASCII 文字列で入力する場合，16 文字に満たない部分は ”0” で入力されます。

注意 4. ID コードを含んだプログラムをダウンロードした場合，ID コードは FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF に書き換わります。

- ［ワーク RAM 開始アドレス］
デバッガが使用するワーク RAM の配置アドレスを指定します。なお，アドレス値は 4 バイト単位の値を指定してください。（入力値が 4 バイト単位の値でなかった場合は，自動的に補正されます。）
指定したワーク RAM の配置アドレスから［ワーク RAM サイズ [バイト]］プロパティの表示サイズ分は，デバッガのファームウェアが使用します。

注意 1. E1 と接続中にこのプロパティを変更することはできません。

注意 2. メモリ内容の退避・復帰が行われるためユーザ・プログラムでもこの領域を使用できますが，ワーク RAM に配置する領域は，以下の用途では使用できません。

- DMA または DTC 機能の転送元・転送先
- DTC ベクタテーブル，および転送情報配置アドレス
- DMAC，DTC 起動要因の割り込みベクタ

- ［ワーク RAM サイズ [バイト]］
  デバッガが使用するワーク RAM のサイズを表示します。

(6)　［CPU の動作モード］
このカテゴリでは，エミュレーションするマイクロコントローラの動作モードに関する設定を行います。

注意　E1 と接続中にこのカテゴリ内のプロパティを変更することはできません。

図 2.16　[CPU の動作モード] カテゴリ

CPUの動作モード	
モード端子設定	シングルチップモード
レジスタ設定	シングルチップモード
エンディアン	Little-endianデータ
起動バンクを変更する	はい
起動バンク	バンク0

- ［モード端子設定］
  マイクロコントローラのモード端子で設定している動作モードを指定します。

- ［USB ブートプログラムの消去を許可する］
  USB ブートプログラムが格納されたマイクロコントローラを使用してユーザブートモードで起動する際に，ユーザブート領域上の USB ブートプログラムを消去するか否かを指定します。
  このプロパティは，［モード端子設定］プロパティにおいて［ユーザブートモード］を指定した場合のみ表示されます。ただし，ユーザブートモードで起動する際に，ユーザブート領域上の USB ブートプログラムを消去する必要のないマイクロコントローラの場合は表示されません。

- ［レジスタ設定］
  レジスタで設定する動作モードを指定します。
  指定可能な動作モードは，マイクロコントローラによって異なります。

- ［エンディアン］
  プロジェクトのエンディアンを表示します。
  プロジェクトからエンディアンの設定情報を取得し，その値を表示します。
  デバッグ・ツールと切断中のみ指定することができます。

- ［起動バンクを変更する］【RX651-2M, RX65N-2M グループ】
  デバッグ・ツールと接続する際に起動バンクを変更するか否かを指定します。
  起動バンクを変更して接続する場合は［はい］を選択してください（デフォルトでは［いいえ］が指定されます）。

- ［起動バンク］【RX65N-2M, RX651-2M グループ】
  デバッグ・ツールと接続する際に変更する起動バンクを指定します。
  このプロパティは，［起動バンクを変更する］【RX651-2M, RX65N-2M グループ】プロパティにおいて［はい］を指定した場合のみ表示されます。

(7)　［外部フラッシュ］
このカテゴリでは，外部フラッシュに関する設定を行います。
このカテゴリ内の設定は，外部フラッシュ・メモリへダウンロードする場合に必要となります。各プロパティ内の設定についての詳細は，「2.5.2.5 外部フラッシュ・メモリへダウンロードする【E1】【E20】」を参照してください。

注意　E1 と接続中にこのカテゴリ内のプロパティを変更することはできません。

2.3.2.2 ［デバッグ・ツール設定］タブ

［デバッグ・ツール設定］タブでは，次に示すカテゴリごとに，デバッグ・ツールの基本設定を行います。

(1)　［メモリ］
(2)　［実行中のメモリ・アクセス］
(3)　［レジスタ］
(4)　［ブレーク］

(5) ［システム］
(6) ［トレース］
(7) ［タイマ］【RX600 シリーズ】

(1) ［メモリ］
このカテゴリでは，メモリに関する設定を行います。

図 2.17 ［メモリ］カテゴリ

◢ メモリ	
◢ メモリ・マッピング	[23]
▷ [0]	内蔵RAM領域
▷ [1]	予約領域
▷ [2]	I/Oレジスタ領域
▷ [3]	予約領域
▷ [4]	I/Oレジスタ領域
▷ [5]	I/Oレジスタ領域
▷ [6]	予約領域
▷ [7]	I/Oレジスタ領域
▷ [8]	I/Oレジスタ領域
▷ [9]	I/Oレジスタ領域
▷ [10]	データフラッシュ領域
▷ [11]	予約領域
▷ [12]	その他のメモリ領域
▷ [13]	予約領域
▷ [14]	I/Oレジスタ領域
▷ [15]	予約領域
▷ [16]	I/Oレジスタ領域
▷ [17]	予約領域
▷ [18]	その他のメモリ領域
▷ [19]	予約領域
▷ [20]	その他のメモリ領域
▷ [21]	予約領域
▷ [22]	内蔵ROM領域
メモリ書き込み時にベリファイを行う	はい

- ［メモリ・マッピング］
現在のメモリ・マッピングの状況が，メモリ領域の種別ごとに詳細表示されます。
このパネル上でマッピング値を変更することはできません。
I/O 保護領域を追加，または削除する必要がある場合は，［メモリ・マッピング］プロパティを選択することで設定欄右端に表示される [...] ボタンのクリックによりオープンするメモリ・マッピング ダイアログで行います。
設定方法についての詳細は，メモリ・マッピング ダイアログの項を参照してください。
このプロパティでは，メモリ領域の個数のみを表示します。
[メモリ・マッピング] プロパティを展開すると，次のサブ項目が表示されます。

- [メモリ種別]
当該領域のメモリ種別を表示します。
各メモリ種別は次の領域に該当します。

内蔵 ROM 領域	プログラム ROM 注 1 およびデータフラッシュ注 2 が該当します。
内蔵 RAM 領域	内蔵 RAM が該当します。注 4
I/O レジスタ領域	周辺 I/O レジスタが該当します。 エンディアンが異なる領域で分割して表示します。
外部領域 (CS7/CS6/.../CS0)	外部アドレス空間が該当します。 CS0 - CS7 を区別して表示します。
その他のメモリ領域	FCU-RAM 注 3, FCU ファーム注 3, ユーザーブートが該当します。
予約領域	上記以外の領域を表示します。

I/O 保護領域	外部領域でデバッガからリードしないアドレス範囲が該当します。 メモリ・マッピング ダイアログで登録します。

注 1.　ダウンロード以外の操作（ メモリ パネルからの操作やライン・アセンブル）で内蔵フラッシュ ROM 領域を編集した場合，編集内容は次の実行操作時にフラッシュ ROM に書き込まれます。

注 2.　消去された状態のデータフラッシュ ROM の内容を参照すると，マイクロコントローラの仕様どおり不定値が表示されます。なお，デバッガからデータフラッシュ ROM の内容を書き換えた場合は，256 バイト単位で書き込みを行います。書き込まれた領域は不定値とはなりません。

注 3.　FCU-RAM 領域の内容をデバッガから書き換えないでください。また，FCU ファーム領域はデバッガから書き換えできません。

注 4.　【RX71M, RX64M グループ】
下記のオプション設定メモリを含む領域（0x00120040 ～ 0x0012006F 番地）は，内蔵 RAM 領域として表示されます。なお，本領域は，メモリパネルで値を設定することはできません。

- シリアルプログラマコマンド制御レジスタ（SPCC）
- OCD/ シリアルプログラマ ID 設定レジスタ（OSIS）
- エンディアン選択レジスタ（MDE）
- オプション機能選択レジスタ 0（OFS0）
- オプション機能選択レジスタ 1（OFS1）

- [開始アドレス]
  当該領域の開始アドレスを表示します。
- [終了アドレス]
  当該領域の終了アドレスを表示します。
- [アドレス幅 [ビット]]
  当該領域のアクセス幅を表示します。
  [メモリ種別] が外部領域については，デバッグ・ツールと切断中のみ変更可能です。
- [エンディアン]
  外部領域および I/O レジスタ領域のエンディアンを表示します。
  [メモリ種別] が外部領域については，デバッグ・ツールと切断中のみ変更可能です。

注意　デバッグ・ツールと接続することにより（「2.4.1 CS+ にデバッグ・ツールを接続する」参照），各メモリ種別ごとの詳細表示を行います。

- ［メモリ書き込み時にベリファイを行う］
  メモリ値の初期化を行う際に，ベリファイを行うか否かをドロップダウン・リストにより指定します。
  ［はい］を指定した場合，ダウンロードした際，ウォッチ パネルやメモリ パネルで値を変更した際にベリファイを行います。

(2)　［実行中のメモリ・アクセス］
このカテゴリでは，プログラム実行中におけるメモリ・アクセスに関する設定を行います。
このカテゴリ内の設定は，リアルタイム表示更新機能を使用する場合に必要となります。リアルタイム表示更新機能についての詳細は，「2.11.1.4 プログラム実行中にメモリの内容を表示／変更する」を参照してください。

図 2.18　［実行中のメモリ・アクセス］カテゴリ I)】

実行中のメモリ・アクセス	
実行を一瞬停止してアクセスする	いいえ
実行中に表示更新を行う	はい
表示更新間隔[ms]	500

- ［実行を一瞬停止してアクセスする］
  プログラム実行中に，メモリに対してアクセスを許可するか否かをドロップダウン・リストにより指定します。
  アクセスを許可する場合は［はい］を選択してください（デフォルトでは［いいえ］が指定されます）。
- ［実行中に表示更新を行う］

プログラム実行中に，ウォッチ パネル／メモリ パネルの表示内容を更新するか否かをドロップダウン・リストにより指定します。
表示内容の更新を行う場合は［はい］を選択してください（デフォルト）。

注意　プログラム実行中にこのプロパティを変更することはできません。

- ［表示更新間隔 [ms]］
このプロパティは，［実行中に表示更新を行う］プロパティにおいて［はい］を指定した場合にのみ表示されます。
プログラム実行中に，ウォッチ パネル／メモリ パネルの表示内容を更新する間隔を 100 ms 単位で指定します。
直接入力により，100 ～ 65500 の範囲の整数（100 ms 未満の端数切り上げ）を指定してください（デフォルトでは［500］が指定されます）。
なお，［実行中に表示更新を行う］プロパティにおいて指定値を［いいえ］から［はい］へ切り替えた場合，このプロパティには前回の設定値が表示されます。

注意　プログラム実行中にこのプロパティを変更することはできません。

(3)　［レジスタ］
このカテゴリでは，プログラム実行中におけるステータス・バーへの PC 表示に関する設定を行います。

図 2.19　［レジスタ］カテゴリ

◢ レジスタ	
実行中にPC表示を行う	はい
PC表示更新間隔[ms]	500

- ［実行中に PC 表示を行う］
このプロパティは，プログラム実行中にステータス・バーへ PC 値を表示するかどうかを指定します。
[いいえ] を選択した場合，実行中のステータス・バーには " 実行中 " と表示されます。

注意 1.　プログラム実行中にこのプロパティを変更することはできません。

注意 2.　【RX100 シリーズ】
プログラム実行中の PC 表示機能をサポートしていないため，非表示となります。

- ［PC 表示更新間隔 [ms]］
このプロパティは，［実行中に PC 表示を行う］プロパティにおいて［はい］を指定した場合にのみ表示されます。
プログラム実行中にステータス・バーの PC 表示を更新する間隔を 100ms 単位で指定します。
直接入力により，100 ～ 65500 の整数（100ms 未満は端数切り上げ）を指定してください。（デフォルトでは［500］が指定されます。）
なお，［実行中に PC 表示を行う］プロパティにおいて指定値を［いいえ］から［はい］へ切り替えた場合，このプロパティには前回の設定値が表示されます。

注意　プログラム実行中にこのプロパティを変更することはできません。

(4)　［ブレーク］
このカテゴリでは，ブレーク機能に関する設定を行います。

図 2.20　［ブレーク］カテゴリ

◢ ブレーク	
優先的に使用するブレークポイントの種類	ハードウエア・ブレーク

- ［優先的に使用するブレークポイントの種類］
エディタ パネル／逆アセンブル パネルにおいて，マウスのワンクリック操作でブレークポイントを設定する際に使用するブレークポイントの種別を次のドロップダウン・リストより指定します。
優先して使用するブレークポイントを使い切った状態で，ブレークポイントを設定した場合，自動的にもう片方のブレークポイントを設定します。
なお，ブレークポイントについての詳細は，「2.10.2 任意の場所で停止する（ブレークポイント）」を参照してください。

ソフトウエア・ブレーク	ソフトウエア・ブレークポイントを優先的に設定します。
ハードウエア・ブレーク	ハードウエア・ブレークポイントを優先的に設定します（デフォルト）。

(5)　［システム］

このカテゴリでは，エミュレーションシステムに関する設定を行います。
実行開始前およびブレーク後の指定ルーチン実行についての詳細は，「2.9.4 指定ルーチンを実行する【E1】【E20】【EZ Emulator】」を参照してください。

図 2.21　　[システム] カテゴリ

システム	
内蔵プログラムROMを書き換えるプログラムをデバッグする	いいえ
内蔵データフラッシュを書き換えるプログラムをデバッグする	いいえ
実行開始前に指定ルーチンを実行する	はい
実行開始前に実行するルーチン	
ブレーク後に指定ルーチンを実行する	はい
ブレーク後に実行するルーチン	
指定ルーチン実行機能用ワークRAM開始アドレス	HEX 1FDD0
指定ルーチン実行機能用ワークRAMサイズ[バイト]	560

- [内蔵プログラム ROM を書き換えるプログラムをデバッグする]
  内蔵プログラム ROM 領域を書き換えるプログラム (ROM P/E モードを使用したプログラムなど) をデバッグするか否かを指定します。

  注意　　E1 と接続中にこのプロパティを変更することはできません。

- [内蔵データフラッシュを書き換えるプログラムをデバッグする]
  内蔵データフラッシュ領域を書き換えるプログラム (データフラッシュ P/E モードを使用したプログラムなど) をデバッグするか否かを指定します。

  注意　　E1 と接続中にこのプロパティを変更することはできません。

- [実行開始前に指定ルーチンを実行する]
  ユーザ・プログラム実行前に，指定ルーチンを実行する必要があるか否かを指定します。

  注意　　プログラム実行中にこのプロパティを変更することはできません。

- [実行開始前に実行するルーチン]
  ユーザ・プログラム実行直前に実行するアドレスを指定します。
  このプロパティは，[実行開始前に指定ルーチンを実行する] プロパティにおいて，[はい] が指定された場合のみ表示されます。

  注意　　プログラム実行中にこのプロパティを変更することはできません。

- [ブレーク後に指定ルーチンを実行する]
  ユーザ・プログラムブレーク後に，指定ルーチンを実行する必要があるか否かを指定します。

  注意　　プログラム実行中にこのプロパティを変更することはできません。

- [ブレーク後に実行するルーチン]
  ユーザ・プログラムブレーク直後に実行するアドレスを指定します。
  このプロパティは，[ブレーク後に指定ルーチンを実行する] プロパティにおいて，[はい] が指定された場合のみ表示します。

  注意　　プログラム実行中にこのプロパティを変更することはできません。

- [指定ルーチン実行機能用ワーク RAM 開始アドレス]
  指定ルーチン実行機能用のワーク RAM 配置アドレスを指定します。なお，アドレス値は 4 バイト単位の値を指定してください。(入力値が 4 バイト単位の値でなかった場合は，自動的に補正されます。)
  指定したワーク RAM の配置アドレスから [指定ルーチン実行機能用ワーク RAM サイズ [バイト]] プロパティの表示サイズ分は，デバッガのファームウェアが使用します。
  このプロパティは，[実行開始前に指定ルーチンを実行する] プロパティまたは，[ブレーク後に指定ルーチンを実行する] プロパティのいずれかにおいて [はい] が指定された場合のみ表示します。

  注意 1.　　プログラム実行中にこのプロパティを変更することはできません。

  注意 2.　　ユーザ・プログラムで使用しない領域を指定してください。

- [指定ルーチン実行機能用ワーク RAM サイズ [バイト]]
  指定ルーチン実行機能用のワーク RAM のサイズを表示します。
  このプロパティは，[実行開始前に指定ルーチンを実行する] プロパティまたは，[ブレーク後に指定ルーチンを実行する] プロパティのいずれかにおいて [はい] が指定された場合のみ表示します。

(6)　[トレース]
このカテゴリでは，トレース機能に関する設定を行います。

図 2.22 [トレース] カテゴリ

トレース	
トレース機能の用途	トレース
トレース・メモリを使い切った後の動作	トレース・メモリを上書きし実行を続ける
トレース・データ種別	分岐＋データアクセス
データアクセスのバス・マスタ	CPU
タイム・スタンプ出力	はい
トレース・クロック・カウント・ソース[MHz]	
トレース・クロック・カウント・ソースの分周比	分周なし

- [トレース機能の用途]
  トレース機能の用途として [トレース] を表示します。
  このプロパティ値を変更することはできません。

- [トレース・メモリを使い切った後の動作]
  トレースの取得モードを指定します。
  取得モードは，次のドロップダウン・リストより指定します。

トレース・メモリを上書きし実行を続ける	トレース・メモリがいっぱいになっても，トレース・データの上書きを続けます。
トレースを停止する	トレース・メモリがいっぱいになったら，トレース・データの書き込みを中止します。
停止する	トレース・メモリがいっぱいになった時点でブレークします。

- [トレース・データ種別]
  トレース取得するデータ種別をドロップダウン・リストより指定します。
  指定可能なデータ種別はマイクロコントローラのシリーズにより異なります。
  ドロップダウン・リストには，次のデータ種別が表示されます。

  - 【RX600 シリーズ】
    分岐 , 分岐＋データアクセス , データアクセス

  - 【RX100, RX200 シリーズ】
    分岐 , データアクセス

- [データアクセスのバス・マスタ]【RX71M, RX64M グループ】
  このプロパティは，[トレース・データ種別] プロパティにおいて [分岐＋データアクセス] または [データアクセス] を指定した場合にのみ表示されます。
  データアクセスのバス・マスタをドロップダウン・リストより指定します。
  ドロップダウン・リストには，次のバス・マスタが表示されます。

  - CPU（デフォルト）, DMAC/DTC

  データアクセスのトレース結果については，指定したバス・マスタからのトレース結果のみをトレースパネルに出力します。

  注意 1. プログラム実行中にこのプロパティを変更することはできません。

  注意 2. データアクセスのバス・マスタ選択機能が無いマイクロコントローラの場合は，[データアクセスのバス・マスタ]【RX71M, RX64M グループ】プロパティは表示されません。その場合，バス・マスタは [CPU] 固定となります。

- [タイム・スタンプ出力]
  収集するトレース・データにタイム・スタンプ情報を付加するか否かを指定します。
  このプロパティは，マイクロコントローラのシリーズや [トレース・データ種別] プロパティでの指定値によって変更可否が次のように切り替わります。

マイクロコントローラ	データ種別	変更可否
RX600 シリーズ	分岐，分岐 + データアクセス，データアクセス	プログラム停止時のみ変更可能
RX200 シリーズ	分岐	変更不可 [いいえ] が固定表示
	データアクセス	プログラム停止時のみ変更可能

マイクロコントローラ	データ種別	変更可否
RX100 シリーズ	分岐，データアクセス	変更不可 [いいえ] が固定表示

- ［トレース・クロック・カウント・ソース [MHz]］
このプロパティは，［タイム・スタンプ出力］プロパティにおいて［はい］を指定した場合にのみ表示されます。
カウント値からタイム・スタンプを算出するためのカウント・ソースを入力します。
直接入力により，0.0001 - 999.999 の間で指定してください。
なお，プロパティが空欄の場合，［接続用設定］タブの［クロック］カテゴリ内の［動作周波数 [MHz]］プロパティの設定値がカウント・ソースとして代用されます。

注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. プロパティの設定値は , 収集済みのトレース・データには反映しません。プロパティ設定後に収集したトレース・データから反映します。

- ［トレース・クロック・カウント・ソースの分周比］【RX71M, RX64M グループ】
このプロパティは，［タイム・スタンプ出力］プロパティにおいて［はい］を指定した場合にのみ表示されます。
タイム・スタンプのカウント・ソースの分周比をドロップダウン・リストより指定します。
ドロップダウン・リストには，次の分周比が表示されます。

- 分周なし（デフォルト）, 16 分周 , 256 分周 , 4096 分周

［トレース・クロック・カウント・ソース [MHz]］プロパティで指定した周波数をさらに指定した値で分周（周波数を 1/n 倍に変換）した値が実際のタイム・スタンプの単位（1 カウントあたりの周波数）になります。

注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. トレースのタイム・スタンプ分周機能が無いマイクロコントローラの場合は，［トレース・クロック・カウント・ソースの分周比］【RX71M, RX64M グループ】プロパティは表示されません。その場合，［分周なし］固定となります。

(7) ［タイマ］【RX600 シリーズ】
このカテゴリでは，タイマ機能に関する設定を行います。

図 2.23 [タイマ] カテゴリ

- ［カウンタを 64 ビットで使用する］
計測カウンタを 32 ビット ×2 本で使用するか，64 ビット ×1 本で使用するかを指定します。

2.3.2.3 ［ダウンロード・ファイル設定］タブ

［ダウンロード・ファイル設定］タブでは，デバッグ・ツールにダウンロードを実行する際の設定を行います。
各カテゴリ内の設定についての詳細は，「2.5.1 ダウンロードを実行する」を参照してください。

2.3.2.4 ［フック処理設定］タブ

［フック処理設定］タブでは，デバッグ・ツールにフック処理の設定を行います。
フック処理，および各カテゴリ内の設定についての詳細は，「2.18 フック処理を設定する」を参照してください。

2.3.3 【E20】の場合

E20 を使用する場合の動作環境の設定を次のプロパティ パネルで行います。
なお，ターゲット・システムとの通信方式（FINE 通信【E20(Serial)】/JTAG 通信【E20(JTAG)】）により，プロパティ パネルの内容は異なります。

図 2.24　　動作環境設定【E20(Serial)】（プロパティ パネル）

図 2.25　　動作環境設定【E20(JTAG)】（プロパティ パネル）

プロパティ パネル上の該当するタブを選択し，次の設定を順次行ってください。

2.3.3.1　［接続用設定］タブ
2.3.3.2　［デバッグ・ツール設定］タブ
2.3.3.3　［ダウンロード・ファイル設定］タブ
2.3.3.4　［フック処理設定］タブ

2.3.3.1　［接続用設定］タブ

［接続用設定］タブでは，次に示すカテゴリごとに，デバッグ・ツールとの接続に関する設定を行います。

(1)　［内蔵 ROM/RAM］
(2)　［クロック］
(3)　［エミュレータとの接続］
(4)　［ターゲット・ボードとの接続］
(5)　［フラッシュ］
(6)　［CPU の動作モード］
(7)　［外部フラッシュ］

(1)　［内蔵 ROM/RAM］
このカテゴリでは，内蔵 ROM/RAM に関する設定を表示します。

図 2.26　　［内蔵 ROM/RAM］カテゴリ

- ［内蔵 ROM サイズ [K バイト]］
  エミュレーションする内蔵 ROM サイズを表示します（単位：K バイト）。
  このプロパティ値を変更することはできません。

- ［内蔵 RAM サイズ [バイト]］
  エミュレーションする内蔵 RAM サイズを表示します（単位：バイト）。
  このプロパティ値を変更することはできません。

- ［データフラッシュ・メモリ・サイズ [K バイト]］
  選択しているマイクロコントローラのデータフラッシュ・メモリ領域のサイズを表示します（単位：K バイト）。
  選択しているマイクロコントローラがデータフラッシュ・メモリ非搭載品の場合は，［0］を表示します。
  このプロパティ値を変更することはできません。

(2)　［クロック］
このカテゴリでは，クロックに関する設定を行います。

注意　E20 と接続中にこのカテゴリ内のプロパティを変更することはできません。

図 2.27　［クロック］カテゴリ【HOCO】

◢ クロック	
メイン・クロック・ソース	HOCO
動作周波数[MHz]	25.0000
内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する	いいえ

図 2.28　［クロック］カテゴリ【EXTAL】

◢ クロック	
メイン・クロック・ソース	EXTAL
メイン・クロック周波数[MHz]	12.5000
動作周波数[MHz]	25.0000
内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する	いいえ

- ［メイン・クロック・ソース］
  メイン・クロックを EXTAL 周波数と内蔵 HOCO から指定します。
  内蔵 HOCO を持たないマイクロコントローラの場合は，[EXTAL] を表示します。

- ［メイン・クロック周波数 [MHz]］
  メイン・クロックの周波数（逓信前）を指定します。
  EXTAL 周波数値として 0.0001 ～ 99.9999（単位：MHz）の範囲で数値を直接入力します。
  入力値の少数第 5 位以下は切り捨てられ，指定可能範囲外の入力値は，0.0001(0 以下だった時) または 99.9999(100 以上だった時) に丸め込みます。
  このプロパティは，［メイン・クロック・ソース］プロパティで [EXTAL] を指定した場合のみ表示されます。

- ［動作周波数 [MHz]］
  動作周波数（ICLK）を指定します。
  0.0001 ～ 999.9999（単位：MHz）の範囲で数値を直接入力します。
  入力値の少数第 5 位以下は切り捨てられ，指定可能範囲外の入力値は，0.0001(0 以下だった時) または 999.9999(1000 以上だった時) に丸め込みます。

- ［内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する］
  内蔵フラッシュ・メモリ書き換え時に，デバッガによるクロック操作を許可するか否かを指定します。

  注意　【E20【RX630, RX631, RX63N, RX63T, RX210, RX21A, RX220, RX110, RX111, RX113】】
  [はい] を指定した場合，マイクロコントローラの FlashIF クロック（FCLK）が動作保証範囲外（LOCO やサブクロック動作中など）でデバッガによる内蔵フラッシュ ROM の書き換えを実施した時，E20 がクロック・ソースを切り替えます。なお，書き換え完了後は，元のクロック・ソースに復帰します。
  クロック操作を許可した場合，クロック・ソースを切り替えるため，内蔵フラッシュ・メモリ書き換え中の周辺クロック動作周波数が変更されます。
  クロック操作の許可設定は，プログラム実行やステップ実行などの実行系機能を実施した後の内蔵フラッシュ ROM を書き換える時に有効となります。なお，デバッグ・ツールの接続直後や CPU リセット発行時は，クロック操作の許可設定に関わらず FCLK が動作保証範囲外の場合は，強制的にクロック・ソースを切り替えます。

(3)　［エミュレータとの接続］
このカテゴリでは，E20 とホスト・マシンとの接続に関する設定を行います。

注意　E20 と接続中にこのカテゴリ内のプロパティを変更することはできません。

図 2.29　　[エミュレータとの接続] カテゴリ

◢ エミュレータとの接続	
エミュレータシリアルNo.	

- ［エミュレータシリアル No.］
  接続されている E20 のシリアル番号をドロップダウン・リストに全て表示します。
  接続する E20 のシリアル番号を指定してください。
  なお，ドロップダウン・リストはドロップダウン時に毎回更新します。

(4)　［ターゲット・ボードとの接続］
このカテゴリでは，E20 とターゲット・ボードとの接続に関する設定を行います。

図 2.30　　［ターゲット・ボードとの接続］カテゴリ【E20(Serial)】

◢ ターゲット・ボードとの接続	
エミュレータから電源供給をする(最大200mA)	いいえ
通信方式	FINE
FINEボーレート[bps]	2000000

図 2.31　　［ターゲット・ボードとの接続］カテゴリ【E20(JTAG)】

◢ ターゲット・ボードとの接続	
エミュレータから電源供給をする(最大200mA)	いいえ
通信方式	JTAG
JTAGクロック[MHz]	16.5

- ［エミュレータから電源供給をする (最大 200mA)］
  プロパティ値は，[いいえ] を表示します。
  E20 は電源供給機能をサポートしていません。

- ［通信方式］
  E20 がターゲット・システム上のマイクロコントローラと通信する際の通信方式を表示します。
  プロジェクト・ツリー パネルにて使用するデバッグ・ツールを選択する際に ”RX E20(Serial)” を選択した場合，このプロパティには [FINE] が表示されます。また，”RX E20(JTAG)” を選択した場合，[JTAG] が表示されます。
  このプロパティ値を変更することはできません。
  なお，デバッグ・ツールの選択についての詳細は，「2.3.1 使用するデバッグ・ツールを選択する」を参照してください。

- [JTAG クロック [MHz]]
  E20 がターゲット・システム上のマイクロコントローラと通信する際の通信速度 (JTAG クロック) をドロップダウン・リストより指定します。
  このプロパティは，［通信方式］プロパティにおいて [JTAG] の場合のみ表示されます。
  ドロップダウン・リストには，次の通信速度が表示されます。

  - 16.5(デフォルト), 12.38, 6.188, 3.094, 1.547

  注意 1.　E20 と接続中にこのプロパティを変更することはできません。

  注意 2.　ターゲット・システム上での JTAG 信号の配線長や配線方法によっては，選択した JTAG クロックで通信できない場合があります。この JTAG クロックを下げることによって，通信が出来るようになる場合があります。

- [FINE ボーレート [bps]]
  E20 がターゲット・システム上のマイクロコントローラと通信する際の通信速度 (FINE ボーレート) をドロップダウン・リストより指定します。
  このプロパティは，［通信方式］プロパティにおいて [FINE] の場合のみ表示されます。
  ドロップダウン・リストには，次の通信速度が表示されます。

  - 2000000(デフォルト), 750000, 500000, 250000

  注意 1.　E20 と接続中にこのプロパティを変更することはできません。

  注意 2.　ターゲット・システム上での FINE 信号の配線長や配線方法によっては，選択した FINE ボーレートで通信できない場合があります。この FINE 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。

(5) ［フラッシュ］
このカテゴリでは，フラッシュ書き換えに関する設定を行います。

図 2.32 ［フラッシュ］カテゴリ

フラッシュ	
IDコードの入力モード	IDコードを16進32桁で指定
IDコード	HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
ワークRAM開始アドレス	HEX 1000
ワークRAMサイズ[バイト]	1024

- ［ID コードの入力モード］
  ID コードの入力モードを指定します。

  注意 E20 と接続中にこのプロパティを変更することはできません。

- ［ID コード］
  ID コードプロテクトを解除するための ID コードを指定します。
  ［ID コードの入力モード］プロパティにおいて，[ID コードを 16 進 32 桁で指定] を指定した場合，ID コードは 32 桁の 16 進数で入力してください。また，[ID コードを 16 文字以内で指定 (ASCII)] を指定した場合，ID コードは 16 文字以内の ASCII 文字列で入力してください。

  注意 1. E20 と接続中にこのプロパティを変更することはできません。

  注意 2. ID コードを 32 桁の 16 進数で入力する場合は，32 ビットのデータ単位に並べて入力してください。
  フラッシュプログラマでは ID コードの並び順が CS+ と異なる場合があります。ご使用のフラッシュプログラマの ID コード入力の仕様を確認してください。

  注意 3. ID コードを ASCII 文字列で入力する場合，16 文字に満たない部分は ”0” で入力されます。

  注意 4. ID コードを含んだプログラムをダウンロードした場合，ID コードは FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF に書き換わります。

- ［ワーク RAM 開始アドレス］
  デバッガが使用するワーク RAM の配置アドレスを指定します。なお，アドレス値は 4 バイト単位の値を指定してください。（入力値が 4 バイト単位の値でなかった場合は，自動的に補正されます。）
  指定したワーク RAM の配置アドレスから［ワーク RAM サイズ [バイト]］プロパティの表示サイズ分は，デバッガのファームウェアが使用します。

  注意 1. E20 と接続中にこのプロパティを変更することはできません。

  注意 2. メモリ内容の退避・復帰が行われるためユーザ・プログラムでもこの領域を使用できますが，ワーク RAM に配置する領域は，以下の用途では使用できません。

  - DMA または DTC 機能の転送元・転送先
  - DTC ベクタテーブル，および転送情報配置アドレス
  - DMAC，DTC 起動要因の割り込みベクタ

- ［ワーク RAM サイズ [バイト]］
  デバッガが使用するワーク RAM のサイズを表示します。

(6) ［CPU の動作モード］
このカテゴリでは，エミュレーションするマイクロコントローラの動作モードに関する設定を行います。

注意 E20 と接続中にこのカテゴリ内のプロパティを変更することはできません。

図 2.33 [CPU の動作モード] カテゴリ

CPUの動作モード	
モード端子設定	シングルチップモード
レジスタ設定	シングルチップモード
エンディアン	Little-endianデータ
起動バンクを変更する	はい
起動バンク	バンク0

- ［モード端子設定］
  マイクロコントローラのモード端子で設定している動作モードを指定します。

- ［USB ブートプログラムの消去を許可する］
  USB ブートプログラムが格納されたマイクロコントローラを使用してユーザブートモードで起動する際に，USB ブートプログラムを消去するか否かを指定します。
  このプロパティは，［モード端子設定］プロパティにおいて［ユーザブートモード］を指定した場合のみ表示されます。ただし，ユーザブートモードで起動する際に，ユーザブート領域上の USB ブートプログラムを消去する必要のないマイクロコントローラの場合は表示されません。
- ［レジスタ設定］
  レジスタで設定する動作モードを指定します。
  指定可能な動作モードは，マイクロコントローラによって異なります。
- ［エンディアン］
  プロジェクトのエンディアンを表示します。
  プロジェクトからエンディアンの設定情報を取得し，その値を表示します。
  デバッグ・ツールと切断中のみ指定することができます。
- ［起動バンクを変更する］【RX651-2M, RX65N-2M グループ】
  デバッグ・ツールと接続する際に起動バンクを変更するか否かを指定します。
  起動バンクを変更して接続する場合は［はい］を選択してください（デフォルトでは［いいえ］が指定されます）。
- ［起動バンク］【RX65N-2M, RX651-2M グループ】
  デバッグ・ツールと接続する際に変更する起動バンクを指定します。
  このプロパティは，［起動バンクを変更する］【RX651-2M, RX65N-2M グループ】プロパティにおいて［はい］を指定した場合のみ表示されます。

(7)　［外部フラッシュ］
このカテゴリでは，外部フラッシュに関する設定を行います。
このカテゴリ内の設定は，外部フラッシュ・メモリへダウンロードする場合に必要となります。各プロパティ内の設定についての詳細は，「2.5.2.5 外部フラッシュ・メモリへダウンロードする【E1】【E20】」を参照してください。

注意　E20 と接続中にこのカテゴリ内のプロパティを変更することはできません。

2.3.3.2 ［デバッグ・ツール設定］タブ

［デバッグ・ツール設定］タブでは，次に示すカテゴリごとに，デバッグ・ツールの基本設定を行います。

(1)　［メモリ］
(2)　［実行中のメモリ・アクセス］
(3)　［レジスタ］
(4)　［ブレーク］
(5)　［システム］
(6)　［トレース］
(7)　［タイマ］【RX600 シリーズ】
(8)　［カバレッジ］【RX71M, RX64M グループ】

(1)　［メモリ］
このカテゴリでは，メモリに関する設定を行います。

図 2.34 ［メモリ］カテゴリ

◢ メモリ	
◢ メモリ・マッピング	[23]
▷ [0]	内蔵RAM領域
▷ [1]	予約領域
▷ [2]	I/Oレジスタ領域
▷ [3]	予約領域
▷ [4]	I/Oレジスタ領域
▷ [5]	I/Oレジスタ領域
▷ [6]	予約領域
▷ [7]	I/Oレジスタ領域
▷ [8]	I/Oレジスタ領域
▷ [9]	I/Oレジスタ領域
▷ [10]	データフラッシュ領域
▷ [11]	予約領域
▷ [12]	その他のメモリ領域
▷ [13]	予約領域
▷ [14]	I/Oレジスタ領域
▷ [15]	予約領域
▷ [16]	I/Oレジスタ領域
▷ [17]	予約領域
▷ [18]	その他のメモリ領域
▷ [19]	予約領域
▷ [20]	その他のメモリ領域
▷ [21]	予約領域
▷ [22]	内蔵ROM領域
メモリ書き込み時にベリファイを行う	はい

- ［メモリ・マッピング］
  現在のメモリ・マッピングの状況が，メモリ領域の種別ごとに詳細表示されます。
  このパネル上でマッピング値を変更することはできません。
  I/O 保護領域を追加，または削除する必要がある場合は，［メモリ・マッピング］プロパティを選択することで設定欄右端に表示される [...] ボタンのクリックによりオープンするメモリ・マッピング ダイアログで行います。
  設定方法についての詳細は，メモリ・マッピング ダイアログの項を参照してください。
  このプロパティでは，メモリ領域の個数のみを表示します。
  [メモリ・マッピング] プロパティを展開すると，次のサブ項目が表示されます。

  - [メモリ種別]
    当該領域のメモリ種別を表示します。
    各メモリ種別は次の領域に該当します。

内蔵 ROM 領域	プログラム ROM 注 1 およびデータフラッシュ注 2 が該当します。
内蔵 RAM 領域	内蔵 RAM が該当します。注 4
I/O レジスタ領域	周辺 I/O レジスタが該当します。 エンディアンが異なる領域で分割して表示します。
外部領域 (CS7/CS6/.../CS0)	外部アドレス空間が該当します。 CS0 - CS7 を区別して表示します。
その他のメモリ領域	FCU-RAM 注 3, FCU ファーム注 3, ユーザーブートが該当します。
予約領域	上記以外の領域を表示します。
I/O 保護領域	外部領域でデバッガからリードしないアドレス範囲が該当します。 メモリ・マッピング ダイアログで登録します。

注 1. ダウンロード以外の操作（ メモリ パネルからの操作やライン・アセンブル）で内蔵フラッシュ ROM 領域を編集した場合，編集内容は次の実行操作時にフラッシュ ROM に書き込まれます。

注 2.　消去された状態のデータフラッシュ ROM の内容を参照すると，マイクロコントローラの仕様どおり不定値が表示されます。なお，デバッガからデータフラッシュ ROM の内容を書き換えた場合は，256 バイト単位で書き込みを行います。書き込まれた領域は不定値とはなりません。

注 3.　FCU-RAM 領域の内容をデバッガから書き換えないでください。また，FCU ファーム領域はデバッガから書き換えできません。

注 4.　【RX71M, RX64M グループ】
下記のオプション設定メモリを含む領域（0x00120040 ～ 0x0012006F 番地）は，内蔵 RAM 領域として表示されます。なお，本領域は，メモリパネルで値を設定することはできません。

- シリアルプログラマコマンド制御レジスタ（SPCC）
- OCD/ シリアルプログラマ ID 設定レジスタ（OSIS）
- エンディアン選択レジスタ（MDE）
- オプション機能選択レジスタ 0（OFS0）
- オプション機能選択レジスタ 1（OFS1）

- [開始アドレス]
  当該領域の開始アドレスを表示します。
- [終了アドレス]
  当該領域の終了アドレスを表示します。
- [アドレス幅 [ビット]]
  当該領域のアクセス幅を表示します。
  [メモリ種別] が外部領域については，デバッグ・ツールと切断中のみ変更可能です。
- [エンディアン]
  外部領域および I/O レジスタ領域のエンディアンを表示します。
  [メモリ種別] が外部領域については，デバッグ・ツールと切断中のみ変更可能です。

注意　デバッグ・ツールと接続することにより（「2.4.1 CS+ にデバッグ・ツールを接続する」参照），各メモリ種別ごとの詳細表示を行います。

- ［メモリ書き込み時にベリファイを行う］
  メモリ値の初期化を行う際に，ベリファイを行うか否かをドロップダウン・リストにより指定します。
  ［はい］を指定した場合，ダウンロードした際，ウォッチ パネルやメモリ パネルで値を変更した際にベリファイを行います。

(2)　［実行中のメモリ・アクセス］
このカテゴリでは，プログラム実行中におけるメモリ・アクセスに関する設定を行います。
このカテゴリ内の設定は，リアルタイム表示更新機能を使用する場合に必要となります。リアルタイム表示更新機能についての詳細は，「2.11.1.4 プログラム実行中にメモリの内容を表示／変更する」を参照してください。

図 2.35　［実行中のメモリ・アクセス］カテゴリ【E20(Serial)】【E20(JTAG)【RX200 シリーズ】】

実行中のメモリ・アクセス	
実行を一瞬停止してアクセスする	いいえ
実行中に表示更新を行う	はい
表示更新間隔[ms]	500

図 2.36　［実行中のメモリ・アクセス］カテゴリ【E20(JTAG)【RX600 シリーズ】】

実行中のメモリ・アクセス	
実行を一瞬停止してアクセスする	はい
実行中に表示更新を行う	はい
表示更新間隔[ms]	500
リアルタイム表示更新を自動設定する	はい

- ［実行を一瞬停止してアクセスする］
  プログラム実行中に，メモリに対してアクセスを許可するか否かをドロップダウン・リストにより指定します。
  アクセスを許可する場合は［はい］を選択してください（デフォルトでは［いいえ］が指定されます）。

- ［実行中に表示更新を行う］
  プログラム実行中に，ウォッチ パネル／メモリ パネルの表示内容を更新するか否かをドロップダウン・リストにより指定します。
  表示内容の更新を行う場合は［はい］を選択してください（デフォルト）。

  注意　　プログラム実行中にこのプロパティを変更することはできません。

- ［表示更新間隔 [ms]］
  このプロパティは，［実行中に表示更新を行う］プロパティにおいて［はい］を指定した場合にのみ表示されます。
  プログラム実行中に，ウォッチ パネル／メモリ パネルの表示内容を更新する間隔を 100 ms 単位で指定します。
  直接入力により，100 ～ 65500 の範囲の整数（100 ms 未満の端数切り上げ）を指定してください（デフォルトでは［500］が指定されます）。
  なお，［実行中に表示更新を行う］プロパティにおいて指定値を［いいえ］から［はい］へ切り替えた場合，このプロパティには前回の設定値が表示されます。

  注意　　プログラム実行中にこのプロパティを変更することはできません。

- ［リアルタイム表示更新を自動設定する］
  RRM 領域を自動的に設定するか否かを指定します。
  このプロパティは，［トレース機能の用途］プロパティにおいて [リアルタイム RAM モニタ] を指定した場合にのみ表示され，[トレース] を指定した場合は [いいえ] の状態で表示されません。

  注意　　プログラム実行中にこのプロパティを変更することはできません。

(3)　［レジスタ］
このカテゴリでは，プログラム実行中におけるステータス・バーへの PC 表示に関する設定を行います。

図 2.37　　［レジスタ］カテゴリ

◢ レジスタ	
実行中にPC表示を行う	はい
PC表示更新間隔[ms]	500

- ［実行中に PC 表示を行う］
  このプロパティは，プログラム実行中にステータス・バーへ PC 値を表示するかどうかを指定します。
  [いいえ] を選択した場合，実行中のステータス・バーには " 実行中 " と表示されます。

  注意 1.　　プログラム実行中にこのプロパティを変更することはできません。

  注意 2.　　【RX100 シリーズ】
  プログラム実行中の PC 表示機能をサポートしていないため，非表示となります。

- ［PC 表示更新間隔 [ms]］
  このプロパティは，［実行中に PC 表示を行う］プロパティにおいて［はい］を指定した場合にのみ表示されます。
  プログラム実行中にステータス・バーの PC 表示を更新する間隔を 100ms 単位で指定します。
  直接入力により，100 ～ 65500 の整数（100ms 未満は端数切り上げ）を指定してください。（デフォルトでは［500］が指定されます。）
  なお，［実行中に PC 表示を行う］プロパティにおいて指定値を［いいえ］から［はい］へ切り替えた場合，このプロパティには前回の設定値が表示されます。

  注意　　プログラム実行中にこのプロパティを変更することはできません。

(4)　［ブレーク］
このカテゴリでは，ブレーク機能に関する設定を行います。

図 2.38　　［ブレーク］カテゴリ

◢ ブレーク	
優先的に使用するブレークポイントの種類	ハードウエア・ブレーク

- ［優先的に使用するブレークポイントの種類］
  エディタ パネル／逆アセンブル パネルにおいて，マウスのワンクリック操作でブレークポイントを設定する際に使用するブレークポイントの種別を次のドロップダウン・リストより指定します。
  優先して使用するブレークポイントを使い切った状態で，ブレークポイントを設定した場合，自動的にもう片方のブレークポイントを設定します。

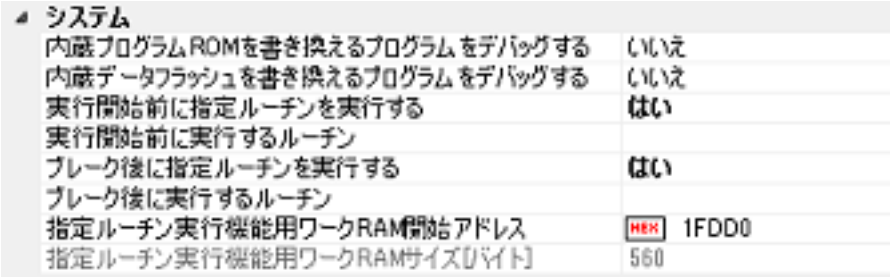
なお，ブレークポイントについての詳細は，「2.10.2 任意の場所で停止する（ブレークポイント）」を参照してください。

ソフトウエア・ブレーク	ソフトウエア・ブレークポイントを優先的に設定します。
ハードウエア・ブレーク	ハードウエア・ブレークポイントを優先的に設定します（デフォルト）。

(5) ［システム］

このカテゴリでは，エミュレーションシステムに関する設定を行います。
実行開始前およびブレーク後の指定ルーチン実行についての詳細は，「2.9.4 指定ルーチンを実行する【E1】【E20】【EZ Emulator】」を参照してください。

図 2.39 [システム] カテゴリ



- ［内蔵プログラム ROM を書き換えるプログラムをデバッグする］
  内蔵プログラム ROM 領域を書き換えるプログラム (ROM P/E モードを使用したプログラムなど) をデバッグするか否かを指定します。

  注意　E20 と接続中にこのプロパティを変更することはできません。

- ［内蔵データフラッシュを書き換えるプログラムをデバッグする］
  内蔵データフラッシュ領域を書き換えるプログラム (データフラッシュ P/E モードを使用したプログラムなど) をデバッグするか否かを指定します。

  注意　E20 と接続中にこのプロパティを変更することはできません。

- ［実行開始前に指定ルーチンを実行する］
  ユーザ・プログラム実行前に，指定ルーチンを実行する必要があるか否かを指定します。

  注意　プログラム実行中にこのプロパティを変更することはできません。

- ［実行開始前に実行するルーチン］
  ユーザ・プログラム実行直前に実行するアドレスを指定します。
  このプロパティは，［実行開始前に指定ルーチンを実行する］プロパティにおいて，[はい] が指定された場合のみ表示されます。

  注意　プログラム実行中にこのプロパティを変更することはできません。

- ［ブレーク後に指定ルーチンを実行する］
  ユーザ・プログラムブレーク後に，指定ルーチンを実行する必要があるか否かを指定します。

  注意　プログラム実行中にこのプロパティを変更することはできません。

- ［ブレーク後に実行するルーチン］
  ユーザ・プログラムブレーク直後に実行するアドレスを指定します。
  このプロパティは，［ブレーク後に指定ルーチンを実行する］プロパティにおいて，[はい] が指定された場合のみ表示します。

  注意　プログラム実行中にこのプロパティを変更することはできません。

- ［指定ルーチン実行機能用ワーク RAM 開始アドレス］
  指定ルーチン実行機能用のワーク RAM 配置アドレスを指定します。なお，アドレス値は 4 バイト単位の値を指定してください。（入力値が 4 バイト単位の値でなかった場合は，自動的に補正されます。）
  指定したワーク RAM の配置アドレスから［指定ルーチン実行機能用ワーク RAM サイズ [バイト]］プロパティの表示サイズ分は，デバッガのファームウェアが使用します。
  このプロパティは，［実行開始前に指定ルーチンを実行する］プロパティまたは，［ブレーク後に指定ルーチンを実行する］プロパティのいずれかにおいて [はい] が指定された場合のみ表示します。

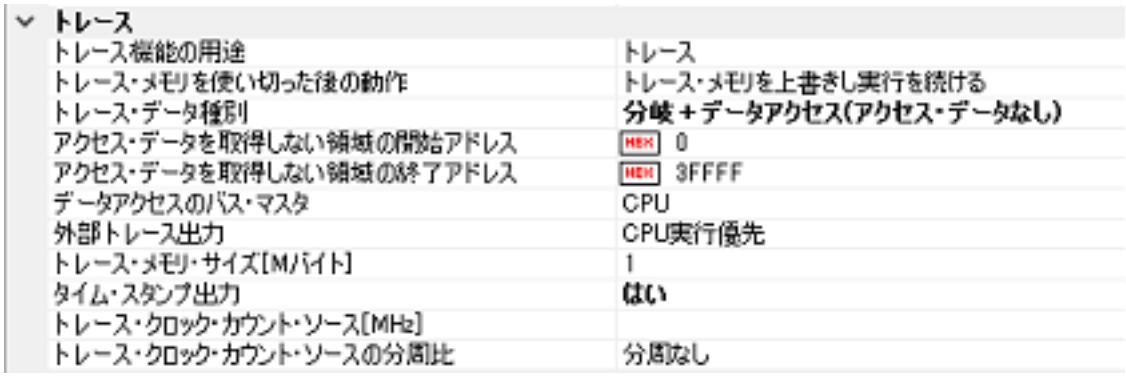
  注意 1.　プログラム実行中にこのプロパティを変更することはできません。

注意 2.　ユーザ・プログラムで使用しない領域を指定してください。

- ［指定ルーチン実行機能用ワーク RAM サイズ [バイト]］
  指定ルーチン実行機能用のワーク RAM のサイズを表示します。
  このプロパティは，［実行開始前に指定ルーチンを実行する］プロパティまたは，［ブレーク後に指定ルーチンを実行する］プロパティのいずれかにおいて [はい] が指定された場合のみ表示します。

(6)　［トレース］
このカテゴリでは，トレース機能に関する設定を行います。

図 2.40　［トレース］カテゴリ



トレース	
トレース機能の用途	トレース
トレース・メモリを使い切った後の動作	トレース・メモリを上書きし実行を続ける
トレース・データ種別	分岐＋データアクセス(アクセス・データなし)
アクセス・データを取得しない領域の開始アドレス	HEX 0
アクセス・データを取得しない領域の終了アドレス	HEX 3FFFF
データアクセスのバス・マスタ	CPU
外部トレース出力	CPU実行優先
トレース・メモリ・サイズ[Mバイト]	1
タイム・スタンプ出力	はい
トレース・クロック・カウント・ソース[MHz]	
トレース・クロック・カウント・ソースの分周比	分周なし

- ［トレース機能の用途］
  トレース機能を利用したリアルタイム RAM モニタ機能 (RRM 機能) として使用するか否かを指定します。
  [リアルタイム RAM モニタ] を指定した場合，一部のトレース機能が使用できなくなります。
  トレース機能の制限事項についての詳細は，「2.11.1.4　プログラム実行中にメモリの内容を表示／変更する」を参照してください。
  なお，このプロパティは，プログラム実行停止中のみ変更可能です。

  注意　リアルタイム RAM モニタ機能を使用する場合は，E20 とターゲット・ボードを 38 ピンで JTAG 接続している必要があります。それ以外の接続の場合，あるいは，38 ピン /14 ピン変換アダプタを使用している場合は [トレース] を選択してください。

- ［トレース・メモリを使い切った後の動作］
  トレースの取得モードを指定します。
  取得モードは，次のドロップダウン・リストより指定します。

トレース・メモリを上書きし実行を続ける	トレース・メモリがいっぱいになっても，トレース・データの上書きを続けます。
トレースを停止する	トレース・メモリがいっぱいになったら，トレース・データの書き込みを中止します。
停止する	トレース・メモリがいっぱいになった時点でブレークします。

  注意　【E20(JTAG)【RX600 シリーズ】】
  ［トレース機能の用途］プロパティにおいて，[リアルタイム RAM モニタ] を指定した場合，プロパティ値は [トレース・メモリを上書きし実行を続ける] と表示され変更不可状態となります。

- ［トレース・データ種別］
  トレース取得するデータ種別をドロップダウン・リストより指定します。
  指定可能なデータ種別はマイクロコントローラのシリーズにより異なります。
  ドロップダウン・リストには，次のデータ種別が表示されます。

  - 【RX65N-2M, RX651-2M グループ】
    分岐 , 分岐＋データアクセス , データアクセス，分岐＋データアクセス (アクセス・データなし)，データアクセス (アクセス・データなし)
  - 【RX600 シリーズ】
    分岐 , 分岐＋データアクセス , データアクセス
  - 【RX100, RX200 シリーズ】
    分岐 , データアクセス

注意　【E20(JTAG)【RX600 シリーズ】】
［トレース機能の用途］プロパティにおいて，[リアルタイム RAM モニタ] を指定した場合，プロパティ値は [データアクセス] と表示され変更不可状態となります。

- ［アクセス・データを取得しない領域の開始アドレス］【RX65N-2M, RX651-2M グループ】
アクセス・データを取得しない領域の開始アドレスを指定します。

- ［アクセス・データを取得しない領域の終了アドレス］【RX65N-2M, RX651-2M グループ】
アクセス・データを取得しない領域の終了アドレスを指定します。

- ［データアクセスのバス・マスタ］【RX71M, RX64M グループ】
データアクセスを発生させたバス・マスタを指定します。
このプロパティは，［トレース・データ種別］プロパティにおいて［分岐＋データアクセス］または［データアクセス］を指定した場合にのみ表示されます。
ドロップダウン・リストには，次のバス・マスタが表示されます。

  - CPU（デフォルト），DMAC/DTC

［トレース機能の用途］プロパティが［トレース］の場合、データアクセスのトレース結果については，指定したバス・マスタからのトレース結果のみをトレースパネルに出力します。

注意 1.　プログラム実行中にこのプロパティを変更することはできません。

注意 2.　［トレース機能の用途］プロパティが［リアルタイム RAM モニタ］の場合，パス・マスタは［CPU］固定となり変更できません。

注意 3.　データアクセスのバス・マスタ選択機能が無いマイクロコントローラの場合は，［データアクセスのバス・マスタ］【RX71M, RX64M グループ】プロパティは表示されません。その場合、バス・マスタは［CPU］固定となります。

- [外部トレース出力]【E20(JTAG)】
トレース取得データの外部出力方法を指定します。
外部出力方法は，次のドロップダウン・リストより指定します。

CPU 実行優先	トレース出力より CPU 実行を優先します。 出力されるトレース情報にロストが発生する場合があります。
トレース出力優先	CPU 実行よりトレース出力を優先します。 トレース出力のために CPU 実行が停止するため，リアルタイム性が損なわれます。
外部出力しない	トレース情報は出力されません。 マイクロコントローラの内蔵バッファのみを使用します。

注意 1.　［トレース機能の用途］プロパティにおいて，[リアルタイム RAM モニタ] が指定された場合，ドロップダウンリストより [外部出力しない] は指定できません。

注意 2.　[CPU 実行優先] または [トレース出力優先] を指定した場合，トレース パネルにトレース・データが表示された状態でステップ・イン実行すると正しいトレース・データが表示されない場合があります。

注意 3.　【RX71M, RX64M グループ】
［外部出力しない］から［CPU 実行優先］，または［トレース出力優先］に変更した場合，タイマ計測結果が初期化されます。

- [トレース・メモリ・サイズ [M バイト]]【E20(JTAG)】
トレース・データを保持するメモリサイズを指定します。
ドロップダウン・リストには，次のメモリサイズが表示されます。

  - 1(デフォルト), 2, 4, 8, 16, 32

注意　［トレース機能の用途］プロパティにおいて，[リアルタイム RAM モニタ] を指定した場合，プロパティ値は [1] と表示され変更不可状態となります。

- ［タイム・スタンプ出力］
収集するトレース・データにタイム・スタンプ情報を付加するか否かを指定します。
このプロパティは，［トレース機能の用途］プロパティにおいて，[トレース] を指定した場合のみ選択可能となります。
また，マイクロコントローラのシリーズや［トレース・データ種別］プロパティでの指定値によって変更可否が次のように切り替わります。

マイクロコントローラ	データ種別	変更可否
RX65N-2M, RX651-2M グループ	分岐，分岐 + データアクセス，データアクセス，分岐＋データアクセス (アクセス・データなし)，データアクセス (アクセス・データなし)	プログラム停止時のみ変更可能
RX600 シリーズ	分岐，分岐 + データアクセス，データアクセス	プログラム停止時のみ変更可能
RX200 シリーズ	分岐	変更不可 [いいえ] が固定表示
	データアクセス	プログラム停止時のみ変更可能
RX100 シリーズ	分岐，データアクセス	変更不可 [いいえ] が固定表示

- ［トレース・クロック・カウント・ソース [MHz]］
このプロパティは，［タイム・スタンプ出力］プロパティにおいて［はい］を指定した場合にのみ表示されます。
カウント値からタイム・スタンプを算出するためのカウント・ソースを入力します。
直接入力により，0.0001 - 999.999 の間で指定してください。
なお，プロパティが空欄の場合，［接続用設定］タブの［クロック］カテゴリ内の［動作周波数 [MHz]］プロパティの設定値がカウント・ソースとして代用されます。

注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. プロパティの設定値は , 収集済みのトレース・データには反映しません。プロパティ設定後に収集したトレース・データから反映します。

- ［トレース・クロック・カウント・ソースの分周比］【RX71M, RX64M グループ】
タイム・スタンプのカウント・ソースの分周比を指定します。
このプロパティは，［タイム・スタンプ出力］プロパティにおいて［はい］を指定した場合にのみ表示されます。
ドロップダウン・リストには，次の分周比が表示されます。

- 分周なし（デフォルト）, 16 分周 , 256 分周 , 4096 分周

［トレース・クロック・カウント・ソース [MHz]］プロパティで指定した周波数をさらに指定した値で分周（周波数を 1/n 倍に変換）した値が実際のタイム・スタンプの単位（1 カウントあたりの周波数）になります。

注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. トレースのタイム・スタンプ分周機能が無いマイクロコントローラの場合は，［トレース・クロック・カウント・ソースの分周比］【RX71M, RX64M グループ】プロパティは表示されません。その場合，［分周なし］固定となります。

(7) ［タイマ］【RX600 シリーズ】
このカテゴリでは，タイマ機能に関する設定を行います。

図 2.41 [タイマ] カテゴリ

- ［カウンタを 64 ビットで使用する］
計測カウンタを 32 ビット ×2 本で使用するか，64 ビット ×1 本で使用するかを指定します。

(8) ［カバレッジ］【RX71M, RX64M グループ】
このカテゴリでは，カバレッジ機能に関する設定を行います。
カバレッジ機能，およびこのカテゴリ内の設定についての詳細は，「2.15 カバレッジの測定【シミュレータ】【E20【RX71M, RX64M グループ】】」を参照してください。

2.3.3.3 ［ダウンロード・ファイル設定］タブ

［ダウンロード・ファイル設定］タブでは，デバッグ・ツールにダウンロードを実行する際の設定を行います。

各カテゴリ内の設定についての詳細は，「2.5.1 ダウンロードを実行する」を参照してください。

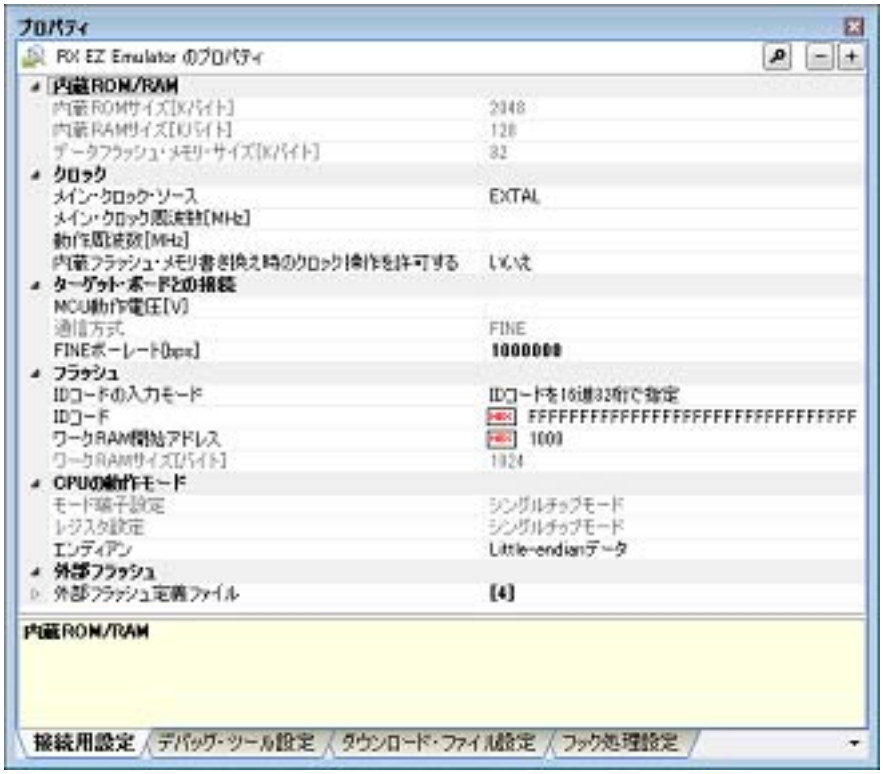
2.3.3.4 ［フック処理設定］タブ

［フック処理設定］タブでは，デバッグ・ツールにフック処理の設定を行います。
フック処理，および各カテゴリ内の設定についての詳細は，「2.18 フック処理を設定する」を参照してください。

2.3.4 【EZ Emulator】の場合

EZ Emulator を使用する場合の動作環境の設定を次のプロパティ パネルで行います。

図 2.42　　動作環境設定（プロパティ パネル）



プロパティ パネル上の該当するタブを選択し，次の設定を順次行ってください。

2.3.4.1 ［接続用設定］タブ
2.3.4.2 ［デバッグ・ツール設定］タブ
2.3.4.3 ［ダウンロード・ファイル設定］タブ
2.3.4.4 ［フック処理設定］タブ

2.3.4.1 ［接続用設定］タブ

［接続用設定］タブでは，次に示すカテゴリごとに，デバッグ・ツールとの接続に関する設定を行います。

(1) ［内蔵 ROM/RAM］
(2) ［クロック］
(3) ［ターゲット・ボードとの接続］
(4) ［フラッシュ］
(5) ［CPU の動作モード］
(6) ［外部フラッシュ］

(1) ［内蔵 ROM/RAM］
このカテゴリでは，内蔵 ROM/RAM に関する設定を表示します。

図 2.43　　［内蔵 ROM/RAM］カテゴリ

- ［内蔵 ROM サイズ [K バイト]］
  エミュレーションする内蔵 ROM サイズを表示します（単位：K バイト）。
  このプロパティ値を変更することはできません。

- ［内蔵 RAM サイズ [バイト]］
  エミュレーションする内蔵 RAM サイズを表示します（単位：バイト）。
  このプロパティ値を変更することはできません。

- ［データフラッシュ・メモリ・サイズ [K バイト]］
  選択しているマイクロコントローラのデータフラッシュ・メモリ領域のサイズを表示します（単位：K バイト）。
  選択しているマイクロコントローラがデータフラッシュ・メモリ非搭載品の場合は，［0］を表示します。
  このプロパティ値を変更することはできません。

(2)　［クロック］
このカテゴリでは，クロックに関する設定を行います。

注意　EZ Emulator と接続中にこのカテゴリ内のプロパティを変更することはできません。

図 2.44　［クロック］カテゴリ【HOCO】

クロック	
メイン・クロック・ソース	HOCO
動作周波数[MHz]	25.0000
内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する	いいえ

図 2.45　［クロック］カテゴリ【EXTAL】

クロック	
メイン・クロック・ソース	EXTAL
メイン・クロック周波数[MHz]	12.5000
動作周波数[MHz]	25.0000
内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する	いいえ

- ［メイン・クロック・ソース］
  メイン・クロックを EXTAL 周波数と内蔵 HOCO から指定します。
  内蔵 HOCO を持たないマイクロコントローラの場合は，[EXTAL] を表示します。

- ［メイン・クロック周波数 [MHz]］
  メイン・クロックの周波数（逓信前）を指定します。
  EXTAL 周波数値として 0.0001 ～ 99.9999（単位：MHz）の範囲で数値を直接入力します。
  入力値の少数第 5 位以下は切り捨てられ，指定可能範囲外の入力値は，0.0001(0 以下だった時) または 99.9999(100 以上だった時) に丸め込みます。
  このプロパティは，［メイン・クロック・ソース］プロパティで [EXTAL] を指定した場合のみ表示されます。

- ［動作周波数 [MHz]］
  動作周波数（ICLK）を指定します。
  0.0001 ～ 999.9999（単位：MHz）の範囲で数値を直接入力します。
  入力値の少数第 5 位以下は切り捨てられ，指定可能範囲外の入力値は，0.0001(0 以下だった時) または 999.9999(1000 以上だった時) に丸め込みます。

- ［内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する］
  内蔵フラッシュ・メモリ書き換え時に，デバッガによるクロック操作を許可するか否かを指定します。

  注意　【EZ Emulator【RX600 シリーズ , RX200 シリーズ , RX100 シリーズ】
  [はい] を指定した場合，マイクロコントローラの FlashIF クロック（FCLK）が動作保証範囲外（LOCO やサブクロック動作中など）でデバッガによる内蔵フラッシュ ROM の書き換えを実施した時，EZ Emulator がクロック・ソースを切り替えます。なお，書き換え完了後は，元のクロック・ソースに復帰します。
  クロック操作を許可した場合，クロック・ソースを切り替えるため，内蔵フラッシュ・メモリ書き換え中の周辺クロック動作周波数が変更されます。
  クロック操作の許可設定は，プログラム実行やステップ実行などの実行系機能を実施した後の内蔵フラッシュ ROM を書き換える時に有効となります。なお，デバッグ・ツールの接続直後や CPU リセット発行時は，クロック操作の許可設定に関わらず FCLK が動作保証範囲外の場合は，強制的にクロック・ソースを切り替えます。

(3)　［ターゲット・ボードとの接続］
このカテゴリでは，EZ Emulator とターゲット・ボードとの接続に関する設定を行います。

図 2.46　　［ターゲット・ボードとの接続］カテゴリ

ターゲット・ボードとの接続	
MCU動作電圧[V]	
通信方式	FINE
FINEボーレート[bps]	1000000

- ［MCU 動作電圧 [V]］
  ユーザ基板に実際に供給しているマイクロコントローラの動作電圧値を指定します。
  0.0001 ～ 9.9999（単位：V）の範囲の数値を直接入力で指定します。

  注意　　EZ Emulator と接続中にこのプロパティを変更することはできません。

- ［通信方式］
  EZ Emulator がターゲット・システム上のマイクロコントローラと通信する際の通信方式を表示します。
  このプロパティ値を変更することはできません。
  なお，デバッグ・ツールの選択についての詳細は，「2.3.1 使用するデバッグ・ツールを選択する」を参照してください。

- ［FINE ボーレート [bps]］
  EZ Emulator がターゲット・システム上のマイクロコントローラと通信する際の通信速度 (FINE ボーレート) をドロップダウン・リストより指定します。
  ドロップダウン・リストには，次の通信速度が表示されます。

  - 1000000(デフォルト), 500000, 250000

  注意 1.　　EZ Emulator と接続中にこのカテゴリ内のプロパティを変更することはできません。

  注意 2.　　ターゲット・システム上での FINE 信号の配線長や配線方法によっては，選択した FINE ボーレートで通信できない場合があります。この FINE 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。

(4)　［フラッシュ］
このカテゴリでは，フラッシュ書き換えに関する設定を行います。

注意　　EZ Emulator と接続中にこのカテゴリ内のプロパティを変更することはできません。

図 2.47　　［フラッシュ］カテゴリ

フラッシュ	
IDコードの入力モード	IDコードを16進32桁で指定
IDコード	HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
ワークRAM開始アドレス	HEX 1000
ワークRAMサイズ[バイト]	1024

- ［ID コードの入力モード］
  ID コードの入力モードを指定します。

  注意　　EZ Emulator と接続中にこのプロパティを変更することはできません。

- ［ID コード］
  ID コードプロテクトを解除するための ID コードを指定します。
  プロパティにおいて，[ID コードを 16 進 32 桁で指定] を指定した場合，ID コードは 32 桁の 16 進数で入力してください。また，[ID コードを 16 文字以内で指定 (ASCII)] を指定した場合，ID コードは 16 文字以内の ASCII 文字列で入力してください。

  注意 1.　　EZ Emulator と接続中にこのプロパティを変更することはできません。

  注意 2.　　ID コードを 32 桁の 16 進数で入力する場合は，32 ビットのデータ単位に並べて入力してください。
  フラッシュプログラマでは ID コードの並び順が CS+ と異なる場合があります。ご使用のフラッシュプログラマの ID コード入力の仕様を確認してください。

  注意 3.　　ID コードを ASCII 文字列で入力する場合，16 文字に満たない部分は ”0” で入力されます。

  注意 4.　　ID コードを含んだプログラムをダウンロードした場合，ID コードは FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF に書き換わります。

- ［ワーク RAM 開始アドレス］
  デバッガが使用するワーク RAM の配置アドレスを指定します。なお，アドレス値は 4 バイト単位の値を指定してください。（入力値が 4 バイト単位の値でなかった場合は，自動的に補正されます。）

指定したワーク RAM の配置アドレスから［ワーク RAM サイズ [バイト]］プロパティの表示サイズ分は，デバッガのファームウェアが使用します。

注意 1. EZ Emulator と接続中にこのプロパティを変更することはできません。

注意 2. メモリ内容の退避・復帰が行われるためユーザ・プログラムでもこの領域を使用できますが，ワーク RAM に配置する領域は，以下の用途では使用できません。

- DMA または DTC 機能の転送元・転送先
- DTC ベクタテーブル，および転送情報配置アドレス
- DMAC，DTC 起動要因の割り込みベクタ

- ［ワーク RAM サイズ [バイト]］
  デバッガが使用するワーク RAM のサイズを表示します。

(5) ［CPU の動作モード］
このカテゴリでは，エミュレーションするマイクロコントローラの動作モードに関する設定を行います。

注意 EZ Emulator と接続中にこのカテゴリ内のプロパティを変更することはできません。

図 2.48 [CPU の動作モード] カテゴリ

CPUの動作モード	
モード端子設定	シングルチップモード
レジスタ設定	シングルチップモード
エンディアン	Little-endianデータ

- ［モード端子設定］
  マイクロコントローラのモード端子で設定している動作モードを指定します。
  EZ Emulator の場合は，シングルチップモード固定となります。
- ［レジスタ設定］
  レジスタで設定する動作モードを指定します。
  EZ Emulator の場合は，シングルチップモード固定となります。
- ［エンディアン］
  プロジェクトのエンディアンを表示します。
  プロジェクトからエンディアンの設定情報を取得し，その値を表示します。
  デバッグ・ツールと切断中のみ指定することができます。

(6) ［外部フラッシュ］
このカテゴリでは，外部フラッシュに関する設定を行います。
EZ Emulator の場合，設定をすることは可能ですが，マイクロコントローラの動作モードがシングルチップモード固定となるため，本カテゴリにて設定した内容は反映されません。

2.3.4.2 ［デバッグ・ツール設定］タブ

［デバッグ・ツール設定］タブでは，次に示すカテゴリごとに，デバッグ・ツールの基本設定を行います。

(1) ［メモリ］
(2) ［実行中のメモリ・アクセス］
(3) ［レジスタ］
(4) ［ブレーク］
(5) ［システム］
(6) ［トレース］
(7) ［タイマ］【RX600 シリーズ】

(1)　［メモリ］
このカテゴリでは，メモリに関する設定を行います。

図 2.49　　［メモリ］カテゴリ

◢ メモリ	
◢ メモリ・マッピング	[23]
▷ [0]	内蔵RAM領域
▷ [1]	予約領域
▷ [2]	I/Oレジスタ領域
▷ [3]	予約領域
▷ [4]	I/Oレジスタ領域
▷ [5]	I/Oレジスタ領域
▷ [6]	予約領域
▷ [7]	I/Oレジスタ領域
▷ [8]	I/Oレジスタ領域
▷ [9]	I/Oレジスタ領域
▷ [10]	データフラッシュ領域
▷ [11]	予約領域
▷ [12]	その他のメモリ領域
▷ [13]	予約領域
▷ [14]	I/Oレジスタ領域
▷ [15]	予約領域
▷ [16]	I/Oレジスタ領域
▷ [17]	予約領域
▷ [18]	その他のメモリ領域
▷ [19]	予約領域
▷ [20]	その他のメモリ領域
▷ [21]	予約領域
▷ [22]	内蔵ROM領域
メモリ書き込み時にベリファイを行う	はい

- [メモリ・マッピング]
現在のメモリ・マッピングの状況が，メモリ領域の種別ごとに詳細表示されます。
このパネル上でマッピング値を変更することはできません。
I/O 保護領域を追加，または削除する必要がある場合は，プロパティを選択することで設定欄右端に表示される [...] ボタンのクリックによりオープンするメモリ・マッピング ダイアログで行います。
設定方法についての詳細は，メモリ・マッピング ダイアログの項を参照してください。
このプロパティでは，メモリ領域の個数のみを表示します。
[メモリ・マッピング] プロパティを展開すると，次のサブ項目が表示されます。

- [メモリ種別]
当該領域のメモリ種別を表示します。
各メモリ種別は次の領域に該当します。

内蔵 ROM 領域	プログラム ROM 注 1 およびデータフラッシュ注 2 が該当します。
内蔵 RAM 領域	内蔵 RAM が該当します。
I/O レジスタ領域	周辺 I/O レジスタが該当します。 エンディアンが異なる領域で分割して表示します。
外部領域 (CS7/CS6/.../CS0)	外部アドレス空間が該当します。 CS0 - CS7 を区別して表示します。
その他のメモリ領域	FCU-RAM 注 3, FCU ファーム注 3, ユーザーブートが該当します。
予約領域	上記以外の領域を表示します。
I/O 保護領域	外部領域でデバッガからリードしないアドレス範囲が該当します。 メモリ・マッピング ダイアログで登録します。

注 1.　　ダウンロード以外の操作（ メモリ パネルからの操作やライン・アセンブル）で内蔵フラッシュ ROM 領域を編集した場合，編集内容は次の実行操作時にフラッシュ ROM に書き込まれます。

注 2.　消去された状態のデータフラッシュ ROM の内容を参照すると，マイクロコントローラの仕様どおり不定値が表示されます。なお，デバッガからデータフラッシュ ROM の内容を書き換えた場合は，256 バイト単位で書き込みを行います。書き込まれた領域は不定値とはなりません。

注 3.　FCU-RAM 領域の内容をデバッガから書き換えないでください。また，FCU ファーム領域はデバッガから書き換えできません。

- [開始アドレス]
  当該領域の開始アドレスを表示します。

- [終了アドレス]
  当該領域の終了アドレスを表示します。

- [アドレス幅 [ビット]]
  当該領域のアクセス幅を表示します。
  [メモリ種別] が外部領域については，デバッグ・ツールと切断中のみ変更可能です。

- [エンディアン]
  外部領域および I/O レジスタ領域のエンディアンを表示します。
  [メモリ種別] が外部領域については，デバッグ・ツールと切断中のみ変更可能です。

注意　デバッグ・ツールと接続することにより（「2.4.1 CS+ にデバッグ・ツールを接続する」参照），各メモリ種別ごとの詳細表示を行います。

- ［メモリ書き込み時にベリファイを行う］
  メモリ値の初期化を行う際に，ベリファイを行うか否かをドロップダウン・リストにより指定します。
  ［はい］を指定した場合，ダウンロードした際，ウォッチ パネルやメモリ パネルで値を変更した際にベリファイを行います。

(2)　［実行中のメモリ・アクセス］
このカテゴリでは，プログラム実行中におけるメモリ・アクセスに関する設定を行います。
このカテゴリ内の設定は，リアルタイム表示更新機能を使用する場合に必要となります。リアルタイム表示更新機能についての詳細は，「2.11.1.4 プログラム実行中にメモリの内容を表示／変更する」を参照してください。

図 2.50　［実行中のメモリ・アクセス］カテゴリ

実行中のメモリ・アクセス	
実行を一瞬停止してアクセスする	いいえ
実行中に表示更新を行う	はい
表示更新間隔[ms]	500

- ［実行を一瞬停止してアクセスする］
  プログラム実行中に，メモリに対してアクセスを許可するか否かをドロップダウン・リストにより指定します。
  アクセスを許可する場合は［はい］を選択してください（デフォルトでは［いいえ］が指定されます）。

- ［実行中に表示更新を行う］
  プログラム実行中に，ウォッチ パネル／メモリ パネルの表示内容を更新するか否かをドロップダウン・リストにより指定します。
  表示内容の更新を行う場合は［はい］を選択してください（デフォルト）。

  注意　プログラム実行中にこのプロパティを変更することはできません。

- ［表示更新間隔 [ms]］
  このプロパティは，［実行中に表示更新を行う］プロパティにおいて［はい］を指定した場合にのみ表示されます。
  プログラム実行中に，ウォッチ パネル／メモリ パネルの表示内容を更新する間隔を 100 ms 単位で指定します。
  直接入力により，100 ～ 65500 の範囲の整数（100 ms 未満の端数切り上げ）を指定してください（デフォルトでは［500］が指定されます）。
  なお，［実行中に表示更新を行う］プロパティにおいて指定値を［いいえ］から［はい］へ切り替えた場合，このプロパティには前回の設定値が表示されます。

  注意　プログラム実行中にこのプロパティを変更することはできません。

(3)　［レジスタ］
このカテゴリでは，プログラム実行中におけるステータス・バーへの PC 表示に関する設定を行います。

図 2.51　　［レジスタ］カテゴリ

◢ レジスタ	
実行中にPC表示を行う	はい
PC表示更新間隔[ms]	500

- ［実行中に PC 表示を行う］
このプロパティは，プログラム実行中にステータス・バーへ PC 値を表示するかどうかを指定します。
[いいえ] を選択した場合，実行中のステータス・バーには " 実行中 " と表示されます。

注意 1.　　プログラム実行中にこのプロパティを変更することはできません。

注意 2.　　【RX100 シリーズ】
プログラム実行中の PC 表示機能をサポートしていないため，非表示となります。

- ［PC 表示更新間隔 [ms]］
このプロパティは，［実行中に PC 表示を行う］プロパティにおいて［はい］を指定した場合にのみ表示されます。
プログラム実行中にステータス・バーの PC 表示を更新する間隔を 100ms 単位で指定します。
直接入力により，100 ～ 65500 の整数（100ms 未満は端数切り上げ）を指定してください。（デフォルトでは［500］が指定されます。）
なお，［実行中に PC 表示を行う］プロパティにおいて指定値を［いいえ］から［はい］へ切り替えた場合，このプロパティには前回の設定値が表示されます。

注意　　プログラム実行中にこのプロパティを変更することはできません。

(4)　［ブレーク］
このカテゴリでは，ブレーク機能に関する設定を行います。

図 2.52　　［ブレーク］カテゴリ

◢ ブレーク	
優先的に使用するブレークポイントの種類	ハードウエア・ブレーク

- ［優先的に使用するブレークポイントの種類］
エディタ パネル／逆アセンブル パネルにおいて，マウスのワンクリック操作でブレークポイントを設定する際に使用するブレークポイントの種別を次のドロップダウン・リストより指定します。
優先して使用するブレークポイントを使い切った状態で，ブレークポイントを設定した場合，自動的にもう片方のブレークポイントを設定します。
なお，ブレークポイントについての詳細は，「2.10.2 任意の場所で停止する（ブレークポイント）」を参照してください。

ソフトウエア・ブレーク	ソフトウエア・ブレークポイントを優先的に設定します。
ハードウエア・ブレーク	ハードウエア・ブレークポイントを優先的に設定します（デフォルト）。

(5)　［システム］
このカテゴリでは，エミュレーションシステムに関する設定を行います。
実行開始前およびブレーク後の指定ルーチン実行についての詳細は，「2.9.4 指定ルーチンを実行する【E1】【E20】【EZ Emulator】」を参照してください。

図 2.53　　[システム] カテゴリ

◢ システム	
内蔵プログラムROMを書き換えるプログラムをデバッグする	いいえ
内蔵データフラッシュを書き換えるプログラムをデバッグする	いいえ
実行開始前に指定ルーチンを実行する	はい
実行開始前に実行するルーチン	
ブレーク後に指定ルーチンを実行する	はい
ブレーク後に実行するルーチン	
指定ルーチン実行機能用ワークRAM開始アドレス	HEX 1FDD0
指定ルーチン実行機能用ワークRAMサイズ[バイト]	560

- [内蔵プログラム ROM を書き換えるプログラムをデバッグする]
  内蔵プログラム ROM 領域を書き換えるプログラム (ROM P/E モードを使用したプログラムなど) をデバッグするか否かを指定します。

  注意　EZ Emulator と接続中にこのプロパティを変更することはできません。

- [内蔵データフラッシュを書き換えるプログラムをデバッグする]
  内蔵データフラッシュ領域を書き換えるプログラム (データフラッシュ P/E モードを使用したプログラムなど) をデバッグするか否かを指定します。

  注意　EZ Emulator と接続中にこのプロパティを変更することはできません。

- [実行開始前に指定ルーチンを実行する]
  ユーザ・プログラム実行前に，指定ルーチンを実行する必要があるか否かを指定します。

  注意　プログラム実行中にこのプロパティを変更することはできません。

- [実行開始前に実行するルーチン]
  ユーザ・プログラム実行直前に実行するアドレスを指定します。
  このプロパティは，[実行開始前に指定ルーチンを実行する] プロパティにおいて，[はい] が指定された場合のみ表示されます。

  注意　プログラム実行中にこのプロパティを変更することはできません。

- [ブレーク後に指定ルーチンを実行する]
  ユーザ・プログラムブレーク後に，指定ルーチンを実行する必要があるか否かを指定します。

  注意　プログラム実行中にこのプロパティを変更することはできません。

- [ブレーク後に実行するルーチン]
  ユーザ・プログラムブレーク直後に実行するアドレスを指定します。
  このプロパティは，[ブレーク後に指定ルーチンを実行する] プロパティにおいて，[はい] が指定された場合のみ表示します。

  注意　プログラム実行中にこのプロパティを変更することはできません。

- [指定ルーチン実行機能用ワーク RAM 開始アドレス]
  指定ルーチン実行機能用のワーク RAM 配置アドレスを指定します。なお，アドレス値は 4 バイト単位の値を指定してください。(入力値が 4 バイト単位の値でなかった場合は，自動的に補正されます。)
  指定したワーク RAM の配置アドレスから [指定ルーチン実行機能用ワーク RAM サイズ [バイト]] プロパティの表示サイズ分は，デバッガのファームウェアが使用します。
  このプロパティは，[実行開始前に指定ルーチンを実行する] プロパティまたは，[ブレーク後に指定ルーチンを実行する] プロパティのいずれかにおいて [はい] が指定された場合のみ表示します。

  注意 1.　プログラム実行中にこのプロパティを変更することはできません。

  注意 2.　ユーザ・プログラムで使用しない領域を指定してください。

- [指定ルーチン実行機能用ワーク RAM サイズ [バイト]]
  指定ルーチン実行機能用のワーク RAM のサイズを表示します。
  このプロパティは，[実行開始前に指定ルーチンを実行する] プロパティまたは，[ブレーク後に指定ルーチンを実行する] プロパティのいずれかにおいて [はい] が指定された場合のみ表示します。

(6)　[トレース]
このカテゴリでは，トレース機能に関する設定を行います。

図 2.54　[トレース] カテゴリ

トレース	
トレース機能の用途	トレース
トレース・メモリを使い切った後の動作	トレース・メモリを上書きし実行を続ける
トレース・データ種別	分岐
タイム・スタンプ出力	はい
トレース・クロック・カウント・ソース[MHz]	

- [トレース機能の用途]
  トレース機能の用途として [トレース] を表示します。
  このプロパティ値を変更することはできません。

- [トレース・メモリを使い切った後の動作]
  トレースの取得モードを指定します。
  取得モードは，次のドロップダウン・リストより指定します。

トレース・メモリを上書きし実行を続ける	トレース・メモリがいっぱいになっても，トレース・データの上書きを続けます。
トレースを停止する	トレース・メモリがいっぱいになったら，トレース・データの書き込みを中止します。
停止する	トレース・メモリがいっぱいになった時点でブレークします。

- ［トレース・データ種別］
  トレース取得するデータ種別をドロップダウン・リストより指定します。
  指定可能なデータ種別はマイクロコントローラのシリーズにより異なります。
  ドロップダウン・リストには，次のデータ種別が表示されます。
  - 【RX600 シリーズ】
    分岐 , 分岐＋データアクセス , データアクセス
  - 【RX100, RX200 シリーズ】
    分岐 , データアクセス
- ［タイム・スタンプ出力］
  収集するトレース・データにタイム・スタンプ情報を付加するか否かを指定します。
  このプロパティは，マイクロコントローラのシリーズや［トレース・データ種別］プロパティでの指定値によって変更可否が次のように切り替わります。

マイクロコントローラ	データ種別	変更可否
RX600 シリーズ	分岐，分岐 + データアクセス，データアクセス	プログラム停止時のみ変更可能
RX200 シリーズ	分岐	変更不可 [いいえ] が固定表示
	データアクセス	プログラム停止時のみ変更可能
RX100 シリーズ	分岐，データアクセス	変更不可 [いいえ] が固定表示

- ［トレース・クロック・カウント・ソース [MHz]］
  このプロパティは，［タイム・スタンプ出力］プロパティにおいて［はい］を指定した場合にのみ表示されます。
  カウント値からタイム・スタンプを算出するためのカウント・ソースを入力します。
  直接入力により，0.0001 - 999.999 の間で指定してください。
  なお，プロパティが空欄の場合，［接続用設定］タブの［クロック］カテゴリ内の［動作周波数 [MHz]］プロパティの設定値がカウント・ソースとして代用されます。

  注意 1. プログラム実行中にこのプロパティを変更することはできません。

  注意 2. プロパティの設定値は , 収集済みのトレース・データには反映しません。プロパティ設定後に収集したトレース・データから反映します。

(7) ［タイマ］【RX600 シリーズ】
このカテゴリでは，タイマ機能に関する設定を行います。

図 2.55 [タイマ] カテゴリ

- ［カウンタを 64 ビットで使用する］
  計測カウンタを 32 ビット ×2 本で使用するか，64 ビット ×1 本で使用するかを指定します。

2.3.4.3 ［ダウンロード・ファイル設定］タブ

［ダウンロード・ファイル設定］タブでは，デバッグ・ツールにダウンロードを実行する際の設定を行います。
各カテゴリ内の設定についての詳細は，「2.5.1 ダウンロードを実行する」を参照してください。

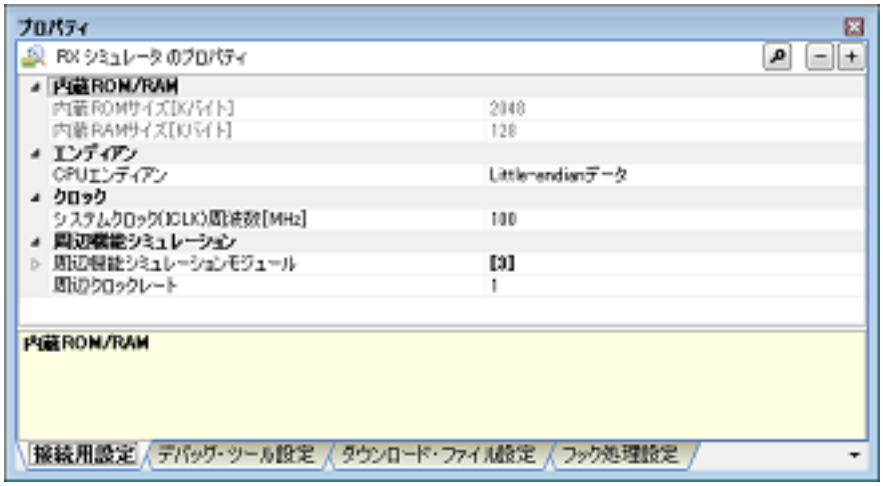
2.3.4.4 ［フック処理設定］タブ

［フック処理設定］タブでは，デバッグ・ツールにフック処理の設定を行います。
フック処理，および各カテゴリ内の設定についての詳細は，「2.18 フック処理を設定する」を参照してください。

2.3.5 【シミュレータ】の場合

シミュレータを使用する場合の動作環境の設定を次のプロパティ パネルで行います。

図 2.56　　動作環境設定【シミュレータ】（プロパティ パネル）



プロパティ パネル上の該当するタブを選択し，次の設定を順次行ってください。

2.3.5.1 ［接続用設定］タブ
2.3.5.2 ［デバッグ・ツール設定］タブ
2.3.5.3 ［ダウンロード・ファイル設定］タブ
2.3.5.4 ［フック処理設定］タブ

2.3.5.1 ［接続用設定］タブ

［接続用設定］タブでは，次に示すカテゴリごとに，デバッグ・ツールとの接続に関する設定を行います。

(1) ［内蔵 ROM/RAM］
(2) ［エンディアン］
(3) ［クロック］
(4) ［周辺機能シミュレーション］

(1) ［内蔵 ROM/RAM］
このカテゴリでは，選択しているマイクロコントローラの内蔵 ROM/RAM サイズが表示されます

図 2.57　　［内蔵 ROM/RAM］カテゴリ【シミュレータ】

- ［内蔵 ROM サイズ [K バイト]］
  シミュレーションする内蔵 ROM サイズを表示します（単位：K バイト）。
  このプロパティ値を変更することはできません。
- ［内蔵 RAM サイズ [K バイト]］
  シミュレーションする内蔵 RAM サイズを表示します（単位：K バイト）。
  このプロパティ値を変更することはできません。

(2) ［エンディアン］
このカテゴリでは，エンディアンに関する設定を行います。

図 2.58　　［エンディアン］カテゴリ【シミュレータ】

- ［CPU エンディアン］
  CPU のエンディアンを指定します。
  デフォルトは，ビルド・ツールのプロパティで選択されたエンディアンを表示します。

  注意　　シミュレータと接続中にこのプロパティを変更することはできません。

(3)　［クロック］
このカテゴリでは，クロックに関する設定を行います。

図 2.59　　［クロック］カテゴリ【シミュレータ】

◢ クロック	
システムクロック(ICLK)周波数[MHz]	100

- ［システムクロック (ICLK) 周波数 [MHz]］
  CPU の動作周波数を指定します（単位：MHz）。
  使用する周波数に合わせて設定してください。
  直接入力により，1 ～ 1,000 の範囲の整数を指定してください。

  注意　　シミュレータと接続中にこのプロパティを変更することはできません。

(4)　［周辺機能シミュレーション］
このカテゴリでは，周辺機能シミュレーションに関する設定を行います。

図 2.60　　［周辺機能シミュレーション］カテゴリ

◢ 周辺機能シミュレーション	
◢ 周辺機能シミュレーションモジュール	[3]
▷ [0]	CMT
▷ [1]	ICU
▷ [2]	MPU
周辺クロックレート	1

- ［周辺機能シミュレーションモジュール］
  使用可能な周辺機能シミュレーションモジュール名を表示，および表示された周辺機能シミュレーションモジュールを使用するか否かをドロップダウン・リストで指定します。
  使用する場合は［はい］を選択してください。（デフォルトでは［いいえ］が指定されます）。
  なお，このパネル上で周辺機能シミュレーションモジュール名を変更することはできません。

- ［周辺クロックレート］
  周辺クロックと内部クロックの比（周辺 1 クロックが内部クロックいくつに相当するか）をドロップダウン・リストにより指定します。
  ドロップダウン・リストには，次のクロックレートが表示されます。
  1（デフォルト），2，3，4，6，8，12，16，24，32，64

  注意　　シミュレータと接続中にこのプロパティを変更することはできません。

2.3.5.2　［デバッグ・ツール設定］タブ

［デバッグ・ツール設定］タブでは，次に示すカテゴリごとに，デバッグ・ツールの基本設定を行います。

(1)　［メモリ］
(2)　［実行中のメモリ・アクセス］
(3)　［レジスタ］
(4)　［トレース］
(5)　［カバレッジ］
(6)　［ストリーム入出力］
(7)　［実行モード］
(8)　［命令デコードキャッシュ］

(1)　［メモリ］
このカテゴリでは，メモリに関する設定を行います。

図 2.61　　［メモリ］カテゴリ【シミュレータ】

メモリ	
メモリ・マッピング	[17]
[0]	内蔵RAM領域
[1]	ノン・マップ領域
[2]	I/Oレジスタ領域
[3]	ノン・マップ領域
[4]	I/Oレジスタ領域
[5]	I/Oレジスタ領域
[6]	ノン・マップ領域
[7]	I/Oレジスタ領域
[8]	I/Oレジスタ領域
[9]	I/Oレジスタ領域
[10]	内蔵ROM領域
[11]	ノン・マップ領域
[12]	I/Oレジスタ領域
[13]	ノン・マップ領域
[14]	I/Oレジスタ領域
[15]	ノン・マップ領域
[16]	内蔵ROM領域

- ［メモリ・マッピング］
現在のメモリ・マッピングの状況が，メモリ領域の種別ごとに詳細表示されます。
このパネル上でマッピング値を変更することはできません。メモリ・マッピングを追加する必要がある場合は，［メモリ・マッピング］プロパティを選択することで設定欄右端に表示される［...］ボタンのクリックによりオープンするメモリ・マッピング ダイアログで行います。
設定方法についての詳細は，メモリ・マッピング ダイアログの項を参照してください。

図 2.62　　メモリ・マッピング ダイアログのオープン

(2)　［実行中のメモリ・アクセス］
このカテゴリでは，プログラム実行中におけるメモリ・アクセスに関する設定を行います。
このカテゴリ内の設定は，リアルタイム表示更新機能を使用する場合に必要となります。リアルタイム表示更新機能についての詳細は，「2.11.1.4 プログラム実行中にメモリの内容を表示／変更する」を参照してください。

図 2.63　　［実行中のメモリ・アクセス］カテゴリ【シミュレータ】

実行中のメモリ・アクセス	
実行中に表示更新を行う	はい
表示更新間隔[ms]	500

- ［実行中に表示更新を行う］
プログラム実行中に，ウォッチ パネル／メモリ パネルの表示内容を更新するか否かをドロップダウン・リストにより指定します。
表示内容の更新を行う場合は［はい］を選択してください（デフォルト）。

注意　　プログラム実行中にこのプロパティを変更することはできません。

- ［表示更新間隔 [ms]］
このプロパティは，［実行中に表示更新を行う］プロパティにおいて［はい］を指定した場合にのみ表示します。
プログラム実行中に，ウォッチ パネル／メモリ パネルの表示内容を更新する間隔を 100 ms 単位で指定します。
直接入力により，100 ～ 65500 の整数（100 ms 未満の端数切り上げ）を指定してください（デフォルトでは［500］が指定されます）。

なお，［実行中に表示更新を行う］プロパティにおいて指定値を［いいえ］から［はい］へ切り替えた場合，このプロパティには前回の設定値が表示されます。

注意　プログラム実行中にこのプロパティを変更することはできません。

(3)　［レジスタ］
このカテゴリでは，プログラム実行中におけるステータス・バーへの PC 表示に関する設定を行います。

図 2.64　[レジスタ] カテゴリ【シミュレータ】

レジスタ	
実行中にPC表示を行う	はい
PC表示更新間隔[ms]	500

- ［実行中に PC 表示を行う］
このプロパティは，プログラム実行中にステータス・バーへ PC 値を表示するかどうかを指定します。
[いいえ] を選択した場合，実行中のステータス・バーは " 実行中 " と表示します。

注意　プログラム実行中にこのプロパティを変更することはできません。

- ［PC 表示更新間隔 [ms]］]
このプロパティは，［実行中に表示更新を行う］プロパティにおいて［はい］を指定した場合にのみ表示します。
プログラム実行中にステータス・バーの PC 表示を更新する間隔を 100ms 単位で指定します。
直接入力により，100 ～ 65500 の整数（100ms 未満は端数切り上げ）を指定してください。（デフォルトでは［500］が指定されます。）
なお，［実行中に表示更新を行う］プロパティにおいて指定値を［いいえ］から［はい］へ切り替えた場合，このプロパティには前回の設定値が表示されます。

注意　プログラム実行中にこのプロパティを変更することはできません。

(4)　［トレース］
このカテゴリでは，トレース機能に関する設定を行います。
トレース機能，およびこのカテゴリ内の設定についての詳細は，「2.13 実行履歴の収集」を参照してください。

(5)　［カバレッジ］
このカテゴリでは，カバレッジ機能に関する設定を行います。
カバレッジ機能，およびこのカテゴリ内の設定についての詳細は，「2.15 カバレッジの測定【シミュレータ】【E20【RX71M, RX64M グループ】】」を参照してください。

(6)　［ストリーム入出力］
このカテゴリでは，ユーザプログラムからの標準入出力またはファイル入出力を行うためのストリーム入出力に関する設定を行います。

図 2.65　［ストリーム入出力］カテゴリ

ストリーム入出力	
ストリーム入出力モードを選択	シミュレータモード
ストリーム入出力機能を使用する	いいえ
ストリーム入出力用アドレス	HEX 0

- ［ストリーム入出力モードを選択］
このプロパティでは，標準入出力を実行するための入出力方式をドロップダウン・リストにより指定します。
エミュレータと同じ低水準インタフェース・ルーチン（アセンブリ言語部分）用のファイルを使用する場合は，［エミュレータモード］を指定してください（デフォルトでは［シミュレータモード］が指定されています）。

備考　ファイル入出力は［シミュレータモード］を指定した場合のみ使用可能です。［エミュレータモード］を指定した場合は，ファイル入出力は使用できません。

注意　シミュレータと接続中にこのプロパティを変更することはできません。

- ［ストリーム入出力機能を使用する］
このプロパティは，［ストリーム入出力モードを選択］プロパティにおいて［シミュレータモード］を指定した場合にのみ表示します。
シミュレータモード使用時のストリーム入出力機能を使用するかどうかをドロップダウン・リストにより指定します。

ストリーム入出力機能を使用する場合は［はい］を選択してください（デフォルトでは［いいえ］が指定されます）。

- ［ストリーム入出力用アドレス］
このプロパティは，［ストリーム入出力モードを選択］プロパティにおいて［シミュレータモード］を指定した場合にのみ表示します。
シミュレータモード使用時のストリーム入出力の開始位置を指定します。
直接入力によりストリーム入出力用アドレスを指定してください。（デフォルトでは［0x00000000］が指定されます。）

(7)　［実行モード］
このカテゴリでは，シミュレーションエラー，または例外が発生した場合の動作に関する設定を行います。

図 2.66　［実行モード］カテゴリ

実行モード	
実行モードを選択	停止
未定義命令例外発生時に停止する	はい
特権命令例外発生時に停止する	はい
アクセス例外発生時に停止する	はい
浮動小数点例外発生時に停止する	はい
割り込み発生時に停止する	はい
INT命令実行時に停止する	はい
BRK命令実行時に停止する	はい

- ［実行モードを選択］
このプロパティでは，実行モードを指定します。
次のドロップダウン・リストより選択します。

停止	シミュレーションを停止します（デフォルト）。
続行	シミュレーションを続行します（下段のプロパティ項目が有効となります）。

- ［未定義命令例外発生時に停止する］
このプロパティは，［実行モードを選択］プロパティにおいて［停止］を指定した場合のみ表示されます。
未定義命令例外発生時に停止するかどうかをドロップダウン・リストにより指定します。
未定義命令例外発生時に停止する場合は，［はい］を選択してください（デフォルト）。

- ［特権命令例外発生時に停止する］
このプロパティは，［実行モードを選択］プロパティにおいて［停止］を指定した場合のみ表示されます。
特権命令例外発生時に停止するかどうかをドロップダウン・リストにより指定します。
特権命令例外発生時に停止する場合は，［はい］を選択してください（デフォルト）。

- ［アクセス例外発生時に停止する］
このプロパティは，［周辺機能シミュレーションモジュール］プロパティに MPU モジュールが存在し，［実行モードを選択］プロパティにおいて［停止］を指定した場合のみ表示されます。
アクセス例外発生時に停止するかどうかをドロップダウン・リストにより指定します。
アクセス例外発生時に停止する場合は，［はい］を選択してください（デフォルト）。

- ［浮動小数点例外発生時に停止する］【FPU 搭載品】
このプロパティは，［実行モードを選択］プロパティにおいて［停止］を指定した場合のみ表示されます。
浮動小数点例外発生時に停止するかどうかをドロップダウン・リストにより指定します。
浮動小数点例外発生時に停止する場合は，［はい］を選択してください（デフォルト）。

- ［割り込み発生時に停止する］
このプロパティは，［実行モードを選択］プロパティにおいて［停止］を指定した場合のみ表示されます。
割り込み発生時に停止するかどうかをドロップダウン・リストにより指定します。
割り込み発生時に停止する場合は，［はい］を選択してください（デフォルト）。

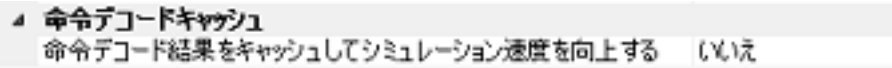
- ［INT 命令実行時に停止する］
このプロパティは，［実行モードを選択］プロパティにおいて［停止］を指定した場合のみ表示されます。
INT 命令実行時に停止するかどうかをドロップダウン・リストにより指定します。
INT 命令実行時に停止する場合は，［はい］を選択してください（デフォルト）。

- ［BRK 命令実行時に停止する］
このプロパティは，［実行モードを選択］プロパティにおいて［停止］を指定した場合のみ表示されます。
BRK 命令実行時に停止するかどうかをドロップダウン・リストにより指定します。

BRK 命令実行時に停止する場合は，［はい］を選択してください（デフォルト）。

(8) ［命令デコードキャッシュ］
このカテゴリでは，命令実行時にデコード結果を保持し，同一アドレス実行時にデコード結果を利用する機能（命令デコードキャッシュ）に関する設定を行います。

図 2.67 ［命令デコードキャッシュ］カテゴリ



- ［命令デコード結果をキャッシュしてシミュレーション速度を向上する］
命令デコードキャッシュ機能を有効にするかどうかをドロップダウン・リストにより指定します。
命令デコードキャッシュ機能を有効にする場合は［はい］を選択してください（デフォルトでは［いいえ］が指定されます）。

注意 命令デコードキャッシュ機能はデコード結果を再利用するため，自己書き換えコードを使用するプログラムには使用できません。また，プログラムの意図しない動作により命令が書き換わると，正しくエラーを検出できない場合があります。

2.3.5.3 ［ダウンロード・ファイル設定］タブ

［ダウンロード・ファイル設定］タブでは，デバッグ・ツールにダウンロードを実行する際の設定を行います。
各カテゴリ内の設定についての詳細は，「2.5.1 ダウンロードを実行する」を参照してください。

2.3.5.4 ［フック処理設定］タブ

［フック処理設定］タブでは，デバッグ・ツールにフック処理の設定を行います。
フック処理，および各カテゴリ内の設定についての詳細は，「2.18 フック処理を設定する」を参照してください。

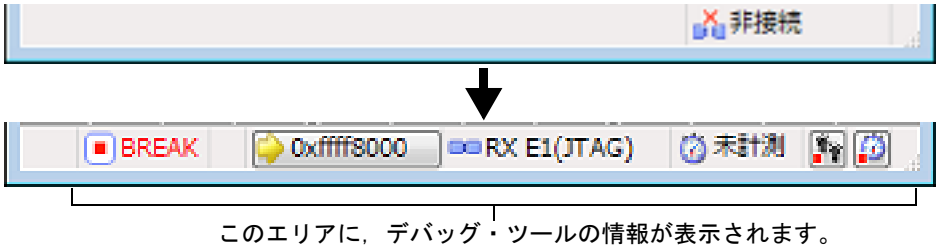
2.4 デバッグ・ツールとの接続／切断

この節では，CS+ とデバッグ・ツールとの接続方法，切断方法，およびホットプラグイン機能を用いた接続方法について説明します。

2.4.1 CS+ にデバッグ・ツールを接続する

［デバッグ］メニュー→［デバッグ・ツールへ接続］を選択することにより，現在のアクティブ・プロジェクトで選択しているデバッグ・ツールとの通信を開始します。
デバッグ・ツールとの接続に成功すると，メイン・ウインドウのステータス・バーが，次のように変化します。
なお，ステータス・バーに表示される各項目についての詳細は，メイン・ウインドウを参照してください。

図 2.68　　デバッグ・ツールとの接続に成功したステータス・バー



備考　　デバッグ・ツールバーの ボタンをクリックすることで，デバッグ・ツールと接続したのち，指定ファイルのダウンロードを実行します（「2.5.1 ダウンロードを実行する」参照）。
また，同ツールバーの ボタンをクリックすることで，プロジェクトのビルドを行い，デバッグ・ツールと接続したのち，指定ファイルのダウンロードを実行します。

(1)　バージョン情報の表示【E1】【E20】【EZ Emulator】
デバッグ・ツールとの接続に成功すると，エミュレータ・ファームウェアなどのバージョン情報および供給電圧などのエミュレータ情報を出力 パネルに表示します。

図 2.69　　接続時の出力 パネル【E1】

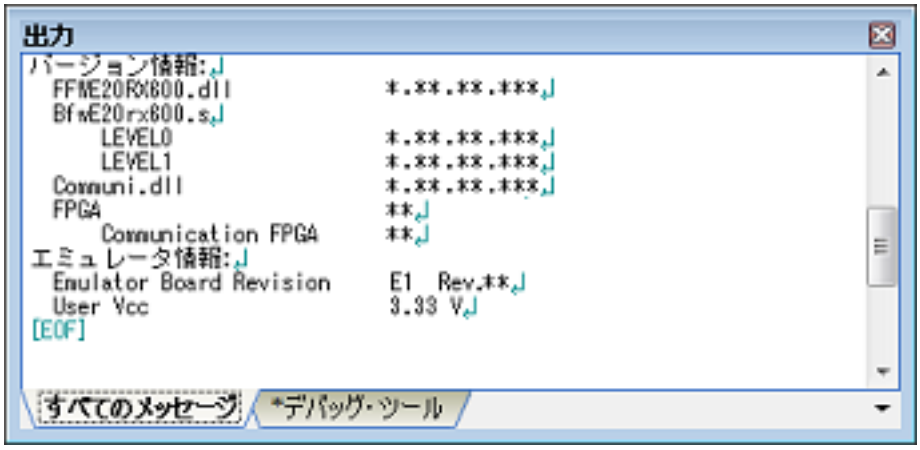


図 2.70 接続時の出力 パネル【E2 Lite】

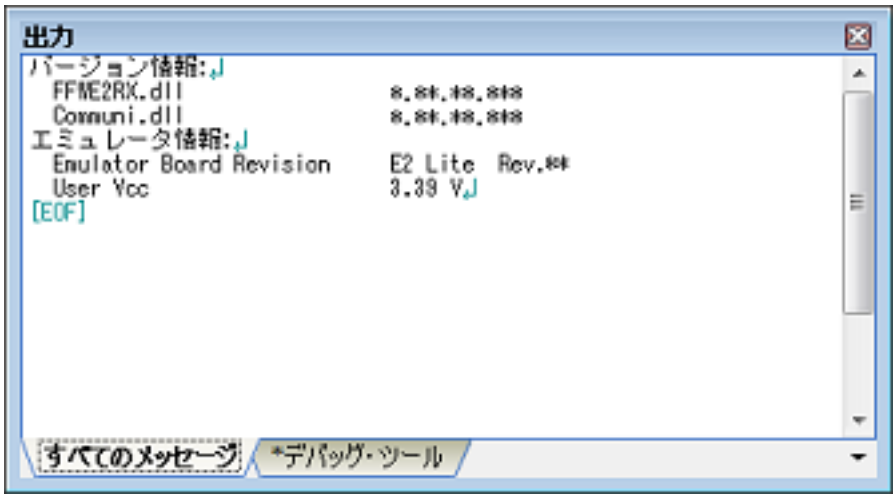


図 2.71 接続時の出力 パネル【E20】

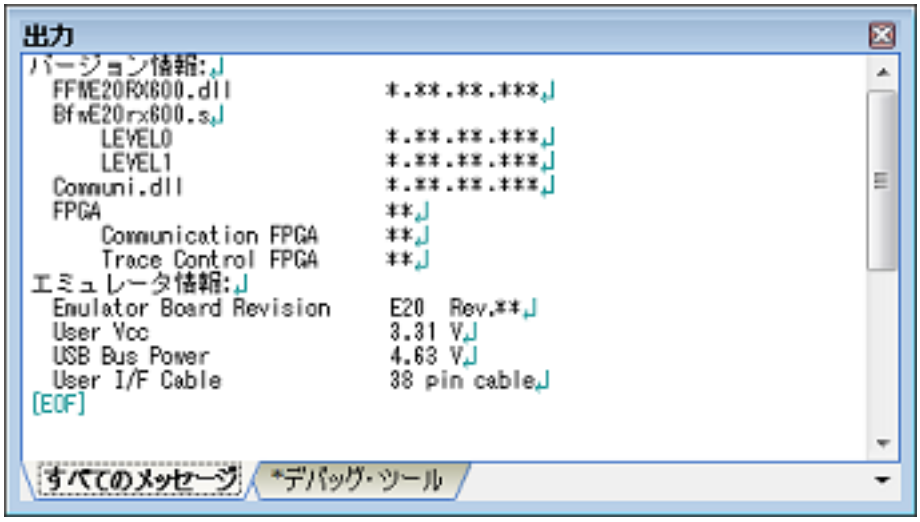
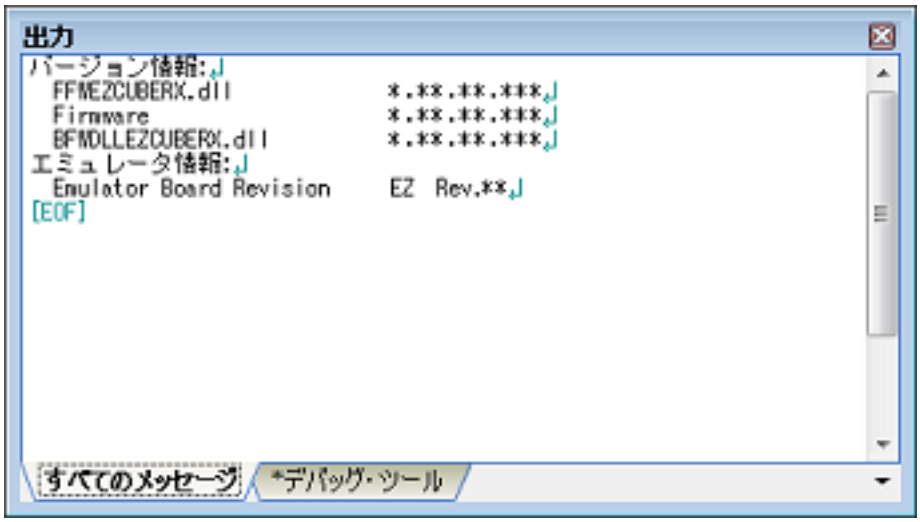
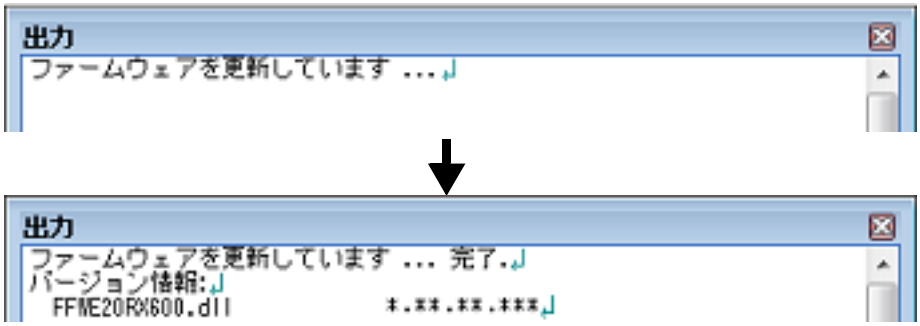


図 2.72 接続時の出力 パネル【EZ Emulator】



注意 【E1】【E20】
エミュレータ・ファームウェアの更新が必要な場合，自動的に更新を行います。そのため，接続が完了するまでは USB および電源を切断しないでください。
なお，エミュレータ・ファームウェアの更新が完了すると，出力 パネルのメッセージが，次のように変化します。

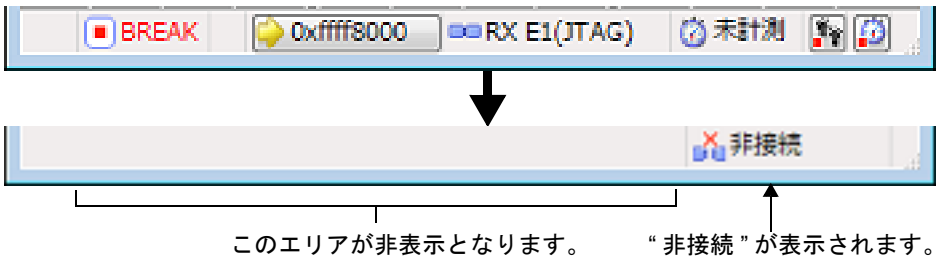
図 2.73 エミュレータ・ファームウェア更新完了時の出力 パネル



2.4.2 CS+ からデバッグ・ツールを切断する

デバッグ・ツールバーの ボタンをクリックすることにより，現在接続しているデバッグ・ツールとの通信を切断します。
デバッグ・ツールと切断すると，メイン・ウインドウのステータス・バーが，次のように変化します。

図 2.74 デバッグ・ツールから切断したステータス・バー



注意 1. プログラム実行中は，デバッグ・ツールと切断することはできません。切断する場合は，実行中のプログラムを停止してから切断してください。

注意 2. 【E1】【E20】【EZ Emulator】
E1/E20/EZ Emulator の切断は，上記の方法もしくは下記 (1)，(2) のいずれかの方法で必ず切断してください。
(1) [デバッグ] メニューにて [デバッグ・ツールから切断] を選択します。
(2) Python コンソール パネルにて “debugger.Disconnect()” と入力します。
強制的にデバッガを終了した場合，次回起動時に正しく起動できません。この状態になった場合は，E20 エミュレータは電源スイッチを切り替え，E1/EZ エミュレータは USB インタフェースケーブルを抜き差しして電源を入れ直してください。なお，EZ エミュレータの場合は，CS+ も再起動してください。

備考 デバッグ・ツールと切断すると，デバッグ・ツールと接続時のみ表示可能なパネル／ダイアログはすべてクローズします。

2.4.3 CS+ にデバッグ・ツールをホットプラグイン接続する【E1(JTAG)】【E20(JTAG)】

ホットプラグイン接続とは，プログラム実行中にデバッグ・ツールを接続して実行中のプログラムをデバッグすることができる接続方式です。
ホットプラグインの接続手順は，次のとおりです。

(1) プログラムの実行
エミュレータと接続しない状態で，予めターゲット・システム上のマイクロコントローラにダウンロードしたプログラムを実行します。

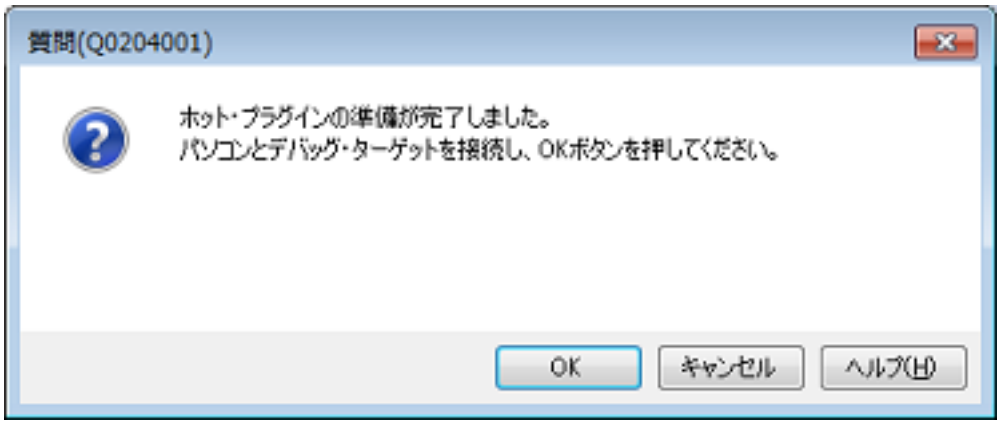
(2) デバッグ・ツールの指定
アクティブ・プロジェクトでホットプラグイン接続可能なデバッグ・ツール (【E1(JTAG)】/【E20(JTAG)】) を指定します。

備考 【E1(Serial)】【E20(Serial)】【EZ Emulator】
ホットプラグイン接続をサポートしていません。

(3) CS+ にデバッグ・ツールをホットプラグイン接続
［デバッグ］メニュー→［ホット・プラグイン］を選択することにより，ホットプラグイン接続の準備を開始します。

(4) ターゲット・システムの接続
ホットプラグイン接続の準備が完了すると，次のメッセージが表示されます。ターゲット・システムとエミュレータを接続してから [OK] ボタンをクリックしてください。現在のアクティブ・プロジェクトで選択しているデバッグ・ツールとの通信を開始します。

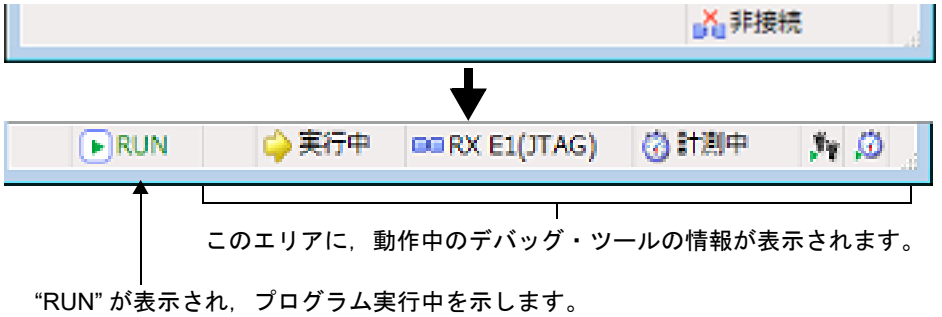
図 2.75　　ホットプラグイン接続準備完了メッセージ



注意　　【E1(JTAG)】
ホットプラグイン接続する場合，エミュレータとターゲット・システム間に別売の " ホットプラグアダプタ "(R0E000010ACB00) を接続する必要があります。

(5)　ホットプラグイン接続完了
デバッグ・ツールとの接続に成功すると，メイン・ウインドウのステータス・バーが，次のように変化します。なお，ステータス・バーに表示される各項目についての詳細は，メイン・ウインドウを参照してください。

図 2.76　　デバッグ・ツールとのホットプラグイン接続に成功したステータス・バー



注意 1.　　ホットプラグイン接続完了後，プログラムをブレークするまではトレース機能を使用できません。

注意 2.　　ホットプラグイン接続完了後，プログラムをブレークするまではリアルタイム RAM モニタ機能を使用できません。そのため，［トレース］カテゴリの [トレース機能の用途] プロパティにおいて，リアルタイム RAM モニタを指定しないでください。

注意 3.　　ホットプラグイン接続をした場合，プロジェクトにユーザ情報として設定されていたイベントは全て削除されます（ビルトイン・イベントは除く）。

注意 4.　　ホットプラグイン接続をする場合，ソフトウエア・ブレークポイントが設定されたプロジェクトを使用しないでください。正しくホットプラグイン接続しない可能性があります。
アクティブ・プロジェクトのイベント パネル上にてソフトウエア・ブレークポイントが全て削除されていることを確認してください（**「2.17.5 イベントを削除する」参照）。**

注意 5.　　ホットプラグイン接続では，ID コードを確認する為，プログラムを一時的に約 800μs 停止させています。（CPU クロック：100MHz，JTAG クロック：16.5MHz）

注意 6.　　【RX630, RX631, RX63N, RX63T グループ】
マイクロコントローラに書かれたエンディアン選択レジスタ（MDEB, MDES）のエンディアン値とプロジェクトのエンディアン情報を合わせてください。エンディアン選択レジスタの詳細については，マイクロコントローラのハードウエア・マニュアルを参照してください。
また，フラッシュ書き換えプログラムが動作中にホットプラグはできません。

2.5 ダウンロード／アップロード

この節では，デバッグ対象となるプログラム（ロード・モジュール・ファイルなど）を CS+ へダウンロードする方法と，デバッグ中のメモリ内容を CS+ からファイルへアップロードする方法を説明します。

2.5.1 ダウンロードを実行する

デバッグ対象となるロード・モジュール・ファイルのダウンロードを実行します。
次に示す手順に従って，プロパティ パネルの［ダウンロード・ファイル設定］タブにおけるダウンロードのための設定を行ったのち，ダウンロードを実行してください。

(1)　［ダウンロード］カテゴリの設定

図 2.77　［ダウンロード］カテゴリ【E1(JTAG)】【E1(Serial)】【E20(JTAG)】【E20(Serial)】【EZ Emulator】

ダウンロード	
ダウンロードするファイル	[1]
[0]	DefaultBuild¥Tutorial.abs
ファイル	DefaultBuild¥Tutorial.abs
ファイルの種類	ロード・モジュール・ファイル
オブジェクトをダウンロードする	はい
シンボルをダウンロードする	はい
PIC/PIDオフセットを指定する	いいえ
入力補完機能用の情報を生成する	はい
ダウンロード後にCPUをリセットする	はい
ダウンロード前にフラッシュROMを消去する	はい
ダウンロード前にデータフラッシュROMを消去する	いいえ
イベント設定位置の自動変更方法	イベントを保留にする

図 2.78　［ダウンロード］カテゴリ【シミュレータ】

ダウンロード	
ダウンロードするファイル	[1]
[0]	DefaultBuild¥Tutorial.abs
ファイル	DefaultBuild¥Tutorial.abs
ファイルの種類	ロード・モジュール・ファイル
オブジェクトをダウンロードする	はい
シンボルをダウンロードする	はい
PIC/PIDオフセットを指定する	いいえ
入力補完機能用の情報を生成する	はい
ダウンロード後にCPUをリセットする	はい
イベント設定位置の自動変更方法	イベントを保留にする

(a)　［ダウンロードするファイル］
ダウンロードの対象となるファイル名，およびダウンロード条件を表示します（プロパティ値の “[ ]” 内の数値は，現在ダウンロードの対象に指定されているファイル数を示す）。
ダウンロードの対象となるファイルは，メイン・プロジェクト／サブプロジェクトでビルド対象に指定しているファイルが自動的に決定されます[注]。
ただし，ダウンロードの対象となるファイル，およびダウンロード条件は，手動で変更することができます。
この場合は，「2.5.2 応用的なダウンロード方法」を参照してください。

注　外部ビルド・ツール（CS+ が提供するビルド・ツール以外のコンパイラ／アセンブラなど）により作成されたロード・モジュール・ファイルをダウンロードする場合，デバッグ専用プロジェクトを作成する必要があります。
デバッグ専用プロジェクトをデバッグの対象とする場合では，ユーザが，プロジェクト・ツリー上のダウンロード・ファイル・ノードにダウンロードするファイルを追加することで，ダウンロードの対象となるファイルがこのプロパティに反映されます。
なお，“ 外部ビルド・ツールの使用 ”，および “ デバッグ専用プロジェクト ” についての詳細は，「CS+ 統合開発環境 ユーザーズマニュアル プロジェクト操作編」を参照してください。

(b)　［ダウンロード後に CPU をリセットする］
ダウンロード完了後に CPU をリセットするか否かをドロップダウン・リストにより指定します。
CPU をリセットする場合は［はい］を選択してください（デフォルト）。

(c) ［ダウンロード前にフラッシュ ROM を消去する］【E1】【E20】【EZ Emulator】
ダウンロード実行前にフラッシュ ROM（プログラム ROM）を消去するか否かをドロップダウン・リストにより指定します。
フラッシュ ROM を消去する場合は［はい］を選択してください。

(d) [ダウンロード前にデータフラッシュ ROM を消去する]【E1】【E20】【EZ Emulator】
ダウンロード実行前にデータフラッシュ ROM を消去するか否かをドロップダウン・リストにより指定します。
データフラッシュ ROM を消去する場合は［はい］を選択してください（デフォルトでは［いいえ］が指定されます）。

(e) ［イベント設定位置の自動変更方法］
デバッグ作業を進めることにより，変更を加えたプログラムを再度ダウンロードした場合，現在設定されているイベントの設定位置（アドレス）が命令の途中になる場合があります。この場合の対象イベントの扱いをこのプロパティで指定します。
次のドロップダウン・リストによりどちらかを選択してください。

命令の先頭に移動する	命令の先頭アドレスに対象イベントを再設定します。
イベントを保留にする	対象イベントを保留状態にします（デフォルト）。

ただし，このプロパティでの指定は，デバッグ情報のないイベント設定位置に対してのみ適用されます。デバッグ情報があるイベント設定位置の場合は，常にソース・テキスト行の先頭に移動します。

(2) ［デバッグ情報］カテゴリの設定

図 2.79 ［デバッグ情報］カテゴリ

(a) ［CPU リセット後に指定シンボル位置まで実行する］
CPU リセット後，またはダウンロード完了後（［ダウンロード後に CPU をリセットする］プロパティで［はい］を指定している場合のみ）に，プログラムを指定シンボル位置まで実行するか否かをドロップダウン・リストにより指定します。
プログラムを指定シンボル位置まで実行する場合は［はい］を選択してください（デフォルト）。

備考　［ダウンロード後に CPU をリセットする］プロパティで［はい］を指定している場合では，このプロパティで［はい］を指定すると，ダウンロード完了後，［指定シンボル］プロパティで指定した位置のソース・テキストを表示した状態でエディタ パネルが自動的にオープンします。
また，［いいえ］を指定した場合では，リセット番地を表示した状態で同パネルがオープンします（リセット番地にソース・テキストが割り付けられていない場合は，逆アセンブル パネルで該当箇所を表示します）。

(b) ［指定シンボル］
このプロパティは，［CPU リセット後に指定シンボル位置まで実行する］プロパティにおいて［はい］を指定した場合のみ表示されます。
CPU リセット後にプログラムを実行して停止する位置を指定します。
0 ～ “ *アドレス空間の終了アドレス* ” の範囲のアドレス式を直接入力で指定してください（デフォルトでは［_main］が指定されます）。
ただし，指定したアドレス式がアドレスに変換できない場合，プログラムは実行されません。

備考　通常，次を指定します。
アセンブラ・ソースの場合：メイン関数に相当する先頭ラベル
C ソースの場合：　メイン関数名の先頭に付与したシンボル

(c) ［デバッグするオーバーレイ・セクションを設定する］
ロード・モジュール内にオーバーレイ・セクション（「2.8 オーバーレイ・セクションの設定」参照）が存在する場合，[はい] となります。
また，ロード・モジュール内にオーバーレイ・セクションが存在しない場合は，[いいえ] となります（変更不可）。

(d) ［オーバーレイ・セクション］
オーバーレイ・セクションが存在するアドレス・グループを表示します。

なお，このプロパティは，［デバッグするオーバーレイ・セクションを設定する］プロパティにおいて [はい] が設定されている場合のみ表示されます。

(e) ［メモリ使用量の上限サイズ［M バイト］］
デバッグ情報の読み込みで使用するメモリ量の上限サイズ［M バイト］を指定します。
直接入力により，100 ～ 1000 の範囲の整数を指定してください（デフォルトでは［500］が指定されます）。
メモリ不足が発生する場合，上限サイズを小さくすることで改善される可能性があります。ただし，上限サイズを小さくするとデバッグ・ツールの応答性が低下する場合があります。

注意　デフォルトの設定では，ダウンロード後に自動的に CPU をリセットし，指定シンボルまで実行します。この動作が不要な場合は，［ダウンロード後に CPU をリセットする］プロパティ，および［CPU リセット後に指定シンボル位置まで実行する］プロパティにおいて［いいえ］を指定してください。

(3) ダウンロードの実行
デバッグ・ツールバーの ボタンをクリックします。
なお，デバッグ・ツールと切断時にこの操作が行われた場合は，自動的にデバッグ・ツールと接続したのち，ダウンロードを実行します。

備考　デバッグ作業を進めることにより，変更を加えたプログラムを再度ダウンロードする場合は，メイン・ウインドウ上の［デバッグ］メニュー→［ビルド & デバッグ・ツールへダウンロード］を選択することにより，ビルド→ダウンロードを容易に行うことができます。

(4) ダウンロードをキャンセルする
ダウンロードの実行をキャンセルする場合，ダウンロードの進捗状況を表示する処理中表示 ダイアログの [キャンセル] ボタンを押下か，または [Esc] キーを押下してください。

ロード・モジュール・ファイルのダウンロードが成功すると，エディタ パネルが自動的にオープンし，ダウンロードしたファイルのソース・テキストが表示されます。

備考　ダウンロードの実行前／実行後に，I/O レジスタ /CPU レジスタ値を指定した値に自動的に書き換える処理を設定することができます（「2.18 フック処理を設定する」参照）。

2.5.2 応用的なダウンロード方法

ダウンロードの対象となるファイル，およびその際のダウンロード条件は変更することができます。
CS+ では，次のファイル形式をダウンロードすることができます。

表 2.2　ダウンロード可能なファイル形式

ファイル形式	拡張子	備考
ロード・モジュール・フォーマット	abs	
インテル・ヘキサ・フォーマット - 標準（オフセットアドレス） - 02（セグメントアドレスレコード） - 04（拡張リニアアドレスコード）	hex	03（スタートセグメントアドレスレコード），05（スタートリニアアドレスレコード）は無視します。
モトローラ S フォーマット -（S0, S1, S9-16 ビット） -（S0, S2, S8-24 ビット） -（S0, S3, S7-32 ビット）	mot	
バイナリ・データ・フォーマット	bin	

ダウンロード・ファイルの変更，およびその際のダウンロード条件の設定は，次のダウンロード・ファイル ダイアログにより行います。
ダウンロード・ファイル ダイアログは，プロパティ パネルの［ダウンロード・ファイル設定］タブ上の［ダウンロード］カテゴリ内 [ダウンロードするファイル］プロパティを選択することで欄内右端に表示される［...］ボタンをクリックするとオープンします。

図 2.80 ダウンロード・ファイル ダイアログのオープン

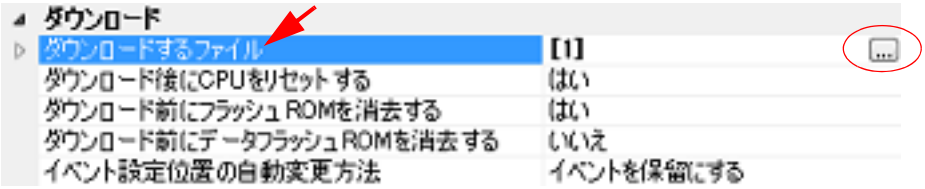
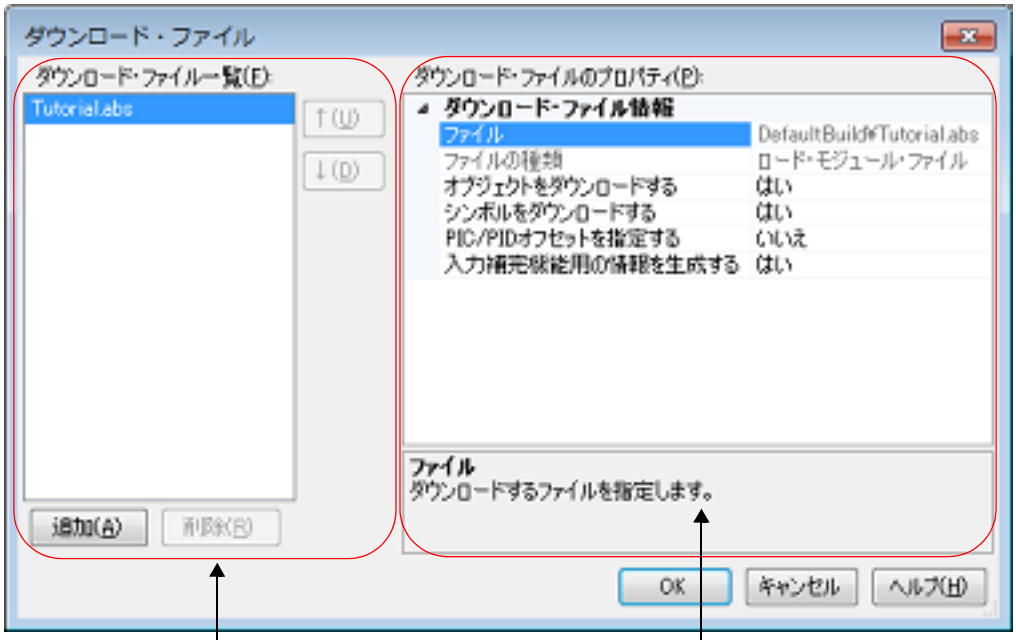


図 2.81 応用的なダウンロード方法（ダウンロード・ファイル ダイアログ）



［ダウンロード・ファイル一覧］エリア ［ダウンロード・ファイルのプロパティ］エリア

ここでは，上記ダウンロード・ファイル ダイアログにおける，次の場合の設定方法を説明します。

2.5.2.1 ロード・モジュール・ファイルのダウンロード条件を変更する
2.5.2.2 ダウンロード・ファイル（*.hex/*.mot/*.bin）を追加する
2.5.2.3 複数のロード・モジュール・ファイルをダウンロードする
2.5.2.4 ヘキサ・フォーマット／モトローラ S フォーマット／バイナリ・データ・フォーマットのファイルでソース・レベル・デバッグを行う
2.5.2.5 外部フラッシュ・メモリへダウンロードする【E1】【E20】

2.5.2.1 ロード・モジュール・ファイルのダウンロード条件を変更する

ダウンロードするロード・モジュール・ファイルのダウンロード条件（オブジェクト情報／シンボル情報の読み込みなど）を変更する場合は，ダウンロード・ファイル ダイアログにおいて，次の手順の操作を行ってください。

(1) ロード・モジュール・ファイルの選択
［ダウンロード・ファイル一覧］エリアにおいて，ダウンロードするロード・モジュール・ファイルを選択します。

(2) ダウンロード条件の変更
［ダウンロード・ファイル一覧］エリアでは，選択しているロード・モジュール・ファイルの現在のダウンロード条件が表示されます。
表示される各項目において，設定の変更を行います。

オブジェクトをダウンロードする	指定したファイルからオブジェクト情報をダウンロードするか否かを指定します。		
	デフォルト	*はい*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	オブジェクト情報をダウンロードします。
		いいえ	オブジェクト情報をダウンロードしません。
シンボルをダウンロードする	指定したファイルからシンボル情報をダウンロードするか否かを指定します[注1]。		
	デフォルト	*はい*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	シンボル情報をダウンロードします。
		いいえ	シンボル情報をダウンロードしません。
PIC/PID オフセットを指定する	ダウンロードするロード・モジュールの PIC（Position Independent Code）領域，および PID（Position Independent Data）領域の位置をロード・モジュール作成時から変更するか否かを指定します。この項目を " はい " に切り替えると，サブ項目に "PIC オフセット "，"PID オフセット " 項目を表示します。		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	PIC/PID オフセットを指定します[注2]。
		いいえ	PIC/PID オフセットを指定しません。
PIC オフセット	ロード・モジュール作成時のアドレスからオフセット値を入力します。 例えば，プログラム・セクションの開始アドレスが 0x1000 番地からだった場合に本項目に "1000" を入力すると，対象のセクションは 0x2000 番地にダウンロードされます。		
	デフォルト	*0*	
	変更方法	キーボードからの直接入力	
	指定可能値	0x0 ～ 0xFFFFFFFF の 16 進数	
PID オフセット	ロード・モジュール作成時に指定した PID レジスタへ設定するオフセット値を入力します。 例えば，ロード・モジュールの実行時に PID レジスタに 0x200 を設定する場合は，本項目に "200" を入力してください。		
	デフォルト	*0*	
	変更方法	キーボードからの直接入力	
	指定可能値	0x0 ～ 0xFFFFFFFF の 16 進数	
入力補完機能用の情報を生成する	この項目は，ダウンロードするファイルがロード・モジュール・フォーマットの場合のみ表示されます。 選択されたファイルからシンボル情報をダウンロードする際に，シンボル名の入力補完機能用の情報を生成するか否かを指定します。 ウォッチパネルやメモリパネルなどでシンボル名の入力補完機能を使う場合は " はい " を選択してください。なお，この情報はシンボルのダウンロード中に生成するため，ダウンロードに時間が掛かる場合があります。		
	デフォルト	はい	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	入力補完機能用の情報を生成します。
		いいえ	入力補完機能用の情報を生成しません。

注 1.　シンボル情報をダウンロードしない場合，ソース・レベル・デバッグを行うことはできません。

注 2.　PIC/PID 機能（「2.7 PIC/PID 機能の使用」参照）を使って作成していないロード・モジュールに対して " はい " を選択した時のデバッグ動作は保証できません。

(3)　［OK］ボタンのクリック
このダイアログでの設定を有効とし，ダウンロード条件が変更されます。

2.5.2.2 ダウンロード・ファイル（*.hex/*.mot/*.bin）を追加する

ダウンロードするファイル（ヘキサ・フォーマット（*.hex）／モトローラ S フォーマット（*.mot）／バイナリ・データ・フォーマット（*.bin））を追加する場合は，ダウンロード・ファイル ダイアログにおいて，次の手順の操作を行ってください。

(1)　［追加］ボタンのクリック
［追加］ボタンをクリックすると，［ダウンロード・ファイル一覧］エリアの最終行に空欄の項目（"-"）が表示されます。

(2)　追加するダウンロード・ファイルのプロパティ設定
［ダウンロード・ファイルのプロパティ］エリアにおいて，追加するダウンロード・ファイルの選択とダウンロード条件を設定します。
表示される各項目において，次の設定を行ってください。
設定を完了すると，［ダウンロード・ファイル一覧］エリア内の空欄の項目に，ここで指定したファイル名が反映されます。

ファイル	追加するダウンロード・ファイル（ヘキサ・フォーマット（*.hex）／モトローラ S フォーマット（*.mot）／バイナリ・データ・フォーマット（*.bin））を指定します（最大指定文字数：259 文字）。	
	デフォルト	*空欄*
	変更方法	キーボードからの直接入力，またはこのプロパティを選択すると右端に表示される［...］ボタンのクリックによりオープンするダウンロードするファイルを選択 ダイアログによる指定
	指定可能値	「表 2.2 ダウンロード可能なファイル形式」参照
ファイルの種類	追加するダウンロード・ファイルのファイル形式を指定します。 ここでは，［ヘキサ・ファイル］，[S レコード・ファイル] または［バイナリ・データ・ファイル］を選択します。	
	デフォルト	*ロード・モジュール・ファイル*
	変更方法	ドロップダウン・リストによる選択
	指定可能値	次のいずれか - ロード・モジュール・ファイル - ヘキサ・ファイル - S レコード・ファイル - バイナリ・データ・ファイル
オフセット	この項目は，ダウンロードするファイルがヘキサ・フォーマット（*.hex）／モトローラ S フォーマット（*.mot）の場合のみ表示されます。 指定したファイルのダウンロードを開始するアドレスからのオフセット値を指定します。	
	デフォルト	*0*
	変更方法	キーボードからの直接入力
	指定可能値	0x0 ～ 0xFFFFFFFF の 16 進数

開始アドレス	この項目は，ダウンロードするファイルがバイナリ・データ・フォーマット（*.bin）の場合のみ表示されます。指定したファイルをダウンロードする開始アドレスを指定します。		
	デフォルト	*0*	
	変更方法	キーボードからの直接入力	
	指定可能値	0x0 ～ 0xFFFFFFFF の 16 進数	

備考　オブジェクト情報，およびシンボル情報をダウンロードするか否かの指定は，ダウンロードするファイルの種類がロード・モジュール・ファイルの場合のみ行うことができます。

(3) ダウンロードの際の実行順序の確認
［ダウンロード・ファイル一覧］エリアでの表示順序が，ダウンロードの際の実行順序となります。
順序を変更する場合は［↓］／［↑］ボタンで変更してください。

(4) ［OK］ボタンのクリック
このダイアログでの設定を有効とし，指定したファイルがダウンロード・ファイルとして追加されます（プロパティ パネルの［ダウンロード・ファイル設定］タブ上の［ダウンロード］カテゴリ内に追加したファイル名とダウンロード条件が表示されます）。

2.5.2.3 複数のロード・モジュール・ファイルをダウンロードする

複数のロード・モジュール・ファイルをダウンロードする場合は，ダウンロード・ファイル ダイアログにおいて，次の手順の操作を行ってください。

注意　複数のロード・モジュール・ファイルから構成されるプログラムをデバッグする際は，配置アドレスが重ならないよう注意が必要です。

備考　複数のロード・モジュール・ファイルをダウンロードした場合，イベント パネルの詳細情報にはロード・モジュール名が表示されます。

(1) ［追加］ボタンのクリック
［追加］ボタンをクリックすると，［ダウンロード・ファイル一覧］エリアの最終行に空欄の項目（“-”）が表示されます。

(2) 追加するダウンロード・ファイルのプロパティ設定
［ダウンロード・ファイルのプロパティ］エリアにおいて，追加するロード・モジュール・ファイルの選択とダウンロード条件を設定します。
表示される各項目において，次の設定を行ってください。
設定を完了すると，［ダウンロード・ファイル一覧］エリア内の空欄の項目に，ここで指定したファイル名が反映されます。

ファイル	追加するロード・モジュール・フォーマットのファイルを指定します（最大指定文字数：259 文字）。		
	デフォルト	*空欄*	
	変更方法	キーボードからの直接入力，またはこのプロパティを選択すると右端に表示される［...］ボタンのクリックによりオープンするダウンロードするファイルを選択 ダイアログによる指定	
	指定可能値	「表 2.2 ダウンロード可能なファイル形式」参照	
ファイルの種類	追加するダウンロード・ファイルのファイル形式を指定します。 ここでは，［ロード・モジュール・ファイル］を選択します（デフォルト）。		
オブジェクトをダウンロードする	指定したファイルからオブジェクト情報をダウンロードするか否かを指定します。		
	デフォルト	*はい*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	オブジェクト情報をダウンロードします。
		いいえ	オブジェクト情報をダウンロードしません。

シンボルをダウンロードする	指定したファイルからシンボル情報をダウンロードするか否かを指定します[注1]。		
	デフォルト	*はい*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	シンボル情報をダウンロードします。
		いいえ	シンボル情報をダウンロードしません。
PIC/PID オフセットを指定する	ダウンロードするロード・モジュールの PIC（Position Independent Code）領域，および PID（Position Independent Data）領域の位置をロード・モジュール作成時から変更するか否かを指定します。この項目を " はい " に切り替えると，サブ項目に "PIC オフセット "，"PID オフセット " 項目を表示します。		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	PIC/PID オフセットを指定します[注2]。
		いいえ	PIC/PID オフセットを指定しません。
PIC オフセット	ロード・モジュール作成時のアドレスからのオフセット値を入力します。 例えば，プログラム・セクションの開始アドレスが 0x1000 番地からだった場合に本項目に "1000" を入力すると，対象のセクションは 0x2000 番地にダウンロードされます。		
	デフォルト	*0*	
	変更方法	キーボードからの直接入力	
	指定可能値	0x0 ～ 0xFFFFFFFF の 16 進数	
PID オフセット	ロード・モジュール作成時に指定した PID レジスタへ設定するオフセット値を入力します。 例えば，ロード・モジュールの実行時に PID レジスタに 0x200 を設定する場合は，本項目に "200" を入力してください。		
	デフォルト	*0*	
	変更方法	キーボードからの直接入力	
	指定可能値	0x0 ～ 0xFFFFFFFF の 16 進数	

注 1.　シンボル情報をダウンロードしない場合，ソース・レベル・デバッグを行うことはできません。

注 2.　PIC/PID 機能（「2.7 PIC/PID 機能の使用」参照）を使って作成していないロード・モジュールに対して " はい " を選択した場合の動作は保証できません。

備考　シンボル情報が不要なロード・モジュール・ファイルの場合では，［シンボルをダウンロードする］項目を［いいえ］に設定することにより，メモリの使用量を軽減することができます（ただし，このファイルをソース・レベルでデバッグすることはできません）。

(3) ダウンロードの際の実行順序の確認
［ダウンロード・ファイル一覧］エリアでの表示順序が，ダウンロードの際の実行順序となります。
順序を変更する場合は［↓］／［↑］ボタンで変更してください。

(4) ［OK］ボタンのクリック
このダイアログでの設定を有効とし，指定したロード・モジュール・ファイルがダウンロード・ファイルとして追加されます（プロパティ パネルの［ダウンロード・ファイル設定］タブ上の［ダウンロード］カテゴリ内に追加したファイル名が表示されます）。

2.5.2.4 ヘキサ・フォーマット／モトローラ S フォーマット／バイナリ・データ・フォーマットのファイルでソース・レベル・デバッグを行う

ヘキサ・フォーマット（*.hex），モトローラ S フォーマット（*.mot），またはバイナリ・データ・フォーマット（*.bin）のファイルをダウンロードの対象ファイルと指定している場合でも，作成元となったロード・モジュール・ファイルのシンボル情報を併せてダウンロードすることにより，ソース・レベル・デバッグを行うことができます。
この場合は，ダウンロード・ファイル ダイアログにおいて，次の手順の操作を行ってください。

(1) ［追加］ボタンのクリック
［追加］ボタンをクリックすると，［ダウンロード・ファイル一覧］エリアの最終行に空欄の項目（“-”）が表示されます。

(2) ロード・モジュール・ファイルのプロパティ設定
［ダウンロード・ファイルのプロパティ］エリアにおいて，各項目を次のとおりに指定します。

ファイル	ダウンロードするヘキサ・フォーマット（*.hex），モトローラ S フォーマット（*.mot），またはバイナリ・データ・フォーマット（*.bin）のファイルを作成する元となったロード・モジュール・ファイルを指定します。 キーボードからの直接入力，または［...］ボタンのクリックによりオープンするダウンロードするファイルを選択 ダイアログにより指定してください。
ファイルの種類	［ロード・モジュール・ファイル］を指定します（デフォルト）。
オブジェクトをダウンロードする	［いいえ］を指定します。
シンボルをダウンロードする	［はい］を指定します（デフォルト）。
PIC/PID オフセットを指定する	［いいえ］を指定します（デフォルト）注。

注　ヘキサ・フォーマット，モトローラ S フォーマット，またはバイナリ・データ・フォーマットのファイルは PIC 領域，PID 領域を区別してダウンロードすることができません。このためロード・モジュール・ファイルで PIC/PID オフセットを指定してもデバッグ機能と一致しなくなり，デバッグできません。

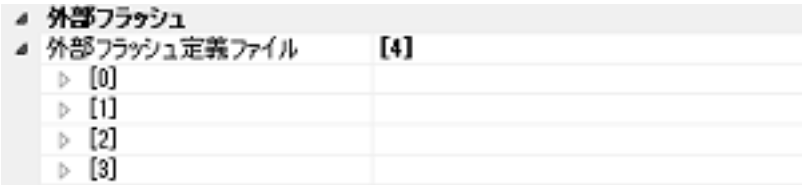
(3) ［OK］ボタンのクリック
このダイアログでの設定を有効とし，指定したロード・モジュール・ファイルがダウンロード・ファイルとして追加されます（ロード・モジュール・ファイル内に含まれるシンボル情報のみがダウンロードの対象となります）。

2.5.2.5 外部フラッシュ・メモリへダウンロードする【E1】【E20】

ダウンロード・ファイル ダイアログで指定したダウンロード・ファイルを，使用するマイクロコントローラの外部バスに接続されたフラッシュ・メモリ（外部フラッシュ・メモリ）にダウンロードすることができます。
この場合は，次の手順の操作を行ってください。

(1) プロパティ パネルの設定
プロパティ パネルの［接続用設定］タブ上の［外部フラッシュ］【E1】【E20】カテゴリ内において，外部フラッシュ定義ファイル（USD ファイル）の登録を行います。
外部フラッシュ・メモリは，［外部フラッシュ］【E1】【E20】カテゴリに USD ファイルを登録することで認識され，最大 4 つの USD ファイルを登録できます。
外部フラッシュ定義ファイルの“＋”マークをクリックすると，インデックス（[1] ～ [4]）毎に定義ファイル名，開始／終了アドレスおよびダウンロード前に外部フラッシュ ROM を消去するか否かの選択状況を表示します。

図 2.82　［外部フラッシュ］カテゴリ



- [外部フラッシュ定義ファイル]
  登録可能な USD ファイルの最大数を表示します。
  USD ファイルを登録する場合，サブプロパティとして展開表示される [ファイル] プロパティに USD ファイル名を指定します。
  ファイル名は直接入力するか（相対パスによる指定の場合はプロジェクト・フォルダ基点），このプロパティを選択することで設定欄右端に表示される［...］ボタンのクリックによりオープンする 外部フラッシュメモリ ダイアログ【E1】【E20】によりファイルを選択することで指定します。
  USD ファイルを登録すると，サブプロパティとして [開始アドレス] ／ [終了アドレス] ／ [ダウンロード前に外部フラッシュ ROM を消去する] プロパティが追加表示されます。
  [開始アドレス] ／ [終了アドレス] プロパティには，登録した USD ファイルに定義されている外部フラッシュメモリの領域が表示されます（編集不可）。
  [ダウンロード前に外部フラッシュ ROM を消去する] プロパティでは，ダウンロード実行前に外部フラッシュ ROM を消去するか否かをドロップダウン・リストにより指定します。
  外部フラッシュ ROM を消去する場合は [はい] を指定してください（デフォルトでは [いいえ] が指定されます）。
  また，USD ファイルの登録を解除する場合，対象となる [ファイル] プロパティのファイル名にキャレットを移動したのち，[Delete] キーをクリックしてください。
  なお，USD ファイルについては，弊社 Web サイトより External Flash Definition Editor のユーザーズ・マニュアルを参照してください。

図 2.83　　外部フラッシュメモリ ダイアログのオープン

注意 1.　USD ファイルのエンディアン情報は，プロパティ パネルの［デバッグ・ツール設定］タブにおける［メモリ］カテゴリ内の [メモリ・マッピング] プロパティに反映されません。そのため，対象となる外部領域のエンディアンを USD ファイルのエンディアン情報に合わせて変更してください。

注意 2.　重複するアドレスの USD ファイルを複数設定した場合，デバッグ・ツールとの接続ができません。

注意 3.　外部フラッシュ・メモリへダウンロードする場合，ワーク RAM として内蔵 RAM と CS 領域の RAM が使用可能です。SDRAM 領域（SDCS）は使用できません。

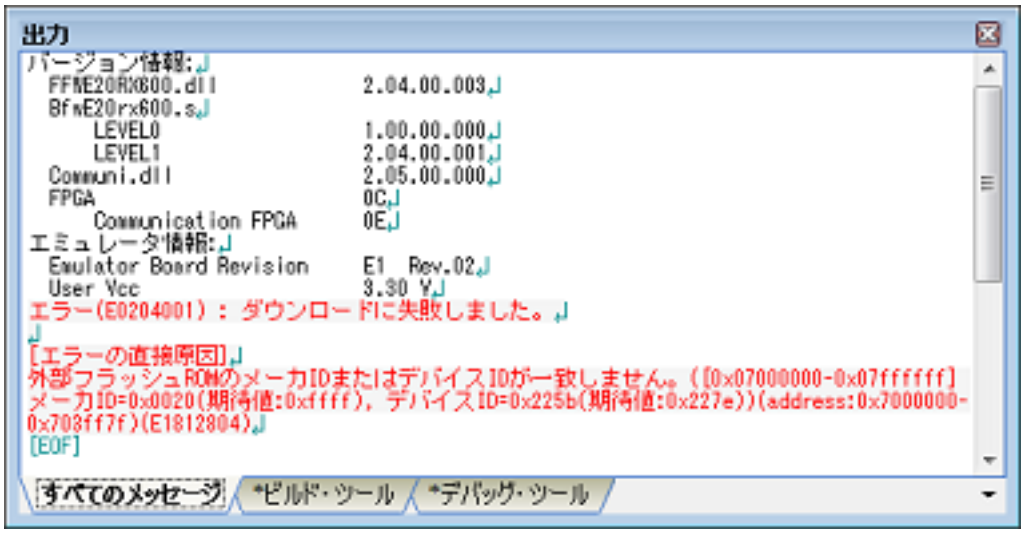
注意 4.　CS 領域をまたいでデバイスを割り当てる場合は，プロパティ パネルの [ダウンロード前にフラッシュ ROM を消去する] を [いいえ] に設定してください。

注意 5.　ワーク RAM 領域は，メモリ内容の退避・復帰を行うため，ユーザ・プログラムでも使用できますが，ワーク RAM に配置する領域は，以下の用途では使用できません。

- DMA または DTC 機能の転送元・転送先
- DTC ベクタテーブル，および転送情報配置アドレス
- DMAC，DTC 起動要因の割り込みベクタ

備考　メーカ ID またはデバイス ID が不一致の場合，リードした値と期待値が出力 パネルに表示されます。

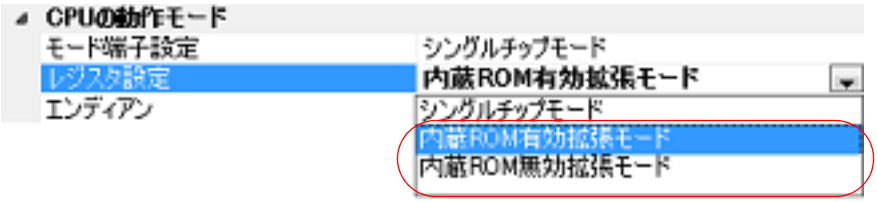
図 2.84 出力 パネルへのエラー表示



(2) CPU 動作モードの設定

外部フラッシュ・メモリへダウンロードする場合，デバッグ・ツールと接続する際にマイクロコントローラの動作モードを設定する必要があります。プロパティ パネルの［接続用設定］タブにおける［CPU の動作モード］【E1】【E20】【EZ Emulator】カテゴリ内の [レジスタ設定] プロパティをプルダウン・リストにて次のように [内蔵 ROM 有効拡張モード] または [内蔵 ROM 無効拡張モード] に変更します。

図 2.85 CPU 動作モードの指定



(3) ダウンロードの実行

ダウンロードを実行します（「(3) ダウンロードの実行」参照）。

注意 1. メモリ パネル上で外部フラッシュ・メモリを書き換えることはできません。

注意 2. 外部フラッシュ・メモリへダウンロード時に外部 RAM をワーク RAM として使用する場合は，CPU と同じエンディアンに設定してください。

注意 3. セクタ数が 4096 以下のフラッシュ・メモリを登録可能です。セクタ数が 4096 を越えるフラッシュ・メモリへの書き込みは保証できません。

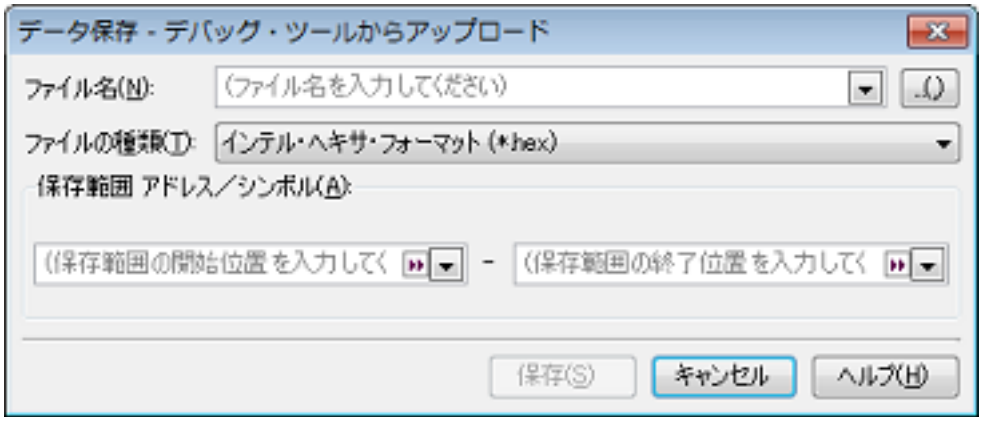
注意 4. ロックコマンドが使用できないフラッシュ・メモリデバイスを使用する場合，USD ファイルの生成時にロックビット解除を指定してください。この指定により，不要なロックビット状態の確認処理を省略することができます。
また，プロパティ パネルの [ダウンロード前にフラッシュ ROM を消去する] を [いいえ] に設定してください。
* ロックコマンドとは，以下のコマンドシーケンスで実行するコマンドを示します。
コマンドシーケンス：

	第 1 サイクル	第 2 サイクル	第 3 サイクル	第 4 サイクル
アドレス	555H	2AAH	555H	ブロックアドレス
データ	AAH	55H	48H	X1H / X0H

2.5.3 アップロードを実行する

現在接続しているデバッグ・ツールのメモリ内容を，任意のファイルに保存（アップロード）することができます。
アップロードは，［デバッグ］メニュー→［デバッグ・ツールからアップロード ...］を選択することによりオープンするデータ保存 ダイアログで行います。
このダイアログにおいて，次の手順で操作を行ってください。

図 2.86　　メモリ内容のアップロード（データ保存 ダイアログ）



(1)　［ファイル名］の指定
保存するファイル名を指定します。
テキスト・ボックスに直接入力するか（最大指定文字数：259 文字），またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
また，［...］ボタンをクリックすることでオープンするデータ保存ファイルを選択 ダイアログにより，ファイルを選択することもできます。

(2)　［ファイルの種類］の指定
保存するファイルの形式を次のドロップダウン・リストにより選択します。
選択できるファイルの形式は次のとおりです。

表 2.3　　アップロード可能なファイル形式

リスト表示	形式
インテル・ヘキサ・フォーマット (*.hex)	ヘキサ・フォーマット注
モトローラ S フォーマット (*.mot)	S レコード・フォーマット
バイナリ・データ (*.bin)	バイナリ・データ・フォーマット

注　　インテル・ヘキサ・フォーマットでは常に“拡張リニアアドレスレコード（32 ビット・アドレス）”で保存します。

(3)　［保存範囲 アドレス / シンボル］の指定
ファイルに保存する範囲を“開始アドレス”と“終了アドレス”で指定します。
それぞれのテキスト・ボックスに 16 進数の数値／アドレス式を直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。

備考　　このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます。（「2.20.2 シンボル名の入力補完機能」参照）。

(4)　［保存］ボタンのクリック
指定したファイルに指定した形式で，メモリ内容をアップロード・データとして保存します。

2.6 プログラムの表示と変更

この節では，デバッグ情報を持ったロード・モジュール・ファイルをデバッグ・ツールにダウンロードした場合のプログラムの表示方法，およびその変更方法について説明します。
ダウンロードしたプログラムの表示は，次の 2 つのパネルで行うことができます。

- エディタ パネル
  ソース・ファイルの表示／編集を行うほか，ソース・レベル・デバッグ（「2.9.3 プログラムをステップ実行する」参照），およびコード・カバレッジ測定結果の表示（「2.15.2 カバレッジ測定結果を表示する」参照）を行います。

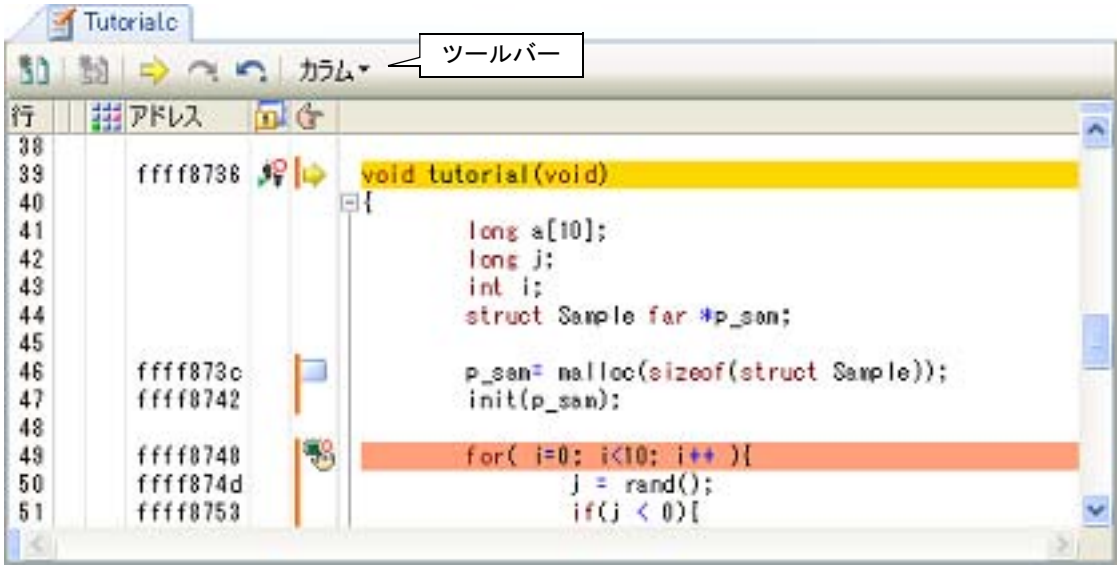
- 逆アセンブル パネル
  ダウンロードしたプログラム（メモリ内容）の逆アセンブル結果の表示／編集（ライン・アセンブル）を行うほか，命令レベル・デバッグ（「2.9.3 プログラムをステップ実行する」参照），およびコード・カバレッジ測定結果の表示（「2.15.2 カバレッジ測定結果を表示する」参照）を行います。
  なお，このパネルでは，逆アセンブル結果の表示とともに，対応するソース・テキストも表示することができます（デフォルト）。

備考　通常，ソース・レベル・デバッグを行うためには，デバッグ情報を持つロード・モジュール・ファイルをダウンロードする必要がありますが，ヘキサ・フォーマット（*.hex），モトローラ S フォーマット（*.mot），またはバイナリ・データ・フォーマット（*.bin）のファイルをダウンロード対象として，ソース・レベル・デバッグを行うことも可能です（「2.5.2.4 ヘキサ・フォーマット／モトローラ S フォーマット／バイナリ・データ・フォーマットのファイルでソース・レベル・デバッグを行う」参照）。

2.6.1 ソース・ファイルを表示する

ソース・ファイルの表示は，次のエディタ パネルで行います。
エディタ パネルは，ロード・モジュール・ファイルが正常にダウンロードされると，指定された位置（「2.5.1 ダウンロードを実行する」参照）のソース・テキストを表示した状態で自動的にオープンします。
手動でエディタ パネルをオープンする場合は，プロジェクト・ツリー パネルにおいてソース・ファイルをダブルクリックしてください。
なお，各エリアの見方，および機能についての詳細は，「CS+ 統合開発環境 ユーザーズマニュアル エディタ編」を参照してください。

図 2.87　ソース・ファイルの表示（エディタ パネル）



2.6.2 逆アセンブル結果を表示する

ダウンロードしたプログラム（メモリ内容）の逆アセンブル結果（逆アセンブル・テキスト）の表示は，次の逆アセンブル パネルで行います。

［表示］メニュー→［逆アセンブル］→［逆アセンブル *1* ～ *4*］を選択してください。

逆アセンブル パネルは，最大 4 個までオープンすることができ，各パネルはタイトルバーの“逆アセンブル 1”，“逆アセンブル 2”，“逆アセンブル 3”，“逆アセンブル 4”の名称で識別されます。

なお，各エリアの見方，および機能についての詳細は，逆アセンブル パネルの項を参照してください。

図 2.88 逆アセンブル結果の表示（逆アセンブル パネル）

備考 ツールバーの [表示] → ボタンをクリックすることによりオープンするスクロール範囲設定 ダイアログにより，このパネルの垂直スクロール・バーのスクロール範囲を設定することができます。

ここでは，次の操作方法について説明します。

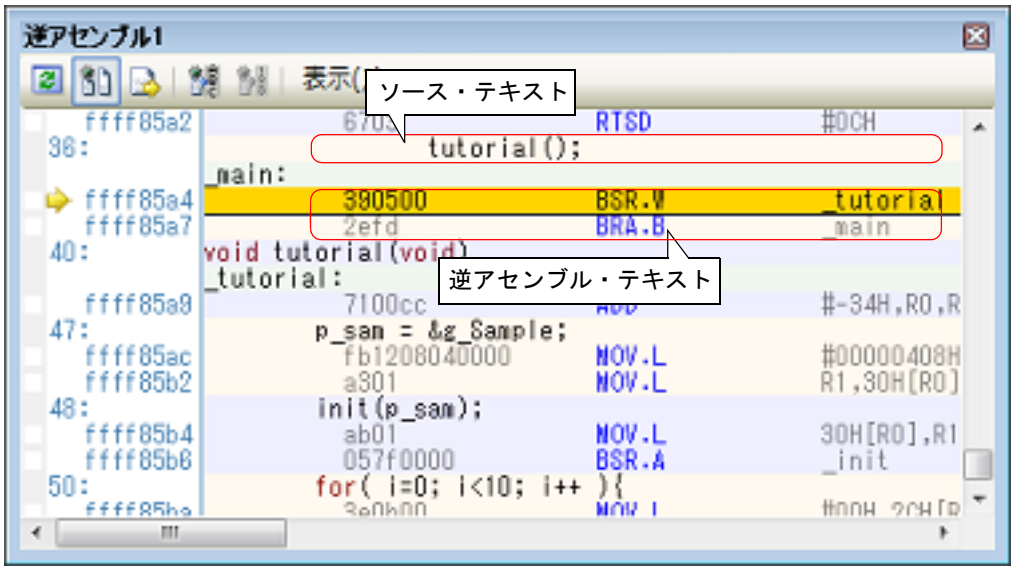
2.6.2.1 表示モードを変更する
2.6.2.2 表示形式を変更する
2.6.2.3 指定アドレスへ移動する
2.6.2.4 シンボル定義箇所へ移動する
2.6.2.5 逆アセンブル結果の表示内容を保存する

2.6.2.1 表示モードを変更する

ツールバーの ボタン（トグル）をクリックすることにより，逆アセンブル パネルの表示モードを切り替えることができます。

- 混合表示モード
  ソース・テキストと逆アセンブル・テキストを併せて表示する，デフォルトの表示モードです。

図 2.89 混合表示モード（逆アセンブル パネル）



- 逆アセンブル表示モード
  ソース・テキストを非表示にし，逆アセンブル・テキストのみを表示します。

図 2.90 逆アセンブル表示モード（逆アセンブル パネル）

2.6.2.2 表示形式を変更する

逆アセンブル結果の表示形式は，ツールバーの次のボタンにより，自由に変更することができます。

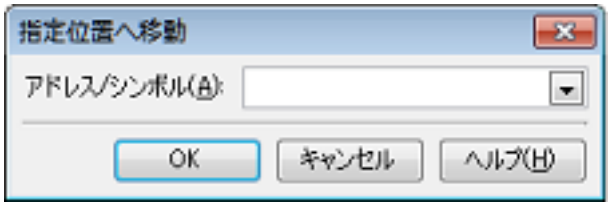
表示	表示形式を変更する次のボタンを表示します。
	ラベルのオフセット値を表示します。アドレスにラベルが定義されていない場合，一番近いラベルからのオフセット値を表示します。
	アドレス値を“シンボル＋オフセット値”で表示します（デフォルト）。 ただし，アドレス値にシンボルが定義されている場合は，シンボルのみを表示します。

2.6.2.3 指定アドレスへ移動する

逆アセンブル・テキスト上の指定アドレスへの移動は，コンテキスト・メニューの［移動 ...］を選択することによりオープンする指定位置へ移動 ダイアログで行います。

このダイアログにおいて，次の手順で操作を行ってください。

図 2.91 逆アセンブル結果内のアドレスへ移動（指定位置へ移動 ダイアログ）



(1) ［アドレス / シンボル］の指定
キャレットを移動したいアドレスを指定します。
テキスト・ボックスにアドレス式を直接入力するか（最大指定文字数：1024 文字），またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。

備考 このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます（「2.20.2 シンボル名の入力補完機能」参照）。

(2) ［OK］ボタンのクリック
指定したアドレスへキャレットを移動します。

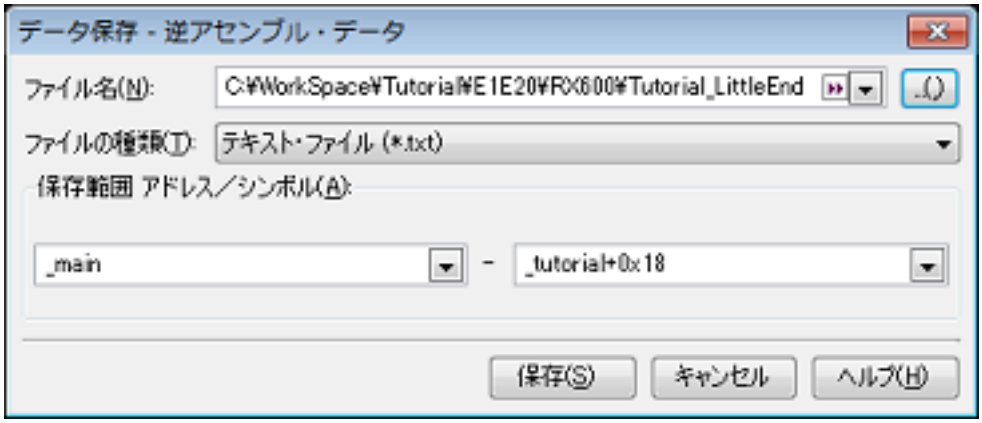
2.6.2.4 シンボル定義箇所へ移動する

シンボルが定義されているアドレスに，キャレット位置を移動することができます。
シンボルを参照している命令にキャレットを移動したのち，ツールバーの ボタンをクリックしてください。
また，この操作に続き，ツールバーの ボタンをクリックすると，キャレット移動前のシンボルを参照している命令にキャレット位置を戻します。

2.6.2.5 逆アセンブル結果の表示内容を保存する

逆アセンブル結果の内容をテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存することができます。
ファイルに保存する際は，デバッグ・ツールから最新の情報を取得し，このパネル上での表示形式に従ったデータで保存します。
［ファイル］メニュー→［名前を付けて逆アセンブル・データを保存 ...］を選択すると，次のデータ保存 ダイアログがオープンします（この際，パネル上で範囲選択した状態でこの操作を行うと選択範囲のみの逆アセンブル・データを保存することができます）。
このダイアログにおいて，次の手順で操作を行ってください。

図 2.92 逆アセンブル・データの保存（データ保存 ダイアログ）



(1) ［ファイル名］の指定
保存するファイル名を指定します。
テキスト・ボックスに直接入力するか（最大指定文字数：259 文字），またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
また，［...］ボタンをクリックすることでオープンするデータ保存ファイルを選択 ダイアログにより，ファイルを選択することもできます。

(2) ［ファイルの種類］の指定
保存するファイルの形式を次のドロップダウン・リストにより選択します。

選択できるファイルの形式は次のとおりです。

リスト表示	形式
テキスト・ファイル (*.txt)	テキスト形式（デフォルト）
CSV(カンマ区切り) (*.csv)	CSV 形式注

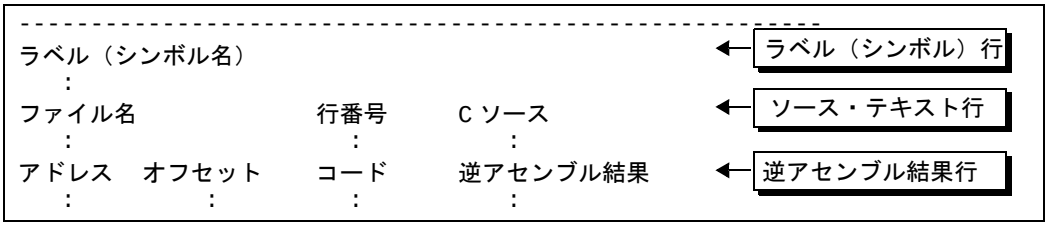
注　各データを “,” で区切り保存します。
なお，データ内に “,” が含まれている際の不正形式を避けるため，各データを “"” （ダブルクォーテーション）で括り出力します。

(3) ［保存範囲 アドレス／シンボル］の指定
ファイルに保存する範囲を “ 開始アドレス ” と “ 終了アドレス ” で指定します。
それぞれのテキスト・ボックスに 16 進数の数値／アドレス式を直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
なお，パネル上で範囲選択している場合は，デフォルトでその選択範囲がテキスト・ボックスに指定されます。
範囲選択していない場合は，現在のパネルの表示範囲が指定されます。

備考　このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます（「2.20.2 シンボル名の入力補完機能」参照）。

(4) ［保存］ボタンのクリック
指定したファイルに，指定した形式で逆アセンブル・データを保存します。

図 2.93　逆アセンブル・データ保存の際の出力イメージ



備考 1.　［ファイル］メニュー→［逆アセンブル・データを保存］の選択によりパネルの内容を上書き保存する場合，逆アセンブル パネル（逆アセンブル 1 ～ 4）はそれぞれ個別に扱われます。
また，保存範囲についても，前回指定したアドレス範囲で保存されます。

備考 2.　[ファイル] メニュー→ [印刷 ...] を選択することにより，現在このパネルで表示している画像イメージを印刷することができます。

2.6.3 他の処理と平行してビルドを実行する

CS+ では，次のタイミングでビルドを自動で開始する機能を提供しています（ラピッド・ビルド機能）。

(1) デバッグ専用プロジェクト以外の場合

- プロジェクトに追加している C ソース・ファイル／ C++ ソース・ファイル／アセンブラ・ソース・ファイル／ヘッダ・ファイル／リンク・ディレクティブ・ファイル／シンボル情報ファイル／オブジェクト・モジュール・ファイル／リロケータブル・モジュール・ファイル／ライブラリ・ファイルのいずれかを更新したとき
- プロジェクトにビルド対象ファイルを追加，または削除したとき
- オブジェクト・モジュール・ファイルのリンク順を変更したとき
- ビルド・ツール，およびビルド対象ファイルのプロパティを変更したとき

(2) デバッグ専用プロジェクトの場合

- デバッグ専用プロジェクトに追加している C ソース・ファイル／ C++ ソース・ファイル／アセンブラ・ソース・ファイル／ヘッダ・ファイルを編集して保存したとき
- デバッグ専用プロジェクトに C ソース・ファイル／ C++ ソース・ファイル／アセンブラ・ソース・ファイル／ヘッダ・ファイルを追加，または削除したとき
- デバッグ専用プロジェクトのプロパティを変更したとき

ラピッド・ビルド機能を有効にすることにより，上記の操作と平行してビルドを行うことができます。

ラピッド・ビルド機能の有効／無効の設定は，［ビルド］メニュー→［ラピッド・ビルド］の選択により切り替えます（デフォルトで有効に設定されています）。

注意　外部エディタを使用する場合，この機能を有効にするためには，オプション ダイアログの［全般 - ビルド／デバッグ］カテゴリの［登録されたファイルの変更を監視する］をチェックする必要があります。

備考 1.　ソース・ファイル編集後，［Ctrl］＋［S］キーの押下により，こまめに上書き保存することを推奨します。

備考 2.　ラピッド・ビルドの有効／無効は，プロジェクト全体（メイン・プロジェクト，およびサブプロジェクト）に対して設定されます。

備考 3.　ラピッド・ビルドの実行中に，ラピッド・ビルドを無効に切り替えた場合は，その場でラピッド・ビルドの実行を中止します。

2.6.4 ライン・アセンブルを行う

逆アセンブル パネルで表示されている命令／命令コードは，編集（ライン・アセンブル）することができます。
ここでは，次の操作方法について説明します。

2.6.4.1 命令を編集する
2.6.4.2 命令コードを編集する

2.6.4.1 命令を編集する

命令を編集する場合は，次の手順で操作を行ってください。

(1) 編集モードへの切り替え
対象命令をダブルクリックするか，または対象命令にキャレットを移動した状態でコンテキスト・メニューの［命令の編集］を選択すると，編集対象が編集モードに切り替わります。

(2) 命令の編集
キーボードから直接命令の文字列を編集します。

(3) メモリへの書き込み
編集終了後，［Enter］キーを押下することにより，変更された命令が自動的にライン・アセンブルされ，コードがメモリに書き込まれます。
ただし，この際に，変更結果が不正な命令となる場合は，編集された文字列が赤色で表示され，メモリへの書き込みは行いません。

なお，表示されている逆アセンブル結果を別の命令で上書きすることによりメモリに空きが生じた場合，次の例のように自動的に nop 命令でバイト数を補います。

例 1.　3 行目の ADD 命令（6 バイト命令）を BCLR 命令（4 バイト命令）で上書きした場合

編集前	5020	AND	[R2].UB,R0
	99e0	MOV.W	0CH[R6],R0
	700fb98a322a	ADD	#2A328AB9H,R0,R15
	5327	AND	R2,R7
編集後	5020	AND	[R2].UB,R0
	99e0	MOV.W	0CH[R6],R0
	f22a45b2	BCLR	#2,0B245H[R2]
	03	NOP	
	03	NOP	
	5327	AND	R2,R7

例 2.　2 行目の MOV.W 命令（2 バイト命令）を BCLR 命令（4 バイト命令）で上書きした場合

編集前	5020	AND	[R2].UB,R0
	99e0	MOV.W	0CH[R6],R0
	700fb98a322a	ADD	#2A328AB9H,R0,R15
	5327	AND	R2,R7

編集後	5020	AND	[R2].UB,R0
	99e0	MOV.W	0CH[R6],R0
	f22a45b2	BCLR	#2,0B245H[R2]
	03	NOP	
	03	NOP	
	03	NOP	
	03	NOP	
	5327	AND	R2,R7

2.6.4.2 命令コードを編集する

命令コードを編集する場合は，次の手順で操作を行ってください。

(1) 編集モードへの切り替え
対象命令コードをダブルクリックするか，または対象命令コードにキャレットを移動した状態で表示されるコンテキスト・メニューの［コードの編集］を選択すると，編集対象が編集モードに切り替わります。

(2) 命令コードの編集
キーボードから直接命令コードの文字列を編集します。

(3) メモリへの書き込み
編集終了後，［Enter］キーを押下することにより，命令コードがメモリに書き込まれます。
ただし，この際に，変更結果が不正な命令となる場合は，編集された文字列が赤色で表示され，メモリへの書き込みは行いません。
命令コードがメモリに書き込まれた場合は，逆アセンブル結果も同時に更新されます。

2.7 PIC/PID 機能の使用

PIC/PID 機能は，一度リンクが完了して配置アドレスが確定した ROM 上のコードやデータを，リンクをやり直すことなく，任意のアドレスに配置して利用できるようにする機能です。
PIC/PID 機能を利用する場合，プログラムは " マスタ " と " アプリケーション " で構成します。
PIC/PID 機能では，ROM 上のコードやデータを PIC や PID にしたプログラムを " アプリケーション "，" アプリケーション " を実行させるのに必要なプログラムを " マスタ " と呼びます。
この節では，ROM 上のコードやデータを PIC や PID にした " アプリケーション " プログラム（ロード・モジュール）を，別のアドレスに配置を変更してデバッグする方法について説明します。

- PIC
  pic オプションを有効にしてコンパイルすると，PIC 機能が有効になり，コード領域（P セクション）が PIC になります。PIC は分岐先アドレスや関数アドレスの取得をすべて PC 相対で行うため，リンク後も任意のアドレスに配置することができます。
- PID
  pid オプションを有効にしてコンパイルすると，PID 機能が有効になり，ROM データ領域（C，C_2，C_1，W，W_2，W_1，および L セクション）が PID になります。プログラムは PID に対し，その先頭アドレスを示すレジスタ (PID レジスタ) から相対のアクセスでアクセスします。" マスタ " で PID レジスタの設定値を変化させることにより，リンク後も PID を任意のアドレスに移動することができます。

備考 1.　PIC/PID 機能については，「CC-RX コンパイラ ユーザーズマニュアル」を参照してください。

備考 2.　ビルド・ツールにおける PIC/PID 機能の設定については，「CS+ 統合開発環境 ユーザーズマニュアル CC-RX ビルド・ツール操作編」を参照してください。

ROM 上のコードやデータを PIC や PID にしたロード・モジュールのアドレスの配置を変更したのち，デバッグを始めるには，次の手順で行ってください。

(1) ダウンロード・ファイルの追加
" アプリケーション " のロード・モジュール・ファイルを " マスタ " のダウンロード・ファイルとして追加します（「2.7.1 PIC/PID 機能を使用したロード・モジュールのアドレスの配置を変更する」参照）。

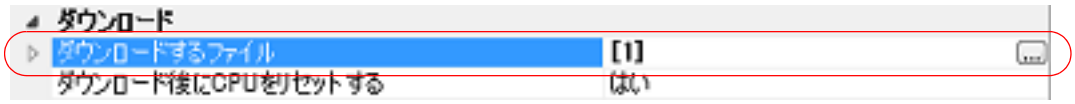
(2) ロード・モジュール・ファイルのダウンロード条件の指定
" アプリケーション " のロード・モジュールに対し，ロード・モジュール作成時のアドレスからのオフセット値（[PIC オフセット]），および作成時に指定した PID レジスタへ設定するオフセット値（[PID オフセット]）を指定します。

(3) ダウンロードの実行
" マスタ " および " アプリケーション " のロード・モジュール・ファイルのダウンロードを実行します（「2.5.1 ダウンロードを実行する」参照）。
新しいアドレスに配置された ROM 上のコードやデータをデバッグできるようになります。

2.7.1 PIC/PID 機能を使用したロード・モジュールのアドレスの配置を変更する

" マスタ " への " アプリケーション " のロード・モジュール・ファイルの追加は，プロパティ パネルの［ダウンロード・ファイル設定］タブ上の［ダウンロード］カテゴリ内 [ダウンロードするファイル］プロパティで行います。

図 2.94　［ダウンロード］カテゴリ



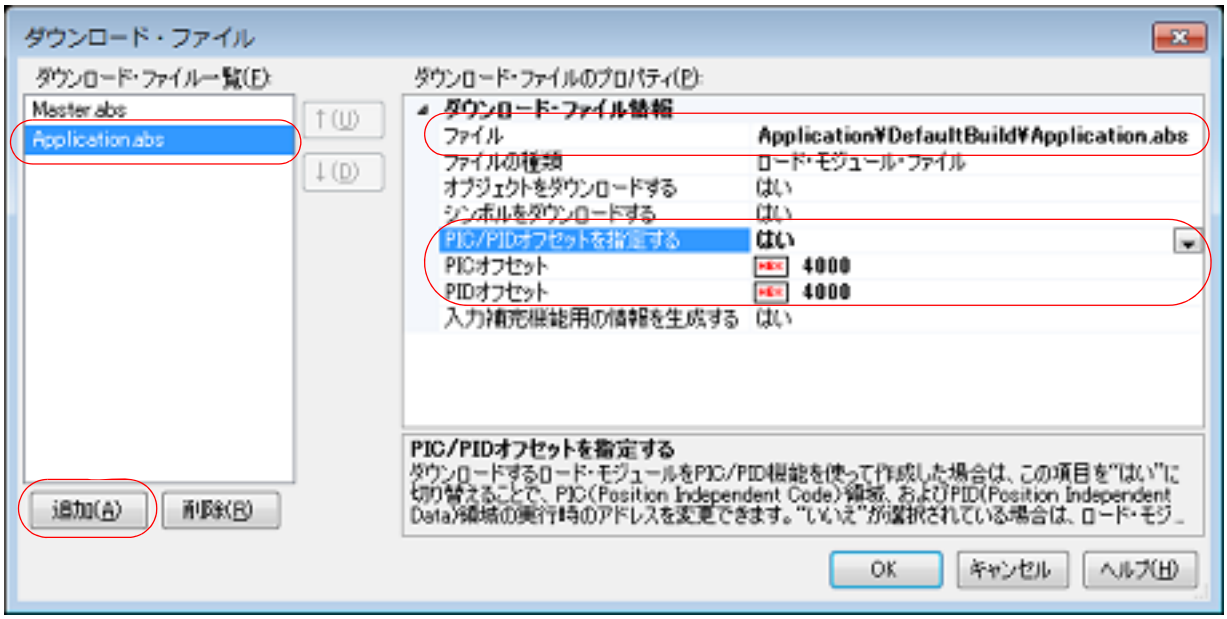
- ［ダウンロードするファイル］
  [...] ボタンをクリックすることにより，ダウンロード・ファイル ダイアログをオープンします。

ダウンロード・ファイル ダイアログの［ダウンロード・ファイル一覧］エリアの［追加］ボタンをクリックし，［ダウンロード・ファイルのプロパティ］エリアで " アプリケーション " のロード・モジュールの情報を設定します。

- ［ファイル］
  ダウンロードする " アプリケーション " のロード・モジュール・ファイルを指定します。
- ［PIC/PID オフセットを指定する］
  ［はい］を選択します。これにより，［PIC オフセット］および［PID オフセット］が表示されます。

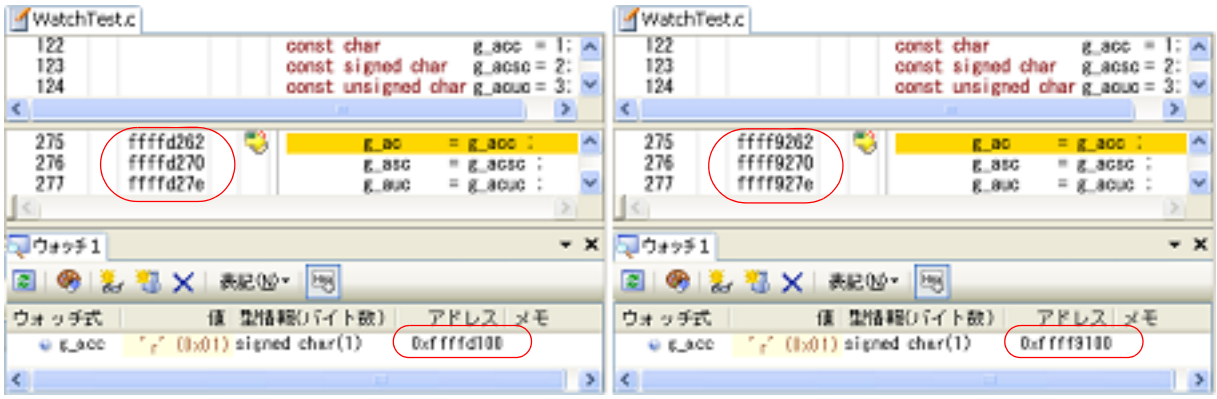
- ［PIC オフセット］
  ロード・モジュール作成時のアドレスからのオフセット値を指定します。
- ［PID オフセット］
  ロード・モジュール作成時に指定した PID レジスタへ設定するオフセット値を指定します。

図 2.95　　ダウンロード・ファイルの追加とダウンロード条件の変更 （ダウンロード・ファイル ダイアログ）



［PIC オフセット］および［PID オフセット］の値を変更したのち，ロード・モジュールのダウンロードを実行すると，以下の図のように，P セクションのアドレス，および，外部変数または静的変数のアドレスの配置が変更されます。
左側の図は，ダウンロード・ファイル ダイアログの［PIC オフセット］および［PID オフセット］に “4000” を指定した場合，右側の図は，“0” を指定した場合に，ダウンロードを実行した例です。
左側の図では，元のアドレスに対して “4000” が加算されています。

図 2.96　　［PIC オフセット］および［PID オフセット］のオフセット値を変更後のダウンロード実行例



2.8 オーバーレイ・セクションの設定

CC-RX が使用する最適化リンケージエディタ（Optlnk）は，プログラム中に定義した複数のセクションを同じアドレスに割り付けることができる start オプションを提供しています。このように割り付けられたセクションを“オーバーレイ・セクション”と呼びます。
この節では，ロード・モジュール内のオーバーレイ・セクションに対する“オーバーレイ・セクションの選択機能”の設定方法について説明します。

プログラムの実行時，ロード・モジュール内の同じアドレスに割り付けられたセクションはいずれか 1 つのセクションしか実行されません。
デバッグ・ツールは，同じアドレスに割り付けられたセクションのうち，デバッグ対象とするオーバーレイ・セクション（以降，優先セクションと呼びます）を選択する“オーバーレイ・セクションの選択機能”を提供しています。
プログラムの実行前に優先セクションを選択することにより，非優先セクションのデバッグ情報が隠蔽されるため，優先セクションをデバッグしやすくなります。

以下は，Optlnk の start オプションを使用したオーバーレイ・セクションの定義例になります。

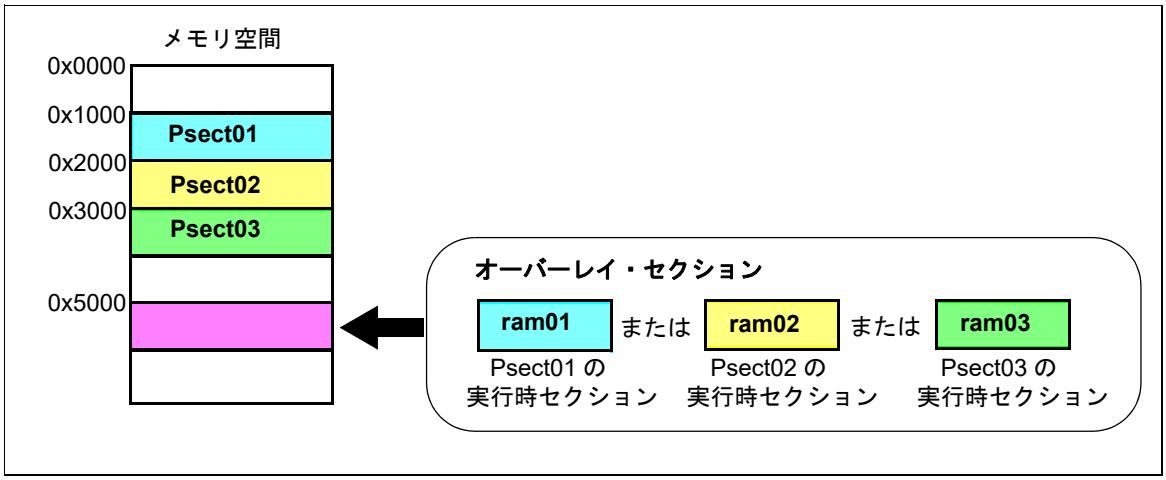
例　　Optlnk によるオーバーレイ・セクションの定義

```
start = Psect01/1000,Psect02/2000,Psect03/3000,ram01:ram02:ram03/5000
rom = Psect01=ram01,Psect02=ram02,Psect03=ram03
```

rom オプションにより，“*PsectXX*” セクションと同サイズの “*ramXX*” セクションを確保し，“*PsectXX*” セクション内定義シンボルを “*ramXX*” セクションのアドレス上でリロケーションします（“*XX*” は，この例では 01，02，または 03 になります）。

備考　　ビルド・ツールにおけるオーバーレイ・セクションの設定については，「CS+ 統合開発環境 ユーザーズマニュアル CC-RX ビルド・ツール操作編」を参照してください。

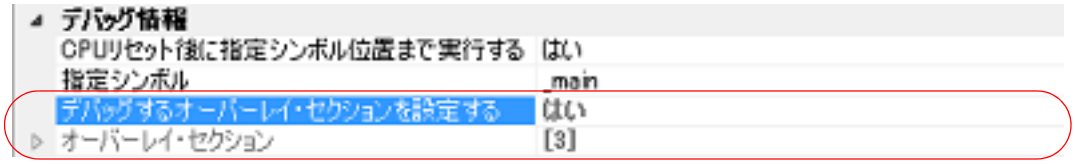
図 2.97　　オーバーレイ・セクションの配置



2.8.1 優先セクションを指定する

優先セクションの指定は，プロパティ パネルの［ダウンロード・ファイル設定］タブ上の［デバッグ情報］カテゴリ内で行います。

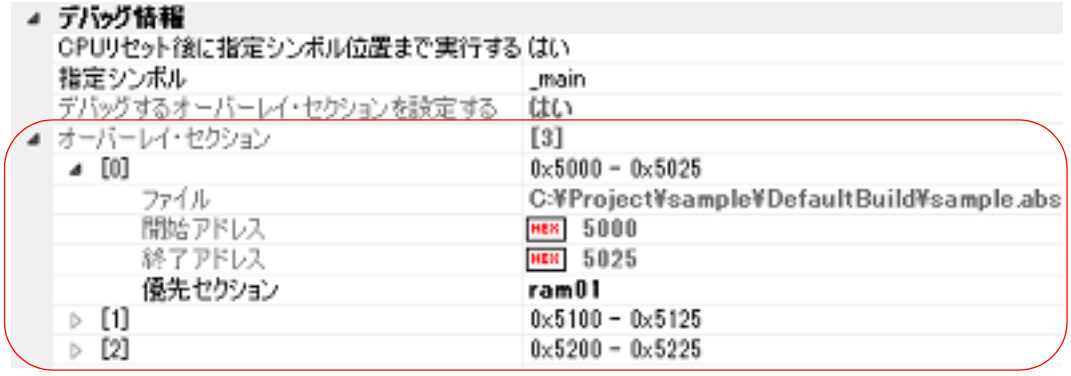
図 2.98　　［デバッグ情報］カテゴリ



(1)　［デバッグするオーバーレイ・セクションを設定する］
ロード・モジュール内にオーバーレイ・セクションが存在する場合に，［はい］となります（変更不可）。

(2) ［オーバーレイ・セクション］
オーバーレイ・セクションが存在するアドレス・グループを表示します。
このプロパティは，［デバッグするオーバーレイ・セクションを設定する］プロパティにおいて［はい］が設定されている場合のみ表示されます。

図 2.99 ［オーバーレイ・セクション］プロパティ



現在のオーバーレイ・セクションの情報をアドレス・グループごとに詳細表示します。なお，［優先セクション］のみ変更可能です。

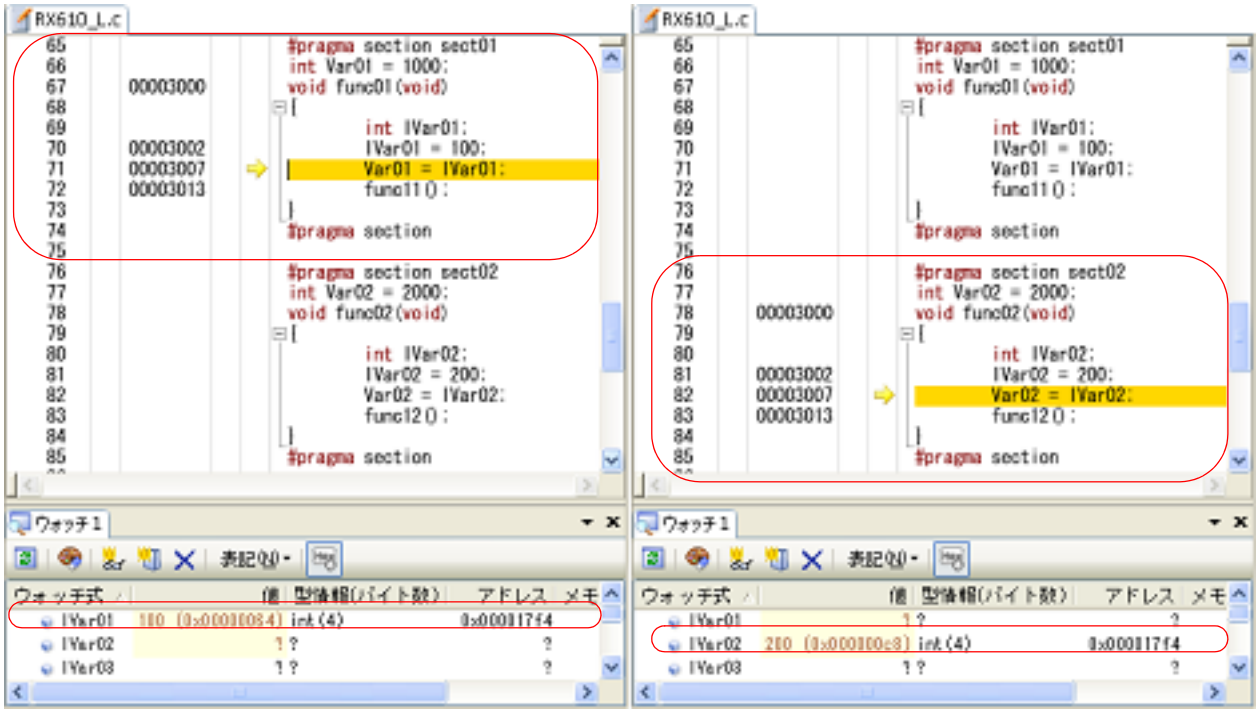
- ［ファイル］
  選択されたアドレス・グループをダウンロードしたロード・モジュール・ファイル名です（変更不可）。
- ［開始アドレス］
  選択されたアドレス・グループの開始アドレスです（変更不可）。
- ［終了アドレス］
  選択されたアドレス・グループの終了アドレスです（変更不可）。
- ［優先セクション］
  選択されたアドレス・グループに定義されたセクションの一覧です。表示された一覧から，デバッグ対象とするセクション（優先セクション）を選択します。

注意 1. オーバーレイ・セクションに関する設定内容は，プロジェクト・ファイルに保存しません。
ロード・モジュールのダウンロード後は，デバッグ対象のセクションを再設定してください。

注意 2. ［優先セクション］項目の変更によって切り替わる情報は，デバッグ情報のみです。
デバッグ・ツールは，対象セクションのデータのコピーは行いません。

次に，［オーバーレイ・セクション］プロパティにおいて，図 2.97 オーバーレイ・セクションの配置に記載している *“ram01”* を優先セクションとして選択した場合（左側），および *“ram02”* へ変更した（右側）場合の例を示します。

図 2.100　［優先セクション］の選択を “*ram01*” → “*ram02*” へ変更



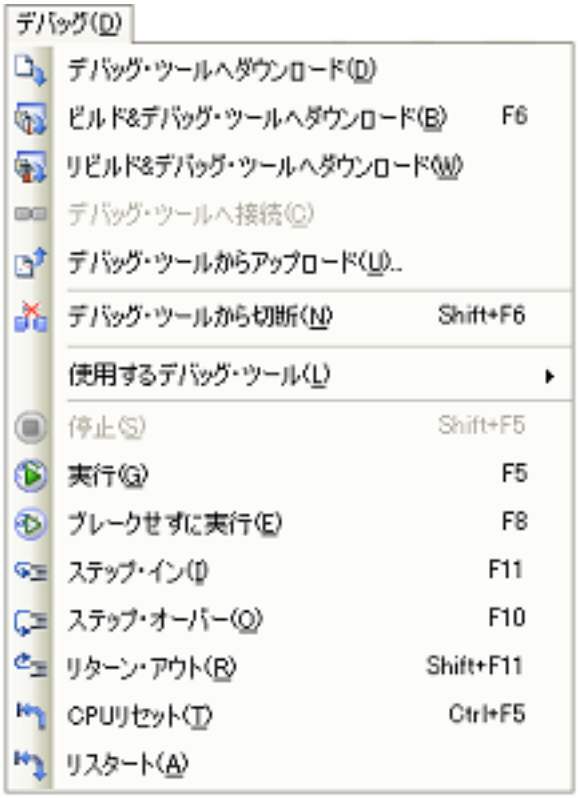
2.9 プログラムの実行

この節では，プログラムの実行方法について説明します。
なお，この節で説明する主な操作は，プログラムの実行を制御するためのコマンドをまとめたメイン・ウインドウ上のデバッグ・ツールバー，または［デバッグ］メニューより行います。

注意　デバッグ・ツールバー，および［デバッグ］メニューの各項目は，デバッグ・ツールと接続時のみ有効となります。

図 2.101　デバッグ・ツールバー

図 2.102　［デバッグ］メニュー



2.9.1 マイクロコントローラ（CPU）をリセットする

デバッグ・ツールバーの ボタンをクリックすることにより，CPU をリセットします。
CPU をリセットすることにより，カレント PC 値をリセット番地に設定します。

備考　CPU リセット後に，I/O レジスタ /CPU レジスタの値を指定した値に自動的に書き換える処理を設定することができます（「2.18 フック処理を設定する」参照）。

2.9.2 プログラムを実行する

プログラムを実行する操作方法には次の種類があります。
デバッグの目的に応じて，操作方法を選択してください。
なお，実行中のプログラムの停止方法については，「2.10 プログラムの停止（ブレーク）」を参照してください。

2.9.2.1 マイクロコントローラ（CPU）をリセットしてから実行する
2.9.2.2 現在のアドレスから実行する
2.9.2.3 PC 値を変更してから実行する

備考 1. プログラムの実行直前に，I/O レジスタ /CPU レジスタ値を指定した値に自動的に書き換える処理を設定することができます（「2.18 フック処理を設定する」参照）。

備考 2. プログラム実行中に，ステータス・バーへ任意の更新間隔で PC 値を表示することができます（「(e) カレント PC 位置」参照）。

2.9.2.1 マイクロコントローラ（CPU）をリセットしてから実行する

CPU をリセットしたのち，リセット番地からプログラムの実行を開始します。
操作は，デバッグ・ツールバーの ボタンをクリックします。
この操作によりプログラムの実行を開始した場合，次のいずれかの状態までその実行を続けます。

- ボタンのクリック（「2.10.1 プログラムの実行を手動で停止する」参照）
- PC がブレークポイントに到達（「2.10.2 任意の場所で停止する（ブレークポイント）」参照）
- ブレーク・イベント条件の成立（「2.10.3 任意の場所で停止する（ブレーク・イベント）【E1】【E20】【EZ Emulator】」，「2.10.4 変数 /I/O レジスタへのアクセスで停止する」参照）
- その他のブレーク要因の発生

備考 この操作は，ボタンをクリックしたのち，ボタンをクリックした場合と同等です。

2.9.2.2 現在のアドレスから実行する

現在のアドレス（カレント PC 値で示されるアドレス）からプログラムの実行を開始する方法には，次の種類があります。

(1) 通常の実行
デバッグ・ツールバーの ボタンをクリックします。
この操作により実行を開始した場合，次のいずれかの状態までその実行を続けます。

- ボタンのクリック（「2.10.1 プログラムの実行を手動で停止する」参照）
- PC がブレークポイントに到達（「2.10.2 任意の場所で停止する（ブレークポイント）」参照）
- ブレーク・イベント条件の成立（「2.10.3 任意の場所で停止する（ブレーク・イベント）【E1】【E20】【EZ Emulator】」，「2.10.4 変数 /I/O レジスタへのアクセスで停止する」参照）
- その他のブレーク要因の発生

(2) ブレーク関連のイベントを無視した実行
デバッグ・ツールバーの ボタンをクリックします。
この操作により実行を開始した場合，次のいずれかの状態までその実行を続けます。

- ボタンのクリック（「2.10.1 プログラムの実行を手動で停止する」参照）
- その他のブレーク要因の発生

備考 この操作により実行を開始した場合，Printf イベントの発生も無視されます。

(3) キャレット位置までの実行
エディタ パネル／逆アセンブル パネルにおいて，プログラムを停止させたい行／命令にキャレットを移動したのち，コンテキスト・メニューの［ここまで実行］を選択します。
この操作により実行を開始した場合，次のいずれかの状態までその実行を続けます。

- PC がキャレット位置のアドレスに到達
- ボタンのクリック（「2.10.1 プログラムの実行を手動で停止する」参照）
- その他のブレーク要因の発生

注意 キャレット位置の行に対応するアドレスが存在しない場合は，下方向の有効な行までプログラムを実行します（有効な行が存在しない場合は，エラーとなります）。

備考 この操作により実行を開始した場合，Printf イベントの発生も無視されます。

2.9.2.3 PC 値を変更してから実行する

カレント PC 値を任意の箇所に強制的に変更したのち，プログラムを実行します。
この操作を行うには，まず，エディタ パネル／逆アセンブル パネルにおいて，プログラムの実行を開始したい行／命令にキャレットを移動したのち，コンテキスト・メニューの［PC をここに設定］を選択します（カレント PC 値が現在キャレットのある行／命令のアドレスに変更されます）。
次に，「2.9.2.2 現在のアドレスから実行する」で示した，いずれかの実行方法を行います。

注意 ソフトウェア・ブレークポイントおよびハードウェア・ブレーク（実行前ブレーク）ポイントで停止後，再度そのアドレスから実行を再開した場合，一度そのアドレスをステップ実行により実行してから実行を継続するので，リアルタイム性はなくなります。なお，このステップ実行で成立したイベントは，その後の実行時にクリアされます。このため，トレースおよびタイマ計測の開始イベントがこのステップ実行で成立した場合，再度イベントが成立するまでトレース取得もタイマ計測も開始しません。また，ブレークおよびトレース開始組み合わせイベントも同様に，ステップ実行で成立したイベントは AND / シーケンシャルの対象になりません。

2.9.3 プログラムをステップ実行する

次のいずれかの操作を行うと，現在のアドレス（カレント PC 値で示されるアドレス）から，ソース・レベル単位（ソース・テキスト 1 行分），または命令レベル単位（1 命令分）でプログラムをステップ実行したのち，自動的に停止します。
プログラムの停止後は逐一各パネルの内容が自動的に更新されるため，ステップ実行は，プログラムの実行遷移をソース・レベル単位／命令単位でデバッグする場合に有効な実行方法です。
なお，ステップ実行を行う際の実行単位は，次に示すようにエディタ パネルにおける設定に依存します。

- ツールバーの ボタンを無効にしている場合（デフォルト）
  ソース・レベル単位によるステップ実行を行います。
  ただし，逆アセンブル パネルにフォーカスがある場合，またはカレント PC 値で示されるアドレスに行情報が存在しない場合は，命令レベル単位によるステップ実行を行います。
- ツールバーの ボタンを有効にしている場合
  命令レベル単位によるステップ実行を行います。

備考 ボタンは，エディタ パネルを混合表示モードに設定している場合のみ有効となります。

ステップ実行には，次の種類があります。

2.9.3.1 関数内にステップ・インする（ステップ・イン実行）
2.9.3.2 関数をステップ・オーバーする（ステップ・オーバー実行）
2.9.3.3 関数内でリターンが完了するまで実行する（リターン・アウト実行）

注意 1. ステップ実行中は，設定されているブレークポイント／ブレーク・イベント／ Printf イベントを発生しません。

注意 2. 【シミュレータ】
ステップ実行中に割り込みハンドラにジャンプすることがあります。

2.9.3.1 関数内にステップ・インする（ステップ・イン実行）

関数呼び出しの場合，呼び出された関数内の先頭で停止するステップ実行です。
操作は，デバッグ・ツールバーの ボタンをクリックします。

注意 1. デバッグ情報がない関数へのステップ・イン実行はできません。

注意 2. longjmp 関数へのステップ・イン実行は，実行処理が完了せずタイムアウト待ちになることがあります。

注意 3. 【E1】【E20】【EZ Emulator】
ステップ実行中は，割り込みが禁止されます。また，JTAG 通信において，ステップ実行中はリセットが禁止されます。

2.9.3.2 関数をステップ・オーバーする（ステップ・オーバー実行）

サブルーチン分岐命令による関数呼び出しの場合，その関数内のソース行／命令すべてを 1 ステップとみなして実行し，関数から戻った箇所で停止するステップ実行です（サブルーチン分岐命令を実行したときと同じネストになるまで，ステップ実行します）。
操作は，デバッグ・ツールバーの ボタンをクリックします。
なお，サブルーチン分岐命令以外の場合は， ボタンのクリックと同じ動作となります。

注意 1. longjmp 関数のステップ・オーバー実行は，実行処理が完了せずタイムアウト待ちになることがあります。

注意 2. 【E1】【E20】【EZ Emulator】
関数やサブルーチンの呼び出し箇所をステップ・オーバーする際は割り込み，およびリセットが許可となります。

2.9.3.3 関数内でリターンが完了するまで実行する（リターン・アウト実行）

現在の関数から，呼び出し元関数に戻った箇所で停止するステップ実行します。
ある関数内において確認が必要なソース行／命令の実行が終了した際などに，この命令によるステップ実行を行うと，残りの関数内の命令をステップ実行せずに呼び出し元の関数に戻ることができます。
操作は，デバッグ・ツールバーの ボタンをクリックします。

注意 1. main 関数内でのリターン・アウト実行は，スタート・アップ・ルーチン内でブレークします。

注意 2. longjmp 関数の呼び出し元関数内でリターン・アウト実行すると，ブレークしないことがあります。

注意 3. 再帰関数からリターン・アウト実行を行うと，フリーラン状態となります。

注意 4. サブルーチン復帰命令以外の命令（ジャンプ命令）で呼び出し元関数に戻る関数については，正しくリターン・アウト実行できません。

注意 5. 【E1】【E20】【EZ Emulator】
関数内やサブルーチン内でリターン・アウトする際は割り込み，およびリセットが許可となります。

2.9.4 指定ルーチンを実行する【E1】【E20】【EZ Emulator】

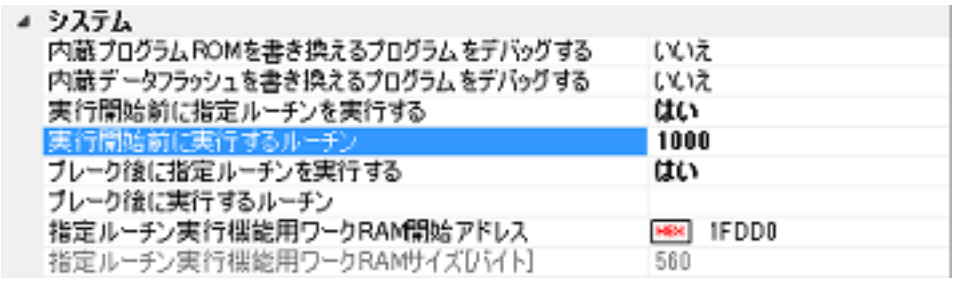
プログラムの実行開始直前および停止直後にプログラムの指定ルーチンを実行することができます。
プログラムの実行や停止に同期してターゲット・システムの制御を行いたい場合に，任意ルーチンを設定します。
なお，実行中のプログラムの停止方法については，「2.10 プログラムの停止（ブレーク）」を参照してください。
ここでは，実行するルーチンを指定する方法について説明します。
指定ルーチンは，プロパティ パネルの［デバッグ・ツール設定］タブにおける［システム］【E1】【E20】【EZ Emulator】カテゴリ内のプロパティにて指定します。

- プログラムの実行開始直前に指定する場合
  [実行開始前に指定ルーチンを実行する] プロパティを [はい] を選択して，この操作で表示される [実行開始前に実行するルーチン] プロパティにおいて実行開始アドレスを入力してください。
  指定されたルーチンがプログラム実行直前に実行されます。

- プログラムの停止直後に指定する場合
  [ブレーク後に指定ルーチンを実行する] プロパティを [はい] を選択して，この操作で表示される [ブレーク後に実行するルーチン] プロパティにおいて実行開始アドレスを入力してください。
  指定されたルーチンがプログラム停止直後に実行されます。

なお，[実行開始前に指定ルーチンを実行する] または [ブレーク後に指定ルーチンを実行する] プロパティのいずれかで [はい] を選択すると，[指定ルーチン実行機能用ワーク RAM 開始アドレス] プロパティが表示されます。指定したワーク RAM の配置アドレスから [指定ルーチン実行機能用ワーク RAM サイズ [バイト]] プロパティの表示サイズ分を，指定ルーチン実行機能用ワーク RAM 領域としてデバッガのファームウェアが使用します。
指定ルーチン実行機能を使用する場合は，このワーク RAM 領域を指定してください。

図 2.103　[システム] カテゴリ



注意 1.　[実行開始前に実行するルーチン]，[ブレーク後に実行するルーチン] および [指定ルーチン実行機能用ワーク RAM 開始アドレス] プロパティのプロパティ値には，C 言語の場合は関数名，アセンブリ言語の場合はラベル名も指定することができます。

注意 2.　指定ルーチン内でスタックを使用する場合，割り込みスタックを使用してください。

注意 3.　指定ルーチン処理の終了は，リターンサブルーチン命令 (RTS) を記述してください。

注意 4.　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，デバッグ・ツールの制御ができない可能性があります。

注意 5.　指定ルーチン実行開始時のレジスタ値は不定です。そのため，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。

注意 6.　指定ルーチンの実行直前および停止直後は，デバッグ・ツールの制御で停止状態となります。なお，停止時間は次のようになります。

- ユーザ・プログラム実行開始前の指定ルーチン実行後，約 100 サイクル（CPU クロック：100MHz の場合，約 1μs）でユーザ・プログラムを実行します。
- ユーザ・プログラム停止後から，約 100 サイクル（CPU クロック：100MHz の場合，約 1μs）で指定ルーチンを実行します。
  ただし，内蔵 ROM 無効拡張モード時は，ユーザ・プログラム停止後から，CPU クロック：100MHz にて約 3ms，CPU クロック：32kHz にて約 8ms で指定ルーチンを実行します。また，ホットプラグイン接続した場合の，最初のユーザ・プログラム停止時も，同様です。

注意 7.　指定ルーチンは開始時にスーパーバイザモードで実行しますので，ユーザモードに切り替えないでください。

注意 8.　指定ルーチンを実行する場合，指定ルーチンのプログラム領域へのメモリアクセス／ダウンロード／ブレークポイント設定を行わないでください。

注意 9.　指定ルーチン実行機能使用時は，[指定ルーチン実行機能用ワーク RAM 開始アドレス] から 230h（560）バイト分の RAM 領域を，エミュレータ専用に使用します。
この領域を，ユーザ・プログラムおよび指定ルーチン内で使用しないでください。
この領域は，内蔵 RAM 領域に配置してください。
なお，[指定ルーチン実行機能用ワーク RAM 開始アドレス] は，4 で割り切れる値にしてください。

注意 10.　指定ルーチンで使用する汎用レジスタおよびフラグについては，次の制限があります。

ISP レジスタ	指定ルーチン終了時に，指定ルーチン実行開始時の値に戻してください。
U フラグ	指定ルーチンを実行中はユーザスタックポインタ（USP）の指定は禁止となります。
I フラグ	指定ルーチンを実行中は割り込み禁止となります。
PM フラグ	指定ルーチンを実行中はユーザモードへの切り替え禁止となります。

注意 11.　指定ルーチン実行時は，トレース／ブレーク／リアルタイム RAM モニタ／タイマ計測は無効となります。

注意 12.　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します。

注意 13.　指定ルーチン実行後にプログラムを開始する場合，マイクロコントローラの状態は次のようになります。

汎用レジスタ	プログラム停止時の状態またはユーザが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クロコントローラ周辺機能動作が継続します。

注意 14. 指定ルーチン内でスタックを使用する場合は，指定ルーチン内で ISP を設定してください。
なお，指定ルーチン開始時の ISP 値は，[指定ルーチン実行機能用ワーク RAM 開始アドレス] に設定されます。

注意 15. ISP について，指定ルーチン終了時に，ISP 値を指定ルーチン実行開始時の値に戻してください。

注意 16. 指定ルーチン実行機能を使用時は，端子リセットやウォッチドッグリセットをしないでください。

注意 17. 指定ルーチン実行機能を使用時は，内蔵 RAM 領域を無効に設定しないでください。

注意 18. 指定ルーチンを実行中は，U フラグに “1（ユーザスタックポインタ（USP）を指定）” を設定しないでください。

注意 19. 指定ルーチン実行中は，RTFI（高速割り込みからの復帰）命令を使用しないでください。

注意 20. 指定ルーチン内で以下が発生しないようにしてください。
[ブレークポイント，ブレークイベント，トレース開始イベント／トレース終了イベント]
例えば，実行開始前に実行するルーチン内でブレークイベントが発生すると，プログラム実行開始直後にプログラムが停止します。また，指定ルーチン内でトレース開始イベントが発生すると，指定ルーチン実行中のトレースは取得されませんが，その後のプログラム実行開始からトレースが取得されることになり，期待するエミュレータ動作にならない場合があります。

注意 21. 指定ルーチン実行機能にて Start ファンクションからユーザプログラムへ移行しない場合，および Stop ファンクションが停止しない場合は，“START/STOP 時の指定ルーチン実行処理でタイムアウトエラーが発生しました。システムリセットを発行しました。” とエラーメッセージが表示され，エミュレータを初期化してユーザ・プログラムが停止します。システムリセット発行後は，トレース記録も初期化します。なお，デバッグを継続して行うことが可能です。

2.10 プログラムの停止（ブレーク）

この節では，実行中のプログラムを停止する方法について説明します。

備考　実行中のプログラムが停止すると，その原因（ブレーク要因）がメイン・ウインドウのステータス・バーに表示されます。

2.10.1 プログラムの実行を手動で停止する

デバッグ・ツールバーの ボタンをクリックすることにより，現在実行中のプログラムを強制的に停止します。

2.10.2 任意の場所で停止する（ブレークポイント）

ブレークポイントを設定することにより，任意の箇所でプログラムの実行を容易に停止させることができます。
ブレークポイントは，マウスのワン・クリックで設定することができます。
ブレークポイントを設定するためには，あらかじめ使用するブレークポイントの種別，および動作の設定を行う必要があります。
ここでは，次の操作方法について説明します。

2.10.2.1 使用するブレークポイントの種別／動作を設定する【E1】【E20】【EZ Emulator】
2.10.2.2 ブレークポイントを設定する
2.10.2.3 ハードウエア・ブレークポイントを編集する
2.10.2.4 ブレークポイントを削除する

注意　【E1/E20/EZ Emulator【RX630, RX631, RX63N, RX63T, RX210, RX110, RX111, RX113 グループ】】
WAIT 命令またはその 1 命令前に設定した実行系イベントでブレークが発生した場合，エラーが発生する可能性があります。ターゲットプログラムを WAIT 命令またはその 1 命令前でブレークさせる場合は，ブレークポイントを使用してください。

2.10.2.1 使用するブレークポイントの種別／動作を設定する【E1】【E20】【EZ Emulator】

使用するブレークポイントの種別／動作の設定は，プロパティ パネルの［デバッグ・ツール設定］タブ上の［ブレーク］【E1】【E20】【EZ Emulator】カテゴリ内で行います。

図 2.104　［ブレーク］カテゴリ【E1】【E20】【EZ Emulator】

［優先的に使用するブレークポイントの種類］プロパティで，マウスのワンクリック操作で設定するブレークポイントの種別を指定します。
ただし，指定した種類のブレークポイントの設定数が制限を越える場合（「2.17.7.1 有効イベント数の制限」参照），もう一方の種類のブレークポイントが使用されます。
ブレークポイントの用途に合わせて，次のドロップダウン・リストから選択します。

ソフトウエア・ブレーク	指定したアドレスの命令コードを一時的にブレーク用の命令に書き換え，その命令を実行した際にプログラムを停止させます。 設定すると，ソフトウエア・ブレーク・イベントとして扱われます。
ハードウエア・ブレーク	デバッグ・ツールが，プログラム実行中にブレーク条件を逐次確認し，条件を満たした際にプログラムを停止させます（デフォルト）注。 設定すると，ハードウエア・ブレーク・イベントとして扱われます。

注 1.　ハードウエア・ブレークは，指定したアドレスの命令実行前にブレークする“実行前ブレーク”であり，デバッグ・ツールの資源を用いて実現している機能です。

注 2.　外部アドレス空間へのソフトウエア・ブレーク・イベントは設定できません。

2.10.2.2 ブレークポイントを設定する

操作は，ソース・テキスト／逆アセンブル・テキストを表示しているエディタ パネル／逆アセンブル パネルで行います。

各パネルのメイン・エリア（エディタ パネル）／イベント・エリア（逆アセンブル パネル）において，ブレークポイントを設定したい箇所をクリックしてください。

ブレークポイントは，クリックした行位置に対応する先頭アドレスの命令に対して設定されます。

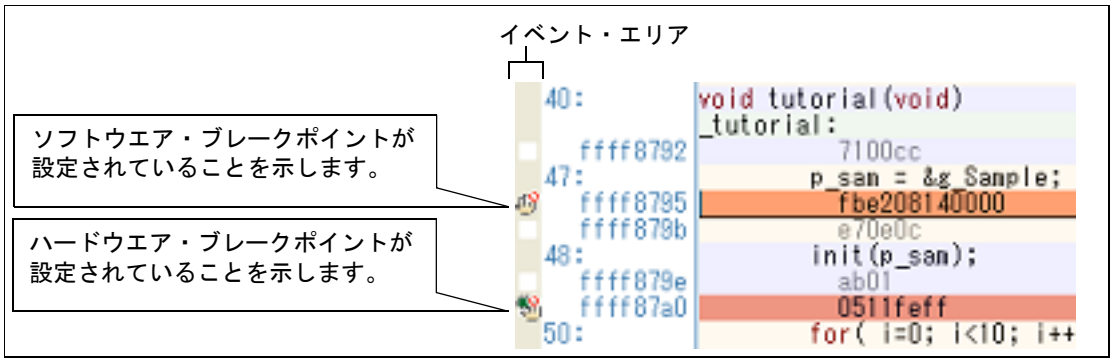
ブレークポイントが設定されると，設定した箇所に次のイベント・マークが表示され，ソース・テキスト行／逆アセンブル・テキスト行が強調表示されます。

また，対象アドレスにブレーク・イベント（ソフトウエア・ブレーク・イベント／ハードウエア・ブレーク・イベント）が設定されたとみなされ，イベント パネルで管理されます（「2.17 イベントの管理」参照）。

表 2.4 ブレークポイントのイベント・マーク

種別	イベント・マーク
ソフトウエア・ブレーク （【E1】【E20】【EZ Emulator】）	
ハードウエア・ブレーク	

図 2.105 ブレークポイントの設定例（逆アセンブル パネルの場合）



注意 1. ブレークポイントはブレーク・イベントとして設定され，イベントとして管理されるため，設定数に制限があります。ブレークポイントの設定に関しては（有効イベント数の制限など），「2.17.7 イベント設定に関する留意事項」も参照してください。

注意 2. ブレークポイントは，アドレス表示がない行に設定することはできません。

注意 3. 【E1】【E20】【EZ Emulator】
ソフトウエア・ブレークポイントは，命令を置き換えることにより実現します。そのため，内蔵 ROM と内蔵 RAM 以外の領域には指定することができません。

備考 1. イベントの設定状態によりイベント・マークは異なります（「2.17.1 設定状態（有効／無効）を変更する」参照）。
また，すでにイベントが設定されている箇所で，新たにイベントを設定した場合は，複数のイベントが設定されていることを示すイベント・マーク（ ）が表示されます。

備考 2. 【シミュレータ】
設定できるブレークポイントは，ハードウエア・ブレークポイント固定です。

備考 3. 【E1】【E20】【EZ Emulator】
次に示す操作により，「2.10.2.1 使用するブレークポイントの種別／動作を設定する【E1】【E20】【EZ Emulator】」の指定に依存することなく，ハードウエア・ブレークポイント／ソフトウエア・ブレークポイントを設定することができます。

種別	操作方法 1	操作方法 2
ハードウエア・ブレークポイント	［Ctrl］キー + クリック	コンテキスト・メニューの［ブレークの設定］→［ハード・ブレークを設定］を選択
ソフトウエア・ブレークポイント	［Shift］キー + クリック	コンテキスト・メニューの［ブレークの設定］→［ソフト・ブレークを設定］を選択

注意 “操作方法 1” は，逆アセンブル パネルでのみ有効です。

2.10.2.3 ハードウエア・ブレークポイントを編集する

設定したハードウエア・ブレークポイントを編集するには，［表示］メニュー→［イベント］の選択でオープンするイベント パネル上において，編集したいハードウエア・ブレークポイントを選択したのち，コンテキスト・メニューの「条件の編集 ...」をクリックしてください。
この操作によりオープンするダイアログで編集します。
ダイアログによる編集の詳細については，「2.17.4.1 実行系イベントを編集する」を参照してください。

2.10.2.4 ブレークポイントを削除する

設定したブレークポイントを削除するには，エディタ パネル／逆アセンブル パネル上において，表示されているイベント・マークを再度クリックします（イベント・マークが消失します）。
また，［表示］メニュー→［イベント］の選択でオープンする イベント パネル上において，削除したいブレークポイントを選択したのち，ツールバーの ボタンをクリックする操作でも削除することができます。（「2.17.5 イベントを削除する」参照）。

2.10.3 任意の場所で停止する（ブレーク・イベント）【E1】【E20】【EZ Emulator】

ブレーク・イベント（実行系）を設定することにより，任意の箇所でプログラムの実行を停止させることができます。ブレーク・イベント（実行系）は，指定したアドレスの命令実行後にブレークする“実行後ブレーク”であり，デバッグ・ツールの資源を用いて実現している機能です。
ここでは，次の操作方法について説明します。

2.10.3.1 ブレーク・イベント（実行系）を設定する
2.10.3.2 ブレーク・イベント（実行系）を編集する
2.10.3.3 ブレーク・イベント（実行系）を削除する

2.10.3.1 ブレーク・イベント（実行系）を設定する

操作は，ソース・テキスト／逆アセンブル・テキストを表示しているエディタ パネル／逆アセンブル パネルで行います。
設定したい行にキャレットを移動したのち，コンテキスト・メニューの「ブレークの設定」→「組み合わせブレークを設定」を選択します。
ブレーク・イベント（実行系）は，選択した行位置に対応する先頭アドレスの命令に対して設定されます。
ブレーク・イベント（実行系）が設定されると，設定した箇所にハードウエア・ブレークポイントと同じイベント・マークが表示され，逆アセンブル・テキスト行が強調表示されます（「2.10.2.2 ブレークポイントを設定する」参照）。
また，対象アドレスにブレーク・イベント（実行系）が設定されたとみなされ，イベント パネルで管理されます（「2.17 イベントの管理」参照）。イベント パネルにおいて，組み合わせブレークの詳細情報内に実行後として登録されます。

注意　ブレーク・イベント（実行系）の設定に関しては（有効イベント数の制限など），「2.17.7 イベント設定に関する留意事項」も参照してください。

図 2.106 逆アセンブル・テキスト上の行に対するブレーク・イベント（実行系）の設定例【E1】【E20】【EZ Emulator】

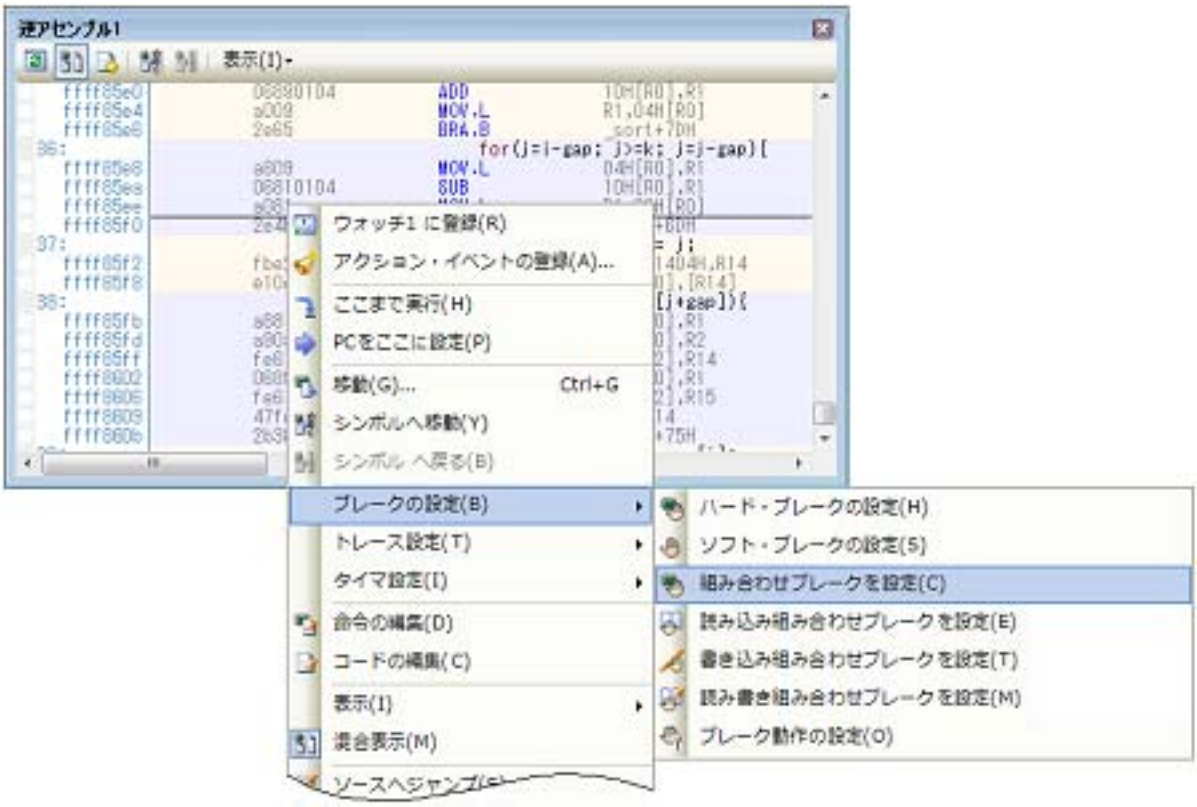
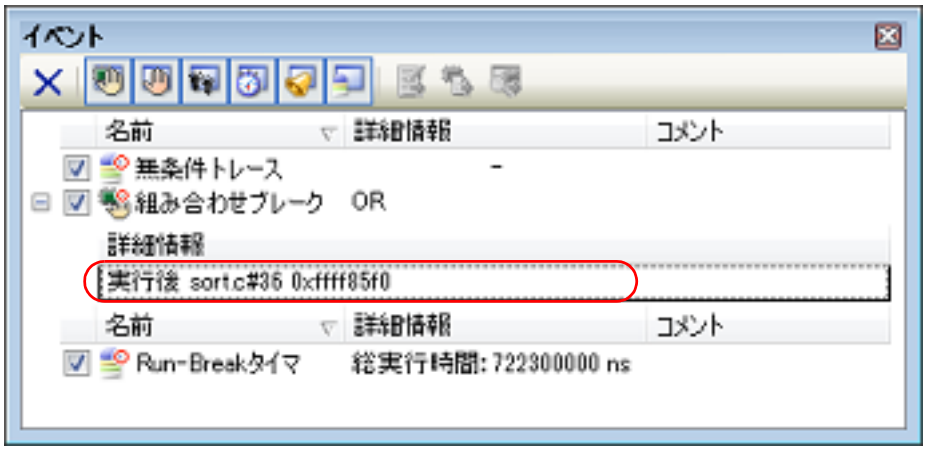


図 2.107 イベントパネル上におけるブレーク・イベント（実行系）の設定例【E1】【E20】【EZ Emulator】



2.10.3.2 ブレーク・イベント（実行系）を編集する

設定したブレーク・イベント（実行系）を編集するには，［表示］メニュー→［イベント］の選択でオープンするイベント パネル上において，組み合わせブレークの詳細情報に表示されたブレーク・イベント（実行系）を選択したのち，コンテキスト・メニューの「条件の編集 ...」をクリックしてください。

この操作によりオープンするダイアログで編集します。

ダイアログによる編集の詳細については，「2.17.4.1 実行系イベントを編集する」を参照してください。

2.10.3.3 ブレーク・イベント（実行系）を削除する

設定したブレーク・イベント（実行系）を削除するには，［表示］メニュー→［イベント］の選択でオープンする イベント パネル上において，組み合わせブレークの詳細情報に表示されたブレーク・イベント（実行系）を選択したのち，ツールバーの ボタンをクリックする操作で削除することができます。（「2.17.5 イベントを削除する」参照）。

また，ソース・テキスト／逆アセンブラ・テキスト上において，表示されているイベント・マークをクリックすることでブレーク・イベント（実行系）を削除することもできます。

2.10.4 変数 /I/O レジスタへのアクセスで停止する

ブレーク・イベント（アクセス系）を設定することにより，任意の変数，または I/O レジスタに対し，指定したアクセスがあった場合にプログラムの実行を停止させることができます。
また，この際に，アクセスした値を限定することもできます。
アクセス系のブレーク・イベントで指定できるアクセス種別は次のとおりです。

表 2.5　　変数へのアクセス種別

アクセス種別	説明
リード	指定した変数 /I/O レジスタに，リード・アクセスした（読み込みを行った）際に実行中のプログラムを停止します。
ライト	指定した変数 /I/O レジスタに，ライト・アクセスした（書き込みを行った）際に実行中のプログラムを停止します。
リード／ライト	指定した変数 /I/O レジスタに，リード・アクセス，またはライト・アクセスした（読み込み，または書き込みを行った）際に実行中のプログラムを停止します。

注意 1.　DMAC（Direct Memory Access Controller）／ DTC（Data Transfer Controller）によるアクセスは対象となりません。

注意 2.　【シミュレータ】
ストリング操作命令，および積和演算命令は，最初と最後のデータアクセスがイベントチェック対象となります。

ここでは，次の操作方法について説明します。

2.10.4.1 変数 /I/O レジスタへのブレーク・イベント（アクセス系）を設定する
2.10.4.2 変数 /I/O レジスタへのブレーク・イベント（アクセス系）を編集する
2.10.4.3 変数 /I/O レジスタへのブレーク・イベント（アクセス系）を削除する

2.10.4.1 変数 /I/O レジスタへのブレーク・イベント（アクセス系）を設定する

変数，または I/O レジスタへのアクセスで，プログラムの実行を停止させるブレーク・イベント（アクセス系）の設定は，次のいずれかの操作により行います。

注意 1.　ブレーク・イベント（アクセス系）の設定に関しては（有効イベント数の制限など），「2.17.7 イベント設定に関する留意事項」も参照してください。

注意 2.　【E1】【E20】【EZ Emulator】
ブレーク・イベント（アクセス系）は，イベント パネルにおいて組み合わせブレークの詳細情報内にリード , ライト , リード / ライトとして登録されます。

(1)　ソース・テキスト／逆アセンブル・テキスト上の変数 /I/O レジスタにブレーク・イベント（アクセス系）を設定する場合
操作は，ソース・テキスト／逆アセンブル・テキストを表示しているエディタ パネル／逆アセンブル パネル上で行います。
ソース・テキスト／逆アセンブル・テキスト上の任意の変数，または I/O レジスタを選択したのち，コンテキスト・メニューより次の操作を行います。ただし，対象となる変数は，グローバル変数／関数内スタティック変数／ファイル内スタティック変数のみとなります。
なお，この操作を行うことにより，対象変数，または I/O レジスタにブレーク・イベント（アクセス系）が設定されたとみなされ，イベント パネルで管理されます（「2.17 イベントの管理」参照）。

アクセス種別	操作方法
リード	-【E1】【E20】【EZ Emulator】 ［ブレークの設定］→［読み込み組み合わせブレークを設定］を選択したのち，［Enter］キーを押下します。 -【シミュレータ】 ［ブレークの設定］→［読み込みブレークを設定］を選択したのち，［Enter］キーを押下します。 この際に，メニュー内のテキスト・ボックスに値を指定した場合，指定した値で読み込みを行った場合のみブレークします。値を指定しない場合は，値にかかわらず，選択している変数に読み込みを行った場合にブレークします。
ライト	-【E1】【E20】【EZ Emulator】 ［ブレークの設定］→［書き込み組み合わせブレークを設定］を選択したのち，［Enter］キーを押下します。 -【シミュレータ】 ［ブレークの設定］→［書き込みブレークを設定］を選択したのち，［Enter］キーを押下します。 この際に，メニュー内のテキスト・ボックスに値を指定した場合，指定した値で書き込みを行った場合のみブレークします。値を指定しない場合は，値にかかわらず，選択している変数に書き込みを行った場合にブレークします。
リード／ライト	-【E1】【E20】【EZ Emulator】 ［ブレークの設定］→［読み書き組み合わせブレークを設定］を選択したのち，［Enter］キーを押下します。 -【シミュレータ】 ［ブレークの設定］→［読み書きブレークを設定］を選択したのち，［Enter］キーを押下します。 この際に，メニュー内のテキスト・ボックスに値を指定した場合，指定した値で読み込み，または書き込みを行った場合のみブレークします。 値を指定しない場合は，値にかかわらず，選択している変数に読み込み，または書き込みを行った場合にブレークします。

注意 1.　カレント・スコープ内の変数が対象となります。

注意 2.　ブレーク・イベントは，アドレス表示がない行上の変数 I/O レジスタを選択しても設定することはできません。

図 2.108　ソース・テキスト上の変数に対するブレーク・イベント（アクセス系）の設定例【E1】【E20】【EZ Emulator】

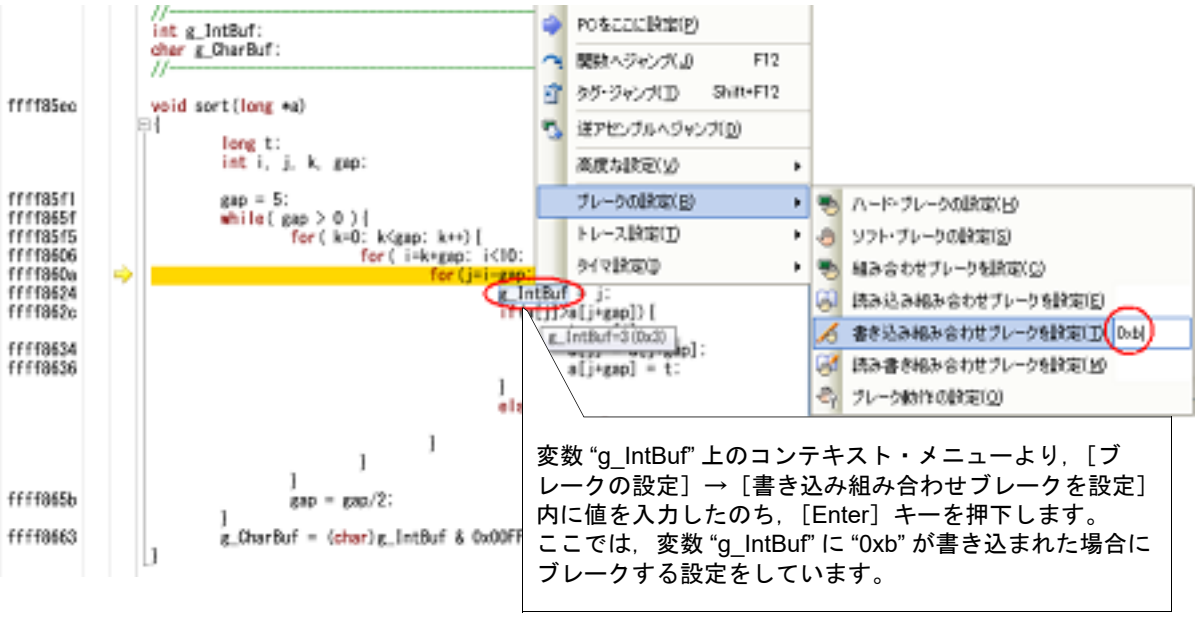


図 2.109 イベントパネル上におけるブレーク・イベント（アクセス系）の設定例【E1】【E20】【EZ Emulator】

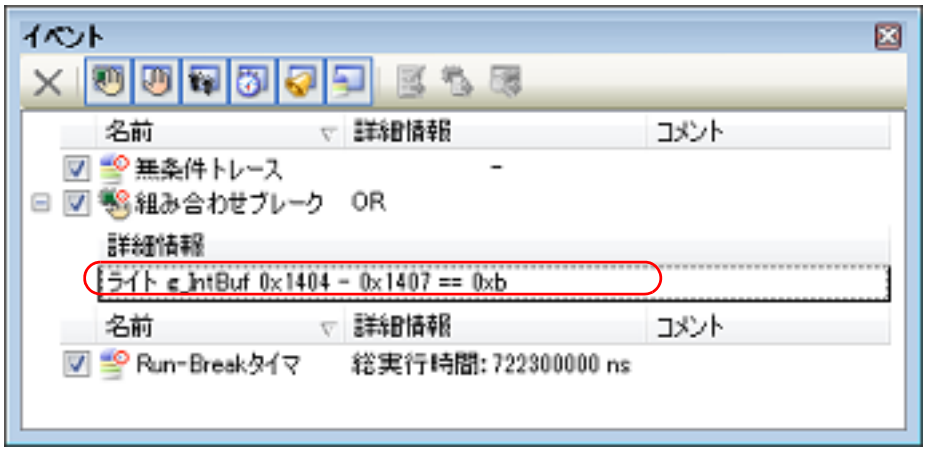


図 2.110 ソース・テキスト上の変数に対するブレーク・イベント（アクセス系）の設定例【シミュレータ】

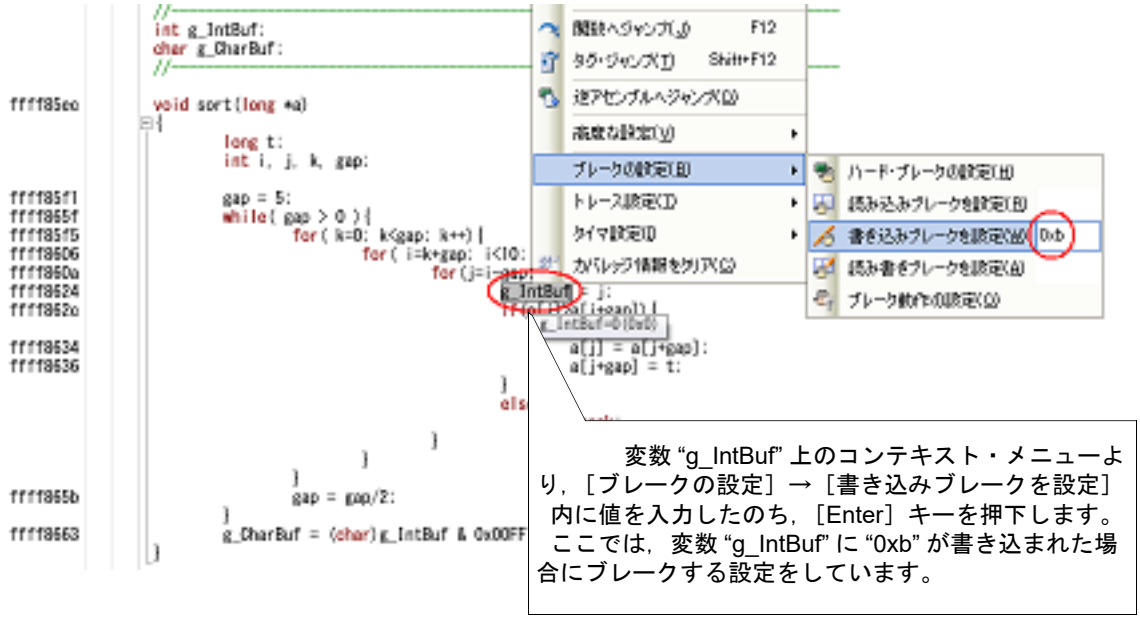
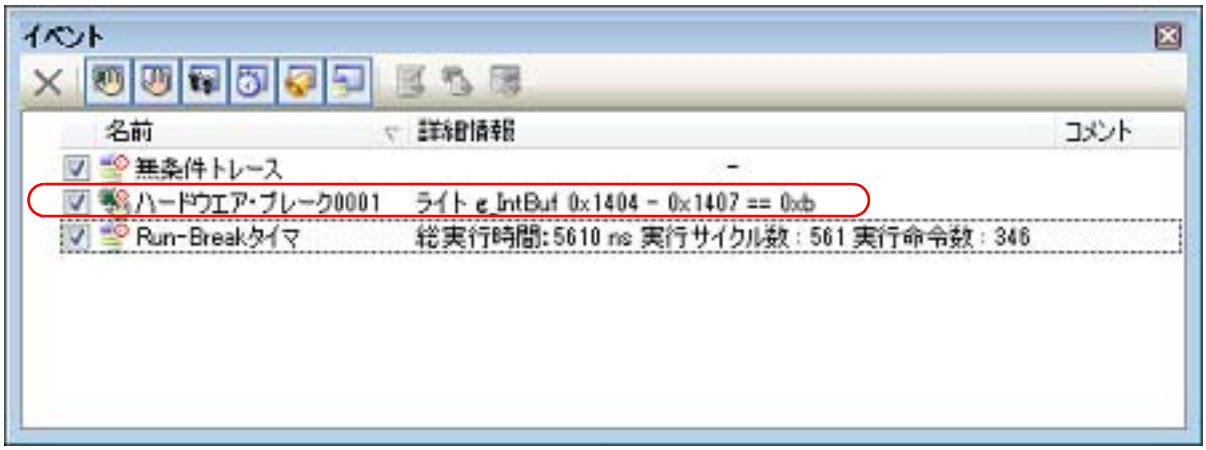


図 2.111 イベントパネル上におけるブレーク・イベント（アクセス系）の設定例【シミュレータ】



(2) 登録したウォッチ式にブレーク・イベント（アクセス系）を設定する場合
操作は，ウォッチ パネル上で行います。
対象となるウォッチ式を選択したのち（複数選択不可），コンテキスト・メニューより次の操作を行います。

ただし，対象となるウォッチ式は，グローバル変数／関数内スタティック変数／ファイル内スタティック変数 / I/O レジスタのみとなります。
なお，この操作を行うことにより，対象ウォッチ式にブレーク・イベント（アクセス系）が設定されたとみなされ，イベント パネルで管理されます（「2.17 イベントの管理」参照）。

アクセス種別	操作方法
リード	-【E1】【E20】【EZ Emulator】 ［アクセス・ブレークの設定］→［読み込み組み合わせブレークを設定］を選択したのち，［Enter］キーを押下します。 -【シミュレータ】 ［アクセス・ブレークの設定］→［読み込みブレークを設定］を選択したのち，［Enter］キーを押下します。 この際に，メニュー内のテキスト・ボックスに値を指定した場合，指定された値で読み込みを行った場合のみブレークします。値を指定しない場合は，値に関わらず，選択しているウォッチ式に読み込みを行った場合にブレークします。
ライト	-【E1】【E20】【EZ Emulator】 ［アクセス・ブレークの設定］→［書き込み組み合わせブレークを設定］を選択したのち，［Enter］キーを押下します。 -【シミュレータ】 ［アクセス・ブレークの設定］→［書き込みブレークを設定］を選択したのち，［Enter］キーを押下します。 この際に，メニュー内のテキスト・ボックスに値を指定した場合，指定された値で書き込みを行った場合のみブレークします。値を指定しない場合は，値に関わらず，選択しているウォッチ式に書き込みを行った場合にブレークします。
リード／ライト	-【E1】【E20】【EZ Emulator】 ［アクセス・ブレークの設定］→［読み書き組み合わせブレークを設定］を選択したのち，［Enter］キーを押下します。 -【シミュレータ】 ［アクセス・ブレークの設定］→［読み書きブレークを設定］を選択したのち，［Enter］キーを押下します。 この際に，メニュー内のテキスト・ボックスに値を指定した場合，指定された値で読み込み，または書き込みを行った場合のみブレークします。値を指定しない場合は，値に関わらず，選択しているウォッチ式に読み書きを行った場合にブレークします。

備考　　カレント・スコープ内のウォッチ式が対象となります。
カレント・スコープ外のウォッチ式を対象とする場合は，スコープ指定したウォッチ式を選択してください。

図 2.112　ウォッチ式に対するブレーク・イベント（アクセス系）の設定例【E1】【E20】【EZ Emulator】

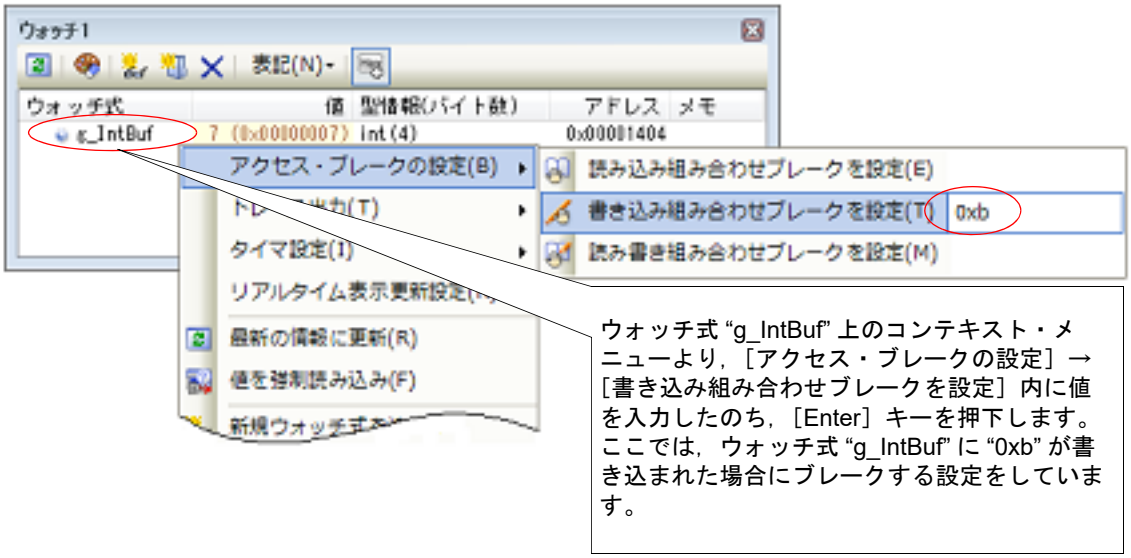
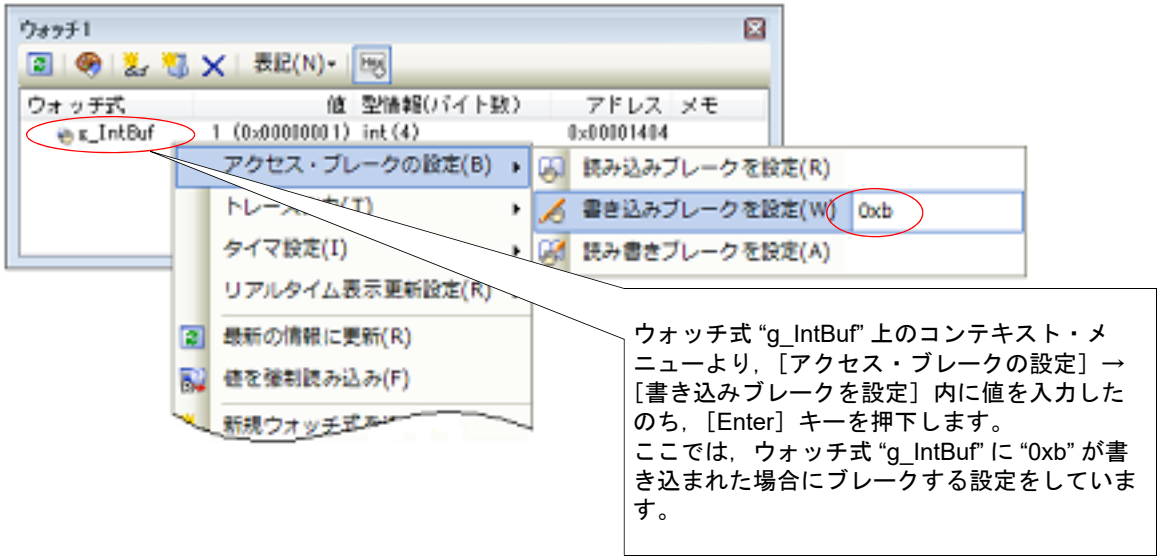


図 2.113 ウォッチ式に対するブレーク・イベント（アクセス系）の設定例【シミュレータ】



2.10.4.2 変数 /I/O レジスタへのブレーク・イベント（アクセス系）を編集する

設定したブレーク・イベント（アクセス系）を編集する場合は，[表示］メニュー→［イベント］の選択でオープンするイベント パネルにおいて，編集したいブレーク・イベント（アクセス系）を選択したのち，コンテキスト・メニューの「条件の編集 ...」をクリックしてください。
ここでオープンするダイアログで編集します。
ダイアログによる編集の詳細については，「2.17.4.2 アクセス系イベントを編集する」を参照してください。

2.10.4.3 変数 /I/O レジスタへのブレーク・イベント（アクセス系）を削除する

設定したブレーク・イベント（アクセス系）を削除する場合は，［表示］メニュー→［イベント］の選択でオープンするイベント パネルにおいて次の操作を行います。なお，使用するデバッグ・ツールにより，イベント名は異なります。

(1) E1/E20/EZ Emulator の場合
削除したい組み合わせブレークの詳細情報に表示されたブレーク・イベント（アクセス系）を選択したのち，同パネルのツールバーの ボタンをクリックします（「2.17.5 イベントを削除する」参照）。
また，逆アセンブラ・テキスト上において，表示されているイベント・マークをクリックすることでブレーク・イベント（アクセス系）を削除することもできます。

(2) シミュレータの場合
削除したいアクセス系のハードウエア・ブレークを選択したのち，同パネルのツールバーの ボタンをクリックします（「2.17.5 イベントを削除する」参照）。

2.10.5 複数のブレーク・イベントを組み合わせる（組み合わせブレーク）【E1】【E20】【EZ Emulator】

2 つ以上のブレーク・イベント（実行系およびアクセス系）を組み合わせて設定することで，設定されたブレーク・イベントが組み合わせ条件を満たした時点でプログラムの実行を停止させることができます。
設定可能な組み合わせブレークは 1 つのみです。そのため，2 つ以上のブレーク・イベントを設定した場合，イベント パネル上では 1 つの組み合わせブレークの詳細情報内にブレーク・イベントが順に追加されます。
組み合わせブレークでは，次の組み合わせ条件が指定できます。

表 2.6 組み合わせブレークに対する組み合わせ条件

組み合わせ条件	説明
OR	設定されたブレーク・イベントの内，いずれかひとつでも条件が成立した時点でブレークします。

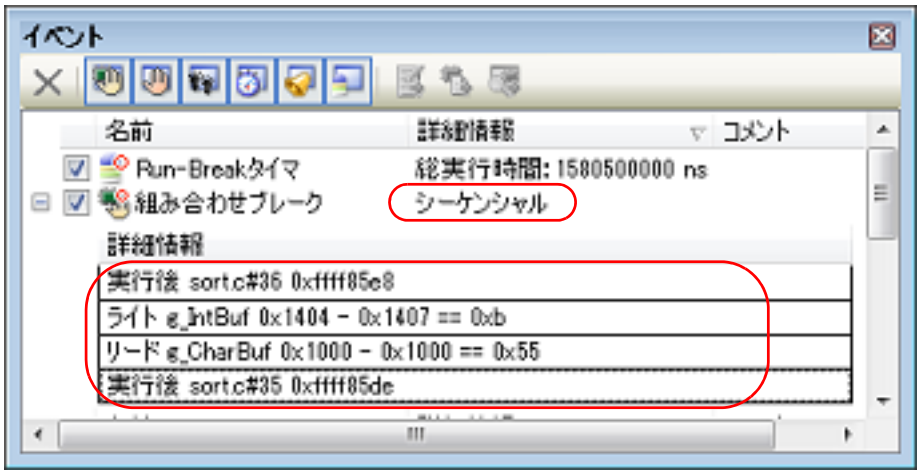
組み合わせ条件	説明
AND	時間軸に関係なく，設定された全てのブレーク・イベントの条件が成立した時点でブレークします。
シーケンシャル	指定した順序で設定されたブレーク・イベントの条件が成立するとブレークします。 また，組み合わせブレーク内のブレーク・イベントを 1 点のみリセットイベント (R イベント) として登録することができます。登録されたブレーク・イベントが成立した場合，その時点まで成立していた他のブレーク・イベントの条件成立は全てクリアされます。

なお，組み合わせブレークを編集する場合，イベント パネルにおいて組み合わせブレークを選択したのち，コンテキスト・メニューの「条件の編集 ...」をクリックしてください。
ここでオープンするダイアログで編集します。
ダイアログによる編集の詳細については，「2.17.4.3 イベントの組み合わせ条件を編集する【E1】【E20】【EZ Emulator】」を参照してください。

注意 1. 組み合わせブレークの設定に関しては（有効イベント数の制限など），「2.17.7 イベント設定に関する留意事項」も参照してください。

注意 2. イベントの組み合わせに関しては，「(1) 組み合わせ条件の編集」の注意も参照してください。

図 2.114 イベントパネル上における組み合わせブレーク（シーケンシャル）の設定例【E1】【E20】【EZ Emulator】



2.10.6 その他のブレーク要因

上記のほか，プログラムの実行が停止する原因（ブレーク要因）には次のものがあります。
なお，ブレーク要因は，プログラム停止時に，メイン・ウインドウのステータス・バーで確認することができます。

表 2.7 その他のブレーク要因

要因	使用するデバッグ・ツール	
	E1(Serial)/E1(JTAG)/E20(Serial)/E20(JTAG)/EZ Emulator	シミュレータ
トレース・メモリを使い切った[注 1]	○	○
テンポラリ・ブレークの発生	○	○
実行の失敗，または不明な原因	○	–
WAIT 命令の実行	–	○
未定義命令例外の発生	–	○[注 2]
特権命令例外の発生	–	○[注 2]

要因	使用するデバッグ・ツール	
	E1(Serial)/E1(JTAG)/E20(Serial)/E20(JTAG)/EZ Emulator	シミュレータ
アクセス例外の発生	―	○注 2
浮動小数点例外の発生	―	○注 2
割り込みの発生	―	○注 2
INT 命令例外の発生	―	○注 2
BRK 命令例外の発生	―	○注 2
周辺機能シミュレーションによるエラーの発生	―	○
不正なメモリ・アクセス	―	○注 2
ストリーム入出力エラーの発生	―	○
カバレッジ・メモリの確保に失敗	―	○
トレース・メモリの確保に失敗	―	○

注 1. プロパティ パネルの［デバッグ・ツール設定］タブ上の［トレース］カテゴリ内［トレース・メモリを使い切った後の動作］プロパティの設定に依存

注 2. 【シミュレータ】
プロパティ パネルの［デバッグ・ツール設定］タブ上の［実行モード］【シミュレータ】カテゴリ内各プロパティの設定に依存

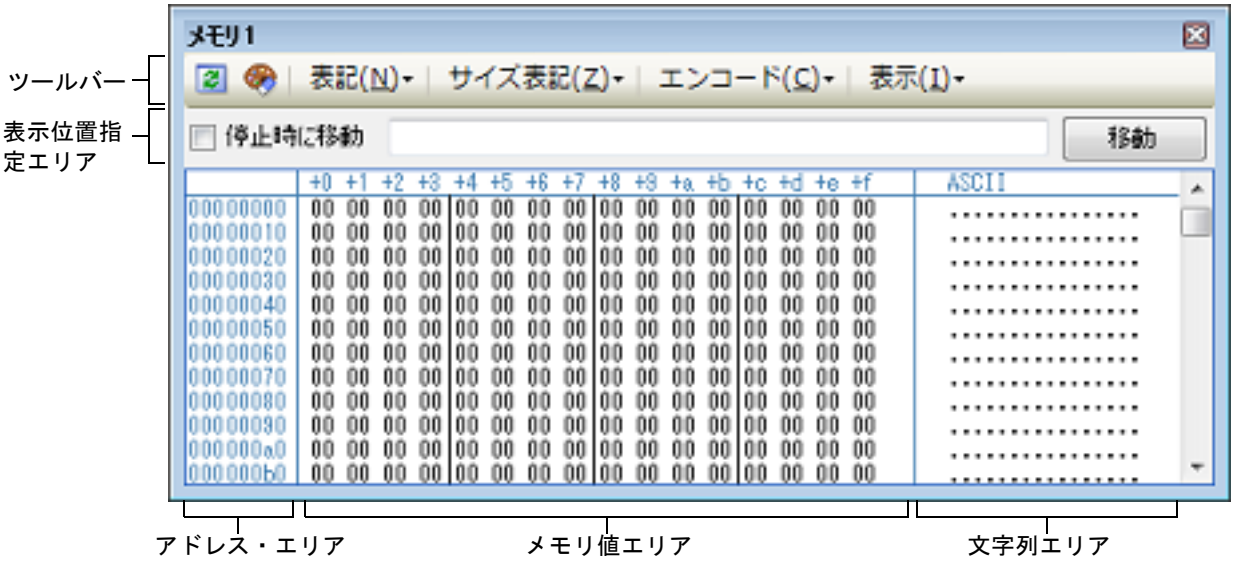
2.11 メモリ，レジスタ，変数の表示／変更

この節では，メモリ，レジスタ，および変数の内容を表示／変更する方法について説明します。

2.11.1 メモリを表示／変更する

メモリの内容の表示，および値の変更は，次のメモリ パネルで行います。
［表示］メニュー→［メモリ］→［メモリ *1* ～ *4*］を選択してください。
メモリ パネルは，最大 4 個までオープンすることができ，各パネルはタイトルバーの“メモリ 1”，“メモリ 2”，“メモリ 3”，“メモリ 4”の名称で識別されます。
なお，各エリアの見方，および機能についての詳細は，メモリ パネルの項を参照してください。

図 2.115 メモリの内容の表示



備考 ツールバーの [表示] → ボタンをクリックすることによりオープンするスクロール範囲設定 ダイアログにより，このパネルの垂直スクロール・バーのスクロール範囲を設定することができます。

ここでは，次の操作方法について説明します。

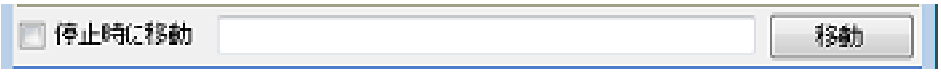
2.11.1.1 表示位置を指定する
2.11.1.2 値の表示形式を変更する
2.11.1.3 メモリの内容を変更する
2.11.1.4 プログラム実行中にメモリの内容を表示／変更する
2.11.1.5 メモリの内容を検索する
2.11.1.6 メモリの内容を一括して変更（初期化）する
2.11.1.7 メモリの表示内容を保存する

2.11.1.1 表示位置を指定する

表示位置指定エリアにアドレス式を指定することにより，メモリ値の表示開始位置を指定することができます（デフォルトでは，0x0 番地より表示を開始します）。

備考 コンテキスト・メニューの [表示アドレス・オフセット値を設定 ...] を選択することでオープンするアドレス・オフセット設定 ダイアログにより，メモリ値の表示開始アドレスにオフセット値を設定することができます。

図 2.116 表示位置指定エリア（メモリ パネル）



(1) アドレス式の指定
表示したいメモリ値のアドレスとなるアドレス式をテキスト・ボックスに直接入力します。最大 1024 文字までの入力式を指定することができ，その計算結果を表示開始位置アドレスとして扱います。

ただし，マイクロコントローラのアドレス空間よりも大きいアドレス式を指定することはできません。

備考 1.　このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます（「2.20.2 シンボル名の入力補完機能」参照）。

備考 2.　指定したアドレス式がシンボルを表現し，サイズが判明する場合では，そのシンボルの先頭アドレスから終了アドレスまで選択状態で表示します。

(2) アドレス式の自動／手動評価の指定
表示開始位置を変更するタイミングは，［停止時に移動］チェック・ボックスの指定，および［移動］ボタンにより決定します。

［停止時に移動］	☑	プログラム停止後，自動的にアドレス式の評価を行い，その計算結果のアドレスにキャレットが移動します。
	☐	プログラム停止後，アドレス式の評価を自動的に行いません。 この場合，［移動］ボタンをクリックすることにより，アドレス式の評価を行います。
［移動］		［停止時に移動］チェック・ボックスのチェックをしなかった場合，このボタンをクリックすることによりアドレス式の評価を行い，その計算結果のアドレスにキャレットが移動します。

2.11.1.2 値の表示形式を変更する

ツールバーの次のボタンにより，このパネルのアドレス・エリア／メモリ値エリア／文字列エリアの表示形式を変更することができます。

表記	メモリ値の表示形式を変更する次のボタンを表示します。
16 進数	メモリ値を 16 進数で表示します（デフォルト）。
符号付き 10 進数	メモリ値を符号付き 10 進数で表示します。
符号無し 10 進数	メモリ値を符号なし 10 進数で表示します。
8 進数	メモリ値を 8 進数で表示します。
2 進数	メモリ値を 2 進数で表示します。
サイズ表記	メモリ値のサイズの表示形式を変更する次のボタンを表示します。
4 ビット	メモリ値を 4 ビット幅で表示します。
1 バイト	メモリ値を 8 ビット幅で表示します（デフォルト）。
2 バイト	メモリ値を 16 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。
4 バイト	メモリ値を 32 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。
8 バイト	メモリ値を 64 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。
エンコード	文字列のエンコードを変更する次のボタンを表示します。

ASCII	文字列を ASCII コードで表示します（デフォルト）。
Shift_JIS	文字列を Shift_JIS コードで表示します。
EUC-JP	文字列を EUC-JP コードで表示します。
UTF-8	文字列を UTF-8 コードで表示します。
UTF-16	文字列を UTF-16 コードで表示します。
Half-Precision Float	文字列を半精度浮動小数点数値で表示します。
Float	文字列を単精度浮動小数点数値注で表示します。
Double	文字列を倍精度浮動小数点数値で表示します。
Float Complex	文字列を単精度浮動小数点数の複素数で表示します。
Double Complex	文字列を倍精度浮動小数点数の複素数で表示します。
Float Imaginary	文字列を単精度浮動小数点数の虚数で表示します。
Double Imaginary	文字列を倍精度浮動小数点数の虚数で表示します。
表示	表示形式を変更する次のボタンを表示します。
スクロール範囲を設定 ...	スクロール範囲を設定するためのスクロール範囲設定 ダイアログがオープンします。
表示桁数を設定	メモリ値エリアの表示桁数を設定するため，表示桁数設定 ダイアログがオープンします。
表示アドレス・オフセット値を設定	アドレス・エリアに表示するアドレスのオフセット値を設定するため，アドレス・オフセット設定 ダイアログをオープンします。

注　浮動小数点数値表示についての詳細は，メモリ パネルの項を参照してください。

2.11.1.3 メモリの内容を変更する

メモリの値は編集することができます。

メモリ値エリア／文字列エリアにおいて，対象メモリ値にキャレットを移動したのち，直接キーボードより編集します。

メモリ値を編集すると変更箇所の表示色が変化し，この状態で［Enter］キーを押下することにより，変更した値がターゲット・メモリに書き込まれます（［Enter］キーの押下前に［Esc］キーを押下すると編集をキャンセルします）。

ただし，変更の際に入力可能な文字列は，現在指定されている表示進数で扱うことができる文字列に限ります。また，文字列エリアでの変更は，文字コードとして “ASCII” が指定されている場合のみ可能です。

なお，メモリの値の編集は，プログラム実行中の状態でも行うことができます。設定方法についての詳細は，「2.11.1.4 プログラム実行中にメモリの内容を表示／変更する」を参照してください。

値を変更する際において，留意する必要がある例を次に示します。

例 1.　表示ビット幅の最大値を越えた場合
10 進数 8 ビット表示において，表示値 “105” の “1” を編集して “3” を入力した場合，変更値は最大値である “127” となります。

例 2.　数値の途中に “-” を入力した場合
符号あり 10 進数 16 ビット表示において，表示値 “32768” を “32-68” と編集した場合，“3” と “2” が空白に変わり，変更値は “-68” となります。

例 3.　数値の途中に空白記号（スペース）を入力した場合
10 進数 16 ビット表示において，表示値 “32767” を “32 67” と編集した場合，“3” と “2” が空白に変わり，変更値は “67” となります。

例 4.　同一の値を入力した場合
現在のメモリ値と同一の値を指定した場合でも，指定した値をメモリに書き込みます。

2.11.1.4 プログラム実行中にメモリの内容を表示／変更する

メモリ パネル／ウォッチ パネルでは，プログラムの実行中に，リアルタイムにメモリ／ウォッチ式の内容を表示更新，および書き換えることができるリアルタイム表示更新機能を備えています。
このリアルタイム表示更新機能を有効化することにより，プログラムが停止している状態時だけでなく，実行中の状態であっても，メモリ／ウォッチ式の値の表示／変更を行うことができます。

注意 1. 【E20(JTAG)【RX600 シリーズ】】
トレース機能／リアルタイム RAM モニタ機能（RRM 機能）は一部排他使用の機能です。このため，以降の設定を行う前に，優先的に使用する機能を指定することを目的として，プロパティ パネルの［デバッグ・ツール設定］タブにおいて次の設定を完了させておく必要があります。

- ［トレース］カテゴリ→［トレース機能の用途］プロパティ→［リアルタイム RAM モニタ］

注意 2. 【E20(JTAG)【RX600 シリーズ】】
プログラム実行中に端子リセットやウォッチドッグタイマなどによるリセットが発生した場合，それ以降のリアルタイム RAM モニタの動作は保証できません。（正しい値を表示できない可能性があります。）

注意 3. 【E20】
リアルタイム RAM モニタ機能を使用する場合は，E20 とターゲット・ボードを 38 ピンで接続している必要があります。
14 ピンで接続している場合は，リアルタイム RAM モニタ機能は使用できません。

備考　リアルタイム表示更新機能は，デバッグ・ツールの RRM（Real-time RAM Monitor）機能，擬似 RRM 機能，DMM（Dynamic Memory Modification）機能により実現されます。
擬似 RRM 機能では，プログラムの実行を一瞬ブレークさせ，ソフトウエア・エミュレーションによる読み込み／書き込みを行います。

リアルタイム表示更新機能を使用した読み込み／書き込みが可能となる対象領域は，使用するデバッグ・ツールとプロパティ パネルの設定内容により異なります。
次に示すプロパティ パネルの［デバッグ・ツール設定］タブ上の［実行中のメモリ・アクセス］カテゴリ，および［トレース］カテゴリ内の各プロパティの設定内容に基づき，リアルタイム表示更新機能の使用目的に応じ，表 2.8/ 表 2.9/ 表 2.10 の設定を行ってください。

設定記号	［トレース］カテゴリ内のプロパティ	設定値	備考
A	トレース機能の用途	リアルタイム RAM モニタ	E20(JTAG)【RX600 シリーズ】

設定記号	［実行中のメモリ・アクセス］カテゴリ内のプロパティ	設定値	備考
B	実行を一瞬停止してアクセスする【E1】【E20】【EZ Emulator】	［いいえ］（デフォルト）	RRM/DMM 機能の有効化
	リアルタイム表示更新を自動設定する	［はい］（デフォルト）	E20(JTAG)【RX600 シリーズ】
C	実行を一瞬停止してアクセスする【E1】【E20】【EZ Emulator】	［はい］	擬似 RRM 機能の有効化
D	実行中に表示更新を行う	［はい］（デフォルト）	
	表示更新間隔 [ms]	［*100 ～ 65500 の整数*］	

(1) 読み込みの場合

表 2.8 リアルタイム表示更新機能（擬似 RRM 機能による読み込み）の対象領域

<table>
<tr><th rowspan="2">領域名</th><th colspan="2">E1(Serial)/ E1(JTAG)/E20(Serial)/ E20(JTAG)/EZ Emulator</th></tr>
<tr><th>設定</th><th>対象</th></tr>
<tr><td>内蔵 ROM</td><td rowspan="4">C+D</td><td rowspan="4">全領域</td></tr>
<tr><td>内蔵 RAM</td></tr>
<tr><td>データフラッシュ</td></tr>
<tr><td>ターゲット・メモリ</td></tr>
<tr><td>エミュレーション・メモリ</td><td colspan="2">－</td></tr>
<tr><td>CPU レジスタ</td><td colspan="2" rowspan="2">不可</td></tr>
<tr><td>I/O レジスタ</td></tr>
</table>

表 2.9 リアルタイム表示更新機能（RRM 機能による読み込み）の対象領域

<table>
<tr><th rowspan="2">領域名</th><th colspan="2">E20(JTAG)[RX600 シリーズ】</th><th colspan="2">シミュレータ</th></tr>
<tr><th>設定</th><th>対象</th><th>設定</th><th>対象</th></tr>
<tr><td>内蔵 ROM</td><td rowspan="4">A+B+D</td><td rowspan="4">自動設定領域注</td><td rowspan="2">D</td><td rowspan="2">全領域</td></tr>
<tr><td>内蔵 RAM</td></tr>
<tr><td>データフラッシュ</td><td colspan="2" rowspan="2">－</td></tr>
<tr><td>ターゲット・メモリ</td></tr>
<tr><td>エミュレーション・メモリ</td><td colspan="2">－</td><td>D</td><td>全領域</td></tr>
<tr><td>CPU レジスタ</td><td colspan="2" rowspan="2">不可</td><td colspan="2">－</td></tr>
<tr><td>I/O レジスタ</td><td>D</td><td>全領域</td></tr>
</table>

注 【E20(JTAG)【RX600 シリーズ】】
RRM 機能の対象領域は，最大 4096 バイト（最大 1024 バイトまでの領域を 4 つ）に限定されます。そのため，CS+ では，次に示す決定規則（優先順位）に従い，上記制限内によるリアルタイム表示更新の対象を自動的に決定します（プログラム実行直前において，前面に表示しているパネルのみが対象となります）。
(1) ウォッチ パネルに表示しているウォッチ式を上から順に設定（複数のウォッチ パネルをオープンしている場合，パネルの番号の小さい順）
(2) メモリ パネルに表示しているメモリをアドレスの小さい順に設定（複数のメモリ パネルをオープンしている場合，パネルの番号の小さい順）
なお，対象領域にリード禁止領域が含まれている場合は，その領域の表示更新は行いません。

(2) 書き込みの場合

表 2.10 リアルタイム表示更新機能（DMM 機能による書き込み）の対象領域

<table>
<tr><th rowspan="2">領域名</th><th colspan="2">E1(Serial)/ E1(JTAG)/E20(Serial)/ E20(JTAG)/EZ Emulator</th><th colspan="2">シミュレータ</th></tr>
<tr><th>設定</th><th>対象</th><th>設定</th><th>対象</th></tr>
<tr><td>内蔵 ROM</td><td colspan="2">不可</td><td rowspan="2">D</td><td rowspan="2">全領域</td></tr>
<tr><td>内蔵 RAM</td><td>C+D</td><td>全領域</td></tr>
<tr><td>データフラッシュ</td><td colspan="2">不可</td><td colspan="2" rowspan="2">－</td></tr>
<tr><td>ターゲット・メモリ</td><td>C+D</td><td>全領域</td></tr>
</table>

領域名	E1(Serial)/ E1(JTAG)/E20(Serial)/ E20(JTAG)/EZ Emulator		シミュレータ	
	設定	対象	設定	対象
エミュレーション・メモリ	―		D	全領域
CPU レジスタ	不可		―	
I/O レジスタ			D	全領域

注意　ローカル変数は，リアルタイム表示更新機能の対象外です。

備考　メモリ パネル／ウォッチ パネルにおける値の書き換え方法についての詳細は，「2.11.1.3 メモリの内容を変更する」／「2.11.6.6 ウォッチ式の内容を変更する」を参照してください。

擬似 RRM 機能で更新しているメモリ値／ウォッチ式は，メモリ パネルにおいてピンク色で強調表示され，RRM 機能で更新しているメモリ値／ウォッチ式は，アクセス状態に従って次のように強調表示されます（表示の際の文字色／背景色は，オプション ダイアログにおける［全般 - フォントと色］カテゴリの設定に依存）。

アクセス状態	表示例
リード／フェッチ	00 00 00 00
ライト	00 00 00 00
リードとライト	00 00 00 00
ロスト注	00 00 00 00

注　ウォッチ パネルにおけるロスト時のウォッチ式の値は，”?” と表示されます。

図 2.117　擬似 RRM 機能によるメモリ表示の例（メモリ パネル）【E1】【E20】【EZ Emulator】

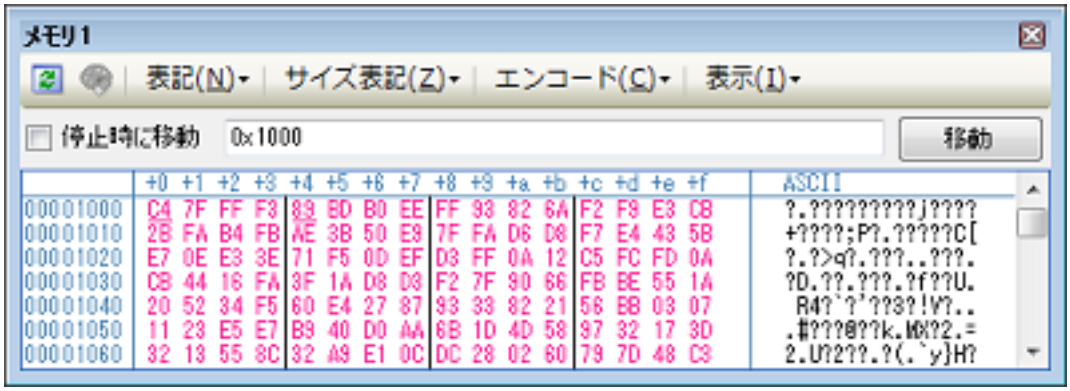


図 2.118　　RRM 機能によるメモリ表示の例（メモリ パネル）【シミュレータ】

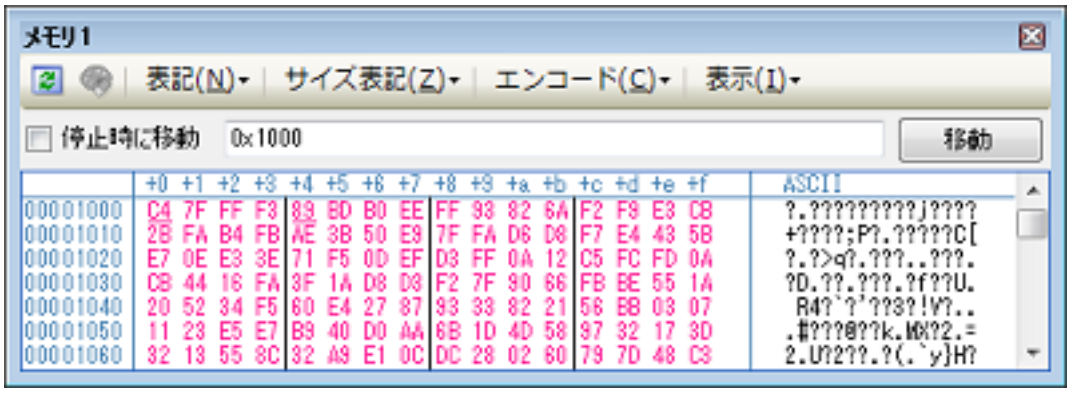
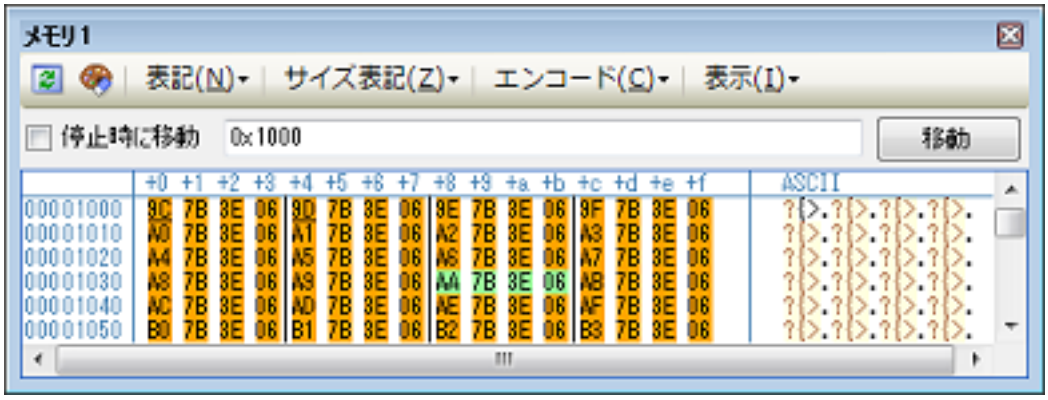


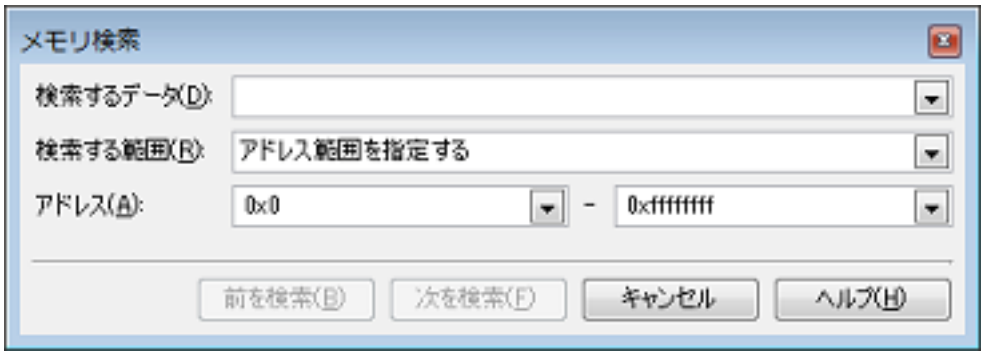
図 2.119　　RRM 機能によるメモリ表示の例（メモリ パネル）【E20(JTAG)【RX600 シリーズ】】



2.11.1.5 メモリの内容を検索する

メモリの値の検索は，コンテキスト・メニューの［検索 ...］を選択することによりオープンするメモリ検索 ダイアログで行います。検索の際は，メモリ値エリアと文字列エリアのうち，キャレットのあるエリアが対象となります。
このダイアログにおいて，次の手順で操作を行ってください。

図 2.120　　メモリ内容の検索（メモリ検索 ダイアログ）



注意 1.　　プログラム実行中に，メモリの内容を検索することはできません。

注意 2.　　浮動少数点数値表示している文字列を検索することはできません。

(1)　［検索するデータ］の指定
検索するデータを指定します。
テキスト・ボックスに直接入力するか（最大指定バイト数：256 バイト），またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
検索の対象がメモリ値エリアの場合，そのエリアと同じ表示形式（表示進数／サイズ）でデータを入力する必要があります。
また，検索の対象が文字列エリアの場合では，検索するデータとして文字列を指定する必要があります。指定した文字列は，そのエリアで表示しているエンコード形式でデータに変換され検索されます。

なお，このダイアログをオープンする直前にメモリ値を選択していた場合は，デフォルトでその値が表示されます。

(2)　［検索する範囲］の指定
検索する範囲を次のドロップダウン・リストより選択します。

アドレス範囲を指定する	［アドレス］で指定するアドレス範囲内で検索を行います。
メモリ・マッピング	選択したメモリ・マッピング範囲内で検索を行います。 このリスト項目は，メモリ・マッピング ダイアログで表示しているメモリ・マッピングを個々に表示します（ノン・マップ領域を除く）。 表示形式：*＜メモリ種別＞ ＜アドレス範囲＞ ＜サイズ＞*

(3)　［アドレス］の指定
この項目は，「(2)　［検索する範囲］の指定」で［アドレス範囲を指定する］を選択した場合のみ有効となります。
メモリ値検索の対象となるアドレス範囲を“開始アドレス - 終了アドレス”で指定します。それぞれのテキスト・ボックスにアドレス式を直接入力するか（最大指定文字数：1024 文字），またはドロップダウン・リストにより入力履歴項目（最大履歴個数：10 個）を選択することにより行います。
入力したアドレス式の計算結果を，それぞれ開始アドレス／終了アドレスとして扱います。
ただし，検索可能なアドレスの上限値は，プログラム空間の上限アドレス（0xFFFFFFFF）です。
また，32 ビットで表現できる値より大きいアドレス値を指定することはできません。

備考 1.　このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます（「2.20.2 シンボル名の入力補完機能」参照）。

備考 2.　“開始アドレス”が空欄の場合は，“0x0”の指定として扱われます。

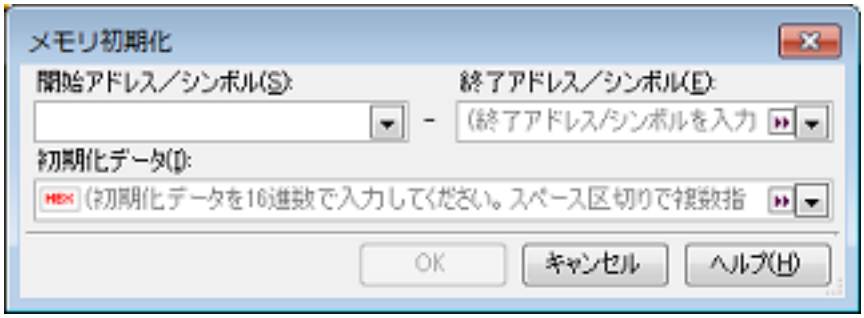
備考 3.　“終了アドレス”が空欄の場合は，マイクロコントローラのアドレス空間の上限値の指定として扱われます。

(4)　［前を検索］／［次を検索］ボタンのクリック
［前を検索］ボタンをクリックすると，指定した範囲内でアドレスの小さい方向に検索を行い，検索結果箇所をメモリ パネル上で選択状態にします。
［次を検索］ボタンをクリックすると，指定した範囲内でアドレスの大きい方向に検索を行い，検索結果箇所をメモリ パネル上で選択状態にします。

2.11.1.6 メモリの内容を一括して変更（初期化）する

メモリの値を一括して変更（初期化）することができます。
コンテキスト・メニューの［初期化 ...］を選択することにより，指定したアドレス範囲のメモリ値を一括して変更するためのメモリ初期化 ダイアログがオープンします。
このダイアログにおいて，次の手順で操作を行ってください。

図 2.121　メモリ内容の一括変更（メモリ初期化 ダイアログ）



(1)　［開始アドレス / シンボル］と［終了アドレス / シンボル］の指定
メモリの内容を初期化するアドレス範囲を［開始アドレス / シンボル］と［終了アドレス / シンボル］に指定します。それぞれのテキスト・ボックスにアドレス式を直接入力するか（最大指定文字数：1024 文字），またはドロップダウン・リストにより入力履歴項目（最大履歴個数：10 個）を選択します。
入力したアドレス式の計算結果を，それぞれ開始アドレス／終了アドレスとして扱います。
なお，マイクロコントローラのアドレス空間よりも大きいアドレス値を指定することはできません。

注意　エンディアンの異なる領域をまたいだアドレス範囲を指定することはできません。

備考　　このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます（「2.20.2 シンボル名の入力補完機能」参照）。

(2)　［初期化データ］の指定
メモリに書き込む初期化データを指定します。
16 進数の数値をテキスト・ボックスに直接入力するか，またはドロップダウン・リストにより入力履歴項目（最大履歴個数：10 個）を選択します。
初期化データを複数指定する場合は，1 個 4 バイト（8 文字）までのデータを最大 16 個まで，半角スペースで区切り指定します。
個々の初期化データは，文字列終端より 2 文字単位で 1 バイトと解釈され，奇数文字数の場合は先頭 1 文字で 1 バイトと解釈されます。
なお，バイト数が 2 バイト以上の場合は，次のように初期化対象となるアドレス範囲のエンディアンに合わせたバイト列に変換してターゲット・メモリへの書き込み処理を行います。

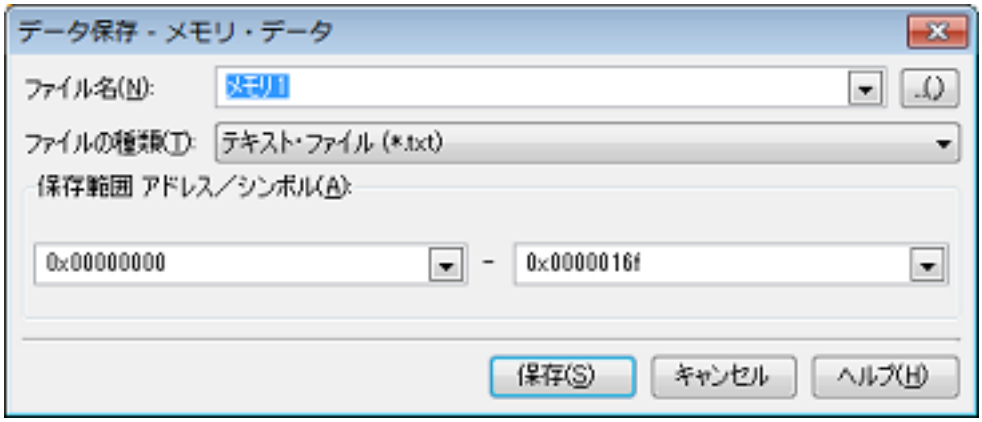
入力文字列（初期化データ）	書き込みイメージ（バイト単位）	
	リトルエンディアン	ビッグエンディアン
1	01	01
0 12	00 12	00 12
00 012 345	00 12 00 45 03	00 00 12 03 45
000 12 000345	00 00 12 45 03 00	00 00 12 00 03 45

(3)　［OK］ボタンのクリック
［OK］ボタンをクリックします。
指定したアドレス範囲のメモリ領域に，指定した初期化データのパターンを繰り返し書き込みます（パターンの途中で終了アドレスに達した場合は書き込みを終了します）。
ただし，不正な値やアドレス式を指定している場合，メッセージを表示し，メモリ値の初期化は行いません。

2.11.1.7 メモリの表示内容を保存する

メモリの内容を範囲指定して，テキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存することができます。
ファイルに保存する際は，デバッグ・ツールから最新の情報を取得し，このパネル上での表示形式に従ったデータで保存します。
［ファイル］メニュー→［名前を付けてメモリ・データを保存 ...］を選択すると，次のデータ保存 ダイアログがオープンします（この際，パネル上で範囲選択した状態でこの操作を行うと選択範囲のみのメモリ・データを保存することができます）。
このダイアログにおいて，次の手順で操作を行ってください。

図 2.122　　メモリ・データの保存（データ保存 ダイアログ）



(1)　［ファイル名］の指定
保存するファイル名を指定します。
テキスト・ボックスに直接入力するか（最大指定文字数：259 文字），またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
また，［...］ボタンをクリックすることでオープンするデータ保存ファイルを選択 ダイアログにより，ファイルを選択することもできます。

(2) ［ファイルの種類］の指定
保存するファイルの形式を次のドロップダウン・リストにより選択します。
選択できるファイルの形式は次のとおりです。

リスト表示	形式
テキスト・ファイル (*.txt)	テキスト形式（デフォルト）
CSV(カンマ区切り) (*.csv)	CSV 形式[注]

注　各データを “,” で区切り保存します。
なお，データ内に “,” が含まれている際の不正形式を避けるため，各データを “"” （ダブルクォーテーション）で括り出力します。

(3) ［保存範囲 アドレス／シンボル］の指定
ファイルに保存する範囲を “ 開始アドレス ” と “ 終了アドレス ” で指定します。
それぞれのテキスト・ボックスに 16 進数の数値／アドレス式を直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
なお，パネル上で範囲選択している場合は，デフォルトでその選択範囲がテキスト・ボックスに指定されます。
範囲選択していない場合は，現在のパネルの表示範囲が指定されます。

備考　このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます（「2.20.2 シンボル名の入力補完機能」参照）。

(4) ［保存］ボタンのクリック
指定したファイルに，指定した形式でメモリ・データを保存します。

図 2.123　メモリ・データ保存の際の出力イメージ

【テキスト・ファイル（*.txt）で保存】
（16 進表記 /8 ビット幅 /ASCII コードの場合の例）

```
         +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +a +b +c +d +e +f
00000000| 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
00000010| 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 | ................
00000020| 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 | """""""""""""""""
00000030| 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 | 3333333333333333
```

【CSV ファイル（*csv）で保存】
（16 進表記 /8 ビット幅 /ASCII コードの場合の例）

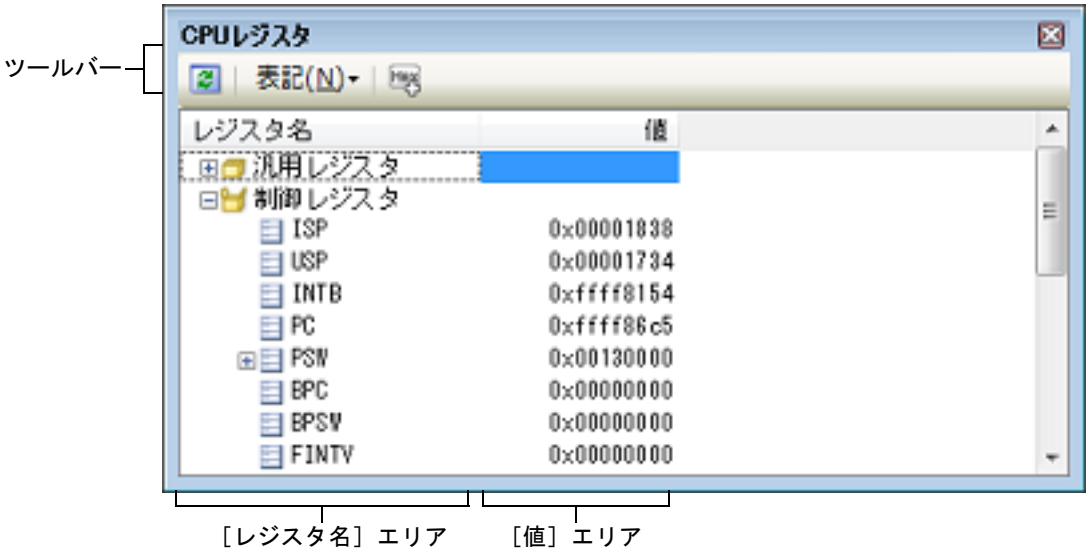
```
00000000,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,................
00000010,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,................
00000020,22,22,22,22,22,22,22,22,22,22,22,22,22,22,22,22,"""""""""""""""""
00000030,33,33,33,33,33,33,33,33,33,33,33,33,33,33,33,33,3333333333333333
```

備考　［ファイル］メニュー→［メモリ・データを保存］の選択によりパネルの内容を上書き保存する場合，メモリ パネル（メモリ 1 ～ 4）はそれぞれ個別に扱われます。
また，保存範囲についても，前回指定したアドレス範囲で保存されます。

2.11.2 CPU レジスタを表示／変更する

CPU レジスタ（汎用レジスタ／制御レジスタ）の内容の表示，および値の変更は，次の CPU レジスタ パネルで行います。
［表示］メニュー→［CPU レジスタ］を選択してください。
なお，各エリアの見方，および機能についての詳細は，CPU レジスタ パネルの項を参照してください。

図 2.124　CPU レジスタの内容の表示（CPU レジスタ パネル）



ここでは，次の操作方法について説明します。

2.11.2.1 値の表示形式を変更する
2.11.2.2 CPU レジスタの内容を変更する
2.11.2.3 CPU レジスタの表示内容を保存する

2.11.2.1 値の表示形式を変更する

［値］エリアの表示形式は，ツールバーの次のボタンにより，自由に変更することができます。

表記	値の表示形式を変更する次のボタンを表示します。
自動	選択している項目（下位項目を含む）の値を規定値で表示します（デフォルト）。
16 進数	選択している項目（下位項目を含む）の値を 16 進数で表示します。
符号付き 10 進数	選択している項目（下位項目を含む）の値を符号付き 10 進数で表示します。
符号なし 10 進数	選択している項目（下位項目を含む）の値を符号なし 10 進数で表示します。
8 進数	選択している項目（下位項目を含む）の値を 8 進数で表示します。
2 進数	選択している項目（下位項目を含む）の値を 2 進数で表示します。
ASCII	選択している項目（下位項目を含む）の文字列を ASCII コードで表示します。 対象が 2 バイト以上ある場合は，1 バイトずつの文字を並べて表示します。
Float	選択している項目を Float で表示します。 ただし，4 バイト・データ以外の場合は，規定値で表示します。
Double	選択している項目を Double で表示します。 ただし，8 バイト・データ以外の場合は，規定値で表示します。
	値表示の末尾に，その値の 16 進数表記を“（）”で囲んで併記します。

2.11.2.2 CPU レジスタの内容を変更する

CPU レジスタの値は，編集することができます。
［値］エリアにおいて，対象 CPU レジスタ値を選択したのち再度クリックすると，値が編集モードになります（［Esc］キーの押下で編集モードをキャンセルします）。
値をキーボードより直接編集したのち，［Enter］キーを押下することにより，変更した値がデバッグ・ツールのターゲット・メモリに書き込まれます。

注意　　この操作は，プログラム実行中に行うことはできません。

2.11.2.3 CPU レジスタの表示内容を保存する

［ファイル］メニュー→［名前を付けて CPU レジスタ・データを保存 ...］を選択することにより，名前を付けて保存ダイアログをオープンし，CPU レジスタのすべての内容をテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存することができます。
ファイルに保存する際は，デバッグ・ツールから最新の情報を取得します。

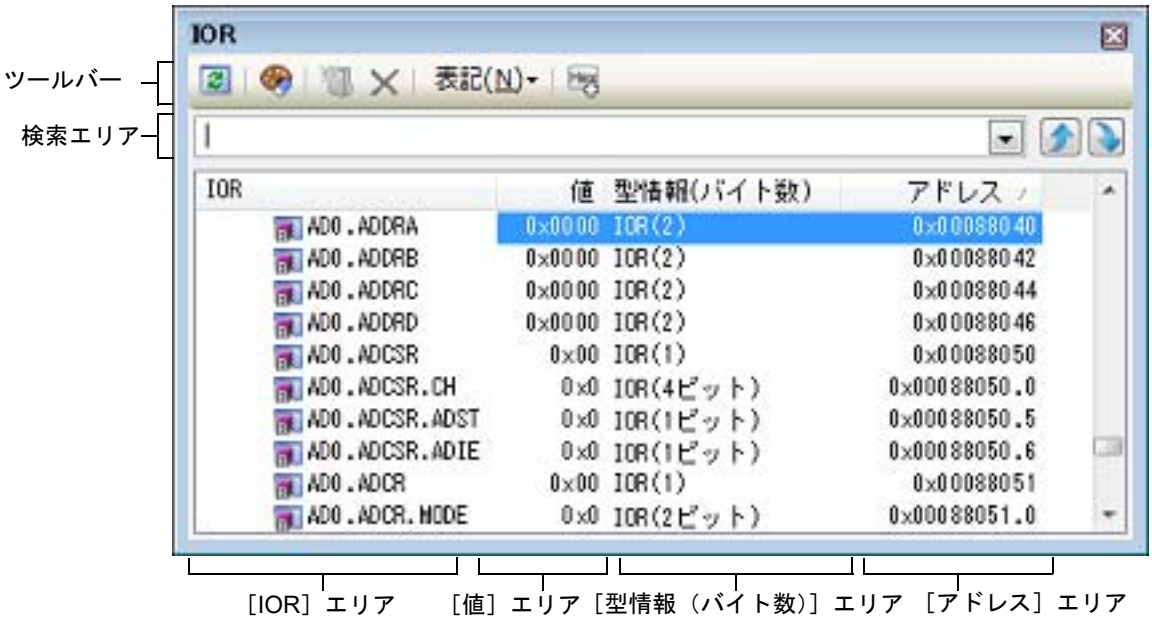
図 2.125　CPU レジスタ保存の際の出力イメージ

```
レジスタ名                値
-----------------------------------------------------------------------
カテゴリ名
 - レジスタ名             値
      :                   :
```

2.11.3 I/O レジスタを表示／変更する

I/O レジスタの内容の表示，および値の変更は，次の IOR パネルで行います。
［表示］メニュー→［IOR］を選択してください。
なお，各エリアの見方，および機能についての詳細は，IOR パネルの項を参照してください。

図 2.126　I/O レジスタの内容の表示（IOR パネル）



ここでは，次の操作方法について説明します。

2.11.3.1 I/O レジスタを検索する
2.11.3.2 I/O レジスタを整理する
2.11.3.3 値の表示形式を変更する
2.11.3.4 I/O レジスタの内容を変更する
2.11.3.5 プログラム実行中に I/O レジスタの内容を表示／変更する

2.11.3.6 I/O レジスタの表示内容を保存する

2.11.3.1 I/O レジスタを検索する

I/O レジスタ名を検索することができます。
検索エリアにおいて，テキスト・ボックスに検索する I/O レジスタ名を指定します（大文字／小文字不問）。キーボードより文字列を直接入力するか（最大指定文字数：512 文字），ドロップダウン・リストより入力履歴項目を選択します（最大履歴：10 個）。
次のいずれかのボタンをクリックします。

	テキスト・ボックスで指定している文字列を含む I/O レジスタ名を上方向に検索し，検索結果を選択状態にします。
	テキスト・ボックスで指定している文字列を含む I/O レジスタ名を下方向に検索し，検索結果を選択状態にします。

備考 1. カテゴリ（フォルダ）により分類されて非表示の状態の I/O レジスタ名も検索します（展開して選択状態となります）。

備考 2. 検索対象の文字列入力後，[Enter] キーを押下することにより，ボタンのクリックと同等の動作を行い，[Shift]+[Enter] キーを押下することにより，ボタンのクリックと同等の動作を行います。

2.11.3.2 I/O レジスタを整理する

各 I/O レジスタを任意のカテゴリ（フォルダ）で分類し，ツリー形式を編集することができます。

注意 1. カテゴリ内にカテゴリを作成することはできません。

注意 2. I/O レジスタの追加／削除はできません。

(1) カテゴリを新規作成する場合
作成したい I/O レジスタ名にキャレットを移動したのち，ツールバーのボタンのクリックし，キーボードより新規カテゴリ名を直接入力します。

(2) カテゴリ名を編集する場合
編集したいカテゴリ名を選択したのち，再度クリックし，キーボードよりカテゴリ名を直接編集します。

(3) カテゴリを削除する場合
削除したいカテゴリを選択したのち，ツールバーのボタンをクリックします。
ただし，削除できるカテゴリは，空のカテゴリのみです。

(4) 表示順を変更する場合
I/O レジスタ名をカテゴリ内に直接ドラッグ・アンド・ドロップすることにより，I/O レジスタはカテゴリで分類されます。
また，カテゴリと I/O レジスタ名の表示の順番（上下位置）も，ドラッグ・アンド・ドロップ操作により自由に変更することができます。

2.11.3.3 値の表示形式を変更する

［値］エリアの表示形式は，ツールバーの次のボタンにより，自由に変更することができます。

表記	値の表示形式を変更する次のボタンを表示します。
16 進数	選択している項目の値を 16 進数で表示します（デフォルト）。
符号付き 10 進数	選択している項目の値を符号付き 10 進数で表示します。
符号なし 10 進数	選択している項目の値を符号なし 10 進数で表示します。
8 進数	選択している項目の値を 8 進数で表示します。
2 進数	選択している項目の値を 2 進数で表示します。
ASCII	選択している項目の値を ASCII コードで表示します。

	選択している項目の値表示の末尾に，その値の 16 進数表記を“（）”で囲んで併記します。

2.11.3.4 I/O レジスタの内容を変更する

I/O レジスタの値は，編集することができます。
［値］エリアにおいて，対象 I/O レジスタ値を選択したのち再度クリックすると，値が編集モードになります（[Esc] キーの押下で編集モードをキャンセルします）。
値をキーボードより直接編集したのち，［Enter］キーを押下することにより，変更した値がデバッグ・ツールのターゲット・メモリに書き込まれます。

注意 1. この操作は，プログラム実行中に行うことはできません。

注意 2. 読み込み専用の I/O レジスタの値を変更することはできません。

備考 1. I/O レジスタのサイズより小さい桁の数値が入力された場合，上位の桁を 0 でパディングします。

備考 2. I/O レジスタのサイズより大きい桁の数値が入力された場合，上位の桁をマスクします。

備考 3. I/O レジスタの値には ASCII 文字による入力も可能です。

- I/O レジスタ名 “CRC.CRCDOR” の値に “0x41” を書き込んだ場合
  → CRC.CRCDOR に，“0x41” が書き込まれます。
- I/O レジスタ名 “CRC.CRCDOR” の値に ASCII 文字 “’A’” を書き込んだ場合
  → CRC.CRCDOR に，“0x41” が書き込まれます。

2.11.3.5 プログラム実行中に I/O レジスタの内容を表示／変更する

対象となる I/O レジスタをウォッチ式としてウォッチ パネルに登録することにより，プログラムが停止状態だけでなく，実行状態であっても I/O レジスタの値をリアルタイムに表示／変更することができます。
ウォッチ式についての詳細は，「2.11.6 ウォッチ式を表示／変更する」を参照してください。

2.11.3.6 I/O レジスタの表示内容を保存する

［ファイル］メニュー→［名前を付けて IOR データを保存 ...］を選択することにより，名前を付けて保存 ダイアログをオープンし，I/O レジスタのすべての内容をテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存することができます（このパネル上での表示／非表示の設定に関わらず，すべての I/O レジスタの値が対象となります）。
ファイルに保存する際は，I/O レジスタの値を再読み込みし，取得した最新の値を保存します。
ただし，読み込み保護対象の I/O レジスタの再読み込みは行いません。最新の内容を保存したい場合は，コンテキスト・メニューの［値を強制読み込み］を選択したのち，ファイルの保存を行ってください。

図 2.127 I/O レジスタ保存の際の出力イメージ

```
IOR 名                    値          型情報（バイト数）      アドレス
-----------------------------------------------------------------------
カテゴリ名
 -IOR 名                  値          型情報（バイト数）      アドレス
    :                      :                :                  :
```

2.11.4 グローバル変数／スタティック変数を表示／変更する

グローバル変数，またはスタティック変数の値の表示／変更は，ウォッチ パネルで行います。
値の表示／変更を行いたい変数をウォッチ式としてウォッチ パネルに登録してください。
ウォッチ式についての詳細は，「2.11.6 ウォッチ式を表示／変更する」を参照してください。

2.11.5 ローカル変数を表示／変更する

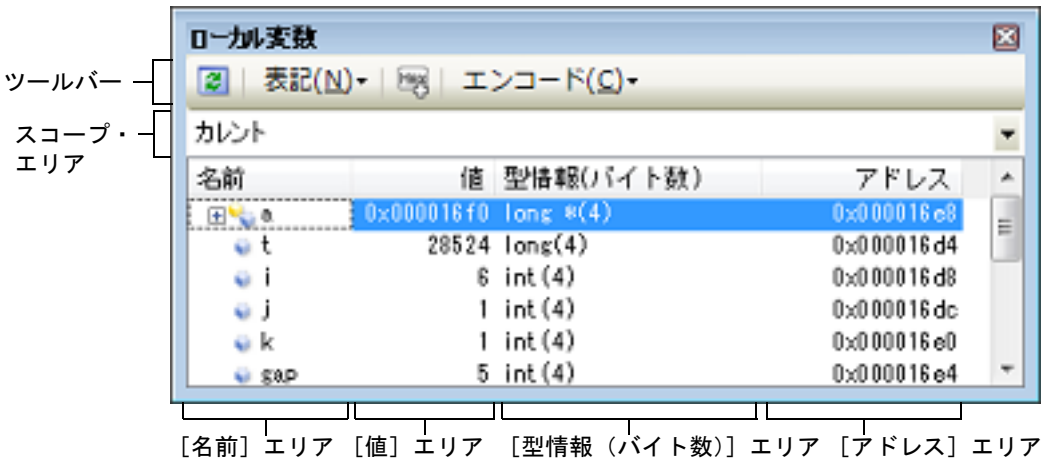
ローカル変数の内容の表示，および値の変更は，次のローカル変数 パネルで行います。
［表示］メニュー→［ローカル変数］を選択してください。
目的のローカル変数の内容を表示するためには，スコープ・エリアでスコープの選択をします。

ローカル変数 パネルでは，ローカル変数名や関数名を表示します。また，関数の引数もローカル変数として表示します。

なお，各エリアの見方，および機能についての詳細は，ローカル変数 パネルの項を参照してください。

注意　プログラム実行中は，このパネルには何も表示されません。
プログラムの実行が停止したタイミングで，各エリアの表示を行います。

図 2.128　ローカル変数の内容の表示（ローカル変数 パネル）



ここでは，次の操作方法について説明します。

2.11.5.1 値の表示形式を変更する
2.11.5.2 ローカル変数の内容を変更する
2.11.5.3 ローカル変数の表示内容を保存する

2.11.5.1 値の表示形式を変更する

［値］エリアの表示形式は，ツールバーの次のボタンにより，自由に変更することができます。

表記	値の表示形式を変更する次のボタンを表示します。
自動	このパネル上の値の表記を変数ごとの規定値で表示します（デフォルト）。
16 進数	このパネル上の値を 16 進数で表示します。
10 進数	このパネル上の値を 10 進数で表示します。
8 進数	このパネル上の値を 8 進数で表示します。
2 進数	このパネル上の値を 2 進数で表示します。
配列のインデックスを 10 進数表記	このパネル上の配列のインデックスを 10 進数で表示します（デフォルト）。
配列のインデックスを 16 進数表記	このパネル上の配列のインデックスを 16 進数で表示します。
Float	このパネル上の値を Float で表示します。 ただし，4 バイト・データ以外，または型情報を持つ場合は，規定値で表示します。
Double	このパネル上の値を Double で表示します。 ただし，8 バイト・データ以外，または型情報を持つ場合は，規定値で表示します。
	値表示の末尾に，その値の 16 進数表記を“（）”で囲んで併記します。
エンコード	文字列変数のエンコードを変更する次のボタンを表示します。

ASCII	文字列変数を ASCII コードで表示します（デフォルト）。
Shift_JIS	文字列変数を Shift_JIS コードで表示します。
EUC-JP	文字列変数を EUC-JP コードで表示します。
UTF-8	文字列変数を UTF-8 コードで表示します。
UTF-16	文字列変数を UTF-16 コードで表示します。

2.11.5.2 ローカル変数の内容を変更する

ローカル変数の値，および引数の値は，編集することができます。
［値］エリアにおいて，対象ローカル変数値／引数値を選択したのち再度クリックすると，値が編集モードになります（［Esc］キーの押下で編集モードをキャンセルします）。
値をキーボードより直接編集したのち，［Enter］キーを押下することにより，変更した値がデバッグ・ツールのターゲット・メモリに書き込まれます。この際に，値のチェックを行い，型に不適合な場合は編集を無効とします。

注意 この操作は，プログラム実行中に行うことはできません。

備考 1. 変数のサイズより小さい桁の数値が入力された場合，上位の桁を 0 でパディングします。

備考 2. 変数のサイズより大きい桁の数値が入力された場合，上位の桁をマスクします。

備考 3. 文字配列（char 型，unsigned char 型）に対しては，表示形式に ASCII が選択されている場合，文字列（ASCII/Shift_JIS/EUC-JP/Unicode(UTF-8/UTF-16)）による値の入力も可能です。

備考 4. ローカル変数の値には，次のように ASCII 文字による入力も可能です。

- ASCII 文字による入力の場合
  変数 “ch” の［値］エリアに “’A’” を入力
  → “ch” が割り当てられているメモリ領域に “0x41” を書き込む
- 数値による入力の場合
  変数 “ch” の［値］エリアに “0x41” を入力
  → “ch” が割り当てられているメモリ領域に “0x41” を書き込む
- 文字列（ASCII）による入力の場合
  文字配列 “str” の表示形式を ASCII に設定し，［値］エリアに “"ABC"” を入力
  → “str” が割り当てられているメモリ領域に “0x41，0x42，0x43，0x00” を書き込む

2.11.5.3 ローカル変数の表示内容を保存する

［ファイル］メニュー→［名前を付けてローカル変数データを保存 ...］を選択することにより，名前を付けて保存 ダイアログをオープンし，ローカル変数のすべての内容をテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存することができます。
ファイルに保存する際は，デバッグ・ツールから最新の情報を取得します。
なお，配列，ポインタ型変数，構造体／共用体，CPU レジスタ（部分を表す名前が付与されているもののみ）を展開表示している場合では，各展開要素の値も保存されます。展開表示していない場合は，先頭に “+” マークが付与され，値は空欄となります。

図 2.129 ローカル変数保存の際の出力イメージ

```
スコープ：現在のスコープ
[V] 変数   [P] 引数   [F] 関数
名前                 値          型情報 ( バイト数 )        アドレス
-----------------------------------------------------------------
[V] 変数名 [1]       値          型情報 ( バイト数 )        アドレス
 -[V] 変数名 [0]     値          型情報 ( バイト数 )        アドレス
        :            :                  :                      :
```

2.11.6 ウォッチ式を表示／変更する

C 言語変数，CPU レジスタ，I/O レジスタ，およびアセンブラ・シンボルなどをウォッチ式として，次のウォッチ パネルに登録することにより，それらの値を常にデバッグ・ツールから取得し，一括して値を監視することができます。

また，ウォッチ式は，プログラムが実行中の状態にあっても値の表示を逐次更新することができます（「2.11.6.7 プログラム実行中にウォッチ式の内容を表示／変更する」参照）。

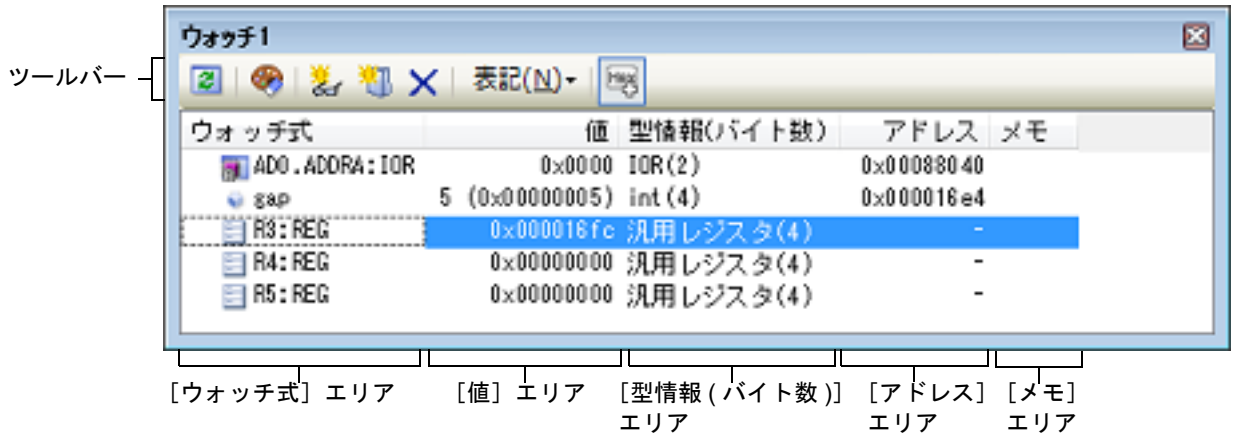
ウォッチ パネルは，［表示］メニュー→［ウォッチ］→［ウォッチ *1* ～ *4*］の選択でオープンします。

ウォッチ パネルは，最大 4 個までオープンすることができます。各パネルは，タイトルバーの“ウォッチ 1”，“ウォッチ 2”，“ウォッチ 3”，“ウォッチ 4”の名称で識別され，それぞれのウォッチ パネルが個別にウォッチ式を登録／管理し，プロジェクトのユーザ情報として保存されます。

なお，各エリアの見方，および機能についての詳細は，ウォッチ パネルの項を参照してください。

注意 プログラム実行中に CPU レジスタの内容を表示／変更することはできません。

図 2.130 ウォッチ式の内容の表示（ウォッチ パネル）



ここでは，次の操作方法について説明します。

2.11.6.1 ウォッチ式を登録する
2.11.6.2 登録したウォッチ式を整理する
2.11.6.3 登録したウォッチ式を編集する
2.11.6.4 ウォッチ式を削除する
2.11.6.5 値の表示形式を変更する
2.11.6.6 ウォッチ式の内容を変更する
2.11.6.7 プログラム実行中にウォッチ式の内容を表示／変更する
2.11.6.8 ウォッチ式をエクスポート／インポートする
2.11.6.9 ウォッチ式の表示内容を保存する

2.11.6.1 ウォッチ式を登録する

ウォッチ式の登録方法には，次の 3 通りがあります（デフォルトでは，ウォッチ式は登録されていません）。

注意 1. 1 つのウォッチ パネルにおいて，ウォッチ式は 128 個まで登録することができます（上限値を越えて登録しようとした場合，メッセージを表示します）。

注意 2. コンパイラによる最適化のため，対象となる変数を使用していないブロックでは変数データがスタック／レジスタに存在しない場合があります。この場合，対象となる変数をウォッチ式として登録しても値の表示は“?”のままとなります。

備考 1. 各ウォッチ パネル（ウォッチ 1 ～ウォッチ 4）上で登録したウォッチ式は，それぞれ個別に管理され，プロジェクトのユーザ情報として保存されます。

備考 2. ウォッチ式は，同名を複数登録することができます。

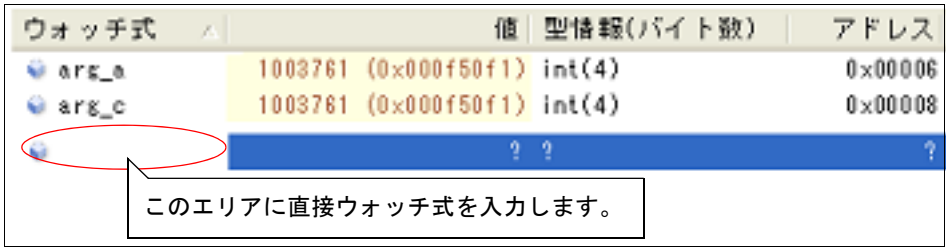
(1) 他のパネルから登録する場合
CS+ の他のパネルから，ウォッチ式を登録することができます。
他のパネルにおいて，ウォッチ式として登録したい対象を任意のウォッチ パネル（ウォッチ 1 ～ウォッチ 4）上に直接ドラッグ・アンド・ドロップします。
なお，この操作が可能なパネルと，ウォッチ式として登録可能な対象との関係についての詳細は，「表 A.4 各パネルとウォッチ式として登録可能な対象の関係」を参照してください。

図 2.131 他のパネルからウォッチ式登録する場合の例

備考 ウォッチ式として登録したい対象を選択したのち，または対象文字列のいずれかにキャレットを移動したのち（対象は自動的に決定されます），コンテキスト・メニューの［ウォッチ 1 に登録］を選択することによっても同様にウォッチ式を登録することができます（ただし，ウォッチ パネル（ウォッチ 1）に限定）。

(2) ウォッチ パネル上で直接登録する場合
任意のウォッチ パネル（ウォッチ 1 ～ウォッチ 4）において，ツールバーの ボタンをクリックすると，［ウォッチ式］エリアに次のエントリ・ボックスが表示されます。

図 2.132 ウォッチ式のエントリ・ボックス



エントリ・ボックス内に，キーボードより直接ウォッチ式を入力したのち，［Enter］キーを押下します。
なお，この際のウォッチ式の入力形式については，次の表を参照してください。

- 「表 A.5 ウォッチ式の入力形式」
- 「表 A.7 C 言語変数をスコープ指定してウォッチ登録した場合の扱い」
- 「表 A.8 CPU レジスタをスコープ指定してウォッチ登録した場合の扱い」
- 「表 A.9 I/O レジスタをスコープ指定してウォッチ登録した場合の扱い」

備考 このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます（「2.20.2 シンボル名の入力補完機能」参照）。

(3) 他のアプリケーションから登録する場合
外部エディタなどから，C 言語変数 /CPU レジスタ /I/O レジスタ／アセンブラ・シンボルの文字列を選択し，ウォッチ パネル（ウォッチ 1 ～ウォッチ 4）に直接ドラッグ・アンド・ドロップします。
この場合，ドロップした文字列がそのままウォッチ式として登録されます。

2.11.6.2 登録したウォッチ式を整理する

登録したウォッチ式をカテゴリ（フォルダ）で分類し，ツリー形式で表示することができます（デフォルトでは，カテゴリは存在しません）。

注意 1. カテゴリ内にカテゴリを作成することはできません。

注意 2. 1 つのウォッチ パネルにおいて，カテゴリは 64 個まで作成することができます（上限値を越えて作成しようとした場合，メッセージを表示します）。

(1) カテゴリを新規作成する場合
作成したい位置にキャレットを移動したのち，ツールバーの ボタンのクリックし，キーボードより新規カテゴリ名を直接入力します。

(2) カテゴリ名を編集する場合
編集したいカテゴリ名を選択したのち，再度クリックし，キーボードよりカテゴリ名を直接編集します。

(3) カテゴリを削除する場合
削除したいカテゴリを選択したのち，ツールバーの ボタンをクリックします。

(4) 表示順を変更する場合
登録済みのウォッチ式を作成したカテゴリ内に直接ドラッグ・アンド・ドロップすることにより，ウォッチ式はカテゴリで分類されます。
また，カテゴリとウォッチ式の表示の順番（上下位置）も，ドラッグ・アンド・ドロップ操作により自由に変更することができます。

備考 ウォッチ式／カテゴリを他のウォッチ パネル（ウォッチ 1 ～ウォッチ 4）にドラッグ・アンド・ドロップすると，ドロップ先のウォッチ パネルにウォッチ式／カテゴリがコピーされます。

2.11.6.3 登録したウォッチ式を編集する

登録したウォッチ式は，編集することができます。
対象ウォッチ式をダブルクリックすると，対象ウォッチ式が編集モードになります（[Esc] キーの押下で編集モードをキャンセルします）。
キーボードより直接内容を編集し，[Enter] キーを押下してください。

2.11.6.4 ウォッチ式を削除する

登録したウォッチ式を削除する場合は，ウォッチ パネルにおいて，削除したいウォッチ式を選択したのち，ツールバーの ボタンをクリックします。

2.11.6.5 値の表示形式を変更する

[値] エリアの表示形式は，ツールバーの次のボタンにより，自由に変更することができます。

表記	値の表示形式を変更する次のボタンを表示します。
自動	選択しているウォッチ式の値の表記を変数ごとの規定値（「表 A.10 ウォッチ式の表示形式（デフォルト）」参照）で表示します（デフォルト）。
16 進数	選択している項目の値を 16 進数で表示します。
符号付き 10 進数	選択している項目の値を符号付き 10 進数で表示します。
符号無し 10 進数	選択している項目の値を符号なし 10 進数で表示します。
8 進数	選択している項目の値を 8 進数で表示します。
2 進数	選択している項目の値を 2 進数で表示します。
ASCII	選択している項目の値を ASCII コードで表示します。
Float	選択している項目の値を Float で表示します。 ただし，選択しているウォッチ式が 4 バイト・データの場合のみ有効となります。
Double	選択している項目の値を Double で表示します。 ただし，選択しているウォッチ式が 8 バイト・データの場合のみ有効となります。
	選択している項目の値表示の末尾に，その値の 16 進数表記を“（）”で囲んで併記します。 ただし，16 進数表記をしている場合は併記しません。

2.11.6.6 ウォッチ式の内容を変更する

ウォッチ式の値は，編集することができます。
［値］エリアにおいて，対象ウォッチ式の値をダブルクリックすると，値が編集モードになります（［Esc］キーの押下で編集モードをキャンセルします）。
値をキーボードより直接編集したのち，［Enter］キーを押下することにより，変更した値がデバッグ・ツールのターゲット・メモリに書き込まれます。
ただし，値を変更できるのは，C 言語変数 /CPU レジスタ /I/O レジスタ／アセンブラ・シンボルと 1 対 1 に対応するウォッチ式のみです。また，読み込み専用の I/O レジスタの値を変更することもできません。
なお，ウォッチ式の値の編集は，プログラム実行中の状態でも行うことができます。設定方法についての詳細は，「2.11.1.4 プログラム実行中にメモリの内容を表示／変更する」を参照してください。

備考 1.　変数のサイズより小さい桁の数値が入力された場合，上位の桁を 0 でパディングします。

備考 2.　変数のサイズより大きい桁の数値が入力された場合，上位の桁をマスクします。

備考 3.　文字配列（char 型，unsigned char 型）に対しては，表示形式に ASCII が選択されている場合，文字列（ASCII/Shift_JIS/EUC-JP/Unicode(UTF-8)/Unicode(UTF-16)）による値の入力も可能です。

備考 4.　ウォッチ式の値には，次のように ASCII 文字による入力も可能です。

- ASCII 文字による入力の場合
  変数 “ch” の［値］エリアに “’A’” を入力
  → “ch” が割り当てられているメモリ領域に “0x41” を書き込む
- 数値による入力の場合
  変数 “ch” の［値］エリアに “0x41” を入力
  → “ch” が割り当てられているメモリ領域に “0x41” を書き込む
- 文字列（ASCII）による入力の場合
  文字配列 “str” の表示形式を ASCII に設定し，［値］エリアに “”ABC”” を入力
  → “str” が割り当てられているメモリ領域に “0x41，0x42，0x43，0x00” を書き込む

2.11.6.7 プログラム実行中にウォッチ式の内容を表示／変更する

メモリ パネル／ウォッチ パネルでは，プログラムの実行中に，リアルタイムにメモリ／ウォッチ式の内容を表示更新，および書き換えることができるリアルタイム表示更新機能を備えています。
このリアルタイム表示更新機能を有効にすることにより，プログラムが停止している状態の時だけでなく，実行中の状態であっても，メモリ／ウォッチ式の値の表示／変更を行うことができます。
設定方法についての詳細は，「2.11.1.4 プログラム実行中にメモリの内容を表示／変更する」を参照してください。

2.11.6.8 ウォッチ式をエクスポート／インポートする

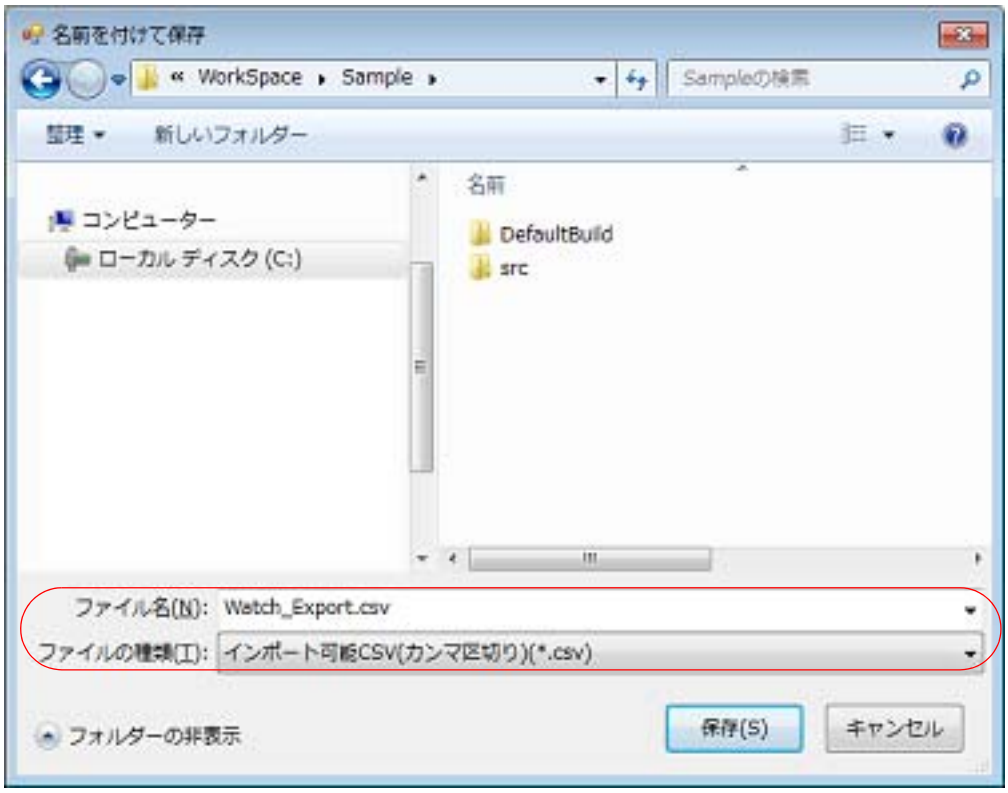
現在登録しているウォッチ式をファイルにエクスポートし，そのファイルをインポートすることにより，ウォッチ式を再登録することができます。
この場合，次の操作を行ってください。

(1) ウォッチ式をエクスポートする
現在登録しているウォッチ式（カテゴリを含む）を，インポート可能なファイル形式で保存します。ウォッチ パネルにフォーカスがある状態で，[ファイル] メニュー→ [名前を付けてウォッチ・データを保存 ...] を選択します。
オープンする名前を付けて保存 ダイアログにおいて，次の指定を行ったのち，[保存] ボタンをクリックします。

- [ファイル名] : 保存するファイル名を指定します（拡張子は “*.csv” に限ります）。
- [ファイルの種類] :“ インポート可能 CSV（カンマ区切り）（*.csv）” を選択します。

注意　値，および型情報は保存されません。
また，配列や構造体などのウォッチ式を解析後に展開される項目は保存されません。

図 2.133 ウォッチ式のエクスポート

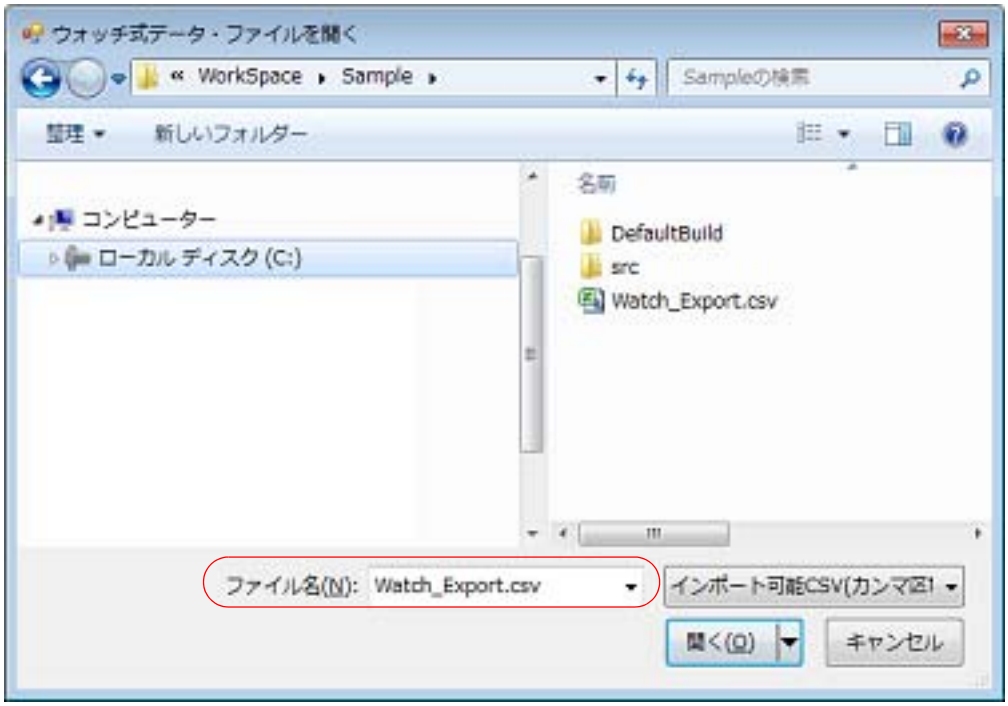


(2) ウォッチ式をインポートする

(1) でエクスポートしたファイルを，ウォッチ パネルにインポートします。
ウォッチ式をインポートしたいウォッチ パネルにおいて，コンテキスト・メニューの [ウォッチ式をインポート ...] を選択します。
オープンするウォッチ式データ・ファイルを開く ダイアログにおいて，先にエクスポートしたファイルを指定したのち，[開く] ボタンをクリックします。

備考 すでにウォッチ式が登録されている場合，最下部のウォッチ式の直後にインポートしたウォッチ式が登録されます。

図 2.134 ウォッチ式のインポート



2.11.6.9 ウォッチ式の表示内容を保存する

［ファイル］メニュー→［名前を付けてウォッチ・データを保存 ...］を選択する，またはコンテキスト・メニューの［ウォッチ・データを展開保存 ...］を選択することにより，名前を付けて保存 ダイアログをオープンし，ウォッチ式と値のすべての内容をテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存することができます。

ファイルに保存する際は，すべてのウォッチ式の値を再読み込みし，取得した最新の値を保存します。

ただし，読み込み保護対象の I/O レジスタの再読み込みは行いません。最新の内容を保存したい場合は，コンテキスト・メニューの［値を強制読み込み］を選択したのち，ファイルの保存を行ってください。

なお，配列，ポインタ型変数，構造体／共用体，レジスタ（部分名がついているもののみ）などの展開表示が可能なウォッチ式の場合，［名前を付けてウォッチ・データを保存 ...］から保存した場合と［ウォッチ・データを展開保存 ...］から保存した場合で動作が異なります。

- ［名前を付けてウォッチ・データを保存 ...］から保存した場合
  展開表示している場合は，各展開要素の値も保存します。展開表示していない場合は，先頭に “+” マークを付与して値は空欄になります。
- ［ウォッチ・データを展開保存 ...］から保存した場合
  展開状態にかかわらず最大 255 階層まで展開して，各展開要素の値も保存します。

図 2.135 ウォッチ・データ保存の際の出力イメージ

```
ウォッチ式      値      型情報 ( バイト数 )     アドレス     メモ
-----------------------------------------------------------------------
変数式          値      型情報 ( バイト数 )     アドレス     メモ
- カテゴリ名
 変数式         値      型情報 ( バイト数 )     アドレス     メモ
   :             :              :                 :           :
```

備考 ［ファイル］メニュー→［ウォッチ・データを保存］の選択によりパネルの内容を上書き保存した場合，ウォッチ パネル（ウォッチ 1 ～ 4）はそれぞれ個別に扱われます。

2.12 スタックからの関数呼び出し情報の表示

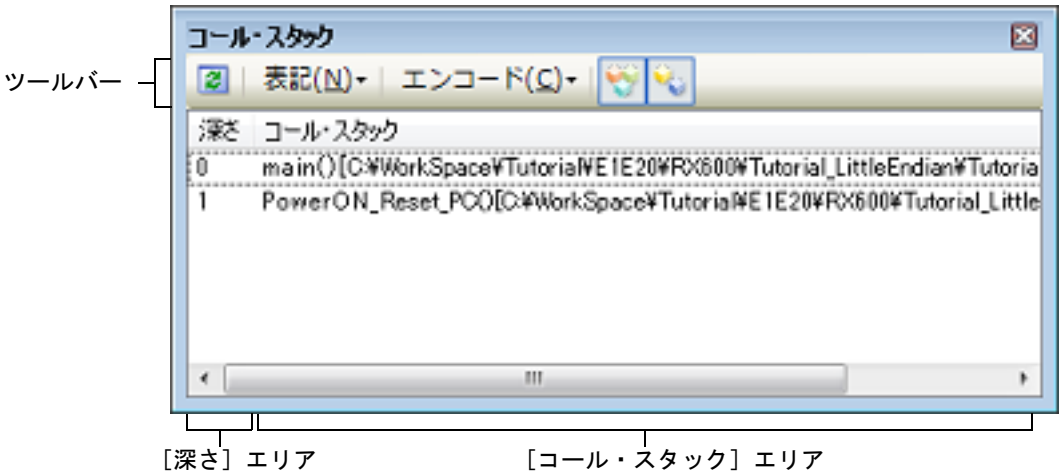
この節では，スタックからの関数呼び出し情報の表示方法について説明します。
CS+ が提供するコンパイラ（CC-RX）は，ANSI 規格に沿って関数呼び出し情報をスタックに積んでいます。この関数呼び出し情報（以降，コール・スタック情報と呼びます）を解析することで，関数の呼び出しの深さ，呼び出し元位置，および引数などを知ることができます。

2.12.1 コール・スタック情報を表示する

コール・スタック情報の表示は，次のコール・スタック パネルで行います。
［表示］メニュー→［コール・スタック］を選択してください。
なお，各エリアの見方，および機能についての詳細は，コール・スタック パネルの項を参照してください。

注意　プログラム実行中は，このパネルには何も表示されません。
プログラムの実行が停止したタイミングで，各エリアの表示を行います。

図 2.136　コール・スタック情報の表示（コール・スタック パネル）



ここでは，次の操作方法について説明します。

2.12.1.1 値の表示形式を変更する
2.12.1.2 ソース行へジャンプする
2.12.1.3 ローカル変数を表示する
2.12.1.4 コール・スタック情報の表示内容を保存する

2.12.1.1 値の表示形式を変更する

ツールバーの次のボタンにより，このパネルの表示形式を変更することができます。

表記	値の表示形式を変更する次のボタンを表示します。
自動	このパネル上の値の表記を変数ごとの規定値で表示します（デフォルト）。
16 進数	このパネル上の値を 16 進数で表示します。
10 進数	このパネル上の値を 10 進数で表示します。
8 進数	このパネル上の値を 8 進数で表示します。
2 進数	このパネル上の値を 2 進数で表示します。

エンコード	文字列変数のエンコードを変更する次のボタンを表示します。
ASCII	このパネル上の文字列変数を ASCII コードで表示します（デフォルト）。
Shift_JIS	このパネル上の文字列変数を Shift_JIS コードで表示します。
EUC-JP	このパネル上の文字列変数を EUC-JP コードで表示します。
UTF-8	このパネル上の文字列変数を UTF-8 コードで表示します。
UTF-16	このパネル上の文字列変数を UTF-16 コードで表示します。

2.12.1.2 ソース行へジャンプする

行をダブルクリックすることにより，選択行が示す関数呼び出し元のソース行にキャレットを移動した状態でエディタ パネルがオープンします（すでにオープンしている場合は，エディタ パネルにジャンプ）。

備考　コンテキスト・メニューの［逆アセンブルへジャンプ］を選択することにより，現在選択している行が示す関数呼び出し元のアドレスにキャレットを移動した状態で逆アセンブル パネル（逆アセンブル 1）がオープンします（すでにオープンしている場合は，逆アセンブル パネル（逆アセンブル 1）にジャンプ）。

2.12.1.3 ローカル変数を表示する

コンテキスト・メニューの［このときのローカル変数を表示］を選択することにより，現在選択している行が示す関数のローカル変数を表示するローカル変数 パネルをオープンします。

2.12.1.4 コール・スタック情報の表示内容を保存する

［ファイル］メニュー→［名前を付けてコール・スタック・データを保存 ...］を選択することにより，名前を付けて保存 ダイアログをオープンし，コール・スタック情報のすべての内容をテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存することができます。
ファイルに保存する際は，デバッグ・ツールから最新の情報を取得します。

図 2.137　コール・スタック情報保存の際の出力イメージ

```
深さ          コール・スタック
-----------------------------------------------------------------------
 0            コール・スタック情報
 1            コール・スタック情報
 :                  :
```

2.13 実行履歴の収集

この節では，プログラムの実行履歴の収集方法について説明します。
一般的に，プログラムの実行履歴をトレースと呼び，以降の記述で使用します。プログラムが暴走した場合，暴走後のメモリ内容やスタック情報などから原因を探ることは非常に困難ですが，収集したトレース・データの内容を解析することにより，暴走するまでの過程を直接探ることができ，プログラムの潜在的バグを発見するために有効です。

注意 1. 【E20(JTAG)【RX600 シリーズ】】
トレース機能／リアルタイム RAM モニタ機能（RRM 機能）は一部排他使用の機能です。

注意 2. 【E1】【E20】【EZ Emulator】
アクション・イベントの Printf イベントを使用している場合，Printf イベントが成立するとそれまでの実行履歴は消去され，イベント成立後の実行履歴が記録されます。

注意 3. 【EZ Emulator【RX100, RX200 シリーズ】】
端子リセットを入れた場合，リセットが入る前のトレース情報は消去されます。EZ Emulator からの再実行後のトレース情報が再取得されます。

注意 4. 【E1】【E20】【EZ Emulator】
ユーザ・プログラム実行中に端子リセットやウォッチドッグタイマなどによるリセットが発生した場合，リセット前後のトレース記録が正しく行えなくなります。

2.13.1 トレース動作の設定をする

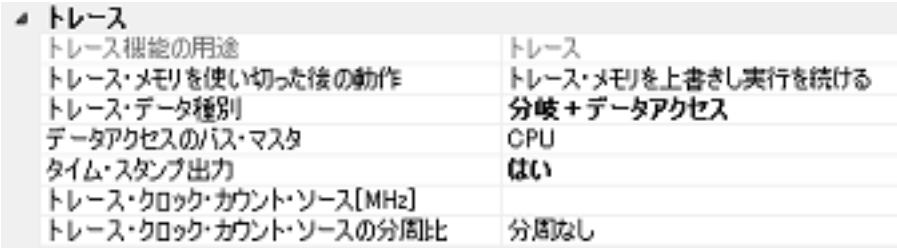
トレース機能が開始すると，現在実行中のプログラムの実行過程を記録したトレース・データがトレース・メモリに収集されます（プログラムの実行が停止すると，自動的にトレース機能も停止します）。
トレース機能を使用するためには，あらかじめトレースの動作に関する設定を行う必要があります。
なお，設定方法は，使用するデバッグ・ツールにより異なります。

2.13.1.1 【E1】の場合
2.13.1.2 【E20】の場合
2.13.1.3 【EZ Emulator】の場合
2.13.1.4 【シミュレータ】の場合

2.13.1.1 【E1】の場合

設定は，プロパティ パネルの［デバッグ・ツール設定］タブ上の［トレース］カテゴリ内で行います。

図 2.138 ［トレース］カテゴリ【E1】



(1) ［トレース機能の用途］
トレース機能のみ使用できます。

(2) ［トレース・メモリを使い切った後の動作］
収集したトレース・データでトレース・メモリがいっぱいになった際の動作を，次のドロップダウン・リストにより指定します。

トレース・メモリを上書きし実行を続ける	トレース・メモリがいっぱいになると，古いトレース・データに上書きを続けます（デフォルト）。
トレースを停止する	トレース・メモリがいっぱいになると，トレース・データの書き込みを停止します（プログラムの実行は停止しません）。
停止する	トレース・メモリがいっぱいになると，トレース・データの書き込みを停止すると同時にプログラムの実行を停止します。

(3) ［トレース・データ種別］
このプロパティは，収集するトレース・データの種別を次のドロップダウン・リストにより指定します。

分岐	プログラム実行中に発生した分岐処理の分岐元，分岐先アドレス情報をトレース・データとして収集します。
分岐＋データアクセス注 1	プログラム実行中に発生した分岐元，分岐先アドレス情報および成立したアクセス・イベントのデータ情報をトレース・データとして収集します。
データアクセス注 2	プログラム実行中に成立したアクセス・イベントのデータ情報をトレース・データとして収集します。

注 1. 【E1(Serial)【RX100, RX200 シリーズ】】
[分岐＋データアクセス] のトレース・データを収集することはできません。そのためドロップダウン・リストには表示されません。

注 2. 【E1(Serial)【RX100, RX200 シリーズ】】
[データアクセス] のトレース・データを収集するには，ポイント・トレースにてアドレス条件を設定する必要があります。設定しない場合，データアクセスのトレース情報が取得されません。
データアクセスのトレース情報には，指定されたアドレスへのアクセスのみ記録します。なお，トレース・パネルのアドレスには，必ずポイント・トレースで設定した開始アドレス値が表示されます。
ポイント・トレースの詳細については，「2.13.4 条件を満たしたときのみの実行履歴を収集する」を参照してください。

(4) ［データアクセスのバス・マスタ］【RX71M, RX64M グループ】
このプロパティは，［トレース・データ種別］プロパティにおいて［分岐＋データアクセス］または［データアクセス］を指定した場合にのみ表示されます。
データアクセスのバス・マスタをドロップダウン・リストより指定します。
ドロップダウン・リストには，次のバス・マスタが表示されます。

CPU	CPU からのデータアクセス結果を，トレースパネルに出力します。
DMAC/DTC	DMAC/DTC からのデータアクセス結果を，トレースパネルに出力します。

データアクセスのトレース結果については，指定したバス・マスタからのトレース結果のみをトレースパネルに出力します。

注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. データアクセスのバス・マスタ選択機能が無いマイクロコントローラの場合は，［データアクセスのバス・マスタ］【RX71M, RX64M グループ】プロパティは表示されません。その場合，バス・マスタは［CPU］固定となります。

(5) ［タイム・スタンプ出力］
このプロパティは，収集するトレース・データにタイム・スタンプ情報を付加するか否かを指定します。タイム・スタンプを出力する場合，［はい］を指定してください。

注意 1. 【E1(Serial)【RX200 シリーズ】】
［トレース・データ種別］プロパティにおいて［分岐］を指定した場合は，タイム・スタンプを出力することができません。出力する場合は，［データアクセス］を指定してください。

注意 2. 【E1【RX100 シリーズ】】
このプロパティの項目を［いいえ］から変更することはできず，タイム・スタンプを出力することができません。

(6) ［トレース・クロック・カウント・ソース [MHz]］
タイム・スタンプのカウント・ソースを 0.0001 - 999.999 の間で入力してください。
なお，プロパティが空欄の場合，［接続用設定］タブの［クロック］カテゴリ内の［動作周波数 [MHz]］プロパティの設定値がカウント・ソースとして代用されます。

(7) ［トレース・クロック・カウント・ソースの分周比］【RX71M, RX64M グループ】
このプロパティは，［タイム・スタンプ出力］プロパティにおいて［はい］を指定した場合にのみ表示されます。
タイム・スタンプのカウント・ソースの分周比をドロップダウン・リストより指定します。
ドロップダウン・リストには，次の分周比が表示されます。

分周なし	タイム・スタンプ用カウント・ソースの周波数をそのままの値で使用する。
16 分周	タイム・スタンプ用カウント・ソースの周波数を 1/16 倍に変換して使用する。

256 分周	タイム・スタンプ用カウント・ソースの周波数を 1/256 倍に変換して使用する。
4096 分周	タイム・スタンプ用カウント・ソースの周波数を 1/4096 倍に変換して使用する。

［トレース・クロック・カウント・ソース [MHz]］プロパティで指定した周波数をさらに指定した値で分周（周波数を 1/n 倍に変換）した値が実際のタイム・スタンプの単位（1 カウントあたりの周波数）になります。

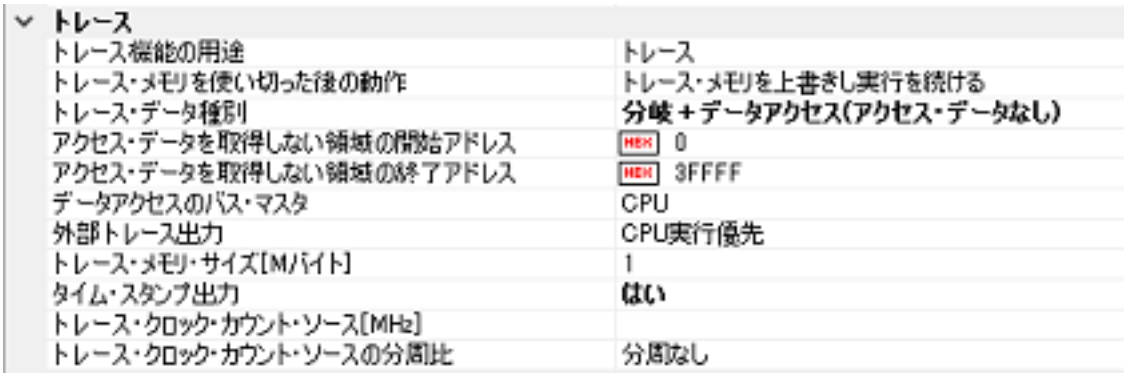
注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. トレースのタイム・スタンプ分周機能が無いマイクロコントローラの場合は，［トレース・クロック・カウント・ソースの分周比］【RX71M, RX64M グループ】プロパティは表示されません。その場合，［分周なし］固定となります。

2.13.1.2 【E20】の場合

設定は，プロパティ パネルの［デバッグ・ツール設定］タブ上の［トレース］カテゴリ内で行います。

図 2.139 ［トレース］カテゴリ【E20】



(1) ［トレース機能の用途］
トレース機能／リアルタイム RAM モニタ機能（RRM 機能）は一部排他使用の機能です。このため，このプロパティで，どちらの機能を優先的に使用するかを指定します。ここでは，次のドロップダウン・リストより，［トレース］を選択してください。

トレース	トレース機能を優先的に使用します（デフォルト）。 - リアルタイム RAM モニタ機能 使用不可となります。
リアルタイム RAM モニタ 注 1	リアルタイム RAM モニタ機能 (RRM) を優先的に使用します。 - トレース機能 制限付きの使用となります。注 2 また，トレース関連のイベントが無効状態となります。

注 1. 【E20(Serial)】
リアルタイム RAM モニタ機能はサポートしていません。そのため，このプロパティ値には [リアルタイム RAM モニタ] を指定しないでください。

注 2. 【E20(JTAG)【RX600 シリーズ】】
トレース機能の一部が使用できなくなります。
制限事項は次のとおりです。

トレース・メモリを使い切った後の動作	[トレースを停止する]，[停止する] は使用不可 [トレース・メモリを上書きし実行を続ける] のみ使用可
トレース・データ種別	[分岐]，[分岐＋データアクセス] は使用不可 [データアクセス] のみ使用可
外部トレース出力	[外部出力しない] は使用不可 [CPU 実行優先]，[トレース出力優先] は使用可

トレース・メモリ・サイズ [M バイト]	1M バイトのみ使用可

(2) ［トレース・メモリを使い切った後の動作］
収集したトレース・データでトレース・メモリがいっぱいになった際の動作を，次のドロップダウン・リストにより指定します。

トレース・メモリを上書きし実行を続ける	トレース・メモリがいっぱいになると，古いトレース・データに上書きを続けます（デフォルト）。
トレースを停止する	トレース・メモリがいっぱいになると，トレース・データの書き込みを停止します（プログラムの実行は停止しません）。
停止する	トレース・メモリがいっぱいになると，トレース・データの書き込みを停止すると同時にプログラムの実行を停止します。

(3) ［トレース・データ種別］
このプロパティは，［トレース機能の用途］プロパティで［トレース］を選択している場合のみ表示されます。
収集するトレース・データの種別を次のドロップダウン・リストにより指定します。

分岐	プログラム実行中に発生した分岐処理の分岐元，分岐先アドレス情報をトレース・データとして収集します。
分岐＋データアクセス注 1	プログラム実行中に発生した分岐元，分岐先アドレス情報および成立したアクセス・イベントのデータ情報をトレース・データとして収集します。
データアクセス注 2	プログラム実行中に成立したアクセス・イベントのデータ情報をトレース・データとして収集します。
分岐＋データアクセス (アクセス・データなし) 注 3	プログラム実行中に発生した分岐元，分岐先アドレス情報および成立したアクセス・イベントのデータ情報をトレース・データとして収集します。 ただし，アクセス・データは収集しません。
データアクセス (アクセス・データなし) 注 3	プログラム実行中に成立したアクセス・イベントのデータ情報をトレース・データとして収集します。 ただし，アクセス・データは収集しません。

注 1. 【E20(Serial)【RX100, RX200 シリーズ】】
[分岐＋データアクセス] のトレース・データを収集することはできません。そのためドロップダウン・リストには表示されません。

注 2. 【E20(Serial)【RX100, RX200 シリーズ】】
[データアクセス] のトレース・データを収集するには，ポイント・トレースにてアドレス条件を設定する必要があります。ポイント・トレースの詳細については，「2.13.4 条件を満たしたときのみの実行履歴を収集する」を参照してください。

注 3. RX65N-2M, RX651-2M グループのみ使用可能です。

(4) ［アクセス・データを取得しない領域の開始アドレス］【RX65N-2M, RX651-2M グループ】
アクセス・データを取得しない領域の開始アドレスを指定します。

(5) ［アクセス・データを取得しない領域の終了アドレス］【RX65N-2M, RX651-2M グループ】
アクセス・データを取得しない領域の終了アドレスを指定します。

(6) ［データアクセスのバス・マスタ］【RX71M, RX64M グループ】
このプロパティは，［トレース・データ種別］プロパティにおいて［分岐＋データアクセス］または［データアクセス］を指定した場合にのみ表示されます。
データアクセスのバス・マスタをドロップダウン・リストより指定します。
ドロップダウン・リストには，次のバス・マスタが表示されます。

CPU	CPU からのデータアクセス結果を，トレースパネルに出力します。
DMAC/DTC	DMAC/DTC からのデータアクセス結果を，トレースパネルに出力します。

データアクセスのトレース結果については，指定したバス・マスタからのトレース結果のみをトレースパネルに出力します。

注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. データアクセスのバス・マスタ選択機能が無いマイクロコントローラの場合は，［データアクセスのバス・マスタ］【RX71M, RX64M グループ】プロパティは表示されません。その場合，バス・マスタは［CPU］固定となります。

(7) ［外部トレース出力］【E20(JTAG)】
収集するトレース・データの外部出力方法を次のドロップダウン・リストにより指定します。

CPU 実行優先	トレース出力より CPU 実行を優先します。 出力されるトレース情報にロストが発生する場合があります。
トレース出力優先	CPU 実行よりトレース出力を優先します。 トレース出力のために CPU 実行が停止するため，リアルタイム性が損なわれます。
外部出力しない	トレース情報は出力されません。 マイクロコントローラの内蔵バッファのみを使用します。

注意 1. ［トレース機能の用途］プロパティにおいて，[リアルタイム RAM モニタ] が指定された場合，ドロップダウンリストより [外部出力しない] は指定できません。

注意 2. [CPU 実行優先] または [トレース出力優先] を指定した場合，トレース パネルにトレース・データが表示された状態でステップ・イン実行すると正しいトレース・データが表示されない場合があります。

注意 3. 【RX71M, RX64M グループ】
［外部出力しない］から［CPU 実行優先］，または［トレース出力優先］に変更した場合，タイマ計測結果が初期化されます。

(8) ［トレース・メモリ・サイズ [M バイト]］【E20(JTAG)】
トレース・データを保持するメモリサイズを次のドロップダウン・リストにより指定します。

- 1(デフォルト), 2, 4, 8, 16, 32

(9) ［タイム・スタンプ出力］
このプロパティは，収集するトレース・データにタイム・スタンプ情報を付加するか否かを指定します。タイム・スタンプを出力する場合，［はい］を指定してください。

注意 1. 【E20(Serial)【RX200 シリーズ】】
［トレース・データ種別］プロパティにおいて［分岐］を指定した場合は，タイム・スタンプを出力することができません。出力する場合は，［データアクセス］を指定してください。

注意 2. 【E20【RX100 シリーズ】】
このプロパティの項目を［いいえ］から変更することはできず，タイム・スタンプを出力することができません。

(10) ［トレース・クロック・カウント・ソース [MHz]］
タイム・スタンプのカウント・ソースを 0.0001 - 999.999 の間で入力してください。
なお，プロパティが空欄の場合，［接続用設定］タブの［クロック］カテゴリ内の［動作周波数 [MHz]］プロパティの設定値がカウント・ソースとして代用されます。

(11) ［トレース・クロック・カウント・ソースの分周比］【RX71M, RX64M グループ】
このプロパティは，［タイム・スタンプ出力］プロパティにおいて［はい］を指定した場合にのみ表示されます。
タイム・スタンプのカウント・ソースの分周比をドロップダウン・リストより指定します。
ドロップダウン・リストには，次の分周比が表示されます。

分周なし	タイム・スタンプ用カウント・ソースの周波数をそのままの値で使用する。
16 分周	タイム・スタンプ用カウント・ソースの周波数を 1/16 倍に変換して使用する。
256 分周	タイム・スタンプ用カウント・ソースの周波数を 1/256 倍に変換して使用する。
4096 分周	タイム・スタンプ用カウント・ソースの周波数を 1/4096 倍に変換して使用する。

［トレース・クロック・カウント・ソース [MHz]］プロパティで指定した周波数をさらに指定した値で分周（周波数を 1/n 倍に変換）した値が実際のタイム・スタンプの単位（1 カウントあたりの周波数）になります。

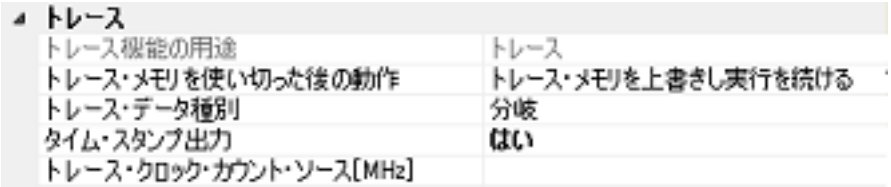
注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. トレースのタイム・スタンプ分周機能が無いマイクロコントローラの場合は，［トレース・クロック・カウント・ソースの分周比］【RX71M, RX64M グループ】プロパティは表示されません。その場合，［分周なし］固定となります。

2.13.1.3 【EZ Emulator】の場合

設定は，プロパティ パネルの［デバッグ・ツール設定］タブ上の［トレース］カテゴリ内で行います。

図 2.140　［トレース］カテゴリ【EZ Emulator】



(1)　［トレース機能の用途］
トレース機能のみ使用できます。

(2)　［トレース・メモリを使い切った後の動作］
収集したトレース・データでトレース・メモリがいっぱいになった際の動作を，次のドロップダウン・リストにより指定します。

トレース・メモリを上書きし実行を続ける	トレース・メモリがいっぱいになると，古いトレース・データに上書きを続けます（デフォルト）。
トレースを停止する	トレース・メモリがいっぱいになると，トレース・データの書き込みを停止します（プログラムの実行は停止しません）。
停止する	トレース・メモリがいっぱいになると，トレース・データの書き込みを停止すると同時にプログラムの実行を停止します。

(3)　［トレース・データ種別］
このプロパティは，収集するトレース・データの種別を次のドロップダウン・リストにより指定します。

分岐	プログラム実行中に発生した分岐処理の分岐元，分岐先アドレス情報をトレース・データとして収集します。
分岐＋データアクセス注 1	プログラム実行中に発生した分岐元，分岐先アドレス情報および成立したアクセス・イベントのデータ情報をトレース・データとして収集します。
データアクセス注 2	プログラム実行中に成立したアクセス・イベントのデータ情報をトレース・データとして収集します。

注 1.　【RX100, RX200 シリーズ】
[分岐＋データアクセス] のトレース・データを収集することはできません。そのためドロップダウン・リストには表示されません。

注 2.　【RX100, RX200 シリーズ】
[データアクセス] のトレース・データを収集するには，ポイント・トレースにてアドレス条件を設定する必要があります。ポイント・トレースの詳細については，「2.13.4 条件を満たしたときのみの実行履歴を収集する」を参照してください。

(4)　［タイム・スタンプ出力］
このプロパティは，収集するトレース・データにタイム・スタンプ情報を付加するか否かを指定します。タイム・スタンプを出力する場合，［はい］を指定してください。

注意 1.　【RX200 シリーズ】
［トレース・データ種別］プロパティにおいて［分岐］を指定した場合は，タイム・スタンプを出力することができません。出力する場合は，［データアクセス］を指定してください。

注意 2.　【RX100 シリーズ】
このプロパティの項目を［いいえ］から変更することはできず，タイム・スタンプを出力することができません。

(5)　［トレース・クロック・カウント・ソース [MHz]］
タイム・スタンプのカウント・ソースを 0.0001 - 999.999 の間で入力してください。
なお，プロパティが空欄の場合，［接続用設定］タブの［クロック］カテゴリ内の［動作周波数 [MHz]］プロパティの設定値がカウント・ソースとして代用されます。

2.13.1.4 【シミュレータ】の場合

設定は，プロパティ パネルの［デバッグ・ツール設定］タブ上の［トレース］カテゴリ内で行います。

図 2.141　［トレース］カテゴリ【シミュレータ】

トレース	
トレース機能を使用する	いいえ
実行前にトレース・メモリをクリアする	はい
トレース・メモリを使い切った後の動作	トレース・メモリを上書きし実行を続ける
トレース・タイム・タグを積算する	いいえ
トレース・メモリ・サイズ[フレーム]	64K

(1)　［トレース機能を使用する］
トレース機能を使用するか否かをドロップダウン・リストにより指定します。
トレース機能を使用する場合は［はい］を指定してください（デフォルトでは［いいえ］が指定されます）。

(2)　［実行前にトレース・メモリをクリアする］
トレース機能を開始する前に，トレース・メモリを一度クリア（初期化）するか否かをドロップダウン・リストにより指定します。
クリアする場合は［はい］を指定してください（デフォルト）。

備考　トレース パネルのツールバーの ボタンをクリックすることにより，トレース・メモリを強制的にクリアすることができます。

(3)　［トレース・メモリを使い切った後の動作］
トレース・メモリが収集したトレース・データでいっぱいになった際の動作を，次のドロップダウン・リストにより指定します。

トレース・メモリを上書きし実行を続ける	トレース・メモリがいっぱいになると，古いトレース・データに上書きを続けます（デフォルト）。 ［実行前にトレース・メモリをクリアする］プロパティで［はい］を指定している場合は，再実行時，トレース・データをクリアしたのちトレース・データの書き込みを行います。
停止する	トレース・メモリがいっぱいになると，トレース・データの書き込みを停止すると同時にプログラムの実行を停止します。 ［実行前にトレース・メモリをクリアする］プロパティで［いいえ］を指定している場合は，再び実行してもプログラムは実行せずに停止します。

(4)　［トレース・タイム・タグを積算する］
トレースの時間表示を積算表示にするか否かをドロップダウン・リストにより指定します。
トレースの時間表示を積算表示にする場合は［はい］を，差分表示にする場合は［いいえ］を指定してください（デフォルト）。

(5)　［トレース・メモリ・サイズ [フレーム]］
トレース・メモリのサイズ（トレース・フレーム数）をドロップダウン・リストにより指定します。
なお，トレース・フレームはトレース・データの一単位を表し，フェッチ／ライト／リードなどで，それぞれ 1 つのトレース・フレームを使用します。
ドロップダウン・リストには，次のトレース・フレーム数が表示されます。
　64K（デフォルト），128K，256K，512K，1M，2M，3M

2.13.2 実行停止までの実行履歴を収集する

デバッグ・ツールには，プログラムの実行開始から実行停止までの実行履歴を収集する機能があらかじめ用意されています。
これにより，プログラムの実行を開始することにより自動的にトレース・データの収集が開始し，実行停止とともにトレース・データの収集も終了します。
なお，収集したトレース・データの確認方法についての詳細は，「2.13.6 実行履歴を表示する」を参照してください。

注意　【シミュレータ】
ストリング操作命令，および積和演算命令のデータアクセスは，最初と最後のアクセス履歴のみ収集します。

備考　この機能は，デバッグ・ツールにデフォルトで設定されているビルトイン・イベントの 1 つである無条件トレース・イベントにより動作します。
この無条件トレース・イベントと後述のトレース・イベント（「2.13.3 任意区間の実行履歴を収集す

る」および「2.13.4 条件を満たしたときのみの実行履歴を収集する」参照）は排他使用のイベントとなります。そのため，トレース・イベントが有効状態で設定されると，無条件トレース・イベントは自動的に無効状態に変更されます。
なお，トレース・イベントが設定されていない場合，無条件トレース・イベントのチェックを外して無効状態としてもトレース・データは収集されます。

2.13.3 任意区間の実行履歴を収集する

トレース・イベントを設定することにより，プログラムの実行過程において，任意の区間の実行履歴のみをトレース・データとして収集することができます。
トレース・イベントは，トレース開始イベントとトレース終了イベントで構成されます。
この機能を使用するためには，次の手順で操作を行います。

2.13.3.1 トレース開始イベント／トレース終了イベントを設定する
2.13.3.2 複数のイベントを組み合わせる【E1】【E20】【EZ Emulator】
2.13.3.3 プログラムを実行する
2.13.3.4 トレース開始イベント／トレース終了イベントを編集する
2.13.3.5 トレース開始イベント／トレース終了イベントを削除する

注意 1. トレース・イベントの設定に関しては（有効イベント数の制限など），「2.17.7 イベント設定に関する留意事項」も参照してください。

注意 2. 【シミュレータ】
ストリング操作命令，および積和演算命令のデータアクセスは，最初と最後のアクセス履歴のみ収集します。

2.13.3.1 トレース開始イベント／トレース終了イベントを設定する

エディタ パネル／逆アセンブル パネルにおいて，トレース・データの収集を開始／終了するイベントを設定します。

(1) 実行系イベントの場合
実行系イベントをトレース開始／終了イベントに設定することにより，任意の箇所でトレース・データの収集を開始／終了させることができます。
実行系イベントをトレース開始イベントとして設定する場合，トレース・データの収集を開始したい行／アドレス[注]にキャレットを移動したのち，コンテキスト・メニューの［トレース設定］→［トレース開始の設定］を選択します。
トレース開始イベントが，キャレット位置の行／アドレスに対応する先頭アドレスの命令に対して設定されます。
また，実行系イベントをトレース終了イベントとして設定する場合，トレース・データの収集を終了したい行／アドレス[注]にキャレットを移動したのち，コンテキスト・メニューの［トレース設定］→［トレース終了の設定］を選択します。
トレース終了イベントが，キャレット位置の行／アドレスに対応する先頭アドレスの命令に対して設定されます。

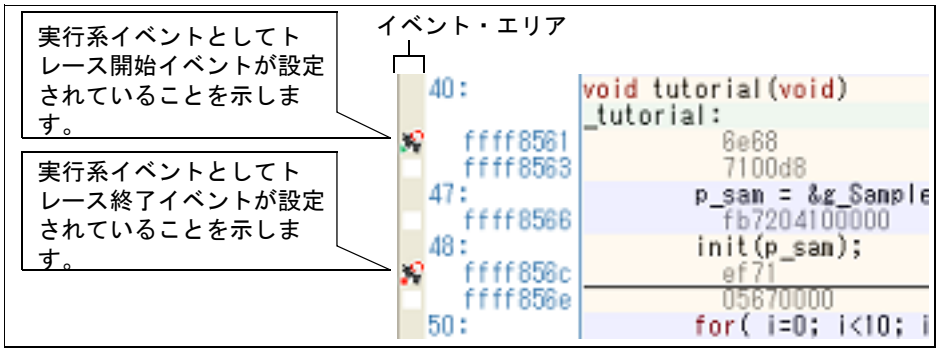
注意 トレース開始イベント／トレース終了イベントは，アドレス表示がない行に設定することはできません。

トレース開始イベント／トレース終了イベントが設定されると，設定した行／アドレスのイベント・エリアに次のイベント・マークが表示されます。

表 2.11 トレース開始イベント／トレース終了イベント・マーク（実行系イベント）

種別	イベント・マーク
トレース開始	
トレース終了	

図 2.142 実行系イベントを用いたトレース開始イベント／トレース終了イベントの設定例（逆アセンブル パネルの場合）



(2) アクセス系イベントの場合

アクセス系イベントをトレース開始／終了イベントに設定することにより，任意の変数，または I/O レジスタに対し，指定したアクセスがあった場合にトレース・データの収集を開始／終了させることができます。
また，この際に，アクセスした値を限定することもできます。
アクセス系イベントで指定できるアクセス種別は次のとおりです。

表 2.12 変数へのアクセス種別

アクセス種別	説明
リード	指定した変数 /I/O レジスタに，リード・アクセスした（読み込みを行った）際にトレース・データの収集を開始／終了します。
ライト	指定した変数 /I/O レジスタに，ライト・アクセスした（書き込みを行った）際にトレース・データの収集を開始／終了します。
リード／ライト	指定した変数 /I/O レジスタに，リード・アクセス，またはライト・アクセスした（読み込み，または書き込みを行った）際にトレース・データの収集を開始／終了します。

備考 DMAC（Direct Memory Access Controller）／ DTC（Data Transfer Controller）によるアクセスは対象となりません。

- ソース・テキスト／逆アセンブル・テキスト上の変数 /I/O レジスタに設定する場合

操作は，ソース・テキスト／逆アセンブル・テキストを表示しているエディタ パネル／逆アセンブル パネル上で行います。
ソース・テキスト／逆アセンブル・テキスト上の任意の変数，または I/O レジスタを選択したのち，コンテキスト・メニューより次の操作を行います。ただし，対象となる変数は，グローバル変数／関数内スタティック変数／ファイル内スタティック変数のみとなります。
なお，この操作を行うことにより，対象変数，または I/O レジスタにトレース・イベント（アクセス系）が設定されたとみなされ，イベント パネルで管理されます（「2.17 イベントの管理」参照）。

アクセス種別	操作方法
リード／ライト	- トレース開始イベント ［トレースの設定］→［読み書き時にトレース開始の設定］を選択したのち，［Enter］キーを押下します。 - トレース終了イベント ［トレースの設定］→［読み書き時にトレース終了の設定］を選択したのち，［Enter］キーを押下します。 この際に，メニュー内のテキスト・ボックスに値を指定した場合，指定した値で読み込み，または書き込みを行った場合のみトレース・データの収集を開始／終了します。値を指定しない場合は，値にかかわらず，選択している変数に読み込み，または書き込みを行った場合にトレース・データの収集を開始／終了します。

注意 1. カレント・スコープ内の変数が対象となります。

注意 2. イベント・エリアにおいて背景色がグレー表示の行（その行が対応するアドレスに変換不可であることを示す）上の変数 /I/O レジスタを選択してもトレース・イベントを設定することはで

きません。このトレース・イベントを設定する際は，イベント・エリアの背景色が白色の行にある変数 /I/O レジスタを選択してください。

注意 3. ソース・テキスト／逆アセンブル・テキスト上から設定可能なアクセス種別は，リード／ライトのみです。アクセス種別をリードおよびライトへ変更する場合は，トレース開始／終了イベントを設定したのち，イベント パネル上からオープンするダイアログで編集します。
（「2.13.3.4 トレース開始イベント／トレース終了イベントを編集する」参照）

- 登録したウォッチ式に設定する場合
操作は，ウォッチ パネル上で行います。
対象となるウォッチ式を選択したのち（複数選択不可），コンテキスト・メニューより次の操作を行います。
ただし，対象となるウォッチ式は，グローバル変数／関数内スタティック変数／ファイル内スタティック変数 /I/O レジスタのみとなります。
なお，この操作を行うことにより，対象ウォッチ式にトレース・イベント（アクセス系）が設定されたとみなされ，イベント パネルで管理されます（「2.17 イベントの管理」参照）。

アクセス種別	操作方法
リード／ライト	- トレース開始イベント ［トレース出力］→［読み書き時にトレース開始の設定］を選択したのち，［Enter］キーを押下します。 - トレース終了イベント ［トレース出力］→［読み書き時にトレース終了の設定］を選択したのち，［Enter］キーを押下します。 この際に，メニュー内のテキスト・ボックスに値を指定した場合，指定された値で読み込み，または書き込みを行った場合のみトレース・データの収集を開始／終了します。 値を指定しない場合は，値に関わらず，選択しているウォッチ式に読み込み，または書き込みを行った場合にトレース・データの収集を開始／終了します。

注意 1. カレント・スコープ内のウォッチ式が対象となります。
カレント・スコープ外のウォッチ式を対象とする場合は，スコープ指定したウォッチ式を選択してください。

注意 2. ウォッチ式から設定可能なアクセス種別は，リード／ライトのみです。アクセス種別をリードおよびライトへ変更する場合は，トレース開始／終了イベントを設定したのち，イベント パネル上からオープンするダイアログから編集します。（「2.13.3.4 トレース開始イベント／トレース終了イベントを編集する」参照）

トレース開始イベント／トレース終了イベントが設定されると，イベント パネル上において，トレース・イベントとして 1 つにまとめて管理されます（トレース・イベント項目の “+” マークをクリックすることにより，設定したトレース開始イベント／トレース終了イベントの詳細情報が表示されます）。

注意 1. トレース開始イベント／トレース終了イベントの両方を設定した場合，開始および終了条件が成立するたびにトレース・データを収集します。この場合，トレース・パネルで正しく逆アセンブリ表示できません。

注意 2. ユーザプログラム実行結果がすべて，“BCND: 条件分岐 ” の場合，トレース情報は表示されません。

注意 3. 【E20(JTAG)【RX600 シリーズ】】
[トレース] カテゴリの [トレース機能の用途] プロパティにおいて，リアルタイム RAM モニタを指定した場合，トレース開始／終了イベントを使用しないでください。また，イベント パネルにおいて，設定されたイベントは削除してください。

備考 1. トレース開始イベント／トレース終了イベントのいずれかが有効状態で設定されると，イベント パネル上の無条件トレース・イベントのチェックが自動的に外れ，設定されたトレース開始イベント成立からトレース終了イベント成立までのトレース・データが収集されます。

備考 2. トレース開始イベントが不要な場合は，未設定でもかまいません。未設定の場合，プログラム実行開始がトレース・データの収集開始となります。同様にトレース終了イベントが未設定の場合，プログラム実行停止がトレース・データの収集停止となります。

備考 3. イベントの設定状態によりイベント・マークは異なります（「2.17.1 設定状態（有効／無効）を変更する」参照）。
また，すでにイベントが設定されている箇所で，新たにイベントを設定した場合は，複数のイベントが設定されていることを示すイベント・マーク（ ）が表示されます。

備考 4.　【シミュレータ】
トレース開始イベント／トレース終了イベントのいずれかが有効状態で設定されると，プロパティ パネルの［デバッグ・ツール設定］タブ上の［トレース］カテゴリ内［トレース機能を使用する］プロパティの指定を自動的に［はい］に変更し，トレース機能が有効化されます。

2.13.3.2 複数のイベントを組み合わせる【E1】【E20】【EZ Emulator】

トレース開始イベントに対して 2 つ以上のイベントを設定した場合，トレース開始イベントが次のような組み合わせ条件を満たした時点でトレース・データの収集を開始させることができます。
また，トレース終了イベントに対して 2 つ以上のイベントを設定した場合，トレース終了イベントのいずれかひとつでも条件が成立した時点でトレース・データの収集を終了します。

表 2.13　トレース開始イベントに対する組み合わせ条件

組み合わせ条件	説明
OR	設定されたトレース開始イベントの内，いずれかひとつでも条件が成立した時点でトレース・データの収集を開始します。
AND	時間軸に関係なく，設定された全てのトレース開始イベントの条件が成立した時点でトレース・データの収集を開始します。
シーケンシャル	指定した順序で設定されたトレース開始イベントの条件が成立するとトレース・データの収集を開始します。 また，トレース開始イベント内のイベントを 1 点のみリセットイベント (R イベント) として登録することができます。 登録されたイベントが成立した場合，その時点まで成立していた他のトレース開始イベントの条件成立は全てクリアされます。

なお，設定したトレース・イベントを編集する場合は，［表示］メニュー→［イベント］の選択でオープンするイベント パネル上において，トレース・イベントを選択したのち，コンテキスト・メニューの「条件の編集 ...」をクリックしてください。
この操作によりオープンするダイアログで編集します。
ダイアログによる編集の詳細については，「2.17.4.3 イベントの組み合わせ条件を編集する【E1】【E20】【EZ Emulator】」を参照してください。

注意　イベントの組み合わせに関しては，「(1) 組み合わせ条件の編集」の注意も参照してください。

2.13.3.3 プログラムを実行する

プログラムを実行します（「2.9 プログラムの実行」参照）。
トレース開始イベント／トレース終了イベントに設定されている条件が成立した時点で，トレース・データの収集を開始／終了します。
なお，収集したトレース・データの確認方法についての詳細は，「2.13.6 実行履歴を表示する」を参照してください。

2.13.3.4 トレース開始イベント／トレース終了イベントを編集する

設定したトレース開始イベント／トレース終了イベントを編集する場合は，［表示］メニュー→［イベント］の選択でオープンするイベント パネル上において，トレースの詳細情報に表示されたトレース開始／終了イベント（実行系およびアクセス系）を選択したのち，コンテキスト・メニューの「条件の編集 ...」をクリックしてください。
この操作によりオープンするダイアログで編集します。
実行系イベントダイアログによる編集の詳細については，「2.17.4.1 実行系イベントを編集する」を参照してください。
また，アクセス系イベントダイアログによる編集の詳細については，「2.17.4.2 アクセス系イベントを編集する」を参照してください。

2.13.3.5 トレース開始イベント／トレース終了イベントを削除する

設定したトレース開始イベント／トレース終了イベントを削除する場合は，イベント・エリア上のイベント・マークを右クリックすることで表示されるコンテキスト・メニューの［イベント削除］を選択します。

また，［表示］メニュー→［イベント］の選択でオープンする イベント パネル上において，トレースの詳細情報に表示されたトレース開始／終了イベント（実行系およびアクセス系）を選択したのち，ツールバーの ボタンをクリックする操作でも削除することができます。（「2.17.5 イベントを削除する」参照）。

注意　イベント・エリア上のイベント・マークからトレース開始／終了イベントのいずれかを削除した場合，対応した全てのイベント・マークが削除されます。

2.13.4 条件を満たしたときのみの実行履歴を収集する

ある条件を満たしたときのみプログラムの実行履歴を収集することができます。

(1) 変数 /I/O レジスタへのアクセスが発生したとき
ポイント・トレース・イベントを設定することにより，任意の変数，または I/O レジスタに対し，指定したアクセスがあった場合にのみ，その情報をトレース・データとして収集します。
次のいずれかの操作により，ポイント・トレース・イベントを設定してください。

注意 1.　ポイント・トレース・イベントの設定に関しては（有効イベント数の制限など），「2.17.7 イベント設定に関する留意事項」も参照してください。

注意 2.　【シミュレータ】
ストリング操作命令，および積和演算命令は，最初と最後のデータアクセスのみがイベントチェック対象となります。

備考 1.　DMAC（Direct Memory Access Controller）／ DTC（Data Transfer Controller）によるアクセスは対象となりません。

備考 2.　【シミュレータ】
ポイント・トレース・イベントのいずれかが有効状態で設定されると，プロパティ パネルの［デバッグ・ツール設定］タブ上の［トレース］カテゴリ内 [トレース機能を使用する] プロパティの指定を自動的に [はい] に変更し，トレース機能が有効化されます。

備考 3.　ポイント・トレース・イベントとトレース開始／終了イベントを同時に設定した場合は，デバッグ・ツールにより動作が異なります。

- 【E1】【E20】【EZ Emulator】
  トレース開始／終了イベントで設定した区間内でポイント・トレース・イベントの条件に一致したデータのみをトレース・データとして収集します。
- 【シミュレータ】
  トレース開始／終了イベントで設定した区間内の実行履歴，およびトレース開始／終了イベントで設定した区間とは関係なくポイント・トレース・イベントの条件に一致したデータをトレース・データとして収集します。

(a) ソース・テキスト／逆アセンブル・テキスト上の変数 /I/O レジスタへのアクセスの場合
操作は，ソース・テキスト／逆アセンブル・テキストを表示しているエディタ パネル／逆アセンブル パネル上で行います。
対象となる変数，または I/O レジスタを選択したのち，指定するアクセス種別にしたがって，コンテキスト・メニューより次の操作を行います。ただし，対象となる変数は，グローバル変数／関数内スタティック変数／ファイル内スタティック変数のみとなります。
なお，この操作を行うことにより，対象変数 /I/O レジスタにポイント・トレース・イベントが設定されたとみなされ，イベント パネルで管理されます（「2.17 イベントの管理」参照）。

アクセス種別	操作方法
リード	［トレース設定］→［値をトレースに記録（読み込み時）］を選択します。
ライト	［トレース設定］→［値をトレースに記録（書き込み時）］を選択します。
リード／ライト	［トレース設定］→［値をトレースに記録（読み書き時）］を選択します。

備考　カレント・スコープ内の変数が対象となります。

(b) 登録したウォッチ式へのアクセスの場合
操作は，ウォッチ パネル上で行います。
対象となるウォッチ式を選択したのち，コンテキスト・メニューより次の操作を行います（「2.11.6 ウォッチ式を表示／変更する」参照）。
ただし，対象となるウォッチ式は，グローバル変数／関数内スタティック変数／ファイル内スタティック変数 /I/O レジスタのみとなります。

なお，この操作を行うことにより，対象ウォッチ式にポイント・トレース・イベントが設定されたとみなされ，イベント パネルで管理されます（「2.17 イベントの管理」参照）。

アクセス種別	操作方法
リード	［トレース出力］→［値をトレースに記録（読み込み時）］を選択します。
ライト	［トレース出力］→［値をトレースに記録（書き込み時）］を選択します。
リード／ライト	［トレース出力］→［値をトレースに記録（読み書き時）］を選択します。

備考　カレント・スコープ内のウォッチ式が対象となります。
カレント・スコープ以外のウォッチ式を対象とする場合は，スコープ指定したウォッチ式を選択してください。

ポイント・トレース・イベントの設定が完了したのち，プログラムを実行します（「2.9 プログラムの実行」参照）。
プログラム実行中，設定したポイント・トレース・イベントの条件が満たされた場合，その情報がトレース・データとして収集されます。トレース・データの確認方法についての詳細は，「2.13.6 実行履歴を表示する」を参照してください。
なお，ポイント・トレース・イベントとして設定したトレース・イベント（アクセス系）を編集する場合は，［表示］メニュー→［イベント］の選択でオープンするイベント パネルにおいて，編集したいポイント・トレースの詳細情報に表示されたトレース・イベント（アクセス系）を選択したのち，コンテキスト・メニューの [条件の編集 ...] をクリックします（「2.17.4.2 アクセス系イベントを編集する」参照）。
また，ポイント・トレース・イベントとして設定したトレース・イベント（アクセス系）を削除する場合は，［表示］メニュー→［イベント］の選択でオープンするイベント パネルにおいて，削除したいポイント・トレース・イベントの詳細情報に表示されたトレース・イベント（アクセス系）を選択したのち，同パネルのツールバーの ボタンをクリックします（「2.17 イベントの管理」参照）。

2.13.5 実行履歴の収集を停止／再開する【E20】【シミュレータ】

プログラム実行中に実行履歴の収集を一時的に停止，および再開することができます。

注意 1. プログラム実行中にトレース機能を停止／再開する場合，リアルタイム性を損なう可能性があります。

注意 2. トレース開始イベント，またはトレース終了イベントを設定した場合，この機能は使用できません。

注意 3. 【E1】【E20【RX200 シリーズ】】【EZ Emulator】
プログラム実行中にトレース機能を停止／再開する機能はサポートしていません。

注意 4. 【E20(JTAG)【RX600 シリーズ】】
プロパティ パネルの［デバッグ・ツール設定］タブにおける［トレース］カテゴリ内の [外部トレース出力] プロパティで [外部出力しない] を選択した場合，この機能は使用できません。使用する場合は，[CPU 実行優先] または [トレース出力優先] を選択してください。

注意 5. 【E20(JTAG)【RX600 シリーズ】】
プロパティ パネルの［デバッグ・ツール設定］タブにおける［トレース］カテゴリ内の [トレース機能の用途] プロパティで [リアルタイム RAM モニタ] を選択した場合，この機能は使用できません。使用する場合は [トレース] を選択してください。

注意 6. アクション・イベントを設定した場合，この機能は使用できません。

2.13.5.1 実行履歴の収集を一時的に停止する場合

プログラム実行中にトレース パネル上のコンテキスト・メニュー→［トレース停止］を選択することにより，トレース・データの収集を一時的に停止することができます。
プログラムを停止せずにトレース機能のみを停止させ，その時点までのトレース・データを確認する場合などに使用します。

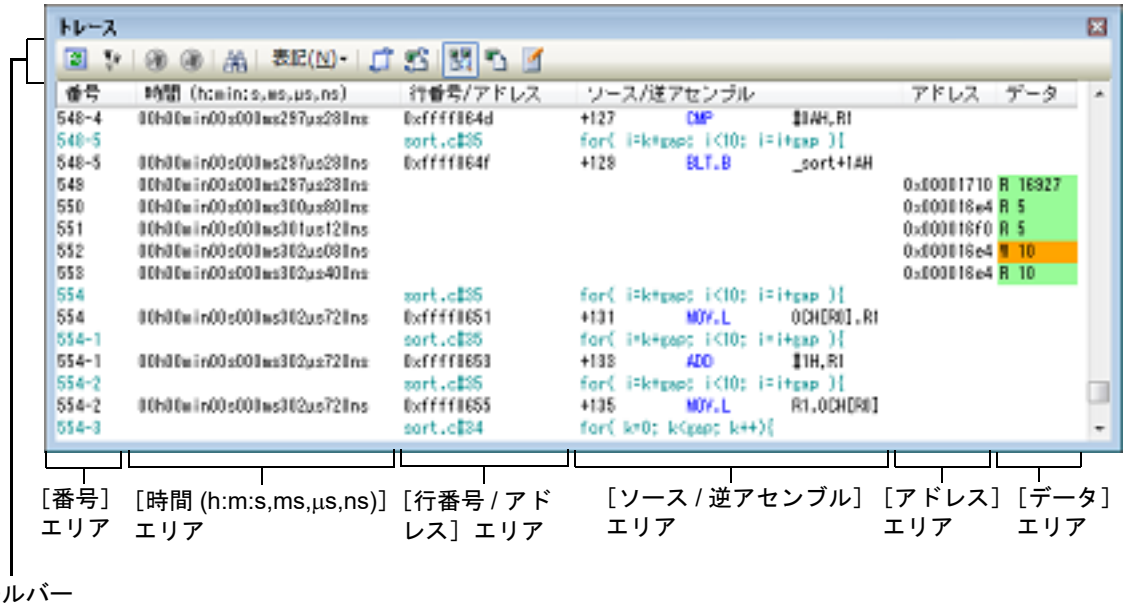
2.13.5.2 実行履歴の収集を再開する場合

プログラム実行中にトレース機能を停止した場合，トレース パネル上のコンテキスト・メニュー→［トレース開始］を選択することにより，再度トレース・データの収集を開始することができます。
なお，再開前に収集したトレース パネルのトレース・データは一度クリアされます。

2.13.6 実行履歴を表示する

収集したトレース・データの表示は，次のトレース パネルで行います。
［表示］メニュー→［トレース］を選択してください。
トレース・データは，デフォルトで逆アセンブル・テキストとソース・テキストを混合して表示しますが，表示モードを選択することにより，そのどちらか一方のみを表示させることもできます。
なお，各エリアの見方，および機能についての詳細は，トレース パネルの項を参照してください。

図 2.143　　トレース・データの表示（トレース パネル）



ここでは，次の操作方法について説明します。

2.13.6.1 表示モードを変更する
2.13.6.2 値の表示形式を変更する
2.13.6.3 他のパネルと連動させる

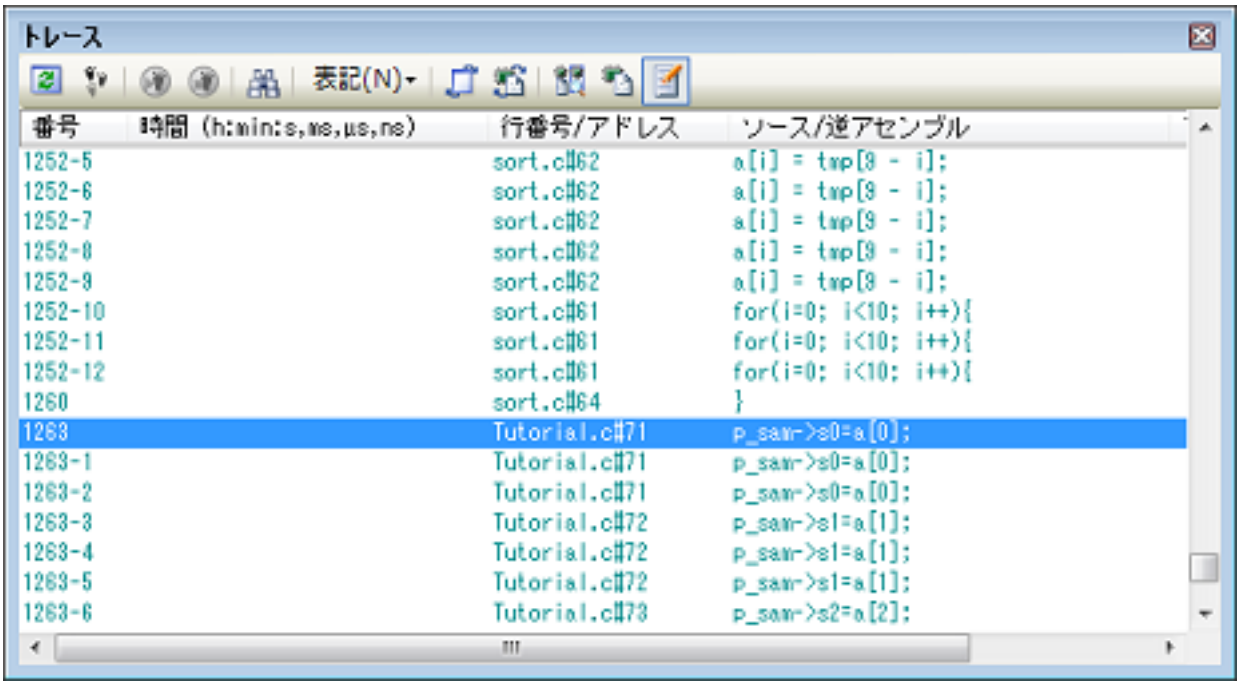
2.13.6.1 表示モードを変更する

次のツールバーのボタンをクリックすることで，用途に応じて表示モードを変更することができます。
ただし，トレース機能が動作中の場合は無効となります。

表 2.14　　トレース パネルの表示モード

ボタン	表示モード	表示内容
	混合表示モード	命令（逆アセンブル）／ラベル名／ソース・テキスト（対応するソース行）／ポイント・トレース結果を表示します（デフォルト）。
	逆アセンブル表示モード	命令（逆アセンブル）／ラベル名／ポイント・トレース結果を表示します。
	ソース表示モード	ソース・テキスト（対応するソース行）を表示します。 ただし，デバッグ情報が存在しない箇所を実行した場合は，“デバッグ情報のない区間の実行”と表示します。

図 2.144 ソース表示モードの例（トレース パネル）



2.13.6.2 値の表示形式を変更する

［行番号 / アドレス］エリア／［アドレス］エリア／［データ］エリアの表示形式は，ツールバーの次のボタンにより，自由に変更することができます。
ただし，プログラム実行中は，ボタンは無効となります。

表記	値の表示形式を変更する次のボタンを表示します。
	このパネル上の値を 16 進数で表示します（デフォルト）。
	このパネル上の値を 10 進数で表示します。
	このパネル上の値を 8 進数で表示します。
	このパネル上の値を 2 進数で表示します。

2.13.6.3 他のパネルと連動させる

現在選択している行のアドレスをポインタとして，他のパネルで対応箇所を連動して表示させることができます（フォーカスの移動は行いません）。
ツールバーの ボタンをクリックすると，エディタ パネルと連動開始します。
ツールバーの ボタンをクリックすると，逆アセンブル パネルと連動開始します。
なお，再度クリックすることにより，連動を中止します。

備考 コンテキスト・メニューの［ソースへジャンプ］／［逆アセンブルへジャンプ］を選択することにより，現在選択している行のアドレスに対応するソース行／アドレスにキャレットを移動した状態で，エディタ パネル／逆アセンブル パネルがオープンします（フォーカスの移動を行います）。

2.13.7 トレース・メモリをクリアする

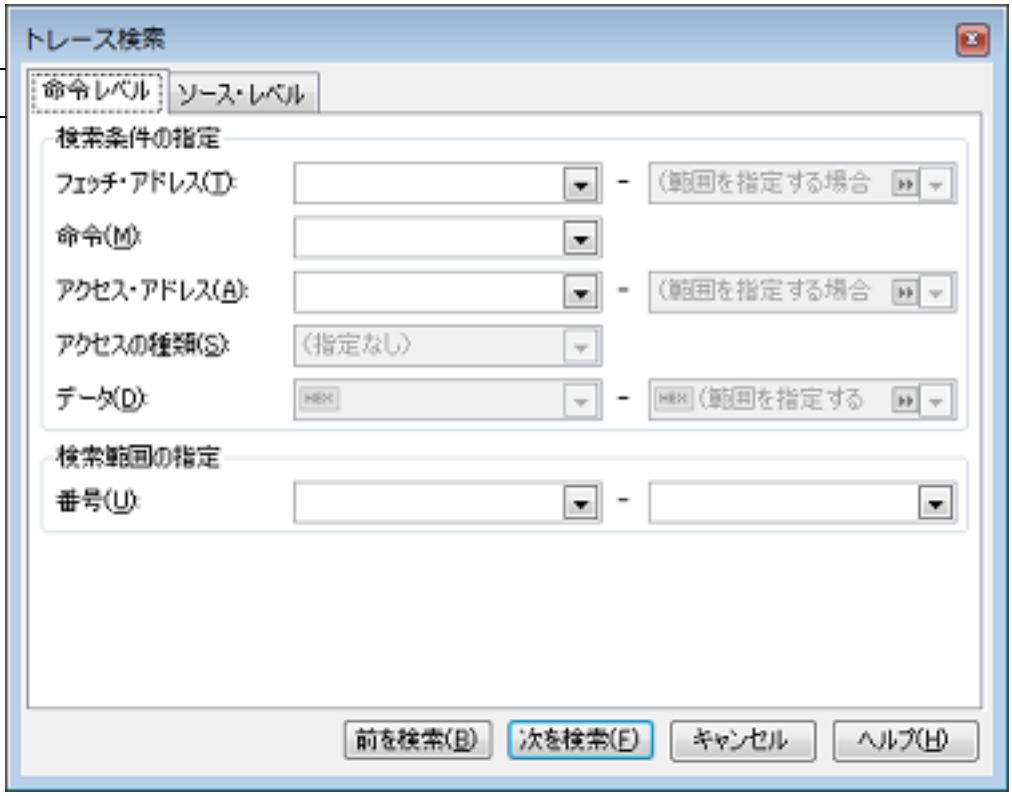
収集したトレース・データの内容をクリアするには，ツールバーの ボタンをクリックします。
ただし，トレース機能が動作中は無効となります。

備考 【シミュレータ】
プロパティ パネルの［デバッグ・ツール設定］タブ上の［トレース］カテゴリ内の［実行前にトレース・メモリをクリアする］プロパティにおいて，［はい］を指定している場合は，プログラムの実行ごとにトレース・メモリがクリアされます。

2.13.8 トレース・データを検索する

収集したトレース・データの検索は，ツールバーの ボタンをクリックすることによりオープンするトレース検索ダイアログにより行います（プログラム実行中は無効）。
このダイアログにおいて，次の操作を行ってください。
なお，タブ選択エリア上のタブを選択することにより，命令レベル，またはソース・レベルでトレース・データを検索することができます。
ただし，命令レベルでトレース・データの検索を行う場合は，トレース パネルを混合表示モード，または逆アセンブル表示モードで表示している必要があります。
また，ソース・レベルで検索を行う場合は，混合表示モード，またはソース表示モードで表示している必要があります。

図 2.145　トレース・データの検索（トレース検索 ダイアログ）



ここでは，次の操作方法について説明します。

2.13.8.1 命令レベルで検索する
2.13.8.2 ソース・レベルで検索する

2.13.8.1 命令レベルで検索する

命令レベルでトレース・データを検索します。
［命令レベル］タブを選択したのち，次の手順で操作を行ってください。

図 2.146 命令レベルでのトレース・データの検索

(1) ［フェッチ・アドレス］の指定
検索条件として必要な場合，フェッチ・アドレスを指定します。
アドレス式をテキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
フェッチ・アドレスの指定は範囲で指定することができます。この場合は，左右両方のテキスト・ボックスにアドレス値を指定することにより範囲を指定します。
右側のテキスト・ボックスが空欄，または“（範囲を指定する場合に入力）”の場合は，左側のテキスト・ボックスに指定された固定アドレスで検索を行います。
なお，マイクロコントローラのアドレス空間よりも大きいアドレス値が指定された場合は，上位のアドレス値をマスクして扱います。
また，32 ビットで表現できる値より大きいアドレス値を指定することはできません。

(2) ［命令］の指定
検索条件として必要な場合，命令の文字列を指定します。
ここで指定した文字列をトレース パネルの［ソース / 逆アセンブル］エリア内より検索します。
命令をテキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
なお，検索の際は，大文字／小文字は区別せず，部分一致も検索の対象とします。

(3) ［アクセス・アドレス］の指定
検索条件として必要な場合，アクセス・アドレスを指定します。
16 進数でアドレス値をテキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
アクセス・アドレスの指定は範囲で指定することができます。この場合は，左右両方のテキスト・ボックスにアドレス値を指定することにより範囲を指定します。
右側のテキスト・ボックスが空欄，または“（範囲を指定する場合に入力）”の場合は，左側のテキスト・ボックスに指定された固定アドレスで検索を行います。
なお，マイクロコントローラのアドレス空間よりも大きいアドレス値が指定された場合は，上位のアドレス値をマスクして扱います。
また，32 ビットで表現できる値より大きいアドレス値を指定することはできません。

(4) ［アクセスの種類］の指定
この項目は［アクセス・アドレス］の指定が指定された場合のみ有効となります。
アクセスの種類（リード／ライト，リード，ライト，ベクタ・リード，DMA）をドロップダウン・リストより選択します。
アクセスの種類を限定しない場合は，［(指定なし)］を選択してください。

注意　【E1】【E20】【EZ Emulator】
トレース パネル上で収集可能なアクセスの種類は，リード，ライトのみです。そのため，ドロップダウン・リストよりリード／ライト，ベクタ・リードおよび DMA を選択しないでください。
【シミュレータ】
トレース パネル上で収集可能なアクセスの種類は，リード，ライト，ベクタ・リードです。そのため，ドロップダウン・リストよりリード／ライトおよび DMA を選択しないでください。

(5)　［データ］の指定
この項目は［アクセス・アドレス］の指定が指定された場合のみ有効となります。
アクセスした数値を指定します。
16 進数値をテキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
数値の指定は範囲で指定することができます。この場合は，左右両方のテキスト・ボックスにデータを指定することにより範囲を指定します。
右側のテキスト・ボックスが空欄，または“（範囲を指定する場合に入力）”の場合は，左側のテキスト・ボックスに指定された固定数値で検索を行います。

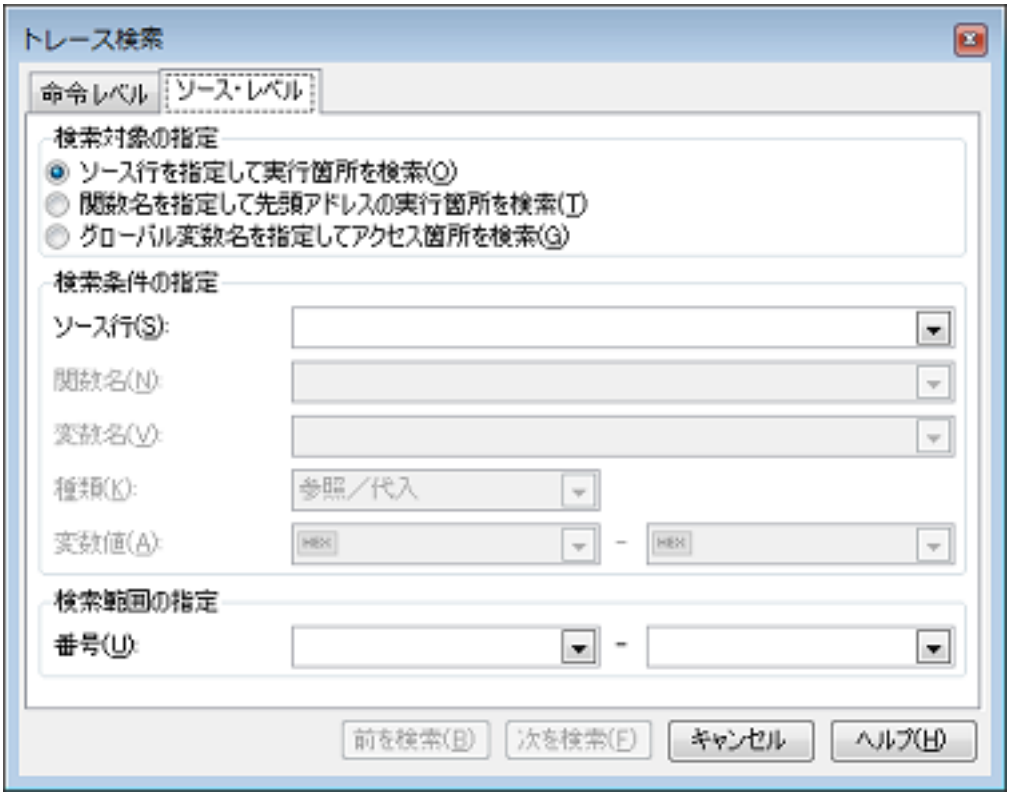
(6)　［番号］の指定
検索するトレース・データの範囲を，トレース パネルの［番号］エリアに表示されている番号で指定します。
左右のテキスト・ボックスに，それぞれ開始番号と終了番号を指定します（デフォルトでは，“0”～“*最終番号*”が指定されます）。
10 進数で番号をテキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
左側のテキスト・ボックスが空欄の場合は，“0”の指定として扱われます。
右側のテキスト・ボックスが空欄の場合は，最終番号の指定として扱われます。

(7)　［前を検索］／［次を検索］ボタンのクリック
［前を検索］ボタンをクリックすると，番号の小さい方向に検索を行い，検索結果箇所をトレース パネル上で選択状態にします。
［次を検索］ボタンをクリックすると，番号の大きい方向に検索を行い，検索結果箇所をトレース パネル上で選択状態にします。

2.13.8.2 ソース・レベルで検索する

ソース・レベルでトレース・データを検索します。
［ソース・レベル］タブを選択してください。

図 2.147　　ソース・レベルでのトレース・データの検索



(1)　ソース行を指定して検索する場合（デフォルト）
［検索対象の指定］エリアにおいて，“ ソース行を指定して実行箇所を検索 ” を選択したのち，次の操作を行います。

- ［ソース行］の指定
ここで指定した文字列をトレース パネルの［行番号 / アドレス］エリア内より検索します。
検索するソース行に含まれる文字列を，テキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
なお，検索の際は，大文字／小文字は区別せず，部分一致も検索の対象とします。

例 1.　　main.c#40

例 2.　　main.c

例 3.　　main

- ［番号］の指定
検索するトレース・データの範囲を，トレース パネルの［番号］エリアに表示されている番号で指定します。
左右のテキスト・ボックスに，それぞれ開始番号と終了番号を指定します（デフォルトでは，“0” ～ “ *最終番号* ” が指定されます）。
10 進数で番号をテキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
左側のテキスト・ボックスが空欄の場合は，“0” の指定として扱われます。
右側のテキスト・ボックスが空欄の場合は，最終番号の指定として扱われます。

- ［前を検索］／［次を検索］ボタンのクリック
［前を検索］ボタンをクリックすると，番号の小さい方向に検索を行い，検索結果箇所をトレース パネル上で選択状態にします。
［次を検索］ボタンをクリックすると，番号の大きい方向に検索を行い，検索結果箇所をトレース パネル上で選択状態にします。

(2)　関数名を指定して検索する場合
［検索対象の指定］エリアにおいて，“ 関数名を指定して先頭アドレスの実行箇所を検索 ” を選択したのち，次の操作を行います。

- ［関数名］の指定
検索する関数名を，テキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。

なお，検索の際は，大文字／小文字を区別し，完全一致のみを検索の対象とします。

- ［番号］の指定
検索するトレース・データの範囲を，トレース パネルの［番号］エリアに表示されている番号で指定します。
左右のテキスト・ボックスに，それぞれ開始番号と終了番号を指定します（デフォルトでは，“0” ～ “ *最終番号* ” が指定されます）。
10 進数で番号をテキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
左側のテキスト・ボックスが空欄の場合は，“0” の指定として扱われます。
右側のテキスト・ボックスが空欄の場合は，最終番号の指定として扱われます。

- ［前を検索］／［次を検索］ボタンのクリック
［前を検索］ボタンをクリックすると，番号の小さい方向に検索を行い，検索結果箇所をトレース パネル上で選択状態にします。
［次を検索］ボタンをクリックすると，番号の大きい方向に検索を行い，検索結果箇所をトレース パネル上で選択状態にします。

(3) グローバル変数名を指定して検索する場合
［検索対象の指定］エリアにおいて，“ グローバル変数名を指定してアクセス箇所を検索 ” を選択したのち，次の操作を行います。

- ［変数名］の指定
検索する変数名を，テキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
なお，検索の際は，大文字／小文字を区別し，完全一致のみを検索の対象とします。

- ［種類］の指定
アクセスの種類（参照／代入（デフォルト），参照，代入）をドロップダウン・リストより選択します。

- ［変数値］の指定
アクセスした変数値をテキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
変数値の指定は範囲で指定することができます。この場合は，左右両方のテキスト・ボックスに変数値を指定することにより範囲を指定します。
右側のテキスト・ボックスが空欄の場合は，左側のテキスト・ボックスに指定された固定変数値でアクセス箇所を検索を行います。

- ［番号］の指定
検索するトレース・データの範囲を，トレース パネルの［番号］エリアに表示されている番号で指定します。
左右のテキスト・ボックスに，それぞれ開始番号と終了番号を指定します（デフォルトでは，“0” ～ “ *最終番号* ” が指定されます）。
10 進数で番号をテキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
左側のテキスト・ボックスが空欄の場合は，“0” の指定として扱われます。
右側のテキスト・ボックスが空欄の場合は，最終番号の指定として扱われます。

- ［前を検索］／［次を検索］ボタンのクリック
［前を検索］ボタンをクリックすると，番号の小さい方向に検索を行い，検索結果箇所をトレース パネル上で選択状態にします。
［次を検索］ボタンをクリックすると，番号の大きい方向に検索を行い，検索結果箇所をトレース パネル上で選択状態にします。

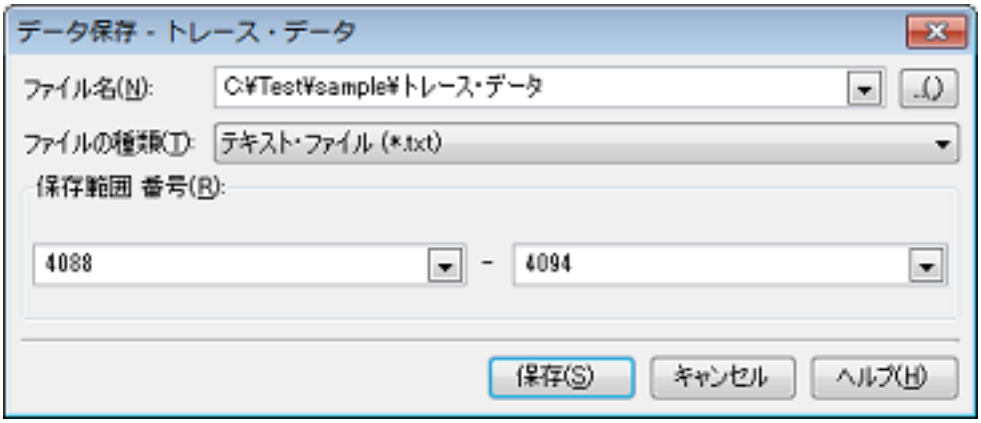
2.13.9 実行履歴の表示内容を保存する

収集したトレース・データの内容を範囲指定して，テキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存することができます。ファイルに保存する際は，デバッグ・ツールから最新の情報を取得し，このパネル上での表示形式に従ったデータで保存します。

［ファイル］メニュー→［名前を付けてトレース・データを保存 ...］を選択すると，次のデータ保存 ダイアログがオープンします。

このダイアログにおいて，次の手順で操作を行ってください。

図 2.148 実行履歴の保存（データ保存 ダイアログ）



(1) ［ファイル名］の指定
保存するファイル名を指定します。
テキスト・ボックスに直接入力するか（最大指定文字数：259 文字），またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
また，［...］ボタンをクリックすることでオープンするデータ保存ファイルを選択 ダイアログにより，ファイルを選択することもできます。

(2) ［ファイルの種類］の指定
保存するファイルの形式を次のドロップダウン・リストにより選択します。
選択できるファイルの形式は次のとおりです。

リスト表示	形式
テキスト・ファイル (*.txt)	テキスト形式（デフォルト）
CSV(カンマ区切り) (*.csv)	CSV 形式[注]

注 各データを “,” で区切り保存します。
なお，データ内に “,” が含まれている際の不正形式を避けるため，各データを “""”（ダブルクォーテーション）で括り出力します。

(3) ［保存範囲 番号］の指定
ファイルに保存する範囲を “ 開始トレース番号 ” と “ 終了トレース番号 ” で指定します。
それぞれのテキスト・ボックスに 10 進数の数値を直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
なお，すべてのトレース・データを保存する場合は，左側のドロップダウン・リストにおいて，［すべてのトレース・データ］を選択してください（右側のテキスト・ボックスは無効）。
パネル上で範囲選択している場合は，デフォルトでその選択範囲がテキスト・ボックスに指定されます。範囲選択していない場合は，現在のパネルの表示範囲が指定されます。

(4) ［保存］ボタンのクリック
指定したファイルに，指定した形式でトレース・データを保存します。

図 2.149 トレース・データ保存の際の出力イメージ

番号	時間	行番号 / アドレス	ソース / 逆アセンブル	アドレス	データ
番号	*時間*	*行番号 / アドレス*	*ソース / 逆アセンブル*	*アドレス*	*データ*
:	:	:	:	:	:

2.14 実行時間の計測

この節では，プログラムの実行時間の計測方法について説明します。

注意 1. 【EZ Emulator【RX200 シリーズ】】
端子リセットを入れた場合，EZ Emulator からの再実行時に実行時間はクリアされます。

注意 2. 【E1/E20/EZ Emulator【RX600 シリーズ】】
割り込み / 例外発生回数と RTE/RTFI 命令の発行回数が同じで，それぞれ対応付けが取れる場合であることを前提としています。もし，割り込み / 例外発生回数と RTE/RTFI 命令の発生回数が異なる場合，それぞれの対応付けが取れないため，正しく処理サイクルを計測することができません。
なお，ネストは 16 回分までしか保持できません。したがって，17 回以上の割り込み・例外でネストが発生した場合，正しく計測ができません。

2.14.1 タイマ計測動作の設定をする【E1】【E20】【EZ Emulator】

タイマ計測機能を行うためには，あらかじめ次のタイマ計測に関する設定を行う必要があります。
設定は，プロパティ パネルの［接続用設定］タブ上の［クロック］カテゴリ内の［動作周波数 [MHz] 【E1】【E20】【EZ Emulator】］プロパティ，およびプロパティ パネルの［デバッグ・ツール設定］タブ上の［タイマ］【E1】【E20】【EZ Emulator】カテゴリ内で行います。

注意 【RX100 シリーズ】
タイマ機能は未サポートとなります。

図 2.150 ［クロック］カテゴリ

クロック	
メイン・クロック・ソース	EXTAL
メイン・クロック周波数[MHz]	12.5000
動作周波数[MHz]	25.0000
内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する	いいえ

図 2.151 ［タイマ］カテゴリ【RX600 シリーズ】

タイマ	
カウンタを64ビットで使用する	いいえ

(1) [動作周波数 [MHz]]
カウント値を時間に変換する際に使用する動作周波数を指定します。
0.0001 ～ 999.9999（単位：MHz）の範囲の数値を直接入力で指定します。

注意 1. タイマ計測イベントの計測結果は，このプロパティで入力した動作周波数とカウント値より算出されます。そのため，実行中に動作周波数を切り替えるプログラムを用いた場合，正しい計測結果は表示されません。

注意 2. タイマ測定中にリセットが入った場合は，リセット期間中も計測が行われます。但し，計測用のクロック設定も初期化されるため正しい計測が行えない場合があります。

(2) [カウンタを 64 ビットで使用する]【RX600 シリーズ】
計測カウンタを 32 ビット ×2 本で使用するか，64 ビット ×1 本で使用するかを指定します。
[はい] を指定した場合，64 ビットの計測カウンタが使用可能ですが，計測区間は 1 区間となります。

注意 【RX200 シリーズ】
計測カウンタは 24 ビット ×1 本のため，このプロパティは表示されません。

2.14.2 実行開始から停止までの実行時間を計測する

デバッグ・ツールには，プログラムの実行開始から実行停止までの実行時間（Run-Beak 時間）を測定する機能があらかじめ用意されています。
したがって，プログラムの実行を開始することにより，自動的に実行時間の計測を行います。
計測結果は，次のいずれかの方法で確認することができます。
ここでは，次の確認方法について説明します。

2.14.2.1 ステータス・バーでの確認
2.14.2.2 イベント パネルでの確認

注意 1. Run-Break タイマ・イベントの計測結果は，計測開始からプログラム実行までの時間，および，プログラム実行停止から計測停止までの時間を含みます。また，計測クロックソースのクロック精度誤差を含む参考値です。

注意 2. ステータス・バーおよびイベント パネルに表示される実行時間は，プログラム実行から停止の間の時間です。なお，ステップ・イン，ステップ・オーバー，およびリターン・アウトでは計測できません。
【E1】【E20】
100us 未満の値は切り捨てられます。

注意 3. EZ Emulator では，Run-Break タイマ機能は未サポートとなります。

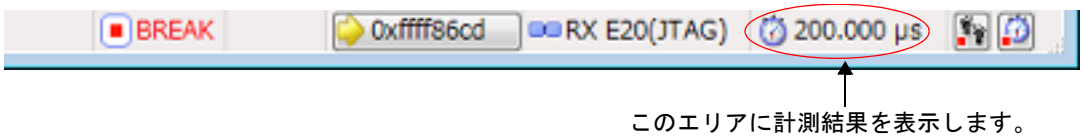
備考 この機能は，デバッグ・ツールにデフォルトで設定されているビルトイン・イベントの 1 つである Run-Break タイマ・イベントにより動作します。
Run-Break タイマ・イベントは，常に有効状態に設定されているイベントです（変更不可）。

2.14.2.1 ステータス・バーでの確認

プログラムの実行停止後，メイン・ウインドウ上のステータス・バーにおいて計測結果を表示します（計測をしていない場合は " 未計測 " と表示）。

備考 EZ Emulator では Run-Break タイマ機能をサポートしていないため，常に " 未計測 " と表示されます。

図 2.152 Run-Break タイマ・イベントの測定結果例（ステータス・バー）



2.14.2.2 イベント パネルでの確認

プログラムの実行停止後，［表示］メニュー→［イベント］の選択によりオープンするイベント パネル上において，Run-Break タイマ・イベントとして計測結果を表示します。

図 2.153 Run-Break タイマ・イベントの測定結果例（イベント パネル）【E1】【E20】

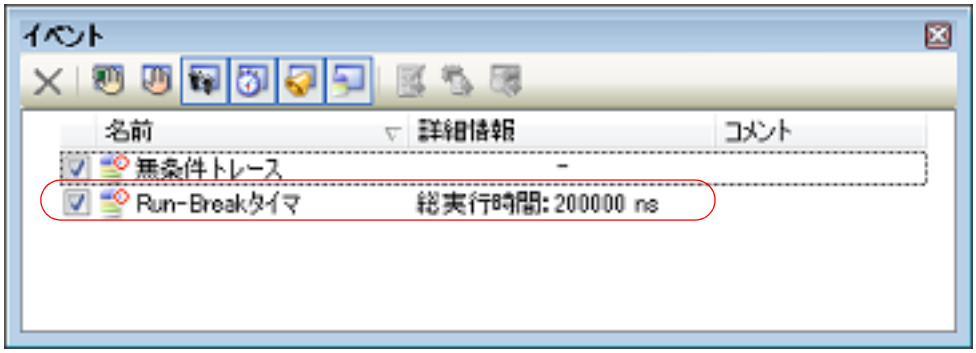
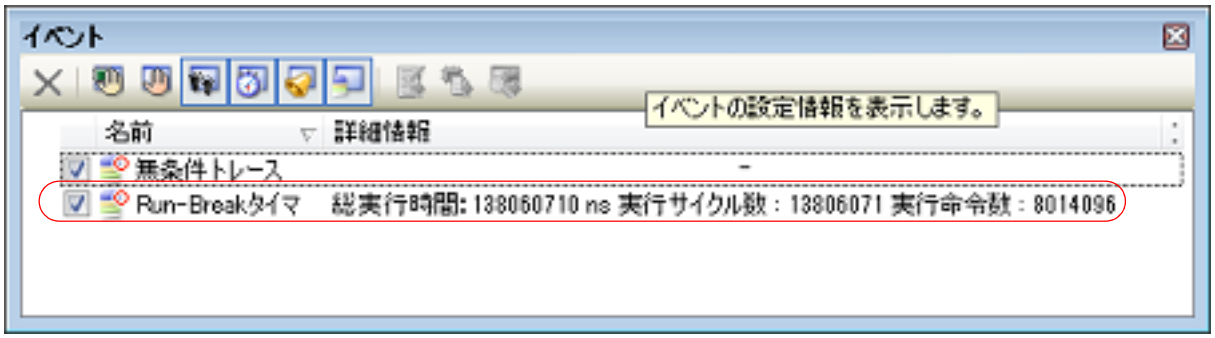


図 2.154 Run-Break タイマ・イベントの測定結果例（イベント パネル）【シミュレータ】



2.14.3 任意区間の実行時間を計測する

タイマ計測イベント（タイマ開始イベント／タイマ終了イベント）を設定することにより，プログラムの実行過程において，任意の区間の実行時間を計測することができます。
この機能を使用するためには，次の手順で操作を行います。

2.14.3.1 タイマ開始イベント／タイマ終了イベントを設定する
2.14.3.2 プログラムを実行する
2.14.3.3 タイマ計測イベントを編集する【E1】【E20】【EZ Emulator】
2.14.3.4 タイマ開始イベント／タイマ終了イベントを編集する
2.14.3.5 タイマ開始イベント／タイマ終了イベントを削除する

注意 1. タイマ計測イベントの設定に関しては（有効イベント数の制限など），「2.17.7 イベント設定に関する留意事項」も参照してください。

注意 2. 【E1(Serial) / E20(Serial) / EZ Emulator【RX200 シリーズ】】
トレース パネルのタイム・スタンプ情報は , タイマ計測用カウンタを用いて実現しています。そのため , イベント パネルにてタイマ計測イベントを設定した場合 , 期待したタイム・スタンプ情報が表示されません。

注意 3. 【E1】【E20】【EZ Emulator】
タイマ開始／終了イベントの追加／削除，またはタイマ計測イベントの編集を行った場合，次回計測時の計測結果は累積されません。
【シミュレータ】
タイマ開始／終了イベントの追加／削除／編集を行った場合，次回計測時の計測結果は累積されません。

2.14.3.1 タイマ開始イベント／タイマ終了イベントを設定する

エディタ パネル／逆アセンブル パネル／ウォッチ パネルにおいて，タイマ計測を開始／終了するイベントを設定します。

(1) タイマ開始イベントの設定方法

- エディタ パネル／逆アセンブル パネルから設定する場合
  タイマ計測を開始したい行／アドレス[注1]にキャレットを移動したのち，コンテキスト・メニューの［タイマ設定］→［実行時にタイマ開始］→ [タイマ＊[注2]に設定] を選択します。
  タイマ開始イベントが，キャレット位置の行／アドレスに対応する先頭アドレスの命令に対して設定されます。
  タイマ計測を開始したい変数にキャレットを移動したのち，コンテキスト・メニューの［タイマ設定］→［読み書き時にタイマ開始］→ [タイマ＊[注2]に設定] を選択します。
  タイマ開始イベントが，キャレット位置の変数に対して設定されます。

- ウォッチ パネルから設定する場合
  タイマ計測を開始したいウォッチ式[注3]にキャレットを移動したのち，コンテキスト・メニューの [タイマ設定] → [読み書き時にタイマ開始] → [タイマ＊[注2]に設定] を選択します。
  タイマ開始イベントが，ウォッチ パネルのウォッチ式に対して設定されます。

(2) タイマ終了イベントの設定方法

- エディタ パネル／逆アセンブル パネルから設定する場合
  タイマ計測を終了したい行／アドレス[注1]にキャレットを移動したのち，コンテキスト・メニューの［タイマ設定］→［実行時にタイマ終了］→ [タイマ＊[注2]に設定] を選択します。
  タイマ終了イベントが，キャレット位置の行／アドレスに対応する先頭アドレスの命令に対して設定されます。
  タイマ計測を終了したい変数にキャレットを移動したのち，コンテキスト・メニューの［タイマ設定］→［読み書き時にタイマ終了］→ [タイマ＊[注2]に設定] を選択します。
  タイマ終了イベントが，キャレット位置の変数に対して設定されます。

- ウォッチ パネルから設定する場合
  タイマ計測を終了したいウォッチ式[注3]にキャレットを移動したのち，コンテキスト・メニューの [タイマ設定] → [読み書き時にタイマ終了] → [タイマ＊[注2]に設定] を選択します。
  タイマ終了イベントが，ウォッチ パネルのウォッチ式に対して設定されます。

注 1.　タイマ開始イベント／タイマ終了イベントは，アドレス表示がない行に設定することはできません。

注 2.　【E1】【E20】【EZ Emulator】
[タイマ＊に設定] メニューの ＊には，タイマ計測区間の番号としてチャネル番号が表示されます。タイマ開始イベント／タイマ終了イベントを 1 区間で設定する場合，同じチャネル番号を選択してください。
なお，選択可能なチャネル番号は，マイクロコントローラおよびプロパティ パネルの［デバッグ・ツール設定］タブ上の［タイマ］【E1】【E20】【EZ Emulator】カテゴリの次のプロパティ設定により異なります。

マイクロコントローラ	[カウンタを 64bit で使用する] プロパティ	機能
RX600 シリーズ	いいえ	[タイマ 1 に設定] および [タイマ 2 に設定] の 2 区間 (32bit) から指定可能
	はい	[タイマ 1 に設定] の 1 区間 (64bit) のみを指定可能
RX200 シリーズ	―	[タイマ 1 に設定] の 1 区間 (24bit) のみを指定可能

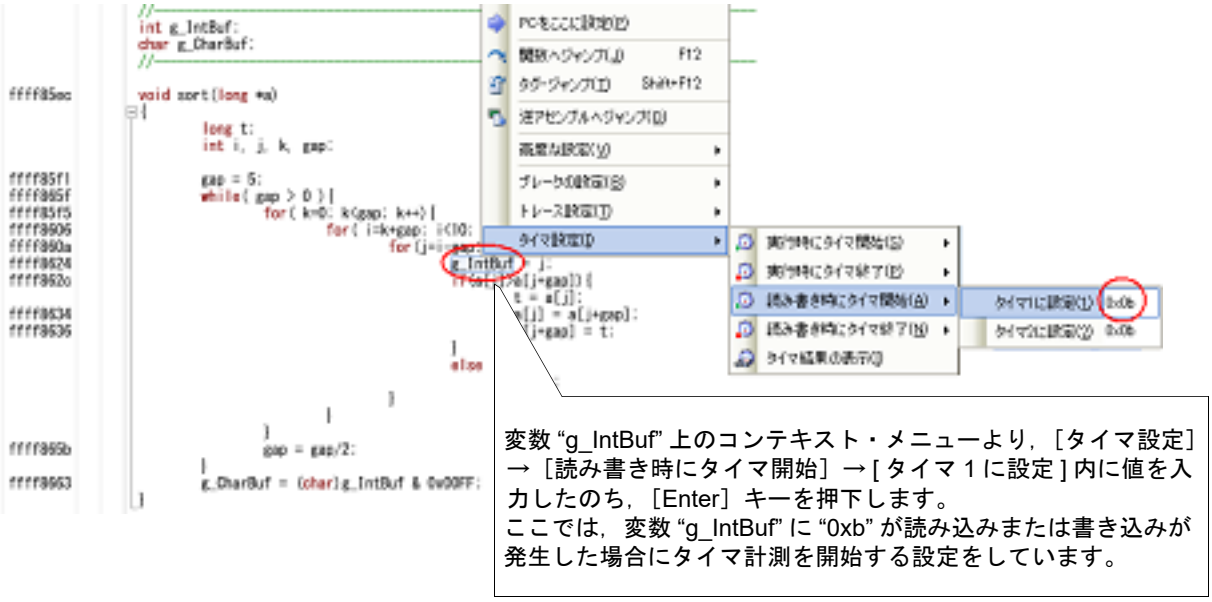
【シミュレータ】
1 区間固定で区間指定がないため，[タイマ＊に設定] メニューは表示されません。

注 3.　対象となるウォッチ式は，グローバル変数／関数内スタティック変数／ファイル内スタティック変数／IO レジスタのみとなります。

注意 1.　【E1】【E20】【EZ Emulator】
タイマ開始／終了イベントのどちらか片方の設定でもタイマ計測できます。タイマ開始イベントのみを設定した場合，プログラムの実行停止でタイマ計測は終了します。また，タイマ終了イベントのみを設定した場合，プログラムの実行開始でタイマ計測を開始します。
【シミュレータ】
タイマ開始／終了イベントの両方の設定でタイマ計測が有効になります。そのため，タイマ開始／終了イベントのどちらか片方の設定ではタイマ計測できません。

注意 2.　【シミュレータ】
タイマ開始／終了イベントはそれぞれ 1 点のみ設定可能です。複数のイベントを組み合わせることはできません。

図 2.155　ソース・テキスト上の変数に対するタイマ開始イベント（アクセス系）の設定例【E1】【E20】【EZ Emulator】



変数 “g_IntBuf” 上のコンテキスト・メニューより，［タイマ設定］→［読み書き時にタイマ開始］→ [タイマ 1 に設定] 内に値を入力したのち，［Enter］キーを押下します。
ここでは，変数 “g_IntBuf” に “0xb” が読み込みまたは書き込みが発生した場合にタイマ計測を開始する設定をしています。

図 2.156　　ソース・テキスト上の変数に対するタイマ開始イベント（アクセス系）の設定例【シミュレータ】

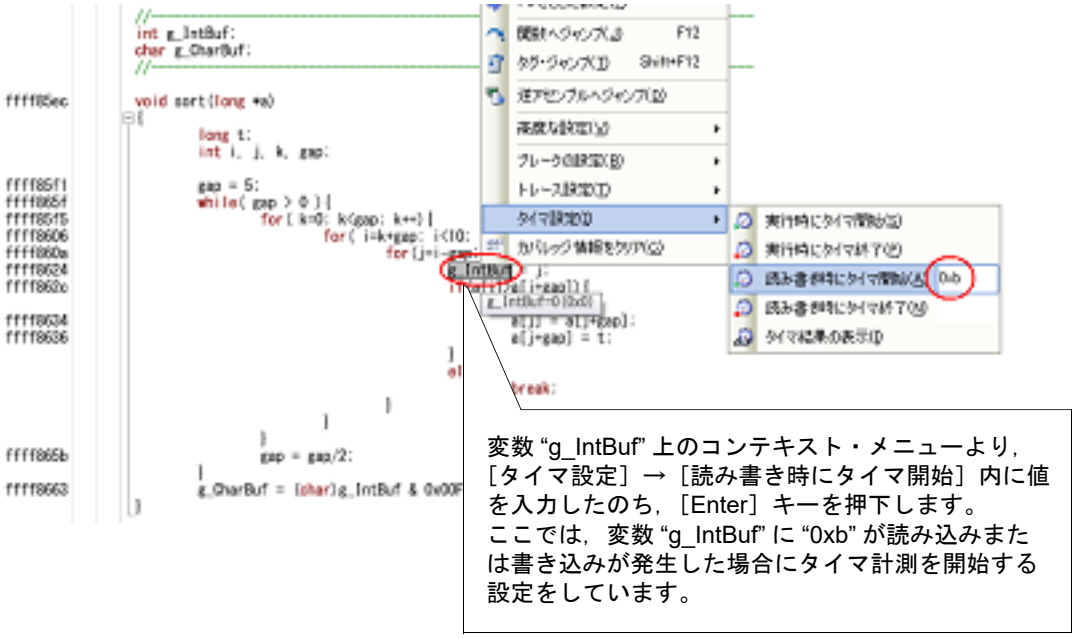


図 2.157　　ウォッチ式に対するタイマ開始イベント（アクセス系）の設定例【E1】【E20】【EZ Emulator】

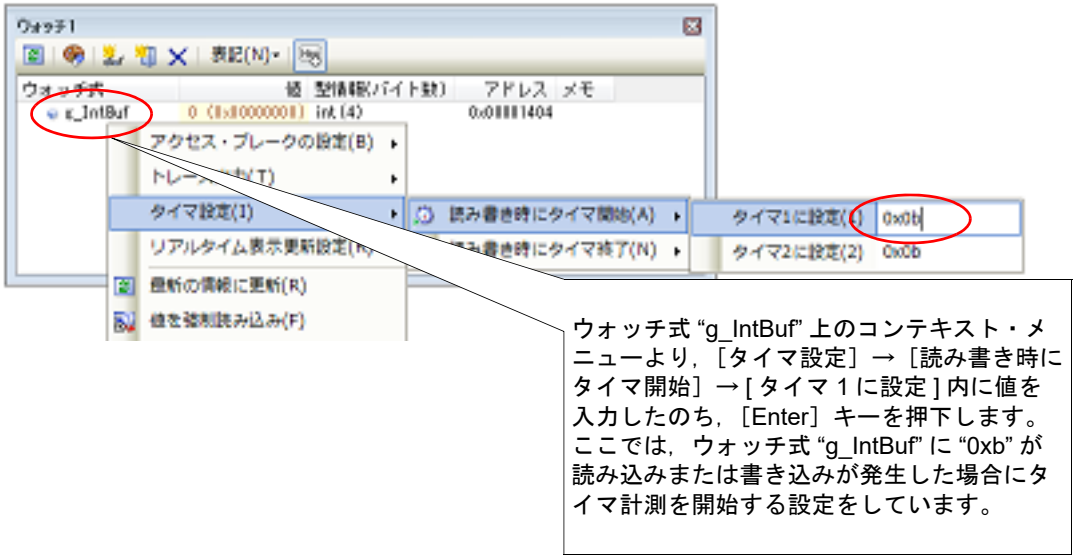
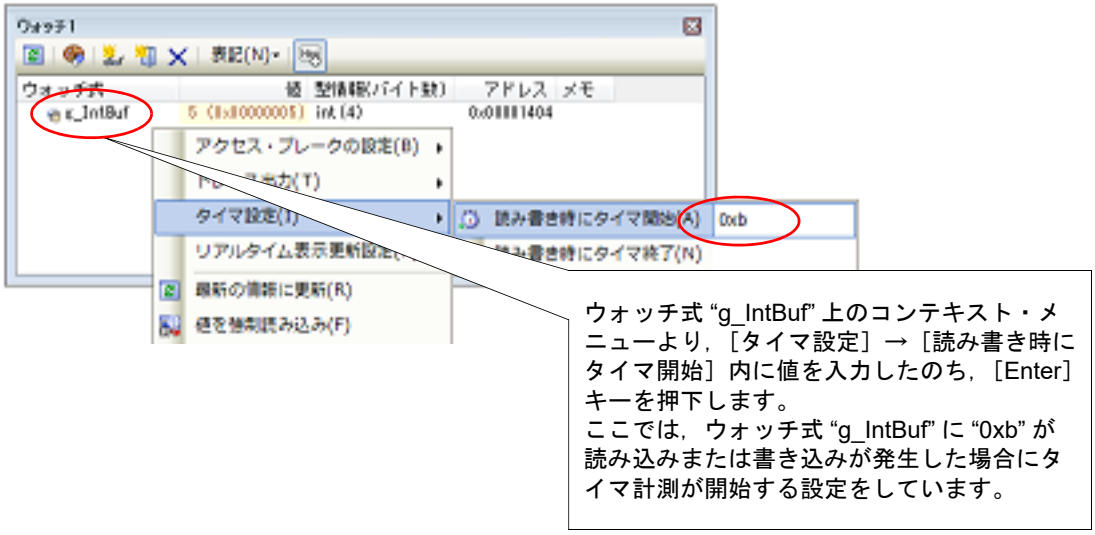


図 2.158　ウォッチ式に対するタイマ開始イベント（アクセス系）の設定例【シミュレータ】



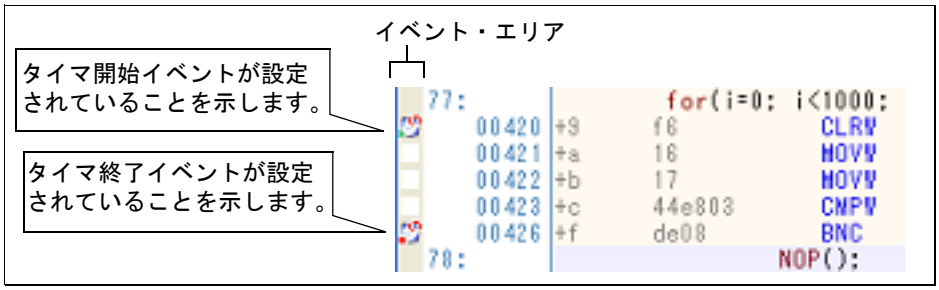
タイマ開始イベント／タイマ終了イベントが設定されると，該当行／アドレス／ウォッチ式のイベント・エリアに次のイベント・マークが表示されます。

また，イベント パネル上において，タイマ計測イベントとして 1 つにまとめて管理されます（タイマ計測イベント項目の “+” マークをクリックすることにより，設定したタイマ開始イベント／タイマ終了イベントの情報を確認することができます）。

表 2.15　タイマ開始イベント／タイマ終了のイベント・マーク

種別	イベント・マーク
タイマ開始	
タイマ終了	

図 2.159　タイマ開始イベント／タイマ終了イベントの設定例（逆アセンブル パネルの場合）



注意　【シミュレータ】
タイマ開始／終了イベントが同じ命令実行で成立した場合，計測状態によって次の動作となります。

- タイマ計測が開始されている場合
  タイマ計測を停止し，計測結果に反映します。
- タイマ計測が開始されていない場合
  タイマ計測を行いません。

備考　イベントの設定状態によりイベント・マークは異なります（「2.17.1 設定状態（有効／無効）を変更する」参照）。
また，すでにイベントが設定されている箇所で，新たにイベントを設定した場合は，複数のイベントが設定されていることを示すイベント・マーク（　）が表示されます。

2.14.3.2 プログラムを実行する

プログラムを実行します（「2.9 プログラムの実行」参照）。
タイマ開始イベント／タイマ終了イベントが設定されている命令が実行された際に，タイマ計測を開始／終了します。

計測結果は，プログラムの実行停止後，［表示］メニュー→［イベント］の選択によりオープンするイベント パネルにおいて，タイマ計測イベントとして次のように確認することができます。
なお，このタイマ計測イベントは，タイマ開始イベント，またはタイマ終了イベントのいずれかが設定された場合に，イベント パネルでのみ表示されるイベント種別です。

図 2.160　タイマ計測イベント（タイマ開始イベント／タイマ終了イベント）の測定結果例【E1】【E20】

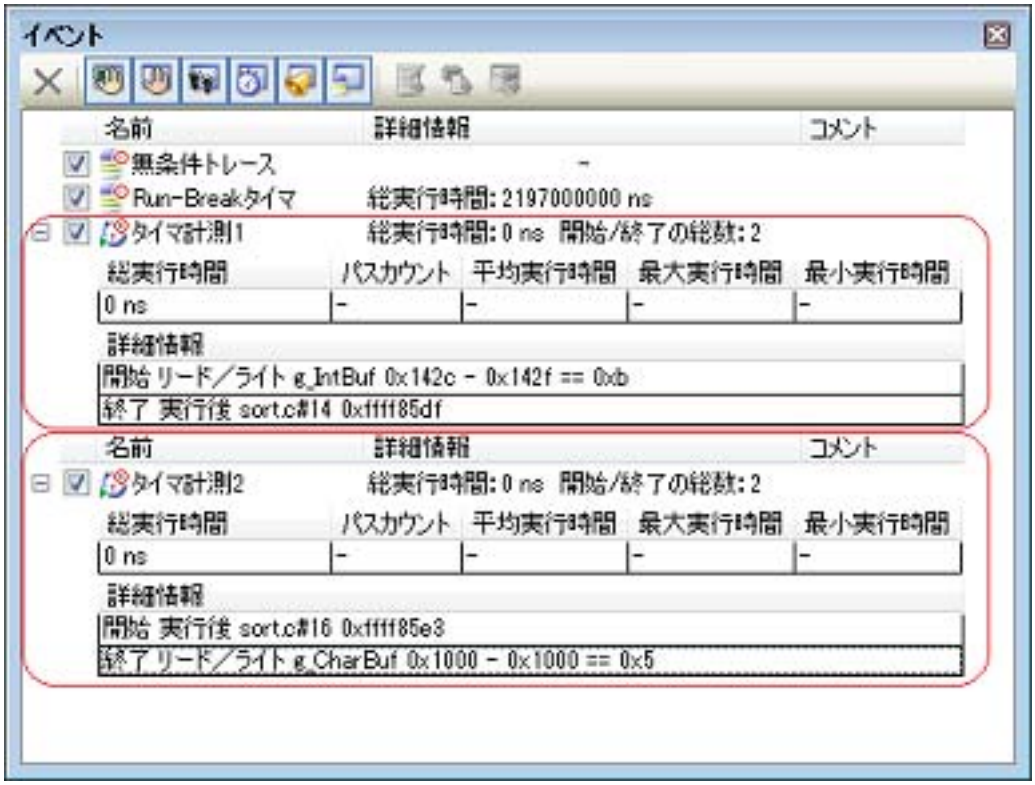


図 2.161　タイマ計測イベント（タイマ開始イベント／タイマ終了イベント）の測定結果例【EZ Emulator】

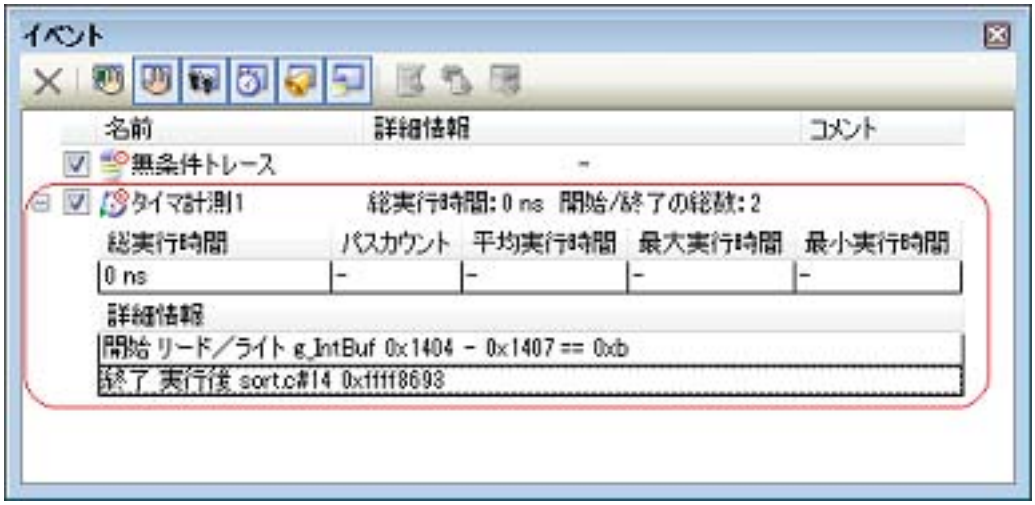


図 2.162 タイマ計測イベント（タイマ開始イベント／タイマ終了イベント）の測定結果例【シミュレータ】

2.14.3.3 タイマ計測イベントを編集する【E1】【E20】【EZ Emulator】

開始イベント，および終了イベントを設定したタイマ計測イベントの詳細情報は，タイマ計測詳細設定ダイアログ【E1】【E20】【EZ Emulator】にて編集します。このダイアログは，イベント パネル上にて編集したいタイマ計測イベントを選択してコンテキスト・メニューの [条件の編集 ...] を選択することで表示します。

(1) 計測項目を編集する場合
タイマ計測詳細設定ダイアログでは，計測項目として次から指定できます。

計測項目	機能
経過サイクル注	指定区間の経過サイクル数 (ICLK) を計測します。
経過サイクル（スーパバイザモードサイクル）	スーパバイザモードにいる間の経過サイクル数 (ICLK) を計測します。
例外・割り込み処理サイクル	例外処理を含む割り込み処理に要したサイクル数 (ICLK) を計測します。
例外処理サイクル	例外処理に要したサイクル数 (ICLK) を計測します。
割り込み処理サイクル	割り込み処理に要したサイクル数 (ICLK) を計測します。
有効命令発行回数	実行完了した命令数を計測します。
例外・割り込み受け付け回数	例外処理を含む割り込みを受け付けた回数を計測します。
例外受付回数	例外処理が発生した回数を計測します。
割り込み受付回数	割り込みを受け付けた回数を計測します。

注 【RX200 シリーズ】
計測項目には，[経過サイクル] のみが表示されます。

備考 1. 経過サイクル，例外・割り込み処理サイクル，例外処理サイクル，割り込み処理サイクルを指定した場合は，総実行時間を計測します。

備考 2. 有効命令発行回数，例外・割り込み受付回数，例外受付回数，割り込み受付回数を指定した場合は，通過回数（パスカウント）を計測します。

備考 3. 平均実行時間，最大実行時間，最小実行時間は測定できません。

(2) 一度だけ計測を編集する場合
[一度だけ計測] プロパティを [はい] と指定することで指定区間を一度だけ計測してタイマ計測を終了します。
なお，指定区間を通過した合計値を計測する場合，[いいえ] を指定してください。

備考　【RX600 シリーズ】
以下の計測実施中にタイマ開始イベントおよびタイマ終了イベントが成立しても，一度も計測条件が成立しなかった場合，計測結果は表示されません。
- 有効命令発行回数
- 例外・割り込み受け付け回数
- 例外受付回数
- 割り込み受付回数

注意　【RX600 シリーズ】
[はい] を指定した場合，タイマ開始イベントは，イベントが 2 回成立するとタイマ終了イベントが成立していなくても計測を停止します。

2.14.3.4 タイマ開始イベント／タイマ終了イベントを編集する

設定したタイマ開始イベント／タイマ終了イベントを編集する場合は，［表示］メニュー→［イベント］の選択でオープンするイベント パネル上において，タイマ計測の詳細情報に表示されたタイマ開始／終了イベント（実行系およびアクセス系）を選択したのち，コンテキスト・メニューの「条件の編集 ...」をクリックしてください。
この操作によりオープンするダイアログで編集します。
実行系イベントダイアログによる編集の詳細については，「2.17.4.1 実行系イベントを編集する」を参照してください。
また，アクセス系イベントダイアログによる編集の詳細については，「2.17.4.2 アクセス系イベントを編集する」を参照してください。

2.14.3.5 タイマ開始イベント／タイマ終了イベントを削除する

設定したタイマ開始イベント／タイマ終了イベントを削除する場合は，イベント・エリア上のイベント・マークを右クリックすることで表示されるコンテキスト・メニューの［イベント削除］を選択します。
また，［表示］メニュー→［イベント］の選択でオープンする イベント パネル上において，タイマ計測の詳細情報に表示されたタイマ開始／終了イベント（実行系およびアクセス系）を選択したのち，ツールバーの ボタンをクリックする操作でも削除することができます。（「2.17.5 イベントを削除する」参照）。

注意　イベント・エリア上のイベント・マークからタイマ開始／終了イベントのいずれかを削除した場合，対応した全てのイベント・マークが削除されます。

2.14.4 測定可能時間の範囲

Run-Break タイマ・イベント（「2.14.2 実行開始から停止までの実行時間を計測する」参照），またはタイマ計測イベント（「2.14.3 任意区間の実行時間を計測する」参照）によるタイマ計測の測定可能時間の範囲は次のとおりです。

表 2.16　測定可能時間の範囲

デバッグ・ツール	Run-Break タイマ・イベント		タイマ計測イベント	
E1(Serial)/E1(JTAG)/ E20(Serial)/E20(JTAG) 【RX600 シリーズ】	最小	100 マイクロ秒	最小	―
	最大	約 72 時間	最大	CPU クロック周波数に依存 32 ビットカウンタ ×2 チャネル or 64 ビットカウンタ ×1 チャネル オーバフロー検出あり
E1(Serial)/E1(JTAG)/ E20(Serial)/E20(JTAG) 【RX200 シリーズ】	最小	100 マイクロ秒	最小	―
	最大	約 72 時間	最大	CPU クロック周波数に依存 24 ビットカウンタ ×1 チャネル オーバフロー検出あり
シミュレータ	CPU クロック周波数に依存 64 ビットカウンタ オーバフロー検出あり		CPU クロック周波数に依存 64 ビットカウンタ オーバフロー検出あり	

2.15 カバレッジの測定【シミュレータ】【E20【RX71M, RX64M グループ】】

この節では，カバレッジ機能を使用した，カバレッジ測定について説明します。
カバレッジ測定の方法にはいくつかの種類がありますが，CS+ では次の領域を対象に，ソース行／関数に対するフェッチ系のコード・カバレッジ測定（C0 カバレッジ），および変数に対するアクセス系のデータ・カバレッジ測定を行います。
カバレッジ測定の対象となる領域は次のとおりです。

表 2.17　　カバレッジ測定の対象領域

デバッグ・ツール	対象領域
シミュレータ	内蔵 ROM/RAM，エミュレーション ROM/RAM
E20【RX71M, RX64M グループ】	任意の領域（4M バイト単位で最大 4 箇所）

備考　　C0 カバレッジ：命令網羅率（ステートメント・カバレッジ）
たとえば，コード内のすべての命令（ステートメント）を 少なくとも 1 回は実行した場合，C0 = 100 % となります。

注意 1.　　E20【RX71M, RX64M グループ】では，トレース機能やリアルタイム RAM モニタ機能と同時に使用することはできません。

注意 2.　　E20【RX71M, RX64M グループ】では，データ・カバレッジは測定できません。

2.15.1 カバレッジ測定の設定をする

カバレッジ機能を使用するためには，あらかじめカバレッジ測定に関する設定を行う必要があります。
設定は，プロパティ パネルの［デバッグ・ツール設定］タブ上の［カバレッジ］【シミュレータ】，または［カバレッジ］【E20【RX71M, RX64M グループ】】カテゴリ内で行います。

図 2.163　　［カバレッジ］カテゴリ【シミュレータ】

図 2.164　　［カバレッジ］カテゴリ【E20【RX71M, RX64M グループ】】

(1)　［カバレッジ機能を使用する］
カバレッジ機能を使用するか否かをドロップダウン・リストにより指定します。
カバレッジ機能を使用する場合は［はい］を選択してください（デフォルトでは［いいえ］が指定されます）。

(2)　［カバレッジ結果を再利用する］
このプロパティは，［カバレッジ機能を使用する］プロパティにおいて［はい］を指定した場合のみ表示されます。
デバッグ・ツールと切断時に，現在取得しているコード・カバレッジ測定結果を自動保存し，次回デバッグ・ツールと接続した際に，保存した測定結果の内容を再現するか否かをドロップダウン・リストにより指定します。
前回取得したコード・カバレッジ測定結果の内容を再現する場合は，［はい］を選択してください（デフォルトでは［いいえ］が指定されます）。

(3)　［コード・カバレッジ機能を使用する］

コード・カバレッジ機能を使用するか否かをドロップダウン・リストにより指定します。
コード・カバレッジ機能を使用する場合は［はい］を選択してください（デフォルトでは［いいえ］が指定されます）。
デバッグ・ツールと切断中のみ指定することができます。

(4) ［カバレッジ計測方法］
このプロパティは，［コード・カバレッジ機能を使用する］プロパティにおいて［はい］を指定した場合のみ表示されます。
カバレッジデータの計測方法をドロップダウン・リストにより指定します。
ドロップダウン・リストには，次のカバレッジ計測方法が表示されます。

- CPU 実行優先 , カバレッジ計測優先

［CPU 実行優先］を指定した場合，データ出力時のロストによりカバレッジデータが失われる可能性があります。
［カバレッジ計測優先］を指定した場合，カバレッジデータ出力のために CPU 実行が停止し，プログラム実行のリアルタイム性が損なわれる可能性があります。

(5) ［カバレッジ計測範囲］
このプロパティは，［コード・カバレッジ機能を使用する］プロパティにおいて［はい］を指定した場合のみ表示されます。
カバレッジを計測する範囲を指定します。
カバレッジ計測範囲は 4M バイト単位で 4 つまで指定できます。
開始アドレスは，4M バイト単位の値を指定してください（入力値が 4M バイト単位の値でなかった場合は，自動的に補正されます）。

(6) ［カバレッジ結果を再利用する］
このプロパティは，［コード・カバレッジ機能を使用する］プロパティにおいて［はい］を指定した場合のみ表示されます。
デバッグ・ツールと切断時に，現在取得しているコード・カバレッジ測定結果を自動保存し，次回デバッグ・ツールと接続した際に，保存した測定結果の内容を再現するか否かをドロップダウン・リストにより指定します。

2.15.2 カバレッジ測定結果を表示する

プログラムの実行が開始すると自動的にカバレッジ測定が開始し，実行停止とともにカバレッジ測定も終了します。

(1) コード・カバレッジ率

(a) ソース行／逆アセンブル行に対するコード・カバレッジ率の表示
対象となるプログラムを表示しているエディタ パネル／逆アセンブル パネルで行われます。
各パネルでは，表 2.18 に示す計算方法で算出されたコード・カバレッジ率を基に，対象ソース・テキスト行／逆アセンブル結果行の背景色が表 2.19 のように色分け表示されます。
ただし，デバッグ・ツールと切断時，またはプログラム実行中は，結果の表示を行いません。
なお，取得したコード・カバレッジ測定結果は，エディタ パネル／逆アセンブル パネル上のコンテキスト・メニューの［カバレッジ情報のクリア］を選択することにより，すべてリセットすることができます（各パネル上の色分け表示もクリアされます）。

表 2.18　ソース行／逆アセンブル行に対するコード・カバレッジ率の計算方法

パネル	計算方法
エディタ パネル	“ ソース行と対応するアドレス範囲内で実行されたバイト数 ”÷“ ソース行と対応するアドレス範囲内の総バイト数 ”
逆アセンブル パネル	“ 逆アセンブル結果行と対応するアドレス範囲内で実行されたバイト数 ” ÷“ 逆アセンブル結果行と対応するアドレス範囲内の総バイト数 ”

表 2.19　コード・カバレッジ測定結果の表示色（デフォルト）

コード・カバレッジ率	背景色
100 %	*ソース・テキスト／逆アセンブル結果*
1 ～ 99 %	*ソース・テキスト／逆アセンブル結果*
0 %（未実行）	*ソース・テキスト／逆アセンブル結果*

備考 1. 各パネルにおけるコード・カバレッジ測定結果の表示更新は，プログラム停止ごとに自動的に行われます。

備考 2. 上記の背景色は，オプション ダイアログにおける［全般 - フォントと色］カテゴリの設定に依存します。

備考 3. 上記の背景表示は，対象領域外（「表 2.17 カバレッジ測定の対象領域」参照）の行に対しては行われません。

備考 4. ダウンロードしているロード・モジュールの更新日時より，現在オープンしているソース・ファイルの更新日時が新しい場合，エディタ パネルではコード・カバレッジ測定結果の表示は行われません。

図 2.165 コード・カバレッジ測定結果の表示例（エディタ パネル）

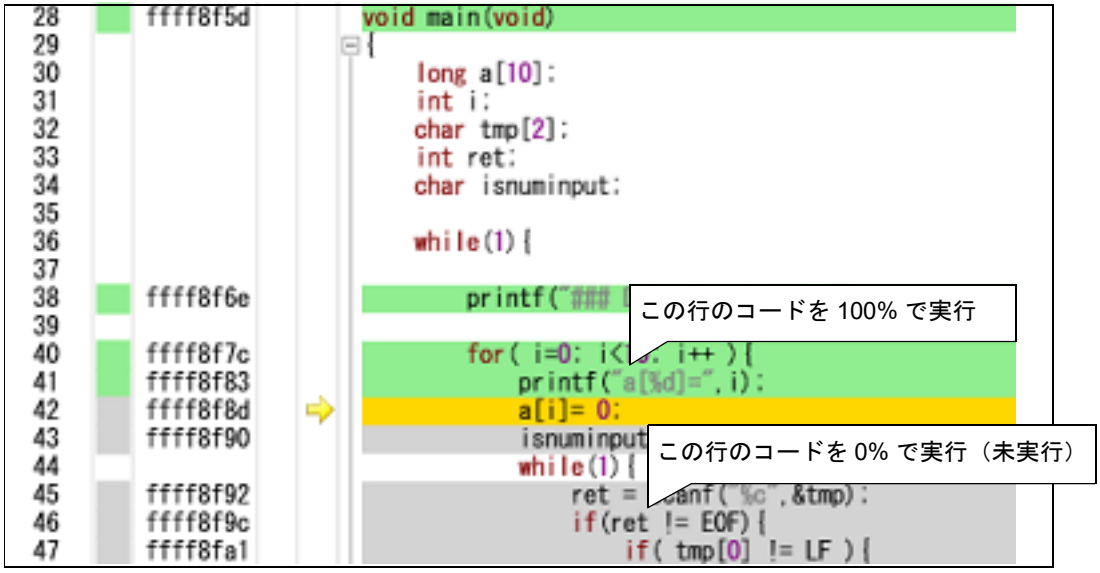
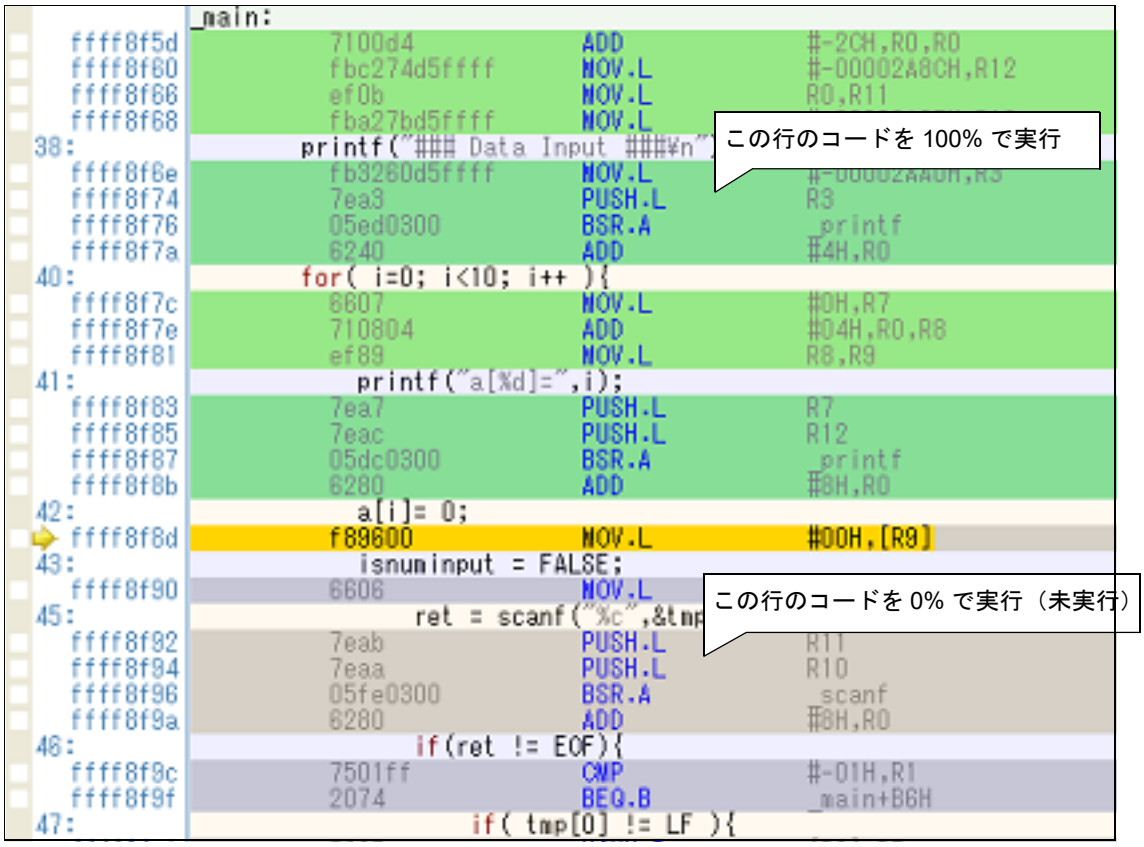


図 2.166 コード・カバレッジ測定結果の表示例（逆アセンブル パネル）



(b) 各関数に対するコード・カバレッジ率の表示
各関数に対するコード・カバレッジ率（関数の網羅率）は，解析ツールの関数一覧 パネル内［コード・カバレッジ [%]］項目で確認することができます。
“ 関数のコード・カバレッジ率 ” についての詳細は，「CS+ 統合開発環境 ユーザーズマニュアル 解析ツール編」を参照してください。

(2) データ・カバレッジ率【シミュレータ】
各変数に対するデータ・カバレッジ率は，解析ツールの変数一覧 パネル内［データ・カバレッジ [%]］項目で確認することができます。
“ 変数のデータ・カバレッジ率 ” についての詳細は，「CS+ 統合開発環境 ユーザーズマニュアル 解析ツール編」を参照してください。

2.16 プログラム内へのアクションの設定

この節では，プログラム内に，指定したアクションを設定する操作方法について説明します。

2.16.1 printf を挿入する

アクション・イベントの 1 つである Printf イベントを設定することにより，プログラムの実行を任意の箇所で一瞬停止させたのち，ソフトウエア処理により printf コマンドを実行させ，指定した変数式の値を出力 パネルに出力することができます。
この機能を使用するためには，次の手順で操作を行ってください。

2.16.1.1 Printf イベントを設定する
2.16.1.2 プログラムを実行する
2.16.1.3 出力結果を確認する
2.16.1.4 Printf イベントを編集する

注意 1. Printf イベントの設定に関しては（有効イベント数の制限など），「2.17.7 イベント設定に関する留意事項」も参照してください。

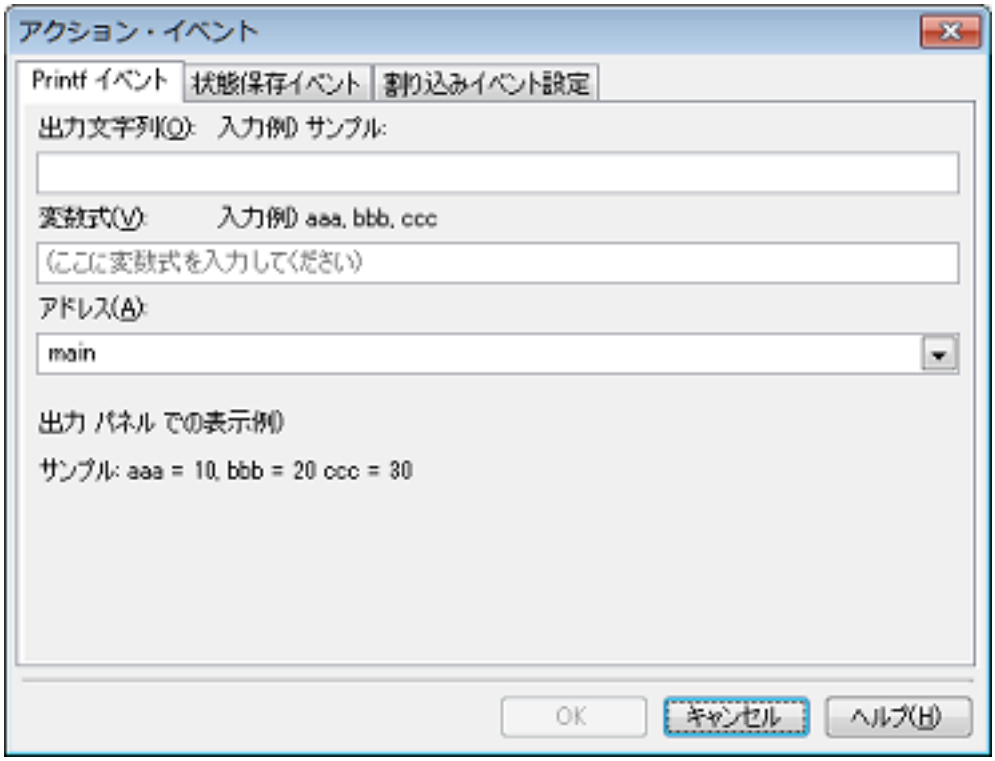
注意 2. ステップ実行中（／／），またはブレーク関連のイベントを無視した実行中（）に Printf イベントは発生しません。

2.16.1.1 Printf イベントを設定する

エディタ パネル／逆アセンブル パネル上で，printf コマンドを実行させたい箇所に Printf イベントを設定します。
エディタ パネル／逆アセンブル パネルにおいて，Printf イベントを設定したい行／アドレス[注]にキャレットを移動したのち，コンテキスト・メニューの［アクション・イベントの登録 ...］を選択すると，次のアクション・イベント ダイアログがオープンします。
このダイアログにおいて，次の操作を行ってください。

注　Printf イベントは，アドレス表示がない行に設定することはできません。

図 2.167　Printf イベントを設定する（アクション・イベント ダイアログ：[Printf イベント] タブ）



(1)　［出力文字列］の指定
出力 パネルに出力する際に付与する文字列をキーボードより直接入力で指定します。
なお，出力する文字列は，1 行分のみ入力可能です（空白可）。

(2) ［変数式］の指定
Printf イベントの対象となる変数式を指定します。
変数式は，テキスト・ボックスに直接入力で指定します（最大指定文字数：1024 文字）。
“,” で区切ることにより，1 つの Printf イベントとして 10 個までの変数式を指定することができます。
エディタ パネル／逆アセンブル パネルにおいて，変数式を選択した状態でこのダイアログをオープンした場合では，選択している変数式がデフォルトで表示されます。
なお，変数式として指定できる基本入力形式と，その際に Printf イベントとして出力される値についての詳細は，「表 A.14 変数式と出力される値の関係（Printf イベント）」を参照してください。

備考　このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます（「2.20.2 シンボル名の入力補完機能」参照）。

(3) ［アドレス］の指定
Printf イベントを指定するアドレスを指定します。
デフォルトで，現在の指定位置のアドレスを表示します。
編集する場合は，テキスト・ボックスにアドレス式を直接入力するか（最大指定文字数：1024 文字），またはドロップダウン・リストにより入力履歴項目（最大履歴個数：10 個）を選択します。

備考　このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます（「2.20.2 シンボル名の入力補完機能」参照）。

(4) ［OK］ボタンのクリック
ここで指定した Printf イベントをエディタ パネル／逆アセンブル パネル上のキャレット位置の行に設定します。
Printf イベントが設定されると，エディタ パネル／逆アセンブル パネルのイベント・エリアに マークが表示され，イベント パネルで管理されます（「2.17 イベントの管理」参照）。

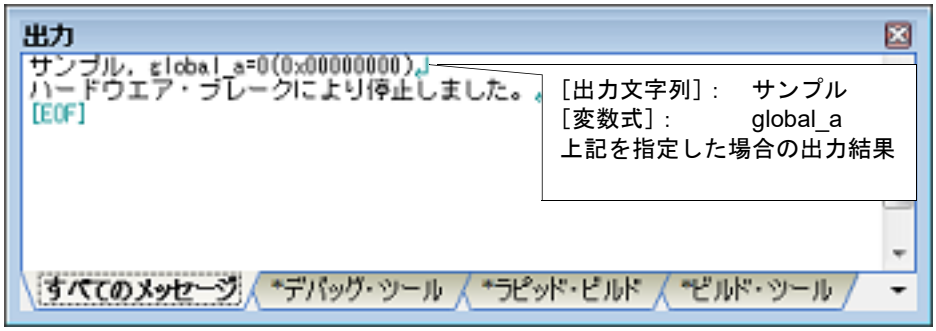
2.16.1.2 プログラムを実行する

プログラムを実行します（「2.9 プログラムの実行」参照）。
プログラムを実行することにより，Printf イベントを設定した箇所の命令実行直前でプログラムを一瞬停止し，指定した変数式の値を出力 パネルに出力します。

2.16.1.3 出力結果を確認する

出力 パネルの［デバッグ・ツール］タブでは，指定した変数式の値が次のように出力されます（「図 A.46 Printf イベントの出力結果フォーマット」参照）。

図 2.168　Printf イベントの出力結果例



2.16.1.4 Printf イベントを編集する

一度設定した Printf イベントを編集することができます。
編集を行う場合は，イベント パネルにおいて，編集対象の Printf イベントを選択したのち，コンテキスト・メニューの［条件の編集 ...］を選択します。オープンするアクション・イベント ダイアログにおいて，編集が必要な項目を編集したのち，［OK］ボタンをクリックします。

2.16.2 割り込みイベントを挿入する【シミュレータ】

アクション・イベントの 1 つである割り込みイベントを設定することにより，プログラムの実行中に任意の箇所で割り込みを発生させることができます。
この機能を使用するためには，次の手順で操作を行ってください。

2.16.2.1 割り込みイベントを設定する
2.16.2.2 プログラムを実行する
2.16.2.3 割り込みイベントを編集する

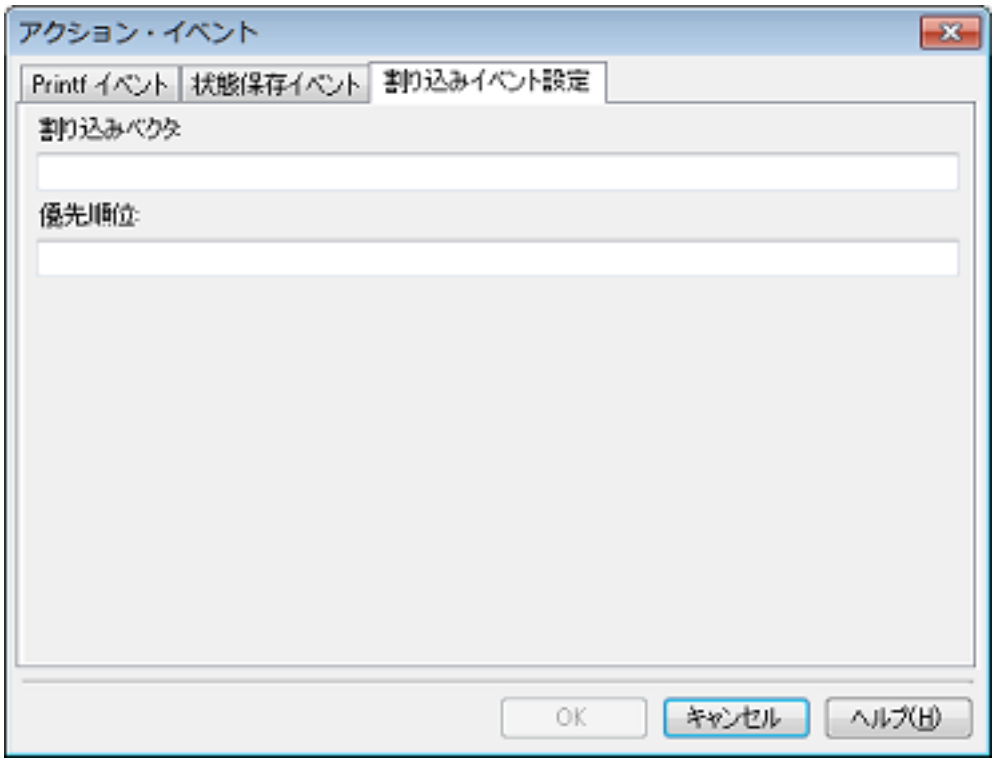
注意　割り込みイベントの設定に関しては（有効イベント数の制限など），「2.17.7 イベント設定に関する留意事項」も参照してください。

2.16.2.1 割り込みイベントを設定する

エディタ パネル／逆アセンブル パネル上で，割り込み要求を発生させたい箇所に割り込みイベントを設定します。
エディタ パネル／逆アセンブル パネルにおいて，割り込みイベントを設定したい行／アドレス注にキャレットを移動したのち，コンテキスト・メニューの［アクション・イベントの登録 ...］を選択すると表示されるアクション・イベント ダイアログで［割り込みイベント設定］タブ【シミュレータ】を選択します。
このダイアログにおいて，次の操作を行ってください。

注　割り込みイベントは，アドレス表示がない行に設定することはできません。

図 2.169　割り込みイベントを設定する（アクション・イベント ダイアログ：［割り込みイベント設定］タブ）



(1)　［割り込みベクタ］の指定
割り込みベクタ番号を指定します。
0 ～ 255 の範囲で直接入力により指定してください。

(2)　［優先順位］の指定
割り込み優先順位を指定します。

- 【RX610 グループ】
  1 ～ 8 の範囲で直接入力により指定してください。
  8 を指定した場合は，高速割り込みとして動作します。
- 【RX610 グループ以外】
  1 ～ 16 の範囲で直接入力により指定してください。

16 を指定した場合は，高速割り込みとして動作します。

(3) ［OK］ボタンのクリック
ここで指定した割り込みイベントをエディタ パネル／逆アセンブル パネル上のキャレット位置の行に設定します。
割り込みイベントが設定されると，エディタ パネル／逆アセンブル パネルのイベント・エリアに マークが表示され，イベント パネルで管理されます（「2.17 イベントの管理」参照）。

2.16.2.2 プログラムを実行する

プログラムを実行します（「2.9 プログラムの実行」参照）。
プログラムを実行することにより，割り込みイベントを設定した箇所の命令実行直前で割り込み要求が発生し，CPU が割り込みを受け付けると割り込み例外処理を実行します。

注意 1. 発生した割り込み要求が受け付けられ，割り込み例外処理が実行されると，割り込み要求はクリアされます。

注意 2. 発生した割り込み要求が保留中に，別の割り込みイベントによる割り込み要求が発生すると，割り込み優先順位が高い割り込みのみ有効となります。

2.16.2.3 割り込みイベントを編集する

一度設定した割り込みイベントを編集することができます。
編集を行う場合は，イベント パネルにおいて，編集対象の割り込みイベントを選択したのち，コンテキスト・メニューの［条件の編集 ...］を選択します。オープンする割り込みイベント詳細設定ダイアログ【シミュレータ】において，編集が必要な項目を編集したのち，［OK］ボタンをクリックします。

(1) ［パス・カウント］の編集
1 ～ 16383 の範囲で直接入力により指定してください。
入力したパス・カウント分のイベント条件を満たした時点で，イベント成立となります。

(2) ［割り込み］の編集
割り込みベクタと優先順位を編集することができます。編集方法はアクション・イベント ダイアログの［割り込みイベント設定］タブ【シミュレータ】と同じです（「(1) ［割り込みベクタ］の指定」，「(2) ［優先順位］の指定」参照）。

2.17 イベントの管理

イベントとは，“アドレス 0x1000 番地をフェッチした”，“アドレス 0x2000 番地にデータを書き込んだ”などのデバッグにおけるマイクロコントローラの特定の状態を指します。

CS+ では，このイベントを任意の箇所でのブレーク，トレース動作の開始／終了，タイマ計測の開始／終了などのデバッグ機能のアクション・トリガとして利用します。

この節では，これらのイベントを管理する方法について説明します。

イベントは，一括して次のイベント パネルで管理されます。

［表示］メニュー→［イベント］を選択してください。

イベント パネルでは，現在設定されているイベントの詳細情報を一覧で確認することができ，各イベントの削除，設定状態（有効／無効）の切り替え，詳細情報の表示，および設定の変更を行うことができます。

なお，各エリアの見方，および機能についての詳細は，イベント パネルの項を参照してください。

備考　複数のロード・モジュール・ファイルをダウンロードした場合，イベント パネルの詳細情報にはロード・モジュール名が表示されます。

図 2.170　設定したイベントの表示（イベント パネル）【E1】【E20】

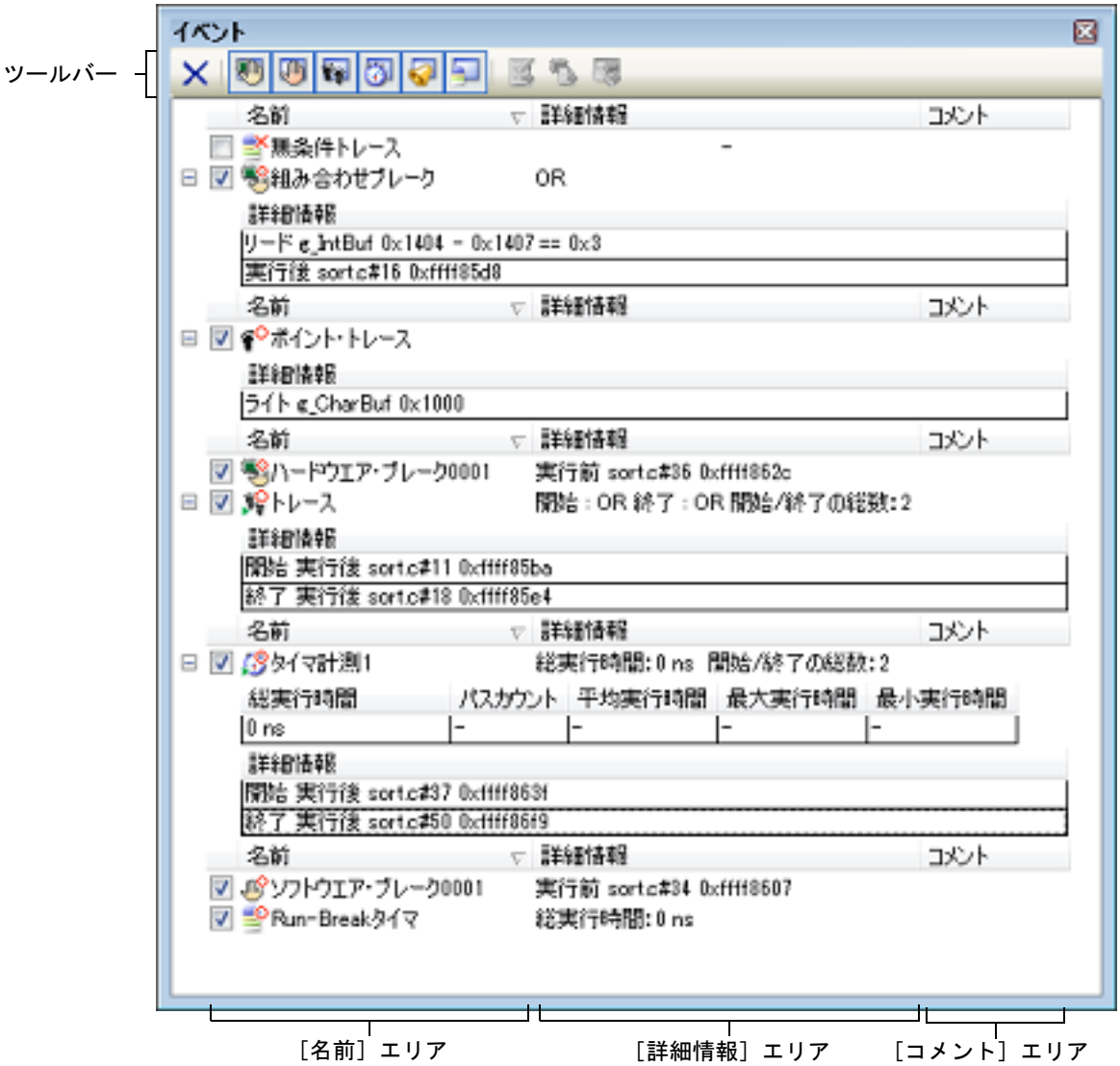


図 2.171 設定したイベントの表示（イベント パネル）【EZ Emulator】

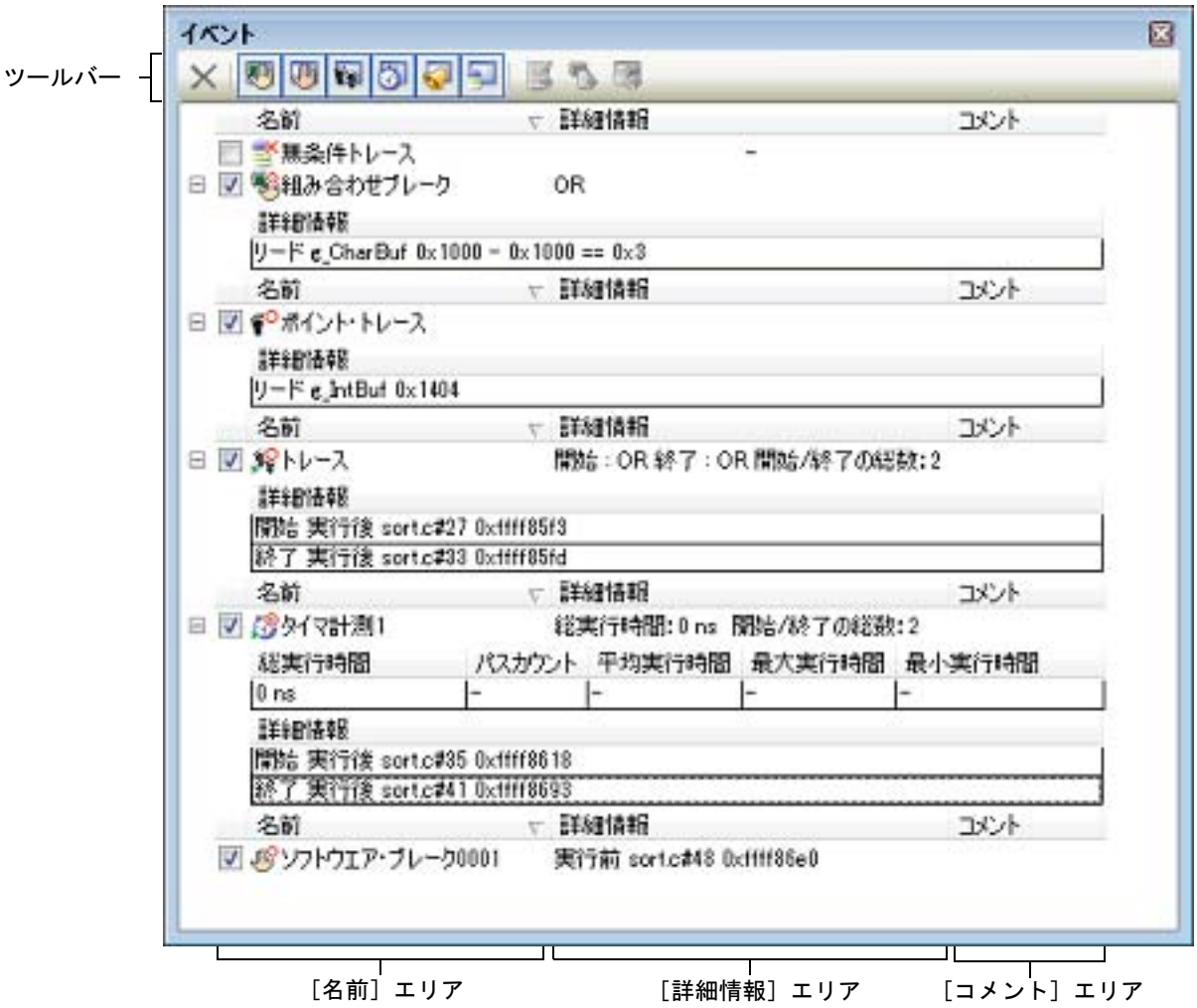
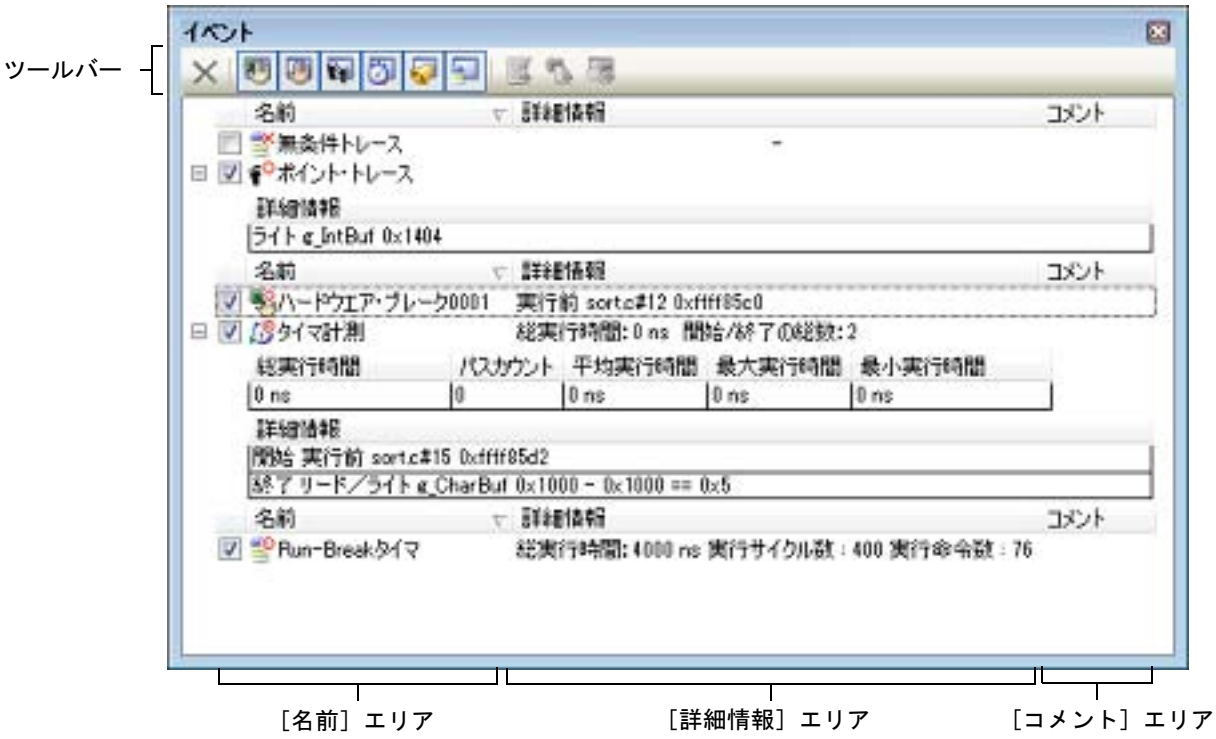


図 2.172 設定したイベントの表示（イベント パネル）【シミュレータ】



2.17.1 設定状態（有効／無効）を変更する

対象となるイベント名のチェック・ボックスのチェックを変更することで，イベントの設定状態を変更することができます（イベントの設定状態を変更すると，対応してイベント・マークも変化します）。

イベントの設定状態には，次の種類があります。

図 2.173 イベント名のチェック・ボックス

表 2.20 イベントの設定状態

	有効状態	指定されている条件の成立で，対象となるイベントが発生します。 チェックを外すことにより，イベントを無効状態にすることができます。
	無効状態	指定されている条件が成立しても，対象となるイベントは発生しません。 チェックすることにより，イベントを有効状態にすることができます。
	保留状態	指定されている条件が，デバッグ対象のプログラムでは設定することができません。チェック・ボックスを操作することはできません。

備考 1. Run-Break タイマ・イベント（EZ Emulator では未サポート）を無効状態／保留状態にすることはできません。

備考 2. イベントの状態は，エディタ パネル／逆アセンブル パネル上のイベント・マークを右クリックすることで表示される，メニューからの選択でも変更することができます。

備考 3. 無条件トレース・イベントとトレース・イベントにおける有効／無効状態の設定は，排他制御となります。このため，ビルトイン・イベントである無条件トレース・イベントは，デフォルトで有効状態で設定されていますが，トレース開始イベント／トレース終了イベントのいずれかが設定されると同時に自動的に無効状態に変更され，トレース開始イベント／トレース終了イベントを 1 つにまとめたトレース・イベントが有効状態になります。

また逆に，設定されているトレース・イベントを無効状態にすると，自動的に無条件トレース・イベントが有効状態となります。

2.17.2 特定のイベント種別のみ表示する

ツールバーの次のボタンをクリックすることで，特定のイベント種別のみを表示することができます。

	ハードウエア・ブレーク関連のイベントを表示します。
【E1】【E20】【EZ Emulator】	ソフトウエア・ブレーク関連のイベントを表示します。
	トレース関連のイベントを表示します。
	タイマ関連のイベントを表示します。
	アクション・イベント（Printf イベント，割り込みイベント【シミュレータ】）を表示します。
	ビルトイン・イベント（無条件トレース・イベント /Run-Break タイマ・イベント）を表示します。注

注　EZ Emulator の場合，Run-Break タイマ・イベントは未サポートとなります。

2.17.3 イベントのアドレスにジャンプする

次のボタンをクリックすることにより，現在選択しているイベントのアドレスに対応して，各パネルにジャンプします。

ただし，トレース・イベント／タイマ計測イベント／ビルトイン・イベント（無条件トレース・イベント /Run-Break タイマ・イベント）を選択している場合は，このボタンは無効となります。

	選択しているイベントが設定されているアドレスに対応するソース行にキャレットを移動した状態で，エディタ パネルがオープンします。
	選択しているイベントが設定されているアドレスに対応する逆アセンブル結果にキャレットを移動した状態で，逆アセンブル パネルがオープンします。
	選択しているイベントが設定されているアドレスに対応するメモリ値にキャレットを移動した状態で，メモリ パネルがオープンします。

2.17.4 イベントの詳細設定を編集する

ここでは，各種イベント詳細設定の編集方法を説明します。なお，タイマ計測イベントの編集に関しては，「2.14.3.3 タイマ計測イベントを編集する【E1】【E20】【EZ Emulator】」を，アクション・イベント（Printf イベント，割り込みイベント）の編集に関しては，「2. 16　プログラム内へのアクションの設定」を参照してください。

2.17.4.1 実行系イベントを編集する
2.17.4.2 アクセス系イベントを編集する
2.17.4.3 イベントの組み合わせ条件を編集する【E1】【E20】【EZ Emulator】

2.17.4.1 実行系イベントを編集する

実行系イベントのアドレス条件およびパス・カウント条件を編集することができます。

実行系イベントは，イベント パネルにおいて次の操作により表示される実行イベント詳細設定ダイアログにて編集します。

ハードウエア・ブレーク	編集したいハードウエア・ブレークにキャレットを移動したのち，コンテキスト・メニューの「条件の編集 ...」を選択
トレース	編集したいトレース内の実行系イベントにキャレットを移動したのち，コンテキスト・メニューの「条件の編集 ...」を選択

組み合わせブレーク 【E1】【E20】【EZ Emulator】	編集したい組み合わせブレーク内の実行系イベントにキャレットを移動したのち，コンテキスト・メニューの「条件の編集 ...」を選択
タイマ計測	編集したいタイマ計測内の実行系イベントにキャレットを移動したのち，コンテキスト・メニューの「条件の編集 ...」を選択

(1) アドレスの編集
[アドレス] カテゴリ内の [アドレス] プロパティに 16 進数の数値を入力して OK ボタンをクリックしてください。

(2) パス・カウントの編集
[パス・カウント] カテゴリ内の [パス・カウント] プロパティに 10 進数の数値を入力して OK ボタンをクリックしてください。入力したパス・カウント分のイベント条件を満たした時点で，イベント成立となります。
パス・カウントの指定可能値は，デバッグ・ツールにより次のように異なります。

- 【E1】【E20】【EZ Emulator】
  1 ～ 256
- 【シミュレータ】
  1 ～ 16383

注意 1. 【E1】【E20】【EZ Emulator】
パス・カウントは，全イベント（アクセス系イベント含む）中で 1 つのイベントに対してのみ 1 以外の値を指定可能です。
2 つのイベントに 1 以外のパス・カウントを設定した場合，エラーとなります。

注意 2. 【E1(Serial) / E20(Serial) / EZ Emulator【RX100, RX200 シリーズ】】
パス・カウントは，1 以外の値を指定できません。

2.17.4.2 アクセス系イベントを編集する

アクセス系イベントのアドレス条件およびデータ条件を編集することができます。
なお，アドレスおよびパス・カウントの編集方法については，「(1) アドレスの編集および (2) パス・カウントの編集」を参照してください。
アクセス系イベントは，イベント パネル上で次の操作により表示されるアクセスイベント詳細設定ダイアログにて編集します。

トレース	編集したいトレース内のアクセス系イベントにキャレットを移動したのち，コンテキスト・メニューの「条件の編集 ...」を選択
ポイント・トレース	編集したいポイント・トレース内のアクセス系イベントにキャレットを移動したのち，コンテキスト・メニューの「条件の編集 ...」を選択
組み合わせブレーク 【E1】【E20】【EZ Emulator】	編集したい組み合わせブレーク内のアクセス系イベントにキャレットを移動したのち，コンテキスト・メニューの「条件の編集 ...」を選択
タイマ計測	編集したいタイマ計測内のアクセス系イベントにキャレットを移動したのち，コンテキスト・メニューの「条件の編集 ...」を選択

(1) アドレス条件の編集
[アドレス条件] カテゴリの [比較条件] プロパティにおいて比較条件をドロップダウン・リストより指定してください。
指定した比較条件により，[アドレス] カテゴリ内の [アドレス] プロパティに表示されたアドレス値と組み合わせて次のようにアドレス条件が設定できます。

- [指定無し] を指定した場合
  アドレス値を条件とします。アドレス値に対するアクセスが発生した場合に，条件成立となります。
- [アドレス範囲] を指定した場合【E1】【E20】【EZ Emulator】
  アドレス値を開始アドレスとします。[アドレス条件] カテゴリ内に追加される [終了アドレス] プロパティにおいて，終了アドレス値を入力してください。設定した開始／終了アドレスの範囲内に対してアクセスが発生した場合に，条件成立となります。
  また，同じく追加される [範囲条件] プロパティにおいて，アドレス範囲外にも設定できます。

注意 1. 【RX100, RX200 シリーズ】
アドレス範囲による比較条件は未サポートです。

注意 2. 【RX600 シリーズ】
設定可能なアドレス範囲による比較条件は，1 点のみです。

注意 3. 【シミュレータ】
アドレス条件は指定できません。

- [マスク付き比較] を指定した場合【E1】【E20】【EZ Emulator】
アドレス値に対するマスク値を設定できます。[アドレス条件] カテゴリ内に追加される [マスク値] プロパティにおいて，16 進数のアドレスマスク値を入力してください。マスク付きアドレスと一致したアドレスに対してアクセスが発生した場合，条件成立となります。
また，同じく追加される [比較] プロパティにおいて，アドレス比較条件を不一致にも設定できます。

注意 マスクは，条件となるアドレス値に対してビット単位でマスク値 “ 0 ” を Don’t Care とします。
例) 0x1000 ～ 0x1FFF のアドレス条件を設定する場合
アドレス値：0x1000
マスク値：0xF000

(2) データ条件の編集

- [アクセス種別]
アクセス種別として次の項目から指定します。

リード	リードアクセスが発生した場合に，条件成立となります。
ライト	ライトアクセスが発生した場合に，条件成立となります。
リード／ライト	リードまたはライトアクセスが発生した場合に，条件成立となります。

- [アクセスサイズ]
アクセスサイズとして次の項目から指定します。

指定無し	全てのアクセスサイズに対して アクセスが発生した場合に，条件成立となります。
バイト	バイトサイズのアクセスが発生した場合に，条件成立となります。
ワード	ワードサイズのアクセスが発生した場合に，条件成立となります。
ロングワード	ロングワードサイズのアクセスが発生した場合に，条件成立となります。

- [比較条件]
データ比較条件を指定します。
デバッグ・ツールにより，比較条件は次のように異なります。

【E1】【E20】【EZ Emulator】	指定値一致 (==)	データ比較条件が一致した場合に，条件成立となります。
	指定値不一致 (!=)	データ比較条件が不一致した場合に，条件成立となります。

【シミュレータ】	一致 (==)	データが指定値と一致で条件成立となります。
	不一致 (!=)	データが指定値と不一致で条件成立となります。
	符号反転注	前回アクセス時のデータと今回アクセス時のデータで符号が反転した場合に条件成立となります。
	差分注	前回アクセス時のデータと今回アクセス時のデータの差分が指定値を超えた場合に条件成立となります。
	指定値より大きい (>)	データが指定値より大で条件成立となります。
	指定値より小さい (<)	データが指定値より小で条件成立となります。
	指定値以上 (>=)	データが指定値以上で条件成立となります。
	指定値以下 (<=)	データが指定値以下で条件成立となります。
	範囲内 (<=Value<=)	データが指定値の範囲内で条件成立となります。 （[比較データ 1] <= データ <= [比較データ 2]）
	範囲外 !(<=Value<=)	データが指定値の範囲外で条件成立となります。 （データ < [比較データ 1] || [比較データ 2] < データ）
	指定値無し	比較データを指定しません。

注　[符号反転]，および [差分] は前回データを比較するため，リセット後，および条件成立 1 回目の判定は常に不成立となります。

備考　【シミュレータ】
このプロパティは，[アクセスサイズ] プロパティにおいて，[指定無し] を指定した場合は表示されません。

- [比較データ]【E1】【E20】【EZ Emulator】
データ値を 16 進数で指定します。

- [比較データ 1]【シミュレータ】
データ値を 16 進数で指定します。
このプロパティは，[比較条件] プロパティにおいて，次の項目を指定した場合に表示されます。
[一致 (==)] ／ [不一致 (!=)] ／ [指定値より大きい (>)] ／ [指定値より小さい (<)] ／ [指定値以上 (>=)] ／ [指定値以下 (<=)] ／ [範囲内 (<=Value<=)] ／ [範囲外 !(<=Value<=)]

- [比較データ 2]【シミュレータ】
データ値を 16 進数で指定します。
このプロパティは，[比較条件] プロパティにおいて，[範囲内 (<=Value<=)] ／ [範囲外 !(<=Value<=)] を指定した場合のみ表示されます。

- [データマスクを指定する]
比較データ値を入力した場合にプロパティが表示され，データ値に対するマスク値を指定するか否かを指定できます。[はい] を指定した場合，データマスクを指定できます。

備考　【シミュレータ】
このプロパティは，[比較条件] プロパティにおいて，次の項目を指定した場合に表示されます。
[一致 (==)] ／ [不一致 (!=)] ／ [指定値より大きい (>)] ／ [指定値より小さい (<)] ／ [指定値以上 (>=)] ／ [指定値以下 (<=)] ／ [範囲内 (<=Value<=)] ／ [範囲外 !(<=Value<=)]

- [マスク値]
[データマスクを指定する] プロパティにおいて，[はい] を指定した場合にプロパティが表示されます。16 進数のデータマスク値を入力して，マスク付きデータと比較条件が一致したデータアクセスが発生した場合，条件成立となります。

- [差分]【シミュレータ】
比較データを 16 進数で指定します。
このプロパティは，[比較条件] プロパティにおいて，[差分] を指定した場合のみ表示されます。

- [符号]【シミュレータ】
比較するデータに符号があるか否かを指定します。
このプロパティは，[比較条件] プロパティにおいて，次の項目を指定した場合に表示されます。

[差分] ／［指定値より大きい (>)］／［指定値より小さい (<)］／［指定値以上 (>=)］／［指定値以下 (<=)］／［範囲内 (<=Value<=)］／［範囲外 !(<=Value<=)］

(3) バス・マスタの編集【E1/E20【 RX71M, RX64M グループ】】

- [バス・マスタ]
バス・マスタとして次の項目から指定します。

CPU	指定したデータアクセスが，CPU からのアクセスにより発生した場合に，条件成立となります。
DMAC/DTC	指定したデータアクセスが，DMAC/DTC からのアクセスにより発生した場合に，条件成立となります。

2.17.4.3 イベントの組み合わせ条件を編集する【E1】【E20】【EZ Emulator】

複数のイベントを設定した組み合わせブレークおよびトレースに対する組み合わせ条件を編集します。トレースの組み合わせ条件は，タブを切り替えることで開始と終了のそれぞれの条件を設定することができます。

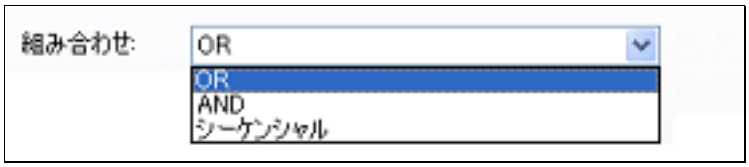
イベントの組み合わせ条件は，イベント パネル上で次の操作により表示される組み合わせ条件設定ダイアログ【E1】【E20】【EZ Emulator】にて編集します。

注意 【EZ Emulator】
端子リセットを入れた場合，EZ Emulator からの再実行時に組み合わせイベント情報はクリアされます。

組み合わせブレーク	組み合わせブレークにキャレットを移動したのち，コンテキスト・メニューの「条件の編集 ...」を選択
トレース	トレースにキャレットを移動したのち，コンテキスト・メニューの「条件の編集 ...」を選択

(1) 組み合わせ条件の編集
組み合わせ条件をプルダウンメニューにより，[OR]，[AND] および [シーケンシャル] から選択します。組み合わせ条件ごとの動作説明については，「22.10.5 複数のブレーク・イベントを組み合わせる（組み合わせブレーク）【E1】【E20】【EZ Emulator】」および「2.13.3.2 複数のイベントを組み合わせる【E1】【E20】【EZ Emulator】」を参照してください。

図 2.174 [組み合わせ] 項目



注意 1. トレース・イベントにおいてトレース終了イベントの組み合わせ条件は，[OR] のみです。

注意 2. 組み合わせ条件における [AND] および [シーケンシャル] は，組み合わせブレークとトレース開始イベントとで兼用です。そのため，どちらか一方で [AND] および [シーケンシャル] を組み合わせ条件に設定した場合，もう片方は，[OR] でのみ設定できます。

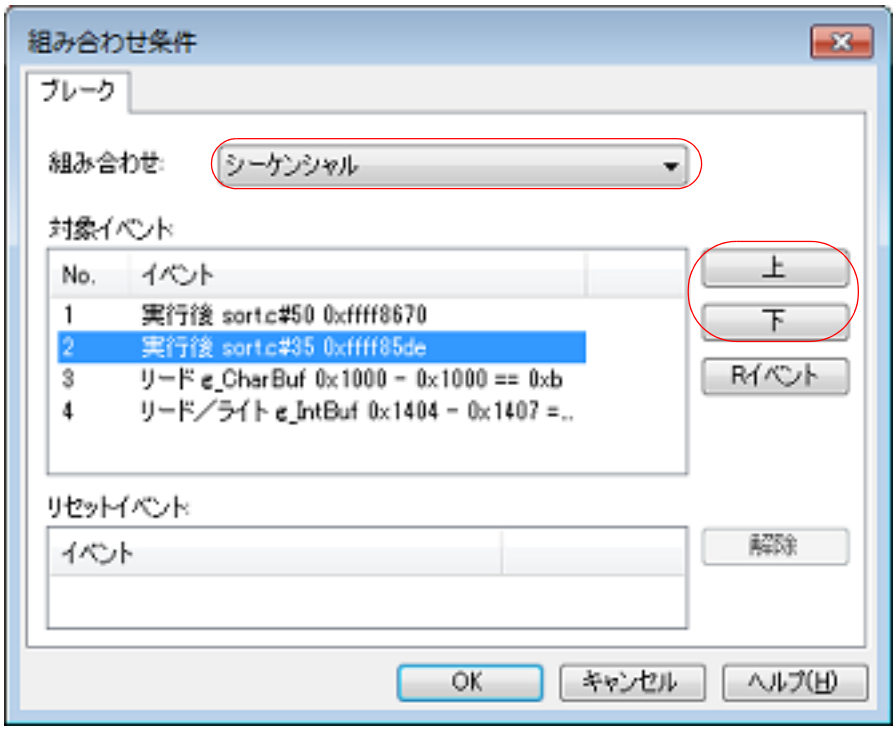
注意 3. 【RX600 シリーズ】
組み合わせ条件がシーケンシャルの場合，アクセス系イベントは，リセットイベント，および，3 番目までのイベントに指定できます。

注意 4. 【RX600 シリーズ】
組み合わせ条件がシーケンシャルの場合，アドレス範囲条件を設定したアクセス系イベントは 1 番目のイベントにのみ指定できます。

注意 5. 【RX200 シリーズ】
組み合わせ条件がシーケンシャルの場合，アクセス系イベントは 2 番目までのイベントに指定できます。

(2) 対象イベントの編集
[↑上] および [↓下] ボタンで対象イベントリスト内のイベントの順序を変更します。このボタンは，組み合わせ条件をシーケンシャルに選択し，対象イベントリスト内のイベントにキャレットを移動した場合のみ有効となります。

図 2.175　　対象イベントリスト内のイベント順序変更（シーケンシャル）



(3)　リセットイベントの編集

リセットイベントを登録する場合，[R イベント] ボタンをクリックします。このボタンは，組み合わせ条件をシーケンシャルに選択して対象イベントリスト内のイベントにキャレットを移動した場合のみ有効となります。また，リセットイベントの登録を解除する場合，[解除] ボタンをクリックします。このボタンは，リセットイベントリスト内のイベントにキャレットを移動して有効となります。

注意 1.　リセットイベントは 1 点のみ指定可能です。
リセットイベントが登録された状態で R イベントボタンを押した場合，対象イベントリスト内で選択したイベントと登録済みのリセットイベントが置き換わります。

注意 2.　リセットイベントが登録された状態で組み合わせ条件を [シーケンシャル] から [OR] または [AND] に変更した場合，リセットイベントリスト内のイベントは対象イベントに含まれません。対象イベントとする場合，リセットイベントを登録解除してから組み合わせ条件を変更してください。

注意 3.　組み合わせ条件がシーケンシャルの場合，リセットイベントに登録したイベントは，パス・カウント条件を指定できません。

注意 4.　【RX600 シリーズ】
アドレス範囲条件を指定したイベントは，リセットイベントに登録できません。また，リセットイベントに登録したイベントは，アドレス範囲条件を指定できません。

図 2.176 リセットイベント登録（シーケンシャル）

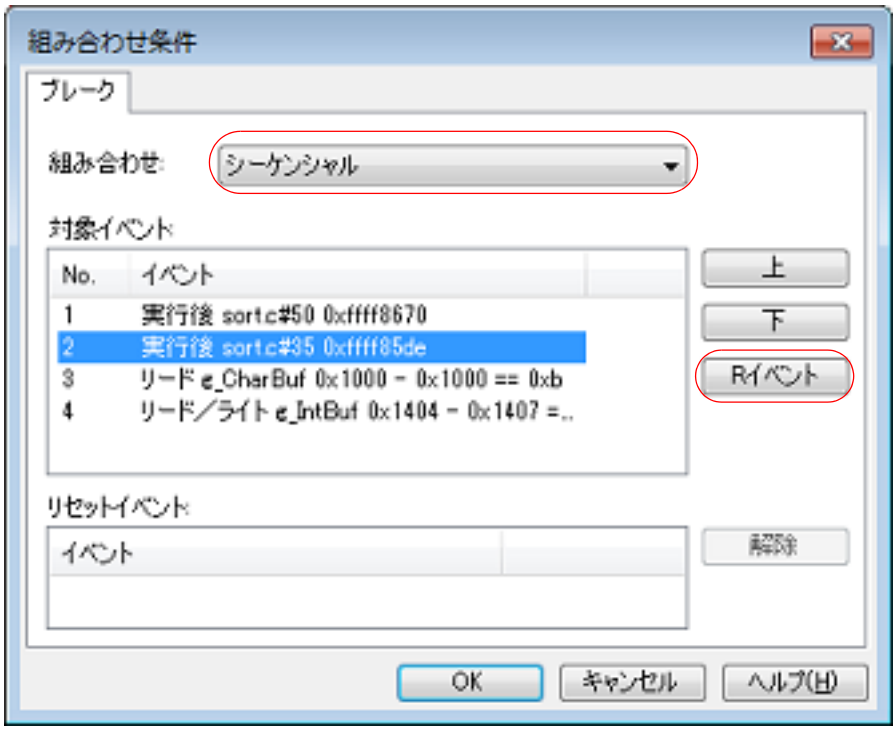
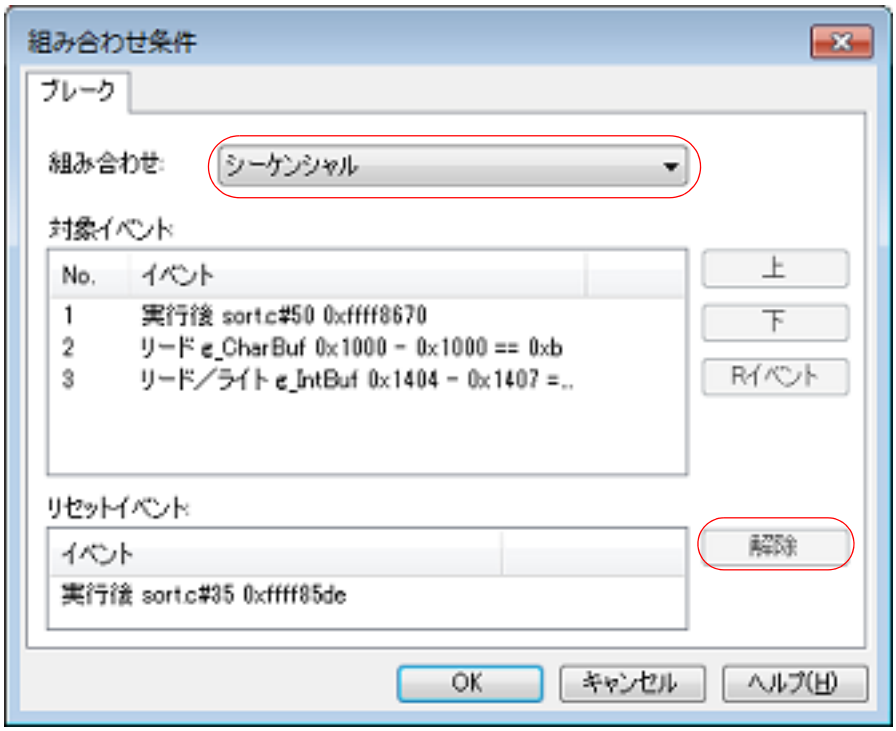


図 2.177 リセットイベント登録解除（シーケンシャル）



2.17.5 イベントを削除する

設定したイベントを削除するには，対象イベントを選択したのち，ツールバーの ボタンをクリックします。
なお，複数のイベントで設定されたイベントの場合，詳細情報内のイベントを個別に削除することもできます。
ただし，ビルトイン・イベントである無条件トレース・イベント /Run-Break タイマ・イベントを削除することはできません。

備考 1.　タイマ開始／終了イベント，トレース開始／終了イベントおよびアクション・イベントについては，エディタ パネル／逆アセンブル パネル上で表示されているイベント・マークからコンテキスト・メニューの [イベント削除] を選択することで，イベントを削除することができます。

備考 2.　ソフトウエア・ブレーク／ハードウエア・ブレークについては，エディタ パネル／逆アセンブル パネル上で表示されているイベント・マークをクリックすることで，イベントを削除することができます。

備考 3.　設定したイベントを一度にすべて削除する場合は，コンテキスト・メニューの［すべて選択］を選択したのち，[X] ボタンをクリックします（ビルトイン・イベントを除く）。

2.17.6 イベントにコメントを入力する

設定した各イベントに対して，ユーザが自由にコメントを入力することができます。
コメントの入力は，コメントを入力したいイベントを選択したのち，［コメント］エリアをクリックし，任意のテキストをキーボードから直接入力します（[Esc］キーの押下で編集モードをキャンセルします）。
コメントを編集したのち，［Enter］キーの押下，または編集領域以外へのフォーカスの移動により，編集を完了します。
なお，コメントは最大 256 文字まで入力することができ，使用中のユーザの設定として保存されます。

2.17.7 イベント設定に関する留意事項

ここでは，各種イベントの設定を行う際の留意事項を示します。

2.17.7.1 有効イベント数の制限
2.17.7.2 実行中に設定／削除可能なイベント種別
2.17.7.3 その他の注意事項

2.17.7.1 有効イベント数の制限

有効状態で同時に設定可能なイベントの個数には，次の制限があります。
したがって，新たに有効状態のイベントを設定する際にこの制限数を越えてしまう場合は，いったん設定しているイベントのいずれかを無効状態にする必要があります。

表 2.21　有効イベント数の制限【RX600 シリーズ】

イベント種別	デバッグ・ツール	
	E1(Serial) / E1(JTAG) / E20(Serial) / E20(JTAG) EZ Emulator	シミュレータ
ソフトウエア・ブレーク	256 注 9	0
ハードウエア・ブレーク（実行系：実行前）	8 注 1	1024 注 8
ブレーク・イベント（実行系：実行後）	8 注 1	0
ブレーク・イベント（アクセス系：リード , ライト , リード / ライト）	4 注 1 注 3	1024
組み合わせブレーク	8+4 注 1 注 2 注 5	0
トレース（トレース開始／トレース終了）	8+4 注 1 注 2 注 6	256
ポイント・トレース	4 注 1 注 3	256
タイマ計測（タイマ開始／タイマ終了）	8+4 注 1 注 2 注 6	1 注 7
アクション（Printf イベント）	256 注 9	1024 注 8
アクション（割り込みイベント）	0	256

表 2.22　　有効イベント数の制限【RX200 シリーズ】

イベント種別	デバッグ・ツール	
	E1(Serial) / E20(Serial) EZ Emulator	シミュレータ
ソフトウエア・ブレーク	256 注 9	0
ハードウエア・ブレーク（実行系：実行前）	4 注 4	1024 注 8
ブレーク・イベント（実行系：実行後）	4 注 4	0
ブレーク・イベント （アクセス系：リード , ライト , リード / ライト）	2 注 4	1024
組み合わせブレーク	4+2 注 4	0
トレース（トレース開始／トレース終了）	4+2 注 4 注 5 注 6	256
ポイント・トレース	4 注 4	256
タイマ計測（タイマ開始／タイマ終了）	4+2 注 4 注 6	1 注 7
アクション（Printf イベント）	256 注 9	1024 注 8
アクション（割り込みイベント）	0	256

表 2.23　　有効イベント数の制限【RX100 シリーズ】

イベント種別	デバッグ・ツール	
	E1(Serial) / E20(Serial) EZ Emulator	シミュレータ
ソフトウエア・ブレーク	256 注 9	0
ハードウエア・ブレーク（実行系：実行前）	4 注 4	1024 注 8
ブレーク・イベント（実行系：実行後）	4 注 4	0
ブレーク・イベント （アクセス系：リード , ライト , リード / ライト）	2 注 4	1024
組み合わせブレーク	4+2 注 4	0
トレース（トレース開始／トレース終了）	4+2 注 4 注 5 注 6	256
ポイント・トレース	4 注 4	256
タイマ計測（タイマ開始／タイマ終了）	0	1 注 7
アクション（Printf イベント）	256 注 9	1024 注 8
アクション（割り込みイベント）	0	256

注 1.　実行アドレスに対するイベント 8 点とデータアクセスに対するイベント 4 点があり，各機能で兼用します。

注 2.　イベントの組み合わせは最大 8 点まで設定可能です。

注 3.　【E20(JTAG)】
トレース機能／リアルタイム RAM モニタ (RRM) 機能は一部排他使用の機能です。そのため，RRM 機能を使用時はポイント・トレースを使用することができません。また，アクセス・イベントは RRM ブロックに空きがある場合のみ使用可能です。

注 4.　実行アドレスに対するイベント 4 点とデータアクセスに対するイベント 2 点があり，各機能で兼用します。

注 5.　組み合わせブレークの組み合わせ条件がシーケンシャルの場合，マイクロコントローラにより次のように最大イベント設定数が異なります。

- 【RX600 シリーズ】
  7 点 + リセットイベント (R イベント)
- 【RX100, RX200 シリーズ】
  3 点 + リセットイベント (R イベント)

注 6. トレース開始／終了イベント，タイマ計測開始／終了イベントにおいて実行系イベントの最大数を設定した場合，エディタ パネル／逆アセンブル パネルにおいてコンテキストメニューの [ここまで実行] は使用できません。また，CPU リセット後に指定シンボル位置まで実行する機能も使用できません。

注 7. タイマ開始／終了イベントは，実行アドレスまたはデータアクセスに対するイベントをそれぞれ 1 点のみ設定可能です。

注 8. ハードウエア・ブレーク (実行系：実行前) とアクション (Printf イベント) は，各機能で兼用します。

注 9. ソフトウエア・ブレークとアクション (Printf イベント) は，各機能で兼用します。

2.17.7.2 実行中に設定／削除可能なイベント種別

プログラム実行中，またはトレーサ／タイマ実行中に設定／削除可能なイベント種別は次のとおりです。

表 2.24 実行中に設定／削除可能なイベント種別

イベント種別	デバッグ・ツール	
	E1(Serial) / E1(JTAG) / E20(Serial) / E20(JTAG) EZ Emulator	シミュレータ
ソフトウエア・ブレーク	△注 1 注 2	―
ハードウエア・ブレーク（実行系：実行前）	△注 2	○
ブレーク・イベント（実行系：実行後）	△注 2	―
ブレーク・イベント （アクセス系：リード , ライト , リード / ライト）	△注 2	○
トレース（トレース開始／トレース終了）	×	□
ポイント・トレース	×	○
タイマ計測（タイマ開始／タイマ終了）	×	×
アクション（Printf イベント）	△注 1 注 2	○
アクション（割り込みイベント）	―	○

○ ：可能
△ ：プログラムの実行を一瞬停止することで可能
□ ：トレーサ動作中は不可
― ：非サポート
× ：不可

注 1. 内蔵 RAM のみ設定可能

注 2. 低消費電力状態中は設定不可

2.17.7.3 その他の注意事項

- ローカル変数にイベントを設定することはできません。

- ステップ実行中（リターン・アウト実行を含む），およびコンテキスト・メニューの［ここまで実行］によるプログラム実行中，イベントは発生しません。

- デバッグ対象のプログラムを再ダウンロードすることにより，既存のイベント設定位置が命令の途中になる場合における該当イベントの再設定方法は次のとおりです。

- デバッグ情報がある場合
  イベント設定位置は常にソース・テキスト行の先頭に移動します。
- デバッグ情報がない場合
  プロパティ パネルの［ダウンロード・ファイル設定］タブ上の［ダウンロード］カテゴリ内 [イベント設定位置の自動変更方法］プロパティの設定に依存します。

2.18 フック処理を設定する

この節では，フック処理機能を使用し，デバッグ・ツールにフックを設定するための操作方法について説明します。
フック処理を設定することで，ロード・モジュールのダウンロード前後や CPU リセット後に，I/O レジスタ /CPU レジスタの値を自動的に変更することができます。
フック処理の設定は，プロパティ パネルの［フック処理設定］タブ上の［フック処理設定］カテゴリ内で行います。

備考　たとえば，［ダウンロード前］プロパティで I/O レジスタを設定することにより，ダウンロードを高速に行うことができます。
また，外部 RAM へのダウンロードも，同様の設定で容易に行うことができます。

図 2.178　［フック処理］カテゴリ

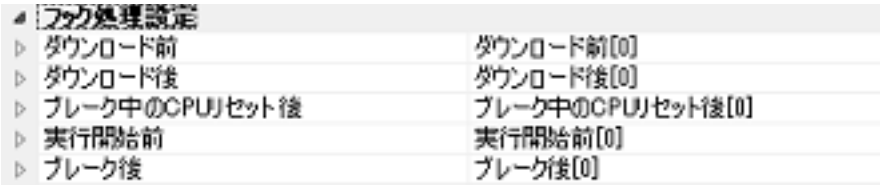


表 2.25　［フック処理］カテゴリのプロパティ

プロパティ	タイミング
ダウンロード前	ロード・モジュール・ファイルをダウンロードする直前に，指定した処理を行います。
ダウンロード後	ロード・モジュール・ファイルをダウンロードした直後に，指定した処理を行います。
ブレーク中の CPU リセット後	CPU リセット直後に，指定した処理を行います。
実行開始前	プログラムの実行開始直前に，指定した処理を行います。
ブレーク後	プログラムの実行がブレークした直後に，指定した処理を行います。

［フック処理設定］カテゴリ内の各プロパティは，フック処理を行うタイミングを示し，プロパティ値の “[ ]” 内は，現在指定されている処理の数を示します（デフォルトで設定されているフック処理はありません）。
フック処理を行いたいプロパティに，目的の処理を次の手順で指定します。
処理の指定は，該当するプロパティを選択すると欄内右端に表示される［...］ボタンをクリックすることでオープンする，次のテキスト編集 ダイアログ上で行います。

図 2.179　テキスト編集 ダイアログのオープン

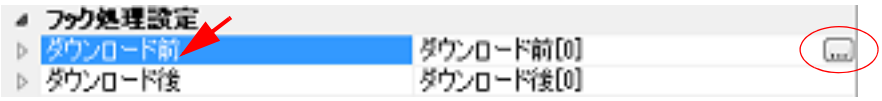
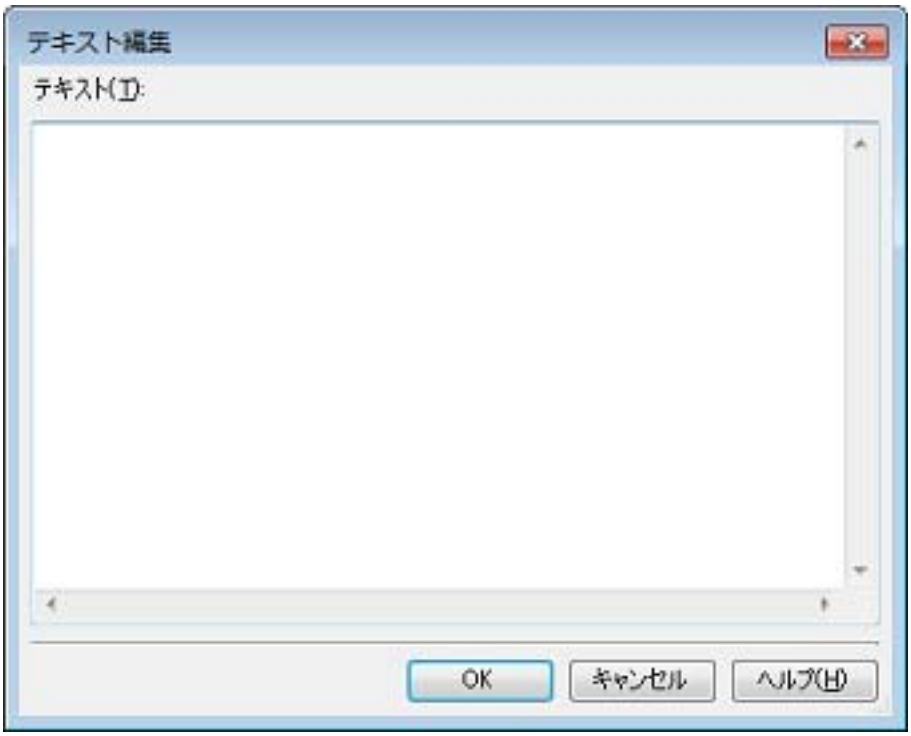


図 2.180　　フック処理を設定する（テキスト編集 ダイアログ）



このダイアログにおいて，目的の処理を直接入力により指定します。
処理の指定形式は次のいずれかです。

【指定形式：1】
*I/O レジスタ*の内容を，*数値*に自動的に書き換えます。

```
I/O レジスタ名 数値
```

【指定形式：2】
*CPU レジスタ*の内容を，*数値*に自動的に書き換えます。

```
CPU レジスタ名 数値
```

【指定形式：3】
Python スクリプト・パス（絶対パス／プロジェクト・フォルダを基点とした相対パス）で指定したスクリプト・ファイルを実行します。

```
Source Python スクリプト・パス
```

備考 1.　　処理の指定の際，行頭に “#” を付与することにより，その行はコメント扱いとなります。

備考 2.　　半角スペースは，タブ文字でも代用可能です。

注意　　デバッガのフック処理から Python スクリプトを実行する場合，以下のコマンドが記載可能です。

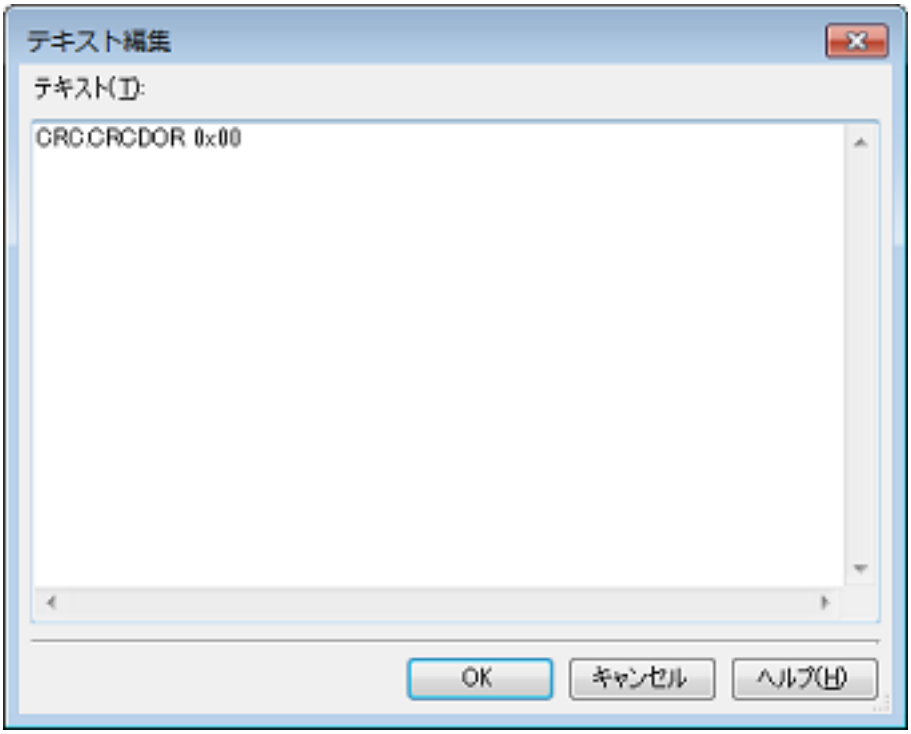
debugger.Register.GetValue
debugger.Register.SetValue
debugger.Memory.GetValue
debugger.Memory.SetValue

それ以外の Python コマンドを使用したい場合，Python コンソールの Hook コマンドを使用してください。

1 処理につき 64 文字まで入力可能で，各プロパティごとに 128 個までの処理を指定することができます（テキスト編集 ダイアログ上の［テキスト］エリア内の 1 行が 1 処理に相当）。

処理の指定が完了したのち，［OK］ボタンをクリックすると，指定した処理がプロパティ パネル上に反映されます。

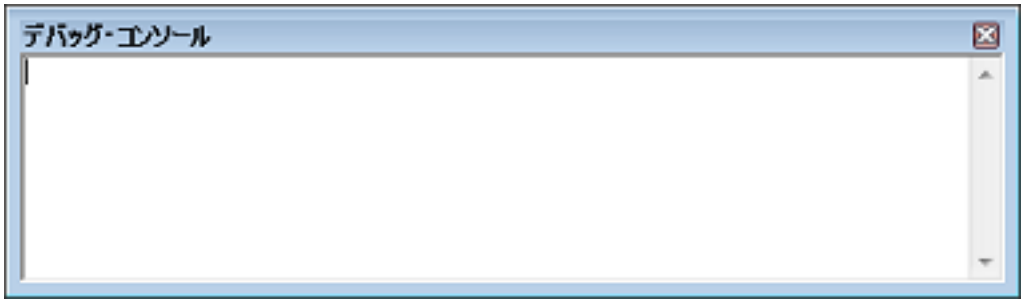
図 2.181 フック処理設定の例



2.19 デバッグ・コンソールの使用

この節では，標準入出力を実行するためのデバッグ・コンソールの使用方法について説明します。
プログラム内の標準ライブラリ関数，例えば，scanf 関数によるキーボード入力データの読み込み，printf 関数によるデータの出力は，次のデバッグ・コンソール パネルで行います。

図 2.182　　デバッグ・コンソール パネル



なお，デバッグ・コンソールの各機能についての詳細は，デバッグ・コンソール パネルを参照してください。

デバッグ・コンソール パネルによる標準入出力を C/C++ プログラムで使用する場合は，プログラム内に低水準インタフェース・ルーチンが必要です。

CS+ では，次のデバッグ・コンソール機能用のファイルを持つ，E1 ／ E20 用のサンプル・プロジェクト「RX610_Tutorial_DebugConsole」を提供しています。

- 低水準インタフェース・ルーチン（C 言語部分）用のファイル（lowsrc.c，lowsrc.h）
  - 低水準インタフェース・ルーチン：open，close，read，write，および lseek
    標準入出力により呼び出されます。
  - 入出力ライブラリの初期設定：_INIT_IOLIB
  - オープンした全関数のクローズ：_CLOSEALL
    それぞれ，初期設定ルーチン・ファイル resetprg.c 内の初期設定ルーチン “PowerON_Reset_PC” により呼び出されます。
- 低水準インタフェース・ルーチン（アセンブリ言語部分）用のファイル（lowlvl.src）
  - 1 文字の出力を行う関数：_charput
  - 1 文字の入力を行う関数：_charget
    それぞれ，低水準インタフェース・ルーチン “write”，“read” により呼び出されます。
- main 関数を含むソース・ファイル（DebugConsole_Sample.c）
  - main 関数内で標準ライブラリ関数 scanf，および printf を呼び出します。

注意　【E1】【E20】【EZ Emulator】
デバッグ・コンソールを使用する場合，プログラム中の charput，charget 関数内でシステムクロックを切り替えないでください。システム・クロックを切り替えるとエミュレータとマイクロコントローラの送受信に影響を与え，データが正常に送受信できない恐れがあります。
デバッグ・コンソール機能を使用する場合は，データの送受信のために，ユーザ・プログラムのリアルタイム性が損なわれます。

備考 1.　低水準インタフェース・ルーチンについては，「CC-RX コンパイラ ユーザーズマニュアル」を参照してください。

備考 2.　【シミュレータ】
標準入出力を実行する方式として，エミュレータと同様の方式とシミュレータ固有の方式をそれぞれサポートしています。シミュレータ固有方式では，標準入出力の他にファイル入出力を使用できます。
各サポート方式は，プロパティ パネル上の［デバッグ・ツール設定］タブの［ストリーム入出力］【シミュレータ】カテゴリ内の［ストリーム入出力モードを選択］プロパティを指定することで切り替わり

ます。
シミュレータ方式の入出力機能については，「B. 入出力機能」を参照してください。

デバッグ・コンソール機能のサンプル・プロジェクトを使用するには，次の手順で操作を行ってください。

(1) サンプル・プロジェクトを読み込む
CS+ のスタート パネル 上の［サンプル・プロジェクトを読み込む］から，E1 ／ E20 用のサンプル・プロジェクト「RX610_Tutorial_DebugConsole」を読み込みます。

備考　“ スタート パネル ” についての詳細は，「CS+ 統合開発環境 ユーザーズマニュアル プロジェクト操作編」を参照してください。

(2) 低水準インタフェース・ルーチン（アセンブリ言語部分）用のファイルを編集する【シミュレータ】
ストリーム入出力の方式をシミュレータモードとして使用する場合，低水準インタフェース・ルーチン（アセンブリ言語部分）用のファイル（lowlvl.src）の内容をシミュレータ用に編集する必要があります。lowlvl.src の内容を以下のシミュレータ用のサンプルへ置き換えてください。

```
          .GLB    _charput
          .GLB    _charget
SIM_IO    .EQU 0h
          .SECTION   P,CODE
;----------------------------------------------------------------------
;  _charput:
;----------------------------------------------------------------------
_charput:
          MOV.L      #IO_BUF,R2
          MOV.B      R1,[R2]
          MOV.L      #1220000h,R1
          MOV.L      #PARM,R3
          MOV.L      R2,[R3]
          MOV.L      R3,R2
          MOV.L      #SIM_IO,R3
          JSR        R3
          RTS
;----------------------------------------------------------------------
;  _charget:
;----------------------------------------------------------------------
_charget:
         MOV.L       #1210000h,R1
         MOV.L       #IO_BUF,R2
         MOV.L       #PARM,R3
         MOV.L       R2,[R3]
         MOV.L       R3,R2
         MOV.L       #SIM_IO,R3
         JSR         R3
         MOV.L       #IO_BUF,R2
         MOVU.B      [R2],R1
         RTS
;----------------------------------------------------------------------
;  I/O Buffer
;----------------------------------------------------------------------
          .SECTION  B,DATA,ALIGN=4
PARM:     .BLKL     1
          .SECTION  B_1,DATA
IO_BUF:   .BLKB     1
          .END
```

(3) 使用するデバッグ・ツールを選択する
使用するデバッグ・ツールの選択／切り替えは，プロジェクト・ツリー パネル上の［*品種名　デバッグ・ツール名* (デバッグ・ツール)］ノードを右クリックすることで表示されるコンテキスト・メニューにより行います。

備考　デバッグ・ツールの選択についての詳細は，「2.3.1 使用するデバッグ・ツールを選択する」を参照してください。

(4) プロパティ パネルの設定を変更する【シミュレータ】
シミュレータを使用する場合，プロパティ パネルの上の［デバッグ・ツール設定］タブの［ストリーム入出力］【シミュレータ】カテゴリ内において，次の設定を行います。

図 2.183　［ストリーム入出力］カテゴリ

(a)　［ストリーム入出力モードを選択］
エミュレータと同じ低水準インタフェース・ルーチン（アセンブリ言語部分）用のファイルを使用する場合は，［エミュレータモード］を選択してください（デフォルトでは［シミュレータモード］が指定されています）。

(b)　［ストリーム入出力機能を使用する］
このプロパティは，［ストリーム入出力モードを選択］プロパティにおいて［シミュレータモード］を選択した場合にのみ表示されます。
標準入出力またはファイル入出力を行う場合，［はい］を選択してください（デフォルトでは［いいえ］が指定されます）。

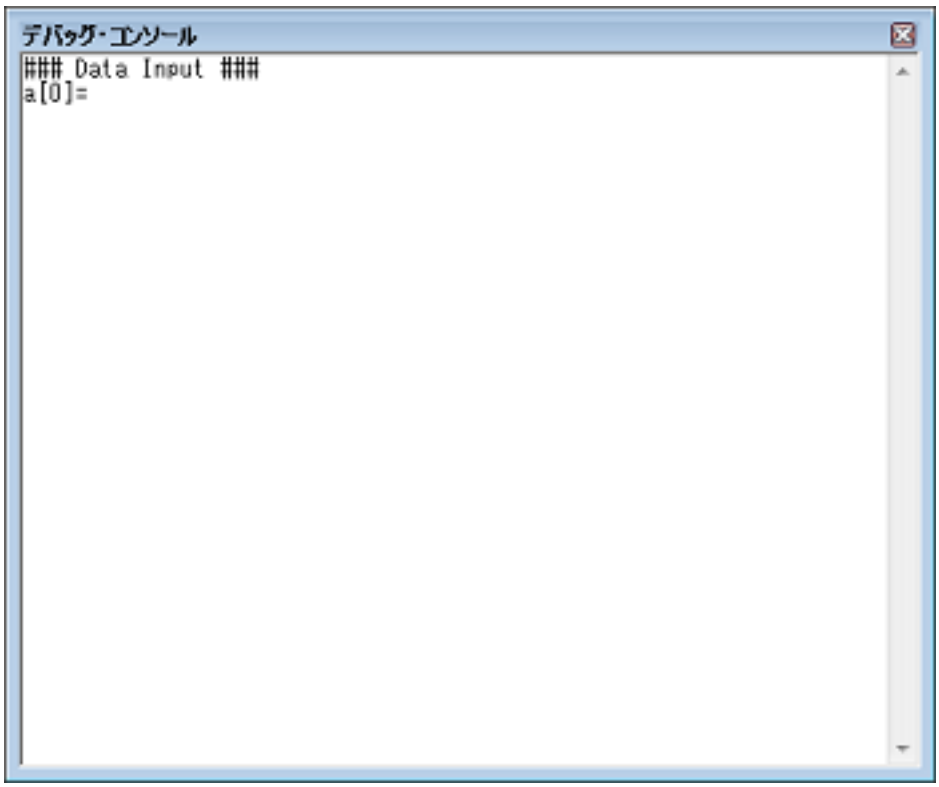
(c)　［ストリーム入出力用アドレス］
このプロパティは，［ストリーム入出力モードを選択］プロパティにおいて［シミュレータモード］を選択した場合にのみ表示されます。
低水準インタフェース・ルーチン（アセンブリ言語部分）に，シミュレータ用のサンプルを使用している場合は，“0”（デフォルト）を指定してください。

(5)　ダウンロードを実行する
メイン・ウインドウ上の［デバッグ］メニュー→［ビルド＆デバッグ・ツールへダウンロード］を選択します。これにより，デバッグ・ツールと接続したのち，ビルド／ダウンロードを行います（「2.5.1 ダウンロードを実行する」参照）。

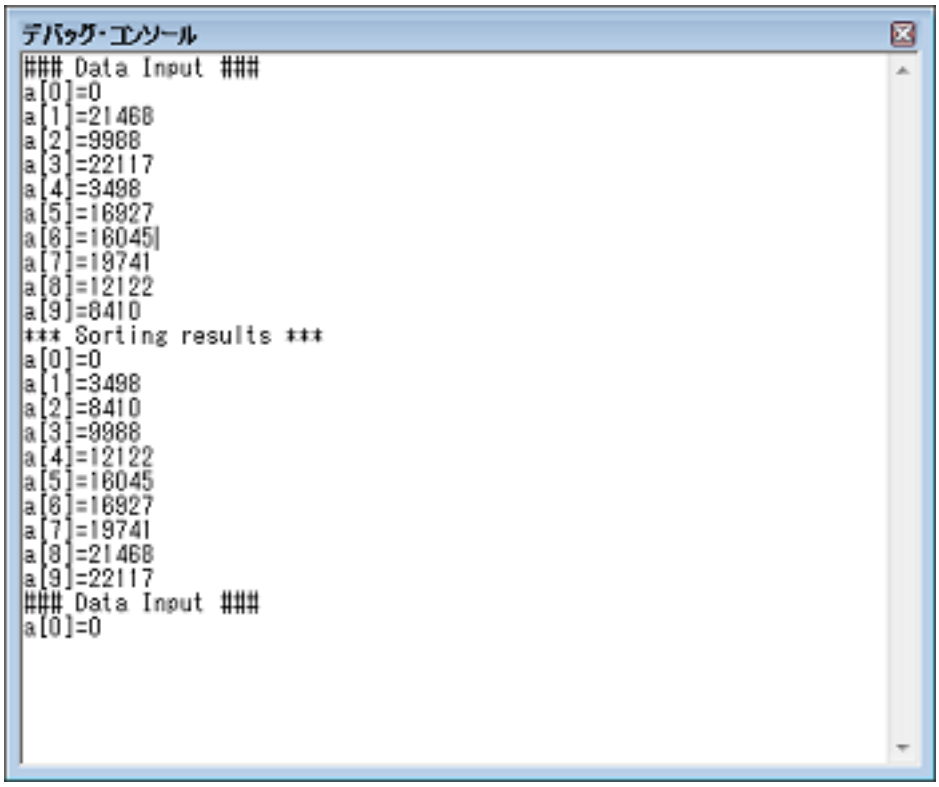
(6)　デバッグ・コンソール パネルをオープンする
デバッグ・コンソール パネルをオープンします。

(7)　プログラムを実行する
プログラムを実行します（「2.9 プログラムの実行」参照）。

(8)　デバッグ・コンソール パネル上でデータの入出力を行う
プログラムを実行することにより，デバッグ・コンソール パネル上でデータの入出力を行います。
本プログラムは，scanf 関数によりキーボードからパネル上へ入力した 10 個のデータを読み込み，入力データを昇順にソートしたのち，printf 関数によりパネル上に出力します。



- main 関数内の printf 関数により，データを格納する配列 “a” の要素名をパネル上に出力します。
- キーボードによりパネル上へデータを入力すると，main 関数内の scanf 関数によりデータを読み込みます。



- main 関数内の printf 関数により，ソートしたデータをパネル上に出力します。

備考　デバッグ・コンソール パネルのコンテキスト・メニューから［ローカル・エコー・バック有効 / 無効］を選択し，ローカル・エコー・バック機能を無効状態にした場合，入力した文字は表示しません（ローカル・エコー・バックを実行しません）。

2.20 入力値について

この節では，各パネル／ダイアログに値を入力する際の留意事項について説明します。

2.20.1 入力規約

各パネル／ダイアログへの入力規約を次に示します。

(1) 文字セット
入力を許可している文字セットは次のとおりです。

表 2.26　　文字セットの一覧

文字セット	概要
ASCII	半角のアルファベット（英字），半角の数字，および半角の記号
Shift-JIS	全角のアルファベット（英字），全角の数字，全角の記号，ひらがな，全角のカタカナ，漢字，および半角のカタカナ
EUC-JP	全角のアルファベット（英字），全角の数字，全角の記号，ひらがな，全角のカタカナ，漢字，および半角のカタカナ
UTF-8	全角のアルファベット（英字），全角の数字，全角の記号，ひらがな，全角のカタカナ，漢字（中国語を含む），および半角のカタカナ
UTF-16 (Unicode)	全角のアルファベット（英字），全角の数字，全角の記号，ひらがな，全角のカタカナ，漢字（中国語を含む），および半角のカタカナ

(2) エスケープ・シーケンス
入力を許可しているエスケープ・シーケンスは次のとおりです。

表 2.27　　エスケープ・シーケンスの一覧

エスケープ・シーケンス	値	意味
¥0	0x00	null 文字
¥a	0x07	アラート
¥b	0x08	バックスペース
¥t	0x09	水平タブ
¥n	0x0A	改行
¥v	0x0B	垂直タブ
¥f	0x0C	フォーム・フィード
¥r	0x0D	キャリッジ・リターン
¥"	0x22	ダブルクォーテーション
¥'	0x27	シングルクォーテーション
¥?	0x3F	疑問符 ? と入力された場合も疑問符として扱います。
\\	0x5C	バックスラッシュ

(3) 数　　値
数値を入力する際に許可している進数は次のとおりです。

表 2.28　　進数の一覧

進数表記	概要
2 進数	0b で始まり，0 ～ 1 の数値が続く数値 （英字の大文字／小文字については，不問）
8 進数	0 で始まり，0 ～ 7 の数字が続く数値
10 進数	0 以外で始まり，0 ～ 9 の数字が続く数値
16 進数	0x で始まり，0 ～ 9 の数字，および a ～ f の英字が続く数値 （英字の大文字／小文字については，不問） ただし，HEX マークが表示されている入力エリアでは，0x の接頭辞は必要ありません。

(4)　式と演算子

式とは，定数，CPU レジスタ名，I/O レジスタ名，シンボル，およびこれらを演算子で結合したものを示します。

式には，アドレス式とウォッチ式があります。シンボルのアドレスを必要とする式をアドレス式，シンボルの値を必要とする式をウォッチ式と呼びます。

(a)　アドレス式と演算子

アドレス式では，シンボルのアドレスを使用して演算します。CPU レジスタ名が記述された場合のみ，値を使用して演算します。

アドレス式の基本入力形式は次のとおりです。

表 2.29　　アドレス式の基本入力形式

式	説明
C/C++ 言語変数名[注 1，注 2]	C/C++ 言語の変数のアドレス
式 [*式* [注 3]]	配列のアドレス
式 . メンバ名[注 4]	構造体／共用体／クラスのメンバのアドレス
式 -> メンバ名[注 4]	ポインタの指し示す構造体／共用体／クラスのメンバのアドレス
ウォッチ式 .* キャスト式	メンバ変数へのポインタのアドレス
ウォッチ式 ->* キャスト式	メンバ変数へのポインタのアドレス
CPU レジスタ名	CPU レジスタの値
I/O レジスタ名	I/O レジスタのアドレス
ラベル名[注 5]，EQU シンボル名[注 5]，[即値]	ラベルのアドレス，EQU シンボルの値，即値アドレス
整定数	アドレス

注 1.　C89，C99，C++ 言語の変数を表します。

注 2.　C 言語変数の値がレジスタに割り付いている場合は，エラーになります。

注 3.　インデックスとして入力された式は，ウォッチ式として解析します。

注 4.　基底クラスのメンバ変数を指定する場合，メンバ名の前にスコープを指定してください（例：variable.BaseClass::member）。

注 5.　ラベル名または EQU シンボル名に “$” が含まれている場合，名前を “{}” で囲んでください（例：{$Label}）。
“I” は虚数のキーワードとなるため，CPU レジスタの “I” を指定する場合は，“:REG” を付加してください（例：I:REG）。

また，「表 2.29 アドレス式の基本入力形式」から，次の演算子を用いたアドレス式を構成することができます。

表 2.30　　演算子を用いたアドレス式の構成

式	説明
(*式*)	演算順序の指定
- *式*	符号反転
! *式*	論理否定
~ *式*	ビット反転
式 * *式*[注]	乗算
式 / *式*[注]	除算
式 % *式*[注]	剰余算
式 + *式*[注]	加算
式 - *式*[注]	減算
式 & *式*[注]	ビットごとの論理積
式 ^ *式*[注]	ビットごとの排他的論理和
式 | *式*[注]	ビットごとの論理和

注　　変数および関数は，変数，関数，整定数以外と演算子で結合することはできません（例：C 言語変数名 +I/O レジスタ名）。

(b)　ウォッチ式と演算子

ウォッチ式ではシンボルの値を使用して演算します。値が存在しない場合のみ，シンボルのアドレスを使用して演算します（例：main( ) + 1）。

ウォッチ式の基本入力形式は次のとおりです。

表 2.31　　ウォッチ式の基本入力形式

式	説明
C/C++ 言語変数名[注 1]	C/C++ 言語の変数の値
式 [*式*]	配列の要素値
式 . メンバ名[注 2]	構造体／共用体／クラスのメンバ値
式 -> メンバ名[注 2]	ポインタの指し示す構造体／共用体／クラスのメンバ値
ウォッチ式 .* キャスト式	メンバ変数へのポインタの値
ウォッチ式 ->* キャスト式	メンバ変数へのポインタの値
* *式*	ポインタの変数の値
& *式*	配置アドレス
CPU レジスタ名	CPU レジスタの値
I/O レジスタ名	I/O レジスタの値
ラベル名[注 3]，EQU シンボル名[注 3]，[即値]	ラベルの値，EQU シンボルの値，即値アドレスの値
整定数	整数の定数値
浮動定数	浮動小数点の定数値
文字定数	文字定数値

注 1.　　C89，C99，C++ 言語の変数を表します。

注 2.　基底クラスのメンバ変数を指定する場合，メンバ名の前にスコープを指定してください。
（例：variable.BaseClass::member）

注 3.　ラベル名または EQU シンボル名に “$” が含まれている場合，名前を “{}” で囲んでください（例：{$Label}）。
虚数の値には，大文字の “I” を掛けてください（例：1.0 + 2.0*I）。“I” は虚数のキーワードとなるため，CPU レジスタの “I” を指定する場合は，“:REG” を付加してください（例：I:REG）。

また，「表 2.31 ウォッチ式の基本入力形式」から，次の演算子を用いたウォッチ式を構成することができます。次表の演算子は，C 言語仕様に従って式を解析します。

表 2.32　演算子を用いたウォッチ式の構成

式	説明
(*式*)	演算順序の指定
! *式*	論理否定
~ *式*	ビット反転
式 * *式*[注]	乗算
式 / *式*[注]	除算
式 % *式*[注]	剰余算
式 + *式*[注]	加算
式 - *式*[注]	減算
式 & *式*[注]	ビットごとの論理積
式 ^ *式*[注]	ビットごとの排他的論理和
式 | *式*[注]	ビットごとの論理和

注　変数および関数は，変数，関数，整定数以外と演算子で結合することはできません（例：C 言語変数名 +I/O レジスタ名）。

2.20.2 シンボル名の入力補完機能

シンボル名の入力補完とは，アドレス式などを入力する際に，プログラム中に存在するシンボル名のリストから 1 つを選択することにより，ユーザのシンボル名の入力作業を補佐する機能です。

シンボル名のリストは，この機能に対応するテキスト・ボックスにおいて，目的のシンボル名の一部が入力されている状態で［Ctrl］+［Space］キーを押下することにより表示されます。リスト内において，目的のシンボル名をダブルクリックすることで（または，［↑］／［↓］キーによりシンボル名を選択したのち［Space］／［Enter］キーを押下），入力中のシンボル名が補完されます。

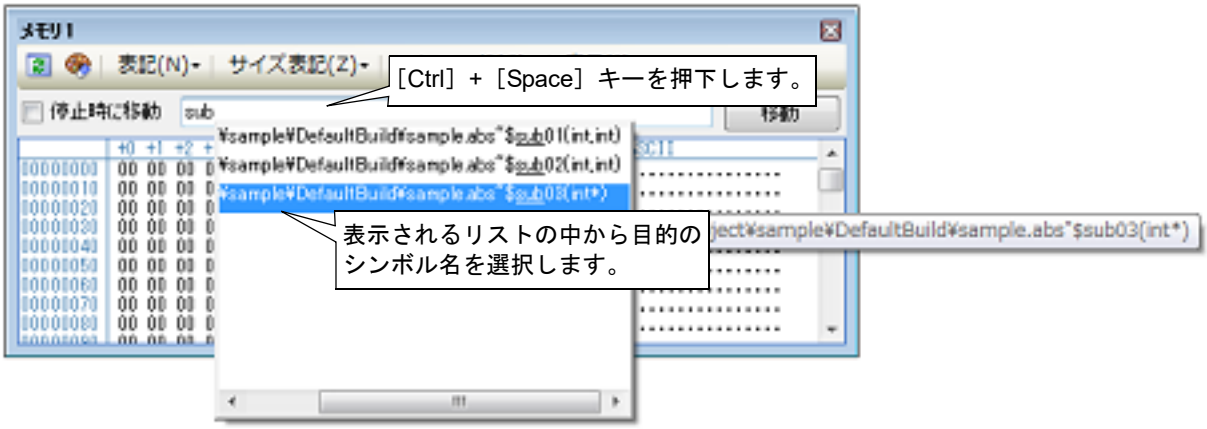
なお，この際に，［Space］／［Enter］キー以外のキーが押下された場合，または現在操作対象としているパネル／ダイアログからフォーカスが移動した場合は，シンボル名のリストは消失します（シンボル名の入力補完は行われません）。

注意 1.　テキスト・ボックスにおいて，1 文字も入力されていない場合，または候補が 1 つも存在しない場合は，シンボル名のリストは表示されません。

注意 2.　コンパイラの最適化により，関数の引数の型がプログラム上で定義した型と異なる型に変換されることがあります（例：func (class Foo obj); → func (class Foo *obj);）。
この場合，シンボル名の入力補完機能では最適化後の関数名を出力します。

備考　シンボル名の入力時にこの機能を使用できるか否かは，該当するパネル／ダイアログの入力エリアの説明を参照してください。

図 2.184 シンボル名の入力補完機能



2.20.3 入力不備箇所に対するアイコン表示

CS+ が提供する一部のダイアログでは，不正な文字列が入力された際，設定すべき値として誤っていることを示すアイコンを該当箇所に表示することにより，入力の不備を警告します。

備考 アイコン上にマウス・カーソルを移動した際には，入力すべき文字列に関する情報がポップアップ表示されます。

2.21 E1/E20/EZ エミュレータとマイクロコントローラの相違点

(1) レジスタ初期値
E1/E20/EZ エミュレータは，システム起動時に汎用レジスタやコントロールレジスタの一部を初期化していますのでご注意ください（表 2.33）。なお，マイクロコントローラの初期値は不定です。

表 2.33　E1/E20/EZ エミュレータでのレジスタ初期値

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

注 1.　ACC レジスタの b15-b0 は，常に “0” が読み出されます。B15-b0 への書き込みは無視されます。

注 2.　FPU を搭載していないマイクロコントローラには，浮動少数点ステータスワード (FPSW) はありません。

注意　[CPU レジスタ] パネルにてレジスタ値を変更した場合，実際にレジスタに反映されるのはユーザプログラムの実行開始直前になります。debugger.Register.SetValue コマンドによる変更も同様です。
[CPU レジスタ] パネルや debugger.Register.SetValue コマンドによりレジスタ値を変更した後にユーザプログラムの実行をせず CPU リセットした場合は変更したレジスタ値は無効となります。

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

(3) リセット信号
ユーザプログラム停止中に端子リセットによるリセットが発生した場合，JTAG 接続ではリセット信号がマスクされます。なお，FINE 接続ではマスクされずマイクロコントローラがリセットされます。

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

(4) ダイレクトメモリアクセスコントローラ（DMAC）
DMAC はブレーク中でも停止しません。このため，転送要求が発生すると，DMA 転送を実行します。使用上の注意事項の「2.22.8 DMAC，DTC に関して」も参照してください。

(5) WDT の使用について
ブレーク中は，WDT のカウントは停止します。プログラム実行再開時に，カウントも再開します。

(6) クロック発生回路関連レジスタの変更とデバッグ機能の競合
ユーザプログラム実行中にクロック発生回路の関連レジスタの値をメモリウィンドウ等から変更しないでください。クロック発生回路の関連レジスタについては，ご使用になるマイクロコントローラのハードウェアマニュアルを参照してください。

(7) マスカブル割り込みに関して

- ユーザプログラム停止中（ランタイムデバッグ中を含む）であっても，タイマなどの機能も動作しています。ユーザプログラム停止中（ランタイムデバッグ中を含む）は，エミュレータで割り込みを禁止しているため，

マスカブル割り込みの要求が発生しても受け付けられません。この割り込み要求は，ユーザプログラムの実行を開始した直後に受け付けられます。

- ユーザプログラム停止中（ランタイムデバッグ中を含む）は，周辺 I/O の割り込み要求が受け付けられませんのでご注意ください。

(8) FCU RAM へのファームウェア転送
E1/E20/EZ エミュレータは，システム起動時に FCU RAM へファームウェアを転送しています。

2.22 その他の使用上の注意事項【E1】【E20】【EZ Emulator】

2.22.1 内蔵フラッシュROM 領域

E1/E20/EZ エミュレータでのデバッグ終了後に，エミュレータを取り外してマイクロコントローラ単体で動作させることは保証しておりません。マイクロコントローラ単体で動作させる場合には，フラッシュプログラマを使用して，改めてプログラムを書き込んでください。

また，E1/E20/EZ エミュレータを接続してデバッグに使用したマイクロコントローラは，エミュレーション時にフラッシュメモリの書き込みを繰り返しておりストレスがかかっています。デバッグに使用したマイクロコントローラはユーザの量産製品には使用しないでください。

2.22.2 FINE インタフェース

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

【RX62x, RX630, RX631, RX63N, RX63T, RX634】

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

【RX71M, RX651, RX65N, RX64M, RX210, RX21A, RX220, RX23x, RX24T, RX110, RX111, RX113, RX130】

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

2.22.3 エンディアン選択レジスタ（MDEB, MDES）について【RX71M, RX651, RX65N, RX64M, RX62x, RX630, RX631, RX63N, RX63T, RX634, RX210, RX21A, RX220, RX23x, RX24T, RX110, RX111, RX113, RX130】

エンディアン選択レジスタ（MDEB, MDES）は，プロパティ パネルの［接続用設定］タブ上の［CPU の動作モード］【E1】【E20】【EZ Emulator】カテゴリ内 [エンディアン] プロパティ（シミュレータの場合は，プロパティ パネルの［接続用設定］タブ上の［エンディアン］【シミュレータ】カテゴリ内 [CPU エンディアン] プロパティ）にて設定するエンディアン情報がデバッグする動作モードに対応したレジスタへ書き込まれます。

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

2.22.4 オプション機能選択レジスタ1（OFS1）の設定について【RX62x, RX630, RX631, RX63N, RX63T, RX634, RX210, RX21A, RX220, RX23x, RX24T, RX130】

- ダウンロード以外の操作（メモリ パネルからの操作やラインアセンブル）でオプション機能選択レジスタ 1（OFS1）を書き換える場合，設定を反映するためには以下の手順が必要です。
  ① OFS1 を書き換える
  ② 実行操作（OFS1 への書き込みが発生）
  ③ CPU のリセットを実施
  ダウンロード操作の場合はダウンロード後に CPU のリセットを実施した時点で反映されます。
- オプション機能選択レジスタ 1（OFS1）では，電圧監視 0 リセットの有効／無効と電圧検出レベルを設定することができます。本レジスタはフラッシュメモリ上にあり，設定によって E1/E20/EZ エミュレータで制御できない可能性があります。
  フラッシュ ROM を書き換える操作（ダウンロードやメモリ パネルからの操作など）の時にオプション機能選択レジスタ 1（OFS1）の設定値が電圧監視 0 リセットを発生させる条件の場合，エラーが発生します。
  オプション機能選択レジスタ 1（OFS1）については，マイクロコントローラのハードウェアマニュアルを参照してください。
- FINE 通信を使用する場合，エミュレータが強制的に OFS1 レジスタ bit25-bit24 を 10b に書き換えます。ユーザ・プログラムで OFS1 レジスタ bit25-bit24 を書き換えないでください。

2.22.5 オプション機能選択レジスタ 1（OFS1）の設定について【RX110, RX111, RX113】

- ダウンロード以外の操作（メモリ パネルからの操作やラインアセンブル）でオプション機能選択レジスタ 1（OFS1）を書き換える場合，設定を反映するためには以下の手順が必要です。
  ① OFS1 を書き換える
  ② 実行操作（OFS1 への書き込みが発生）
  ③ CPU のリセットを実施
  ダウンロード操作の場合はダウンロード後に CPU のリセットを実施した時点で反映されます。
- 電圧監視 1 リセット機能は使用できません。
  ダウンロード操作でオプション機能選択レジスタ 1（OFS1）を電圧監視 1 リセット有効に設定する場合，エミュレータが強制的に電圧監視 1 リセットを無効にした値に書き換えます。
  ダウンロード以外の操作（メモリ パネルからの操作やラインアセンブル）でオプション機能選択レジスタ 1（OFS1）を電圧監視 1 リセット有効に設定する場合，エラーが発生します。
- FINE 通信を使用する場合，エミュレータが強制的に OFS1 レジスタ bit25-bit24 を 10b に書き換えます。ユーザ・プログラムで OFS1 レジスタ bit25-bit24 を書き換えないでください。

2.22.6 オプション機能選択レジスタ 1（OFS1）の設定について【RX71M, RX651, RX65N, RX64M】

- ダウンロード以外の操作（メモリ パネルからの操作やラインアセンブル）でオプション機能選択レジスタ 1（OFS1）を書き換えることはできません。
- 電圧監視 0 リセット機能は使用できません。
  ダウンロード操作で OFS1 レジスタを電圧監視 0 リセット有効に設定する場合，エミュレータが強制的に電圧監視 0 リセットを無効にした値に書き換えます。
- FINE 通信を使用する場合，エミュレータが強制的に OFS1 レジスタ bit25-bit24 を 10b に書き換えます。ユーザ・プログラムで OFS1 レジスタ bit25-bit24 を書き換えないでください。

2.22.7 フラッシュメモリ書き換え時のレジスタ値

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

2.22.8 DMAC，DTC に関して

下記のどちらかの条件もしくは両方を満たすとき，

- DMAC，DTC の転送元がマイクロコントローラ内蔵フラッシュ ROM に指定されている
- DMAC，DTC 起動要因の割り込みベクタまたは DTC ベクタテーブルがマイクロコントローラ内蔵フラッシュ ROM に配置されている

ユーザプログラム停止中に DMAC，DTC 要求が発生する場合は下記操作をしないでください。

- マイクロコントローラ内蔵フラッシュ ROM 領域へのソフトウエア・ブレークポイントの設定
- メモリ パネルや Python コンソール パネルからマイクロコントローラ内蔵フラッシュ ROM 領域への書き込み
- マイクロコントローラ内蔵フラッシュ ROM 領域へのダウンロード

2.22.9 高速クロック発振器（HOCO）に関して【RX71M, RX64M, RX210, RX21A, RX220】

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

- デバッガでユーザブートモード起動する場合，高速オンチップオシレータ電源コントロールレジスタ (HOCOPCR) で，HOCO の電源を OFF に設定しないでください。HOCO の電源を OFF にした場合，エミュレータからの制御ができなくなります。

【RX71M, RX64M】

- 本制限事項は FINE 接続時のみ該当します。

2.22.10 ロックビット【ロックビット実装マイクロコントローラ】

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

(2) ダウンロード完了後のロックビット復帰について
プロパティ パネルの［デバッグ・ツール設定］タブ上の［システム］【E1】【E20】【EZ Emulator】カテゴリ内 [内蔵プログラム ROM を書き換えるプログラムをデバッグする] プロパティの設定により動作が異なります。

- [はい] の場合
  ダウンロード完了後にロックビットを元の設定に復帰します。
- [いいえ] の場合
  ダウンロード完了後にロックビットを再設定せず解除したままとなります。

2.22.11 エリアプロテクション【エリアプロテクション実装マイクロコントローラ】

(1) ダウンロード時のエリアプロテクション解除について
エリアプロテクションの設定により書き換えのできないフラッシュ ROM 領域に対して，ユーザ・プログラムをダウンロードする場合，E1/E20/EZ エミュレータはエリアプロテクションを解除してからダウンロードします。

(2) ダウンロード完了後のエリアプロテクション復帰について
プロパティ パネルの［デバッグ・ツール設定］タブ上の［システム］【E1】【E20】【EZ Emulator】カテゴリ内 [内蔵プログラム ROM を書き換えるプログラムをデバッグする] プロパティの設定により動作が異なります。

- [はい] の場合
  ダウンロード完了後にエリアプロテクションを元の設定に復帰します。
- [いいえ] の場合
  ダウンロード完了後にエリアプロテクションを再設定せず解除したままとなります。

【RX651, RX65N】
[内蔵プログラム ROM を書き換えるプログラムをデバッグする] プロパティの設定に関係なく，ダウンロード完了後にエリアプロテクションを元の設定に復帰します。

2.22.12 動作周波数について

E1/E20/EZ エミュレータにてデバッグ可能なマイクロコントローラの最低動作周波数は 32.768kHz です。

2.22.13 スタートアッププログラム保護機能について【RX23T, RX24T, RX110, RX111, RX113, RX130】

- スタートアッププログラム保護機能を使用するユーザ・プログラムをデバッグする場合は，プロパティ パネルの［デバッグ・ツール設定］タブ上の［システム］【E1】【E20】【EZ Emulator】カテゴリ内 [内蔵プログラム ROM を書き換えるプログラムをデバッグする] プロパティの設定を [はい] にしてください。
- スタートアッププログラム保護機能を使用するユーザ・プログラムをデバッグしている時にシステムリセットを発行した場合，再度ユーザ・プログラムをダウンロードし直してからデバッグを再開してください。
- ユーザ・プログラム実行中に下記いずれかを実施した後は，「CPU のリセット」，「端子リセット」，「内部リセット」はできません。
  - R_FCL_ChangeSwapFlag 関数をコールする
  - フラッシュエクストラ領域制御レジスタ (FEXCR) を操作してスワップする

一度デバッガを終了し，再起動した後はデバッグすることが可能です。

- ユーザ・プログラム実行中に下記いずれかを実施した後は，プログラム停止中の「端子リセット」はできません。プログラム実行中の「端子リセット」は可能です。
  - R_FCL_ChangeSwapState 関数をコールする
  - フラッシュ初期設定レジスタ (FISR) を操作して即座にスワップする
- ユーザ・プログラム停止中にフラッシュ初期設定レジスタ（FISR）を操作して即座にスワップしないでください。

2.22.14 MPU 領域のアクセスについて【MPU 実装マイクロコントローラ】

MPU 領域の参照および設定はプログラム停止中のみ可能です。
プログラム実行中の MPU 領域アクセスは以下になります。

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

MPU 領域に対して以下の操作は実施できません。

- ラインアセンブル
- メモリ検索
- メモリコピー

2.22.15 リセットについて

- 【EZ Emulator】
  内部リセットが発生した場合に，エミュレータからの制御ができなくなります。ウォッチドッグタイマなど内部リセットは発生させないでください。
- 【EZ Emulator】
  ディープソフトスタンバイからの復帰時は内部リセットが発生するため，エミュレータから制御できなくなります。
- 【EZ Emulator】
  実行中に端子リセットを入力した場合に，プログラムが暴走をする場合があります。端子リセット入力を検出した場合，リセットを入力し直して再実行させますが，再実行できずブレークする場合があります。
- 【E1】【E20】【EZ Emulator】
  端子リセットやウォッチドッグタイマなどによるリセットとエミュレータシステムによるマイクロコントローラに対する操作（メモリ パネルでメモリ参照など）が競合した場合，“タイムアウトエラーが発生しました。MCU がリセット状態です。システムリセットを発行しました。”とエラーメッセージが表示され，エミュレータを初期化してユーザ・プログラムが停止します。システムリセット発行後は，トレース記録も初期化します。なお，デバッグを継続して行うことが可能です。
- 以下に示すどちらかの条件で端子リセットまたは内部リセットが発生した場合，マイクロコントローラの動作モードおよびエミュレータ通信インタフェースに依存した注意事項があります。端子リセット発生による注意事項を表 2.34 に，内部リセット発生による注意事項を表 2.35 に示します。

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

表 2.34　　端子リセット発生による注意事項

グループ	インタフェース	動作モード	ユーザ・プログラム実行中の端子リセット発生による注意事項
RX610, RX621, RX62N	JTAG	内蔵 ROM 無効拡張	リセットはエミュレータが解除します。このため，実際のマイクロコントローラ単体動作とはタイミングが異なります。
RX71M, RX651, RX65N, RX64M, RX630, RX631, RX63N, RX63T, RX634	JTAG	ユーザブート 内蔵 ROM 無効拡張	
RX71M, RX651, RX65N, RX64M, RX630, RX631, RX63N, RX63T, RX210, RX21A, RX220, RX231, RX23T, RX24T, RX110, RX111, RX113, RX130	FINE	全モード	
RX210, RX21A, RX220, RX231, RX23T, RX24T, RX110, RX111, RX113	FINE	ユーザブート 内蔵 ROM 無効拡張	ユーザ・プログラム実行中に端子リセットによるリセットが発生した場合，パフォーマンス解析カウンタ値およびトレース取得データが初期化されます。

表 2.35　　内部リセット発生による注意事項

グループ	インタフェース	動作モード	ユーザ・プログラム実行中の内部リセット発生による注意事項
RX610, RX621, RX62N, RX630, RX631, RX63N, RX63T, RX634	JTAG	内蔵 ROM 無効拡張	リセット解除後，ユーザ・プログラムで内蔵 ROM 無効拡張モードに設定された後にデバッグ可能となります。
RX71M, RX651, RX65N, RX64M, RX630, RX631, RX63N, RX63T, RX634	JTAG	ユーザブート	内部リセットが発生した場合，エミュレータからの制御ができなくなります。ウォッチドッグタイマなどによる内部リセットを発生させないでください。
RX71M, RX651, RX65N, RX64M, RX630, RX631, RX63N, RX63T, RX210, RX21A, RX220, RX231, RX23T, RX24T, RX110, RX111, RX113, RX130	FINE	全モード	

- 【E1】【E20】【EZ Emulator】
  ソフトウェアリセットを発生させる命令から以下の操作を実施した場合は，ソフトウェアリセットが発生しません。
  - ステップ実行
  - ソフトウェア・ブレークポイントおよびハードウェア・ブレーク（実行前ブレーク）ポイントが設定されている状態で実行

  上記は JTAG 通信のみ該当します。
  FINE 通信ではソフトウェアリセットを発生させないでください。エミュレータからの制御ができなくなります。

2.22.16 内蔵 ROM 無効拡張モード時のメモリアクセスについて

内蔵 ROM 無効拡張モード時は，メモリパネルや IOR パネルで FCU ファーム領域，フラッシュ関連 I/O レジスタをリード・ライトするとバスエラーが発生するため，リード・ライトしないでください。

2.22.17 イベントの組み合わせ条件について【RX71M, RX64M, RX651, RX65N】

イベントの組み合わせ条件がシーケンシャルの場合，連続した命令にイベントを設定しないでください。
命令によってはイベントが成立しない場合があります。

2.22.18 カバレッジ計測機能について【E20【RX71M, RX64M, RX651, RX65N】】

割り込みが発生した後，割り込み発生元に復帰しない処理の場合，割り込み発生箇所から 1 ～ 2 命令先の未実行命令が実行済みと表示されます。

2.22.19 トレース機能について

割り込みが発生した場合，割り込み発生箇所から 1 ～ 2 命令先の未実行命令が割り込み処理の前に表示されます。

2.22.20 Trusted Memory 機能について

【RX71M, RX64M, RX651, RX65N】
Trusted Memory 機能が有効の場合，Trusted Memory 識別データ TMINF を書き換えることはできません。ダウンロードしても書き換わりません。

【RX651, RX65N】
TrustedMemory 機能が有効の場合，コードフラッシュメモリのデュアルバンク機能でリニアモードとデュアルモードの切り替えができません。また，デュアルモードを選択時に，TrustedMemory 機能が有効（TMEF.TMEF ビット =00）かつ FFEE0000h ～ FFEEFFFFh の TrustedMemory 機能が無効（TMEF.TMEFDB ビット =111b）の場合，起動バンクの切り替えができません。
そのため，TrustedMemory 機能が有効なことが要因でデバッグ・ツールへの接続時にリニアモードとデュアルモードの切り替えやデュアルモード時の起動バンクが切り替えできなかった場合，デバッグ・ツールへの接続ができません。
なお，本注意事項は RX65x の ROM 2MB および 1.5MB のみ該当します。RX65x-1MB 以下は該当しません。

2.22.21 FAW レジスタの FSPR ビットについて【RX651, RX65N】

FAW レジスタの FSPR ビットをエミュレータからプロテクト設定状態に書き換えることはできません。
また，FAW レジスタの FSPR ビットによるプロテクトが設定されたデバイスに対して，FAW レジスタの FAWS と FAWE ビットで指定したフラッシュアクセスウィンドウ対象領域以外へのダウンロードと，FAW レジスタへのダウンロードができません。
このデバイスを使用してデバッグを継続する場合は，プロパティ パネルの［ダウンロード・ファイル設定］タブ上の［ダウンロード］カテゴリ内 [ダウンロード前にフラッシュ ROM を消去する] プロパティの設定を [いいえ] にしてください。

2.22.22 ROM キャッシュ機能について【RX651, RX65N, RX24T】

フラッシュ書き換えを伴うデバッグを行う場合，プログラム実行中に端子リセットまたは内部リセットが入ると，逆アセンブル表示やメモリ表示が古いまま更新されていない場合があります。
ユーザ・プログラム停止後に手動で ROM キャッシュ無効化レジスタ (ROMCIV) を操作し，ROM キャッシュの無効化を行ってください。

2.22.23 コードフラッシュメモリのデュアルバンク機能について【RX651, RX65N】

RX65N/RX651 の ROM 2MB 品および 1.5MB 品に，コードフラッシュメモリのデュアルバンク機能があります。リニアモードまたはデュアルモードの選択はプロジェクト設定で行い，デュアルモード選択時の起動バンクの選択はプロパティ パネルの［接続用設定］タブ上の［CPU の動作モード］【E1】【E20】【EZ Emulator】カテゴリ内 [起動バンク] プロパティで行います。デバッグ・ツールへの接続時は選択したモードに合わせて MDE レジスタや BANKMD レジスタを書き換えて起動します。

2.22.24 デュアルモード時のデバッグについて【RX651, RX65N】

ダウンロードデータ中の BANKMD レジスタの設定値とプロパティ パネルの［接続用設定］タブ上の［CPU の動作モード］【E1】【E20】【EZ Emulator】カテゴリ内 [起動バンク] プロパティで選択した起動バンクが整合していない場合，CS+ の出力 パネルに警告メッセージを表示します。
デバッグ・ツールを切断せずにリセットすると起動バンクが切り替わるため，デバッグ情報の不整合が発生します。そのため，デバッグを継続できません。WDT リセットなどの内部リセットが発生した場合もデバッグを継続できません。

起動バンクが切り替わった後のデバッグを行う場合は，一旦デバッグ・ツールを切断し，その起動バンク用のプロジェクトを読み込んでデバッグしてください。
なお，本注意事項は RX65x の ROM 2MB および 1.5MB のみ該当します。RX65x-1MB 以下は該当しません。

2.22.25 低消費電力モード中のメモリアクセスについて

ユーザプログラムが低消費電力モード中の場合，デバイスからの応答が返らないため，メモリ パネル等で RAM や I/O レジスタの表示を更新できません。
デバッグ・ツールで事前に低消費電力モード中と判断できた場合は，そのメモリアクセスを抑止しますが，低消費電力モードへの移行直前などのタイミングと重なるとデバイスからの応答が返らなくなり，デバイスとの通信が途絶えます。デバイスとの通信を復帰するために内部でリセットを発行するため，低消費電力モードに遷移するプログラムをデバッグする場合は，プロパティ パネルの［デバッグ・ツール設定］タブ上の［実行中のメモリ・アクセス］カテゴリ内 [実行中に表示更新を行う] プロパティを [いいえ] にしてください。

A. ウインドウ・リファレンス

この付録では，CS+ でデバッグを行う際に使用するウインドウ／パネル／ダイアログについての詳細を説明します。

A.1 説　　明

次に，デバッグに関するウインドウ／パネル／ダイアログの一覧を示します。

表 A.1　　ウインドウ／パネル／ダイアログ一覧

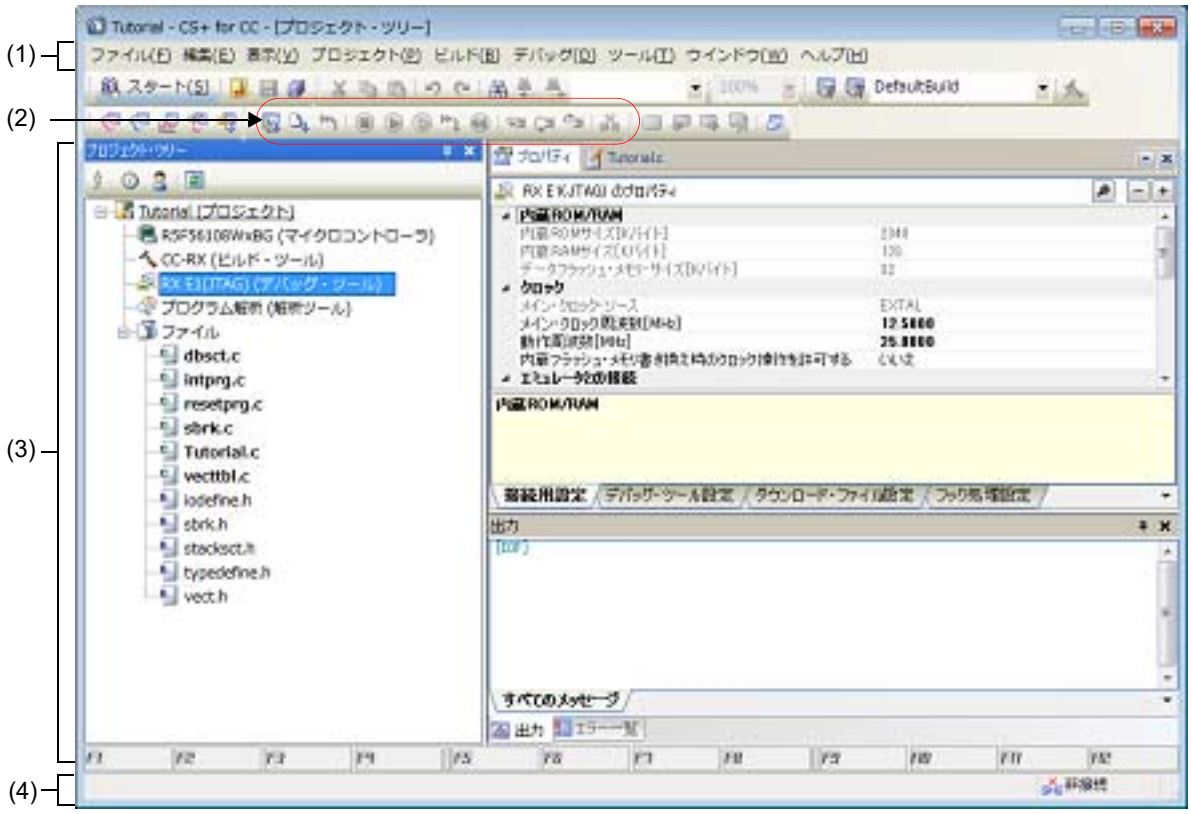
ウインドウ／パネル／ダイアログ名	機能概要
メイン・ウインドウ	プログラムの実行制御，および各パネルのオープン
プロジェクト・ツリー パネル	使用するデバッグ・ツールの選択
プロパティ パネル	プロジェクト・ツリー パネルで選択しているデバッグ・ツールについて，詳細情報の表示，および設定の変更
メモリ パネル	メモリの値の表示，および値の変更
逆アセンブル パネル	メモリ値を逆アセンブルした結果の表示，ライン・アセンブル，および命令レベル・デバッグ
CPU レジスタ パネル	CPU レジスタ（プログラム・レジスタ／システム・レジスタ）の内容の表示，および値の変更
IOR パネル	I/O レジスタの内容の表示，および値の変更
ローカル変数 パネル	ローカル変数の内容の表示，および値の変更
ウォッチ パネル	登録したウォッチ式の内容の表示，および値の変更
コール・スタック パネル	関数呼び出しのコール・スタック情報の表示
トレース パネル	デバッグ・ツールから取得したトレース・データの表示
イベント パネル	設定イベントの詳細情報の表示，有効／無効の切り替え，および削除
出力 パネル	ビルド・ツール／デバッグ・ツール／各プラグインから出力されるメッセージ，または検索・置換 ダイアログ による一括検索を行った際の結果の表示
デバッグ・コンソール パネル	標準入出力
メモリ・マッピング ダイアログ	メモリ・マッピングの設定
ダウンロード・ファイル ダイアログ	ダウンロードする際のファイルの選択，およびダウンロード条件の設定
アクション・イベント ダイアログ	アクション・イベントの設定
表示桁数設定 ダイアログ	メモリ パネルにおけるメモリ値の表示桁数の設定
アドレス・オフセット設定 ダイアログ	メモリ パネルにおけるアドレス表示のオフセット値の設定
メモリ初期化 ダイアログ	メモリの初期化
メモリ検索 ダイアログ	メモリの検索
印刷アドレス範囲設定 ダイアログ	逆アセンブル パネルにおける印刷範囲の設定
トレース検索 ダイアログ	トレース・データの検索
組み合わせ条件設定ダイアログ【E1】【E20】【EZ Emulator】	イベント組み合わせ条件の表示，および設定の変更

ウインドウ／パネル／ダイアログ名	機能概要
タイマ計測詳細設定ダイアログ【E1】【E20】【EZ Emulator】	タイマ・イベントに関する詳細情報の表示，および設定の変更
実行イベント詳細設定ダイアログ	実行系イベントに関する詳細情報の表示，および設定の変更
アクセスイベント詳細設定ダイアログ	アクセス系イベントに関する詳細情報の表示，および設定の変更
割り込みイベント詳細設定ダイアログ【シミュレータ】	割り込みイベントに関する詳細情報の表示，および設定の変更
COM ポート設定 ダイアログ	デバッグ・コンソール パネルにおける COM ポートの設定
スクロール範囲設定 ダイアログ	メモリ パネル／逆アセンブル パネルにおけるスクロール範囲の設定
指定位置へ移動 ダイアログ	指定した位置にキャレットを移動
データ保存 ダイアログ	各パネルの表示内容，およびアップロード・データの保存
関数変数アクセス表 パネル	変数にアクセスしている関数の直交表での表示

メイン・ウインドウ

CS+ を起動した際，最初にオープンするウインドウです。
デバッグを行う際は，このウインドウからプログラムの実行制御，および各パネルのオープン操作を行います。

図 A.1 メイン・ウインドウ



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］

［オープン方法］

- Windows の［スタート］→［すべてのプログラム］→［Renesas Electronics CS+］→［CS+ for CC(RL78,RX,RH850)］を選択

備考 Windows 8.1 の場合は，スタート画面の［CS+ for CC (RL78,RX,RH850)］を選択してください。
Windows 10 の場合は，Windows の［スタート］メニューから［すべてのアプリ］→［Renesas Electronics CS+］→［CS+ for CC (RL78,RX,RH850)］を選択してください。

［各エリアの説明］

(1) メニューバー
デバッグ関連のメニュー項目は次のとおりです。

備考 各メニューから引き出される項目は，ユーザ設定 ダイアログでカスタマイズすることができます。

(a) ［表示］
［表示］メニューの各項目，および機能は次のとおりです（デフォルト）。

ウォッチ	ウォッチ パネルをオープンするために，次のカスケード・メニューを表示します。 ただし，デバッグ・ツールと切断時は無効となります。
ウォッチ 1	ウォッチ パネル（ウォッチ 1）をオープンします。
ウォッチ 2	ウォッチ パネル（ウォッチ 2）をオープンします。
ウォッチ 3	ウォッチ パネル（ウォッチ 3）をオープンします。
ウォッチ 4	ウォッチ パネル（ウォッチ 4）をオープンします。
ローカル変数	ローカル変数 パネルをオープンします。 ただし，デバッグ・ツールと切断時は無効となります。
コール・スタック	コール・スタック パネルをオープンします。 ただし，デバッグ・ツールと切断時は無効となります。
メモリ	メモリ パネルをオープンするために，次のカスケード・メニューを表示します。 ただし，デバッグ・ツールと切断時は無効となります。
メモリ 1	メモリ パネル（メモリ 1）をオープンします。
メモリ 2	メモリ パネル（メモリ 2）をオープンします。
メモリ 3	メモリ パネル（メモリ 3）をオープンします。
メモリ 4	メモリ パネル（メモリ 4）をオープンします。
IOR	IOR パネルをオープンします。 ただし，デバッグ・ツールと切断時は無効となります。
CPU レジスタ	CPU レジスタ パネルをオープンします。 ただし，デバッグ・ツールと切断時は無効となります。
トレース	トレース パネルをオープンします。 ただし，デバッグ・ツールと切断時は無効となります。
逆アセンブル	逆アセンブル パネルをオープンするために，次のカスケード・メニューを表示します。 ただし，デバッグ・ツールと切断時は無効となります。
逆アセンブル 1	逆アセンブル パネル（逆アセンブル 1）をオープンします。
逆アセンブル 2	逆アセンブル パネル（逆アセンブル 2）をオープンします。
逆アセンブル 3	逆アセンブル パネル（逆アセンブル 3）をオープンします。
逆アセンブル 4	逆アセンブル パネル（逆アセンブル 4）をオープンします。
イベント	イベント パネルをオープンします。 ただし，デバッグ・ツールと切断時は無効となります。
デバッグ・コンソール	デバッグ・コンソール パネルをオープンします。 ただし，デバッグ・ツールと切断時は無効となります。
現在の PC 位置を開く	カレント PC 位置（PC レジスタ値）をエディタ パネルで表示します。 ただし，デバッグ・ツールと切断時は無効となります。
ジャンプ前の位置へ戻る	定義箇所へジャンプ（「2.6.2.4 シンボル定義箇所へ移動する」参照）する前の位置へ戻ります。
ジャンプ先の位置へ進む	［ジャンプ前の位置へ戻る］を実行する前の位置へ進みます。
タグ・ジャンプ	エディタ パネル／出力 パネルにおいて，キャレットのある行にファイル名／行／桁の情報がある場合，該当するファイルの該当行／該当桁へジャンプします。

(b) ［デバッグ］
［デバッグ］メニューの各項目，および機能は次のとおりです（デフォルト）。

デバッグ・ツールへダウンロード	アクティブ・プロジェクトで現在選択しているデバッグ・ツールに，指定されたファイルをダウンロードします。 デバッグ・ツールと切断時の場合は，自動的にデバッグ・ツールに接続し，ダウンロードを実行します。 ただし，プログラム実行中，またはビルド（ラピット・ビルドを除く）実行中は無効となります。
ビルド＆デバッグ・ツールへダウンロード	プロジェクトのビルドを行い，ビルド後にアクティブ・プロジェクトで現在選択しているデバッグ・ツールにダウンロードを実行します。 デバッグ・ツールと切断時の場合は，自動的にデバッグ・ツールに接続し，ダウンロードを実行します。 ただし，プログラム実行中，またはビルド（ラピット・ビルドを除く）実行中は無効となります。 なお，ビルドに失敗した場合，ダウンロードは実行しません。
リビルド＆デバッグ・ツールへダウンロード	プロジェクトのリビルドを行い，リビルド後にアクティブ・プロジェクトで現在選択しているデバッグ・ツールにダウンロードを実行します。 デバッグ・ツールと切断時の場合は，自動的にデバッグ・ツールに接続し，ダウンロードを実行します。 ただし，プログラム実行中，またはビルド（ラピット・ビルドを除く）実行中は無効となります。 なお，リビルドに失敗した場合，ダウンロードは実行しません。
デバッグ・ツールへ接続	アクティブ・プロジェクトで現在選択しているデバッグ・ツールに接続します。 ただし，デバッグ・ツールと接続時，ビルド（ラピット・ビルドを除く）実行中，またはサポート範囲外のバージョンのコンパイラを使用している場合は無効となります。
ホット・プラグイン 【E1(JTAG)】 【E20(JTAG)】	現在実行中のターゲット・システムをデバッグするために，アクティブ・プロジェクトで現在選択しているデバッグ・ツールとホット・プラグイン接続します（「2.4.3 CS+ にデバッグ・ツールをホットプラグイン接続する【E1(JTAG)】【E20(JTAG)】」参照）。 ただし，デバッグ・ツールと接続時は無効となります。
デバッグ・ツールからアップロード ...	メモリ内容をファイルに保存するためのデータ保存 ダイアログをオープンします。 ただし，プログラム実行中，ビルド（ラピット・ビルドを除く）実行中，またはデバッグ・ツールと切断時は無効となります。
デバッグ・ツールから切断	現在接続中のデバッグ・ツールとの通信を切断します。 ただし，プログラム実行中，ビルド（ラピット・ビルドを除く）実行中，またはデバッグ・ツールと切断時は無効となります。
使用するデバッグ・ツール	使用するデバッグ・ツールを選択するためのカスケード・メニューを表示します。 なお，プロジェクトで選択しているマイクロコントローラの種類により，表示されるデバッグ・ツールは異なります。
RX E2 Lite	E2 Lite を使用します。
RX E1(Serial)	E1 を Serial 通信方式で使用します。
RX E1(JTAG)	E1 を JTAG 通信方式で使用します。
RX E20(Serial)	E20 を Serial 通信方式で使用します。
RX E20(JTAG)	E20 を JTAG 通信方式で使用します。
RX EZ Emulator	EZ Emulator を使用します。
RX シミュレータ	シミュレータを使用します。

停止	現在実行中のプログラムを強制的に停止します。 ただし，プログラム停止時，またはデバッグ・ツールと切断時は無効となります。
実行	プログラムをカレント PC 位置から実行し，設定されているブレーク・イベントの条件が成立した場合，実行中のプログラムを停止します。 ただし，プログラム実行中，ビルド（ラピット・ビルドを除く）実行中，またはデバッグ・ツールと切断時は無効となります。
ブレークせずに実行	プログラムをカレント PC 位置から実行し，設定されているブレーク・イベント／アクション・イベントを無視してプログラムの実行を続けます。 ただし，プログラム実行中，ビルド（ラピット・ビルドを除く）実行中，またはデバッグ・ツールと切断時は無効となります。
ステップ・イン	カレント PC 位置からステップ実行し[注]，各パネルの内容を更新します。 関数呼び出しの場合は，呼び出された関数の先頭で停止します。 ただし，プログラム実行中，ビルド（ラピット・ビルドを除く）実行中，またはデバッグ・ツールと切断時は無効となります。
ステップ・オーバー	カレント PC 位置からステップ実行し[注]，各パネルの内容を更新します。 サブルーチン分岐命令による関数呼び出しの場合は，その関数内のソース行／命令すべてを 1 ステップとみなして実行し，関数から戻る箇所まで実行します（サブルーチン分岐命令を実行したときと同じネストになるまで，ステップ実行します）。 なお，サブルーチン分岐命令以外の場合，［ステップ・イン］の選択と同じ動作となります。 ただし，プログラム実行中，ビルド（ラピット・ビルドを除く）実行中，またはデバッグ・ツールと切断時は無効となります。
リターン・アウト	現在の関数からリターンするまで（呼び出し元関数に戻るまで）実行します[注]。 ただし，プログラム実行中，ビルド（ラピット・ビルドを除く）実行中，またはデバッグ・ツールと切断時は無効となります。
CPU リセット	CPU をリセットします（プログラムは実行しません）。 ただし，ビルド（ラピット・ビルドを除く）実行中，またはデバッグ・ツールと切断時は無効となります。
リスタート	CPU をリセットしたのち，リセット番地からプログラムを実行します。 ただし，ビルド（ラピット・ビルドを除く）実行中，またはデバッグ・ツールと切断時は無効となります。
デバッグ・ツールの状態を巻き戻す	最後に自動保存されたデバッグ・ツールの状態に巻き戻します。 なお，巻き戻されるデータは読み書き可能なメモリとレジスタの値に限られます。 このデバッグ機能を使用するにはオプション・ダイアログで設定する必要があります。

注　ステップ実行には，ソース・レベル単位と命令レベル単位の実行方法があります。
詳細は，「2.9.3 プログラムをステップ実行する」を参照してください。

(2) デバッグ・ツールバー

デバッグ・ツールバーは，プログラムの実行を制御するためのコマンドをまとめたボタン群です。
各ボタン，および機能は次のとおりです（デフォルト）。

備考 1.　各ツールバーのボタンは，ユーザ設定 ダイアログでカスタマイズすることができます。また，同ダイアログにより，新規にツールバーを作成することもできます。

備考 2.　ツールバー上を右クリックすることで表示されるコンテキスト・メニューにより，ツールバー上に表示／非表示するグループを選択することができます。

	プロジェクトのビルドを行い，ビルド後にアクティブ・プロジェクトのデバッグ・ツールにダウンロードを実行します。 デバッグ・ツールと切断時の場合は，自動的にデバッグ・ツールに接続し，ダウンロードを実行します。 ただし，ビルドに失敗した場合，ダウンロードは実行されません。 ［デバッグ］メニュー→［ビルド＆デバッグ・ツールへダウンロード］の選択と同等です。

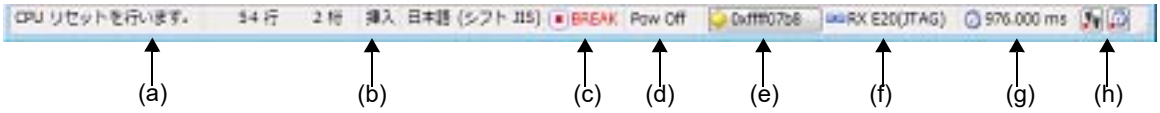
	アクティブ・プロジェクトのデバッグ・ツールに，指定されたファイルをダウンロードします。 デバッグ・ツールと切断時の場合は，自動的にデバッグ・ツールに接続し，ダウンロードを実行します。 ただし，プログラム実行中，またはビルド（ラピット・ビルドを除く）実行中は無効となります。 ［デバッグ］メニュー→［デバッグ・ツールへダウンロード］の選択と同等です。
	CPU をリセットします（プログラムは実行しません）。 ただし，ビルド（ラピット・ビルドを除く）実行中，またはデバッグ・ツールと切断時は無効となります。 ［デバッグ］メニュー→［CPU リセット］の選択と同等です。
	現在実行中のプログラムを強制的に停止します。 ただし，プログラム停止時，またはデバッグ・ツールと切断時は無効となります。 ［デバッグ］メニュー→［停止］の選択と同等です。
	プログラムをカレント PC 位置から実行し，設定されているブレーク・イベントの条件が成立した場合，実行中のプログラムを停止します。 ただし，プログラム実行中，ビルド（ラピット・ビルドを除く）実行中，またはデバッグ・ツールと切断時は無効となります。 ［デバッグ］メニュー→［実行］の選択と同等です。
	プログラムをカレント PC 位置から実行し，設定されているブレーク・イベント／アクション・イベントを無視してプログラムの実行を続けます。 ただし，プログラム実行中，ビルド（ラピット・ビルドを除く）実行中，またはデバッグ・ツールと切断時は無効となります。 ［デバッグ］メニュー→［ブレークせずに実行］の選択と同等です。
	CPU をリセットしたのち，リセット番地からプログラムを実行します。 ただし，ビルド（ラピット・ビルドを除く）実行中，またはデバッグ・ツールと切断時は無効となります。 ［デバッグ］メニュー→［リスタート］の選択と同等です。
	カレント PC 位置からでステップ実行し[注]，各パネルの内容を更新します（ステップ・イン実行）。 関数呼び出しの場合は，呼び出された関数の先頭で停止します。 ただし，プログラム実行中，ビルド（ラピット・ビルドを除く）実行中，またはデバッグ・ツールと切断時は無効となります。 ［デバッグ］メニュー→［ステップ・イン］の選択と同等です。
	カレント PC 位置からステップ実行し[注]，各パネルの内容を更新します（ステップ・オーバー実行）。 サブルーチン分岐命令による関数呼び出しの場合は，その関数内のソース行／命令すべてを 1 ステップとみなして実行し，関数から戻る箇所まで実行します（サブルーチン分岐命令を実行したときと同じネストになるまで，ステップ実行します）。 なお，サブルーチン分岐命令以外の場合，ボタンのクリックと同じ動作となります。 ただし，プログラム実行中，ビルド（ラピット・ビルドを除く）実行中，またはデバッグ・ツールと切断時は無効となります。 ［デバッグ］メニュー→［ステップ・オーバー］の選択と同等です。
	現在の関数からリターンするまで（呼び出し元関数に戻るまで）実行します[注]（リターン・アウト実行）。 ただし，プログラム実行中，ビルド（ラピット・ビルドを除く）実行中，またはデバッグ・ツールと切断時は無効となります。 ［デバッグ］メニュー→［リターン・アウト］の選択と同等です。
	現在接続中のデバッグ・ツールとの通信を切断します。 ただし，ビルド（ラピット・ビルドを除く）実行中，またはデバッグ・ツールと切断時は無効となります。 ［デバッグ］メニュー→［デバッグ・ツールから切断］の選択と同等です。

注　ステップ実行には，ソース・レベル単位と命令レベル単位の実行方法があります。
詳細は，「2.9.3 プログラムをステップ実行する」を参照してください。

(3) パネル表示エリア
各種パネルを表示するエリアです。
表示内容についての詳細は，各パネルの項を参照してください。

(4) ステータス・バー
ステータス・バーは，次の情報を表示します。

図 A.2 ステータス・バー



(a) ステータス・メッセージ
次のメッセージを表示します。

- 選択しているメニュー項目の簡易説明
- パネル／ダイアログにおいて入力値が不正な場合のメッセージ
- 検索・置換 ダイアログにより検索した際に，指定文字列が見つからなかった場合のメッセージ
- ブレークした際のブレーク要因（「2.10 プログラムの停止（ブレーク）」参照）

(b) フォーカス・パネルのステータス情報
現在フォーカスのあるパネルのステータス情報（キャレット位置や上書き／挿入モードなどの情報）を表示します。
ただし，ステータス情報を持たないパネルの場合は非表示となります。

(c) 実行状態
プログラムの現在の実行状態を次のアイコンと文字列で示します。
ただし，デバッグ・ツールと切断時の場合は非表示となります。

プログラムの状態	表示内容
実行中	RUN
停止中	BREAK
ステップ実行中	STEP

(d) CPU 状態【E1(JTAG)】【E20(JTAG)】
デバッグ・ツールの現在の CPU の状態を表示します。
なお，同時に複数の状態になっている場合は “&” で区切って状態を列挙して表示します。
ただし，デバッグ・ツールと切断時の場合は非表示となります。

デバッグ・ツール	表示内容	CPU 状態
E1(JTAG) E20(JTAG)	Reset	リセット状態
	PowOff	ターゲットに電源が供給されていない状態
	Sleep	スリープ・モード中
	Standby	スタンバイ・モード中

備考 CPU が上記以外の状態の場合は何も表示しません。

(e) カレント PC 位置
現在のカレント PC 位置の値を 16 進数で表示します。
このエリアをクリックすると，エディタ パネル上のカレント PC 位置へキャレットを移動します。
なお，プログラム実行中の表示内容は，プロパティ パネルの［デバッグ・ツール設定］タブ上における［レジスタ］カテゴリ内の設定によって次のように切り替わります。

プログラムの状態	［実行中に PC 表示を行う］プロパティの状態	表示内容
停止中	－	現在のカレント PC 位置

プログラムの状態	[実行中に PC 表示を行う] プロパティの状態	表示内容
実行中	はい	[PC 表示更新間隔 [ms]] プロパティにて設定した更新間隔ごとのカレント PC 位置注 1
	いいえ	実行中

また，このエリアにマウスを重ねることにより，次の情報をポップアップ表示します。

- カレント PC：0x *カレント PC 値*（*ソース名 # 行数*注 2）

ただし，デバッグ・ツールと切断時の場合は非表示となります。

注 1. 【RX100 シリーズ】
プログラム実行中の PC 表示機能をサポートしていないため，PC 値は表示されません。

注 2. 情報の取得が不可能な場合は，“ *シンボル名 + オフセット値* ” となります。

(f) デバッグ・ツールとの接続状態
現在のデバッグ・ツールとの接続状態を次のアイコンと文字列で示します。

接続状態	表示内容
接続中	*デバッグ・ツール名*
切断中	非接続

(g) Run-Break タイマ結果
Run-Break タイマの計測結果（「2.14.2 実行開始から停止までの実行時間を計測する」参照）を表示します。
表示単位は計測結果に依存します。
ただし，デバッグ・ツールと切断時の場合は非表示となります。

状態	表示内容
計測していない状態	未計測
計測中	計測中
オーバフローした場合	OVERFLOW

備考 EZ Emulator では Run-Break タイマ機能をサポートしていませんので，常に “ 未計測 ” と表示されます。

(h) デバッグ・ツールの状態
現在のデバッグ・ツールの各機能の状態を次のアイコンで示します。
ただし，デバッグ・ツールと切断時の場合は非表示となります。

機能	動作中	停止中（使用する）	使用しない
トレース			
タイマ			—
カバレッジ【シミュレータ】【E20【RX71M, RX64M グループ】】			

注意 1. 【E1】【E20】【EZ Emulator】
トレース機能は，対象アイコンをクリックしても “ 使用しない ” 状態にはなりません。

注意 2. 【E20(JTAG)】
トレース機能のアイコンをクリックすることにより，プロパティ パネルの［デバッグ・ツール設定］タブ上の［トレース］カテゴリ内［トレース機能の用途］プロパティの設定内容が，トレース／リアルタイム RAM モニタと切り替わります。

注意 3. 【シミュレータ】
トレース機能およびカバレッジ機能は，プログラムが停止中の場合，対象アイコンをクリック

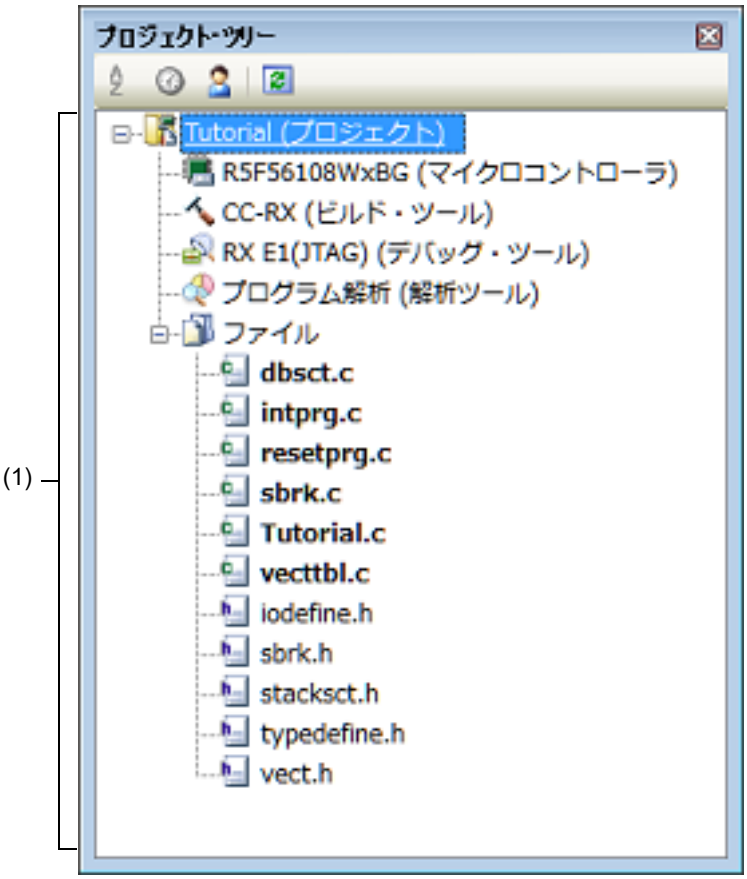
することにより , 使用する／使用しないの状態を変更することができます。
また，プロパティ パネルの［デバッグ・ツール設定］タブにおける次のプロパティの設定内容が，はい／いいえと切り替わります。

- ［トレース］カテゴリ→［トレース機能を使用する］プロパティ
- ［カバレッジ］【シミュレータ】カテゴリ→［カバレッジ機能を利用する］プロパティ

プロジェクト・ツリー パネル

プロジェクトの構成要素（マイクロコントローラ，ビルド・ツール，デバッグ・ツールなど）をツリー形式で表示します。
なお，使用するデバッグ・ツールの選択／切り替えは，このパネル上で行います。

図 A.3 プロジェクト・ツリー パネル



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］
- ［コンテキスト・メニュー］

［オープン方法］

- ［表示］メニュー→［プロジェクト・ツリー］を選択

［各エリアの説明］

(1) プロジェクト・ツリー エリア
プロジェクトの構成要素を次のノードでツリー表示します。

ノード	説明
品種名 デバッグ・ツール名 (デバッグ・ツール)	- *品種名* 選択しているマイクロコントローラの品種名（RX）を表示します。 - *デバッグ・ツール名* プロジェクトで使用するデバッグ・ツール名（E1(Serial), E1(JTAG), E20(Serial), E20(JTAG), EZ Emulator, シミュレータ）を表示します。 新規プロジェクト作成時は，シミュレータが設定されます。

ノードを選択すると，その詳細情報（プロパティ）がプロパティ パネルに表示され，設定の変更を行うことができます（プロパティ パネルがオープンしていない場合は，ノードをダブルクリックすることでオープンします）。

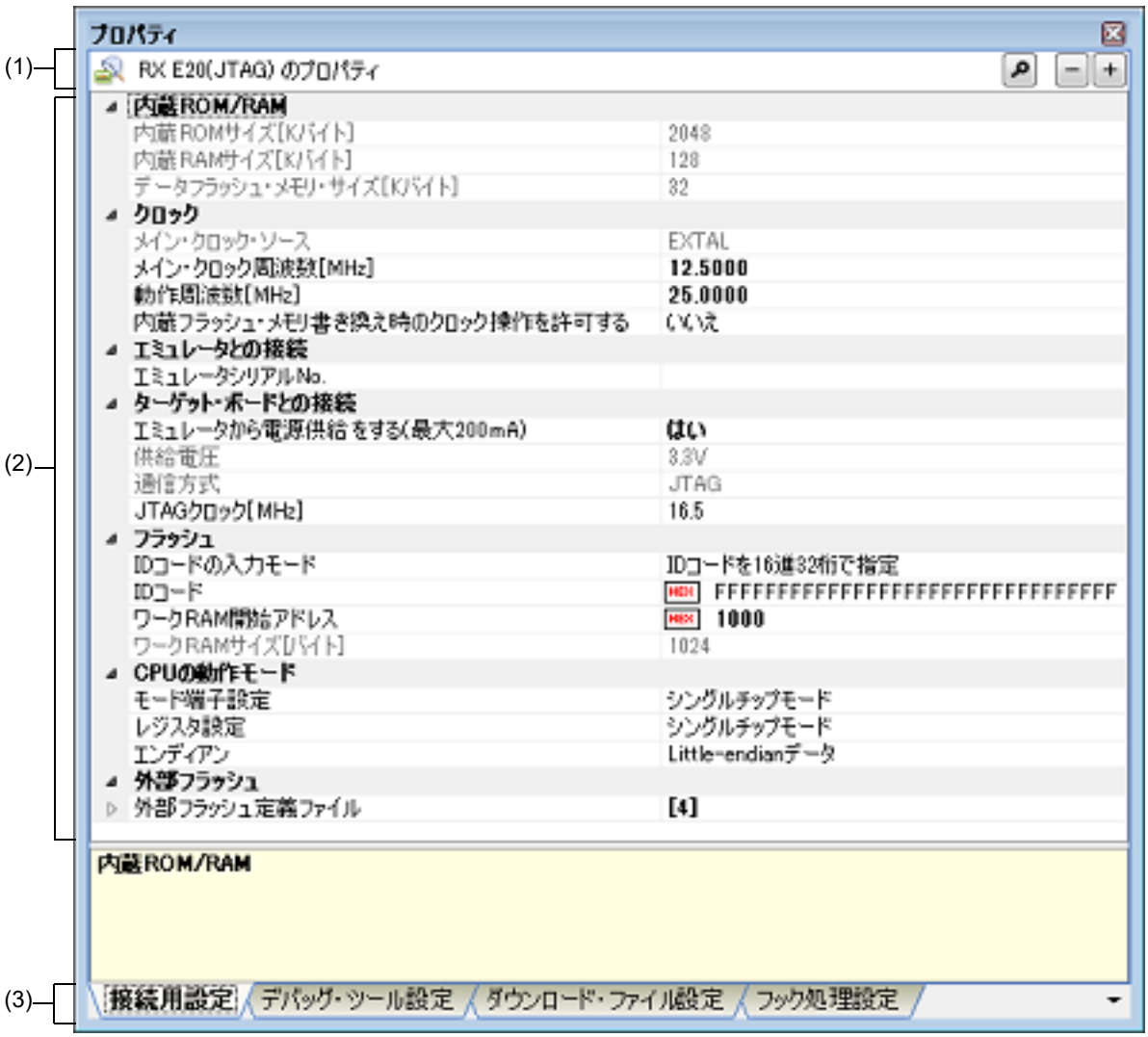
［コンテキスト・メニュー］

使用するデバッグ・ツール	使用するデバッグ・ツールを選択するためのカスケード・メニューを表示します。 なお，プロジェクトで選択しているマイクロコントローラの種類により，表示されるデバッグ・ツールは異なります（「表 2.1 マイクロコントローラの種類と接続可能なデバッグ・ツールの関係」参照）。
品種名 E2 Lite	E2 Lite を使用します。
品種名 E1(Serial)	E1 をシリアル通信方式で使用します。
品種名 E1(JTAG)	E1 を JTAG 通信方式で使用します。
品種名 E20(Serial)	E20 をシリアル通信方式で使用します。
品種名 E20(JTAG)	E20 を JTAG 通信方式で使用します。
品種名 EZ Emulator	EZ Emulator を使用します。
品種名 シミュレータ	シミュレータを使用します。
プロパティ	選択しているデバッグ・ツールのプロパティをプロパティ パネルに表示します。

プロパティ パネル

プロジェクト・ツリー パネルで選択しているデバッグ・ツールについて，カテゴリ別に詳細情報の表示，および設定の変更を行います。

図 A.4 プロパティ パネル（E20(JTAG) を選択した場合の例）



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］
- ［［編集］メニュー（プロパティ パネル専用部分）］
- ［コンテキスト・メニュー］

［オープン方法］

- プロジェクト・ツリー パネルにおいて，使用する［*品種名 デバッグ・ツール名* (デバッグ・ツール)］ノードを選択したのち，［表示］メニュー，またはコンテキスト・メニューより［プロパティ］を選択
- プロジェクト・ツリー パネルにおいて，使用する［*品種名 デバッグ・ツール名* (デバッグ・ツール)］ノードをダブルクリック

備考　このパネルがすでにオープンしている場合，プロジェクト・ツリー パネル上において，使用する［*品種名 デバッグ・ツール名* (デバッグ・ツール)］ノードを選択することにより，選択したデバッグ・ツールの詳細情報を表示します。

［各エリアの説明］

(1) 対象名エリア
プロジェクト・ツリー パネルで選択しているデバッグ・ツールの名称を表示します。

(2) 詳細情報表示／変更エリア
プロジェクト・ツリー パネルで選択しているデバッグ・ツールの詳細情報を，カテゴリ別のリスト形式で表示し，設定の変更を直接行うことができるエリアです。
⊟マークは，そのカテゴリ内に含まれているすべてのプロパティ項目が展開表示されていることを示し，また，⊞マークは，カテゴリ内のプロパティ項目が折りたたみ表示されていることを示します。展開／折りたたみ表示の切り替えは，このマークのクリック，またはカテゴリ名のダブルクリックにより行うことができます。
なお，各プロパティ項目設定欄内に表示される HEX マークは，16 進数入力専用のテキスト・ボックスであることを示します。
カテゴリ，およびそれに含まれるプロパティ項目の表示内容／設定方法についての詳細は，該当するタブの項を参照してください。

(3) タブ選択エリア
タブを選択することにより，詳細情報を表示するカテゴリが切り替わります。
このパネルには，次のタブが存在します（各タブ上における表示内容／設定方法についての詳細は，該当するタブの項を参照してください）。

- ［接続用設定］タブ
- ［デバッグ・ツール設定］タブ
- ［ダウンロード・ファイル設定］タブ
- ［フック処理設定］タブ

［［編集］メニュー（プロパティ パネル専用部分）］

元に戻す	直前に行ったプロパティの値の編集作業を取り消します。
切り取り	プロパティの値を編集中の場合，選択している文字列を切り取ってクリップ・ボードに移動します。
コピー	選択しているプロパティの値の文字列をクリップ・ボードにコピーします。
貼り付け	プロパティの値を編集中の場合，クリップ・ボードの内容を挿入します。
削除	プロパティの値を編集中の場合，選択している文字列を削除します。
すべて選択	プロパティの値を編集中の場合，選択しているプロパティの値の文字列をすべて選択します。

［コンテキスト・メニュー］

【文字列編集中以外の場合】

デフォルトに戻す	選択しているプロパティ項目の設定値をデフォルトに戻します。
すべてデフォルトに戻す	現在選択しているタブ上の設定値をすべてデフォルトに戻します。

【文字列編集中の場合】

元に戻す	直前に行ったプロパティの値の編集作業を取り消します。
切り取り	プロパティの値を編集中の場合，選択している文字列を切り取ってクリップ・ボードに移動します。

コピー	選択しているプロパティの値の文字列をクリップ・ボードにコピーします。
貼り付け	プロパティの値を編集中の場合，クリップ・ボードの内容を挿入します。
削除	プロパティの値を編集中の場合，選択している文字列を削除します。
すべて選択	プロパティの値を編集中の場合，選択しているプロパティの値の文字列をすべて選択します。

［接続用設定］タブ

［接続用設定］タブでは，次に示すカテゴリごとに詳細情報の表示，および設定の変更を行います。

(1) ［内蔵 ROM/RAM］
(2) ［エンディアン］【シミュレータ】
(3) ［クロック］
(4) ［エミュレータとの接続］【E1】【E20】
(5) ［ターゲット・ボードとの接続］【E1】【E20】【EZ Emulator】
(6) ［フラッシュ］【E1】【E20】【EZ Emulator】
(7) ［CPU の動作モード］【E1】【E20】【EZ Emulator】
(8) ［外部フラッシュ］【E1】【E20】
(9) ［周辺機能シミュレーション］【シミュレータ】

図 A.5 プロパティ パネル：［接続用設定］タブ【E1(Serial)】

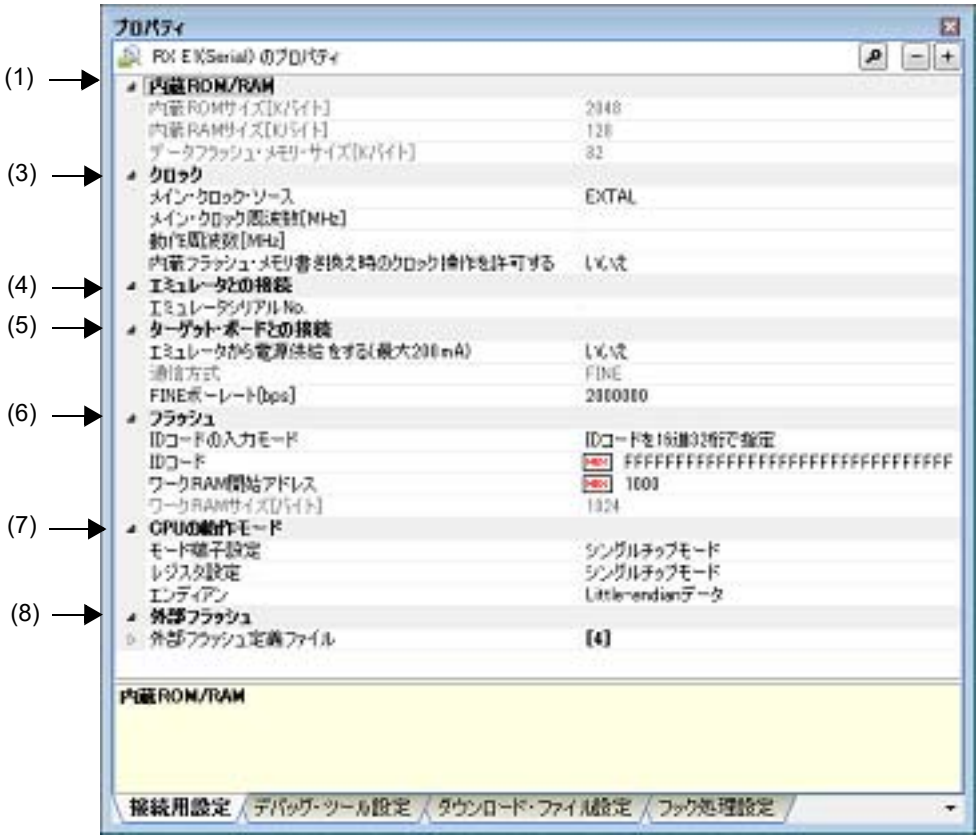


図 A.6 プロパティ パネル：［接続用設定］タブ【E1(JTAG)】

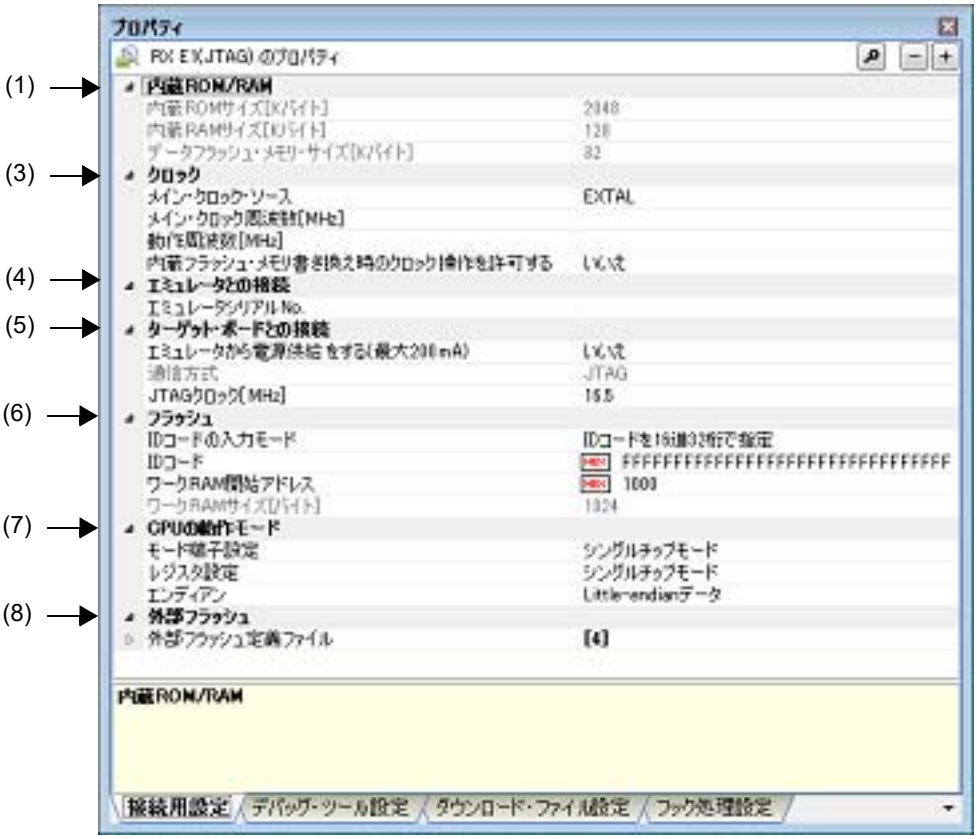


図 A.7 プロパティ パネル：［接続用設定］タブ【E20(Serial)】）

図 A.8 プロパティ パネル：［接続用設定］タブ【E20(JTAG)】）

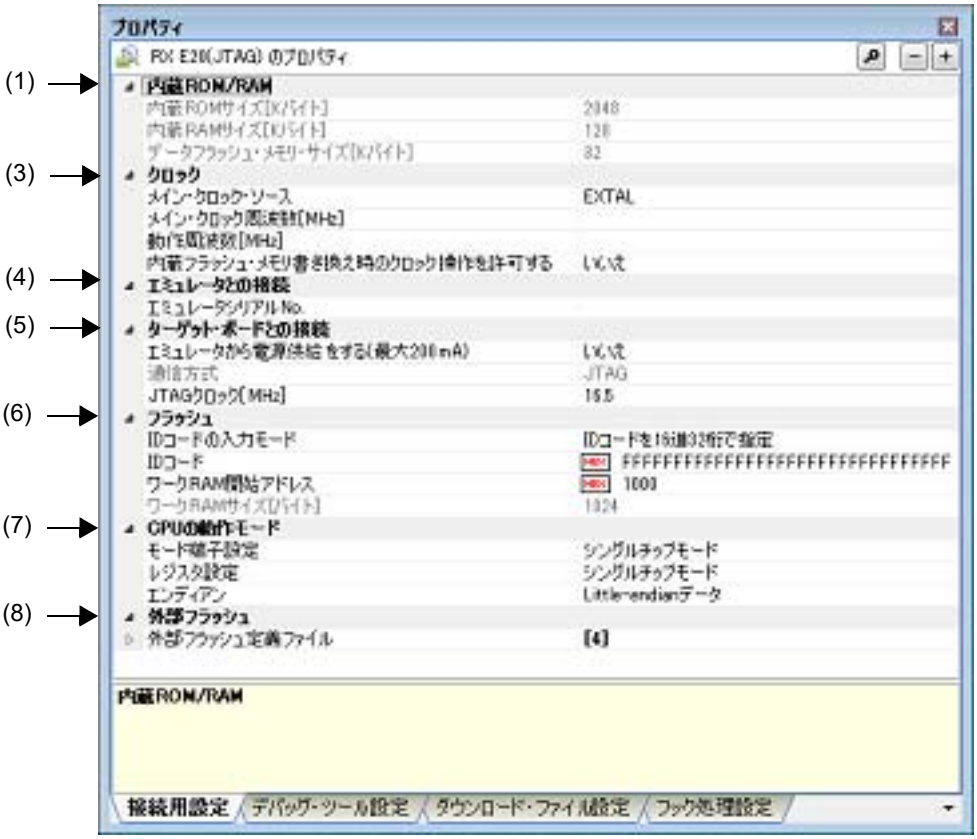
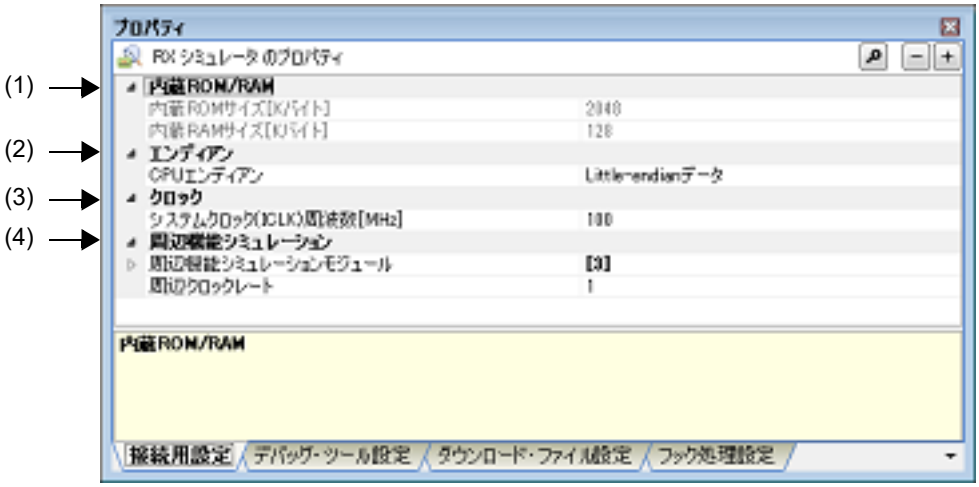


図 A.9 プロパティ パネル：［接続用設定］タブ【EZ Emulator】）

図 A.10 プロパティ パネル：［接続用設定］タブ【シミュレータ】



［各カテゴリの説明］

(1) ［内蔵 ROM/RAM］

内蔵 ROM/RAM に関する詳細情報の表示を行います。

内蔵 ROM サイズ [K バイト]	選択しているマイクロコントローラの内蔵 ROM サイズを表示します。	
	デフォルト	- ROM 内蔵品の場合 *選択しているマイクロコントローラの内蔵 ROM サイズ* - ROM レス品の場合 *0*
	変更方法	変更不可
内蔵 RAM サイズ [K バイト]	選択しているマイクロコントローラの内蔵 RAM サイズを表示します。	
	デフォルト	*選択しているマイクロコントローラの内蔵 RAM サイズ*
	変更方法	変更不可
データフラッシュ・メモリ・サイズ [K バイト] 【E1】【E20】【EZ Emulator】	選択しているマイクロコントローラのデータフラッシュ・メモリ領域のサイズを表示します。	
	デフォルト	*選択しているマイクロコントローラのデータフラッシュ・メモリ・サイズ*
	変更方法	変更不可

(2) ［エンディアン］【シミュレータ】

CPU エンディアンに関する詳細情報の表示，および設定の変更を行います。

CPU エンディアン	マイクロコントローラのエンディアンを指定します。 ビルド・ツールのプロパティで設定されたエンディアン情報を取得して表示します。	
	デフォルト	*Little-endian データ*
	変更方法	- ビルド & デバッグ・ツールの場合 変更不可 - デバッグ・ツールのみの場合 ドロップダウン・リストによる選択 ただし，デバッグ・ツールと切断中の場合のみ変更可
	指定可能値	- ドロップダウン・リストによる次のいずれか *Little-endian データ，Big-endian データ*

(3) ［クロック］
クロックに関する詳細情報の表示，および設定の変更を行います。

メイン・クロック・ソース【E1】【E20】【EZ Emulator】	- 内蔵 HOCO 搭載品の場合 メイン・クロックを EXTAL 周波数と内蔵 HOCO から指定します。 - 内蔵 HOCO 非搭載品の場合 メイン・クロックとして EXTAL 周波数を表示します。		
	デフォルト	*EXTAL*	
	変更方法	ドロップダウン・リストによる選択 ただし，デバッグ・ツールと切断中の場合のみ変更可	
	指定可能値	EXTAL	メイン・クロックが EXTAL 周波数として動作します
		HOCO	メイン・クロックが内蔵 HOCO として動作します。
メイン・クロック周波数 [MHz]【E1】【E20】【EZ Emulator】	EXTAL 周波数を MHz 単位で指定します。 なお，このプロパティは [メイン・クロック・ソース 【E1】【E20】【EZ Emulator】] プロパティにおいて [EXTAL] を指定した場合のみ有効となります。		
	デフォルト	*空欄*	
	変更方法	キーボードからの直接入力 ただし，デバッグ・ツールと切断中の場合のみ変更可	
	指定可能値	0.0001 ～ 99.9999（単位：MHz）	
動作周波数 [MHz]【E1】【E20】【EZ Emulator】	動作周波数（ICLK）を MHz 単位で指定します。		
	デフォルト	*空欄*	
	変更方法	キーボードからの直接入力	
	指定可能値	0.0001 ～ 999.9999（単位：MHz）	
内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する【E1】【E20】【EZ Emulator】	内蔵フラッシュ・メモリを書き換える際，デバッガによるメイン・クロック・ソースの操作を許可するか否かを指定します。		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択 ただし，デバッグ・ツールと切断中の場合のみ変更可	
	指定可能値	はい	メイン・クロック・ソースの操作を許可します。
		いいえ	メイン・クロック・ソースの操作を許可しません。
システムクロック (ICLK) 周波数 [MHz]【シミュレータ】	CPU の動作周波数を指定します。		
	デフォルト	*選択しているマイクロコントローラに依存*	
	変更方法	キーボードからの直接入力 ただし，デバッグ・ツールと切断中の場合のみ変更可	
	指定可能値	1 ～ 1000 の整数（単位：MHz）	

(4) ［エミュレータとの接続］【E1】【E20】

エミュレータシリアル No.	接続するエミュレータのシリアル番号を選択します。注	
	デフォルト	*空欄*
	変更方法	ドロップダウン・リストによる選択 ただし，デバッグ・ツールと切断中の場合のみ変更可
	指定可能値	使用するエミュレータに依存

注　使用するデバッグ・ツールにおいて E1(JTAG)，または E1(Serial) を選択した場合，E1 エミュレータのシリアル No. が列挙され，E20(Serial)，または E20(JTAG) を選択した場合，E20 エミュレータのシリアル No. が列挙されます。
空欄時に接続した場合，検索して最初に見つかったエミュレータのシリアル番号を自動で選択し，接続します。このとき，自動で選択されたエミュレータのシリアル番号はプロジェクト情報には保存されません。

(5) ［ターゲット・ボードとの接続］【E1】【E20】【EZ Emulator】
ターゲット・ボードとの接続状態に関する詳細情報の表示，および設定の変更を行います。

MCU 動作電圧 [V] 【EZ Emulator】	ユーザ基板に実際に供給しているマイクロコントローラの動作電圧値を指定します。		
	デフォルト	*空欄*	
	変更方法	キーボードからの直接入力 ただし，デバッグ・ツールと切断中の場合のみ変更可	
	指定可能値	0.0001 ～ 9.9999（単位：V）	
エミュレータから電源供給をする（最大 200mA【E1】【E20】	E1 からターゲット・システムに電源を供給するか否かを指定します。 E20 は電源供給機能をサポートしていません。そのため，プロパティ値は [いいえ] を表示します。		
	デフォルト	*いいえ*	
	変更方法	-【E1】の場合 ドロップダウン・リストによる選択 ただし，デバッグ・ツールと切断中の場合のみ変更可 -【E20】の場合 変更不可	
	指定可能値	はい	電源を供給します。
		いいえ	電源を供給しません。
供給電圧 【E1】	E1 からターゲット・ボードに供給する電圧を指定します。 なお，このプロパティは [エミュレータから電源供給をする（最大 200mA【E1】【E20】] プロパティにおいて [はい] を指定した場合のみ表示されます。		
	デフォルト	*3.3V*	
	変更方法	ドロップダウン・リストによる選択 ただし，デバッグ・ツールと切断中の場合のみ変更可	
	指定可能値	- ドロップダウン・リストによる次のいずれか注 -【E1】 *3.3V，5.0V* -【E2 Lite】 *3.3V*	

通信方式	エミュレータとターゲット・システム上の CPU と通信を行う際に接続する通信方式を表示します。	
	デフォルト	-【E1(JTAG)】【E20(JTAG)】 *JTAG* -【E1(Serial)】【E20(Serial)】【EZ Emulator】 *FINE* -【E2 Lite】 *JTAG*
	変更方法	-【E1】【E20】【EZ Emulator】 変更不可 -【E2 Lite】 変更不可，またはドロップダウン・リストによる選択 変更可能かどうかは選択しているマイクロコントローラの種類と通信方式によって異なります。マイクロコントローラの種類と通信方式の関係については，「2.2.1 ホスト・マシンとの接続を確認する」を参照してください。
	指定可能値	-【E2 Lite】 *JTAG*，*FINE*
JTAG クロック [MHz] 【E1(JTAG)】 【E20(JTAG)】	エミュレータとターゲット・システム上の CPU 間の JTAG 通信速度を指定します。 なお，このプロパティは [通信方式] プロパティにおいて [JTAG] を指定した場合のみ表示されます。	
	デフォルト	-【E1(JTAG)】【E20(JTAG)】 *16.5* -【E2 Lite】 *6.00*
	変更方法	ドロップダウン・リストによる選択 ただし，デバッグ・ツールと切断中の場合のみ変更可
	指定可能値	- ドロップダウン・リストによる次のいずれか -【E1(JTAG)】【E20(JTAG)】 *16.5*，*12.38*，*6.188*，*3.094*，*1.547* -【E2 Lite】 *6.00*，*3.00*，*1.50*

FINE ボーレート [bps] 【E1(Serial)】 【E20(Serial)】【EZ Emulator】	エミュレータとターゲット・システム上の CPU 間の FINE 通信速度を指定します。 なお，このプロパティは [通信方式] プロパティにおいて [FINE] を指定した場合のみ表示されます。	
	デフォルト	- 【E1(Serial)】【E20(Serial)】 *2000000* - 【EZ Emulator】 *1000000* - 【E2 Lite】 *1500000*
	変更方法	ドロップダウン・リストによる選択 ただし，デバッグ・ツールと切断中の場合のみ変更可
	指定可能値	- ドロップダウン・リストによる次のいずれか - 【E1(Serial)】【E20(Serial)】 *2000000，750000，500000，250000* - 【EZ Emulator】 *1000000，500000，250000* - 【E2 Lite】 *1500000，750000，500000，250000*

注 指定可能な電圧値は，選択するマイクロコントローラによって異なります。

(6) ［フラッシュ］【E1】【E20】【EZ Emulator】

フラッシュ書き換えに関する詳細情報の表示，および設定の変更を行います。

ID コードの入力モード	ID コードの入力モードを指定します。		
	デフォルト	*ID コードを 16 進 32 桁で指定*	
	変更方法	ドロップダウン・リストによる選択 ただし，デバッグ・ツールと切断中の場合のみ変更可	
	指定可能値	ID コードを 16 進 32 桁で指定	ID コードを 16 進数の 32 桁で入力します。
		ID コードを 16 文字以内で指定 (ASCII)	ID コードを ASCII 16 文字以内で入力します。

ID コード	ID コードプロテクトを解除するための ID コードを指定します。	
	デフォルト	- [ID コードを 16 進 32 桁で指定] の場合 *FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF* - [ID コードを 16 文字以内で指定 (ASCII)] の場合 *空欄*
	変更方法	キーボードからの直接入力 ただし，デバッグ・ツールと切断中の場合のみ変更可

	指定可能値	- [ID コードを 16 進 32 桁で指定] の場合
		32 桁の 16 進
		ID コードの入力は，32 ビットのデータ単位で並べて入力します。
		例：下記のように確保した場合，
		00112233445566778899aabbccddeeff と入力します。
		const unsigned long __OSISreg[4] = {
		0x00112233,
		0x44556677,
		0x8899aabb,
		0xccddeeff
		};
		フラッシュプログラマでは ID コードの並び順が CS+ と異なる場合があります。ご使用のフラッシュプログラマの ID コード入力の仕様を確認してください。
		- [ID コードを 16 文字以内で指定 (ASCII)] の場合
		ASCII コードの 16 文字（16 文字に満たない場合は 0 が付加）

ワーク RAM 開始アドレス	デバッガが使用するワーク RAM の配置アドレスを指定します。 アドレス値は 4 バイト単位の値を指定してください。（入力値が 4 バイト単位の値でなかった場合は，自動的に補正されます。） 指定したワーク RAM の配置アドレスから指定バイト分は，デバッガのファームウェアが使用します。注	
	デフォルト	*選択しているマイクロコントローラに依存*
	変更方法	キーボードからの直接入力 ただし，デバッグ・ツールと切断中の場合のみ変更可
	指定可能値	選択しているマイクロコントローラの内蔵 RAM 領域に合わせたアドレス値
ワーク RAM サイズ [バイト]	デバッガが使用するワーク RAM のサイズを表示します。	
	デフォルト	*選択しているマイクロコントローラに依存*
	変更方法	変更不可

注 メモリ内容の退避・復帰が行われるためユーザ・プログラムでもこの領域を使用できますが，ワーク RAM に配置する領域は，以下の用途では使用できません。

- DMA または DTC 機能の転送元・転送先
- DTC ベクタテーブル，および転送情報配置アドレス
- DMAC，DTC 起動要因の割り込みベクタ

(7) ［CPU の動作モード］【E1】【E20】【EZ Emulator】

マイクロコントローラの動作モードに関する詳細情報の表示，および設定の変更を行います。

モード端子設定	モード端子で設定している動作モードを指定します。		
	デフォルト	*シングルチップモード*	
	変更方法	ドロップダウン・リストによる選択 ただし，デバッグ・ツールと切断中の場合のみ変更可	
	指定可能値	- ドロップダウン・リストによる次のいずれか注 1 -【E1】【E20】 *シングルチップモード，ユーザブートモード* -【EZ Emulator】 *シングルチップモード*	
USB ブートプログラムの消去を許可する【E1】【E20】	USB ブートプログラムが格納されたマイクロコントローラを使用してユーザブートモードで起動する際に，ユーザブート領域上の USB ブートプログラムを消去するか否かを指定します。 このプロパティは，[モード端子設定] プロパティにおいて [*ユーザブートモード*] を指定した場合のみ表示します。ただし，ユーザブートモードで起動する際に，ユーザブート領域上の USB ブートプログラムを消去する必要のないマイクロコントローラの場合は表示されません。		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択 ただし，デバッグ・ツールと切断中の場合のみ変更可	
	指定可能値	*はい*	ユーザブート領域上の USB ブートプログラムを消去します。
		いいえ	ユーザブート領域上の USB ブートプログラムを消去しません。
レジスタ設定	レジスタで設定する動作モードを指定します。		
	デフォルト	*シングルチップモード*	
	変更方法	ドロップダウン・リストによる選択 ただし，デバッグ・ツールと切断中の場合のみ変更可	
	指定可能値	- ドロップダウン・リストによる次のいずれか注 1 -【E1】【E20】 *シングルチップモード，内蔵 ROM 有効拡張モード，内蔵 ROM 無効拡張モード* -【EZ Emulator】 *シングルチップモード*	
エンディアン	プロジェクトのエンディアンを表示します。注 2		
	デフォルト	*Little-endian データ*	
	変更方法	- ビルド & デバッグ・ツールの場合 変更不可 - デバッグ・ツールのみの場合 ドロップダウン・リストによる選択 ただし，デバッグ・ツールと切断中の場合のみ変更可	
	指定可能値	- ドロップダウン・リストによる次のいずれか *Little-endian データ，Big-endian データ*	

起動バンクを変更する【RX651-2M, RX65N-2M グループ】	デバッグ・ツールと接続する際に起動バンクを変更するか否かを指定します。		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	起動バンクを変更します。
		いいえ	起動バンクを変更しません。
起動バンク【RX651-2M, RX65N-2M グループ】	デバッグ・ツールと接続する際に変更する起動バンクを指定します。		
	デフォルト	*バンク0*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	- ドロップダウン・リストによる次のいずれか *バンク0*，*バンク1*	

注 1. 指定可能な動作モードは，選択するマイクロコントローラによって異なります。

注 2. MDE 端子があるマイクロコントローラを選択した場合，プロジェクトのエンディアンとマイクロコントローラの MDE 端子の状態を合わせてください。エンディアンが異なる場合，正常に接続できません。

(8) ［外部フラッシュ］【E1】【E20】

外部フラッシュに関する詳細情報の表示，および設定の変更を行います。

外部フラッシュ定義ファイル	外部フラッシュ定義ファイルを指定します。注 メインプロパティには，登録可能な外部フラッシュ定義ファイルの数が表示されます。 サブプロパティには，ファイル名，アドレス範囲およびダウンロード条件が下段に展開表示されます。なお，外部フラッシュ定義ファイルが未登録の場合，ファイル以外のサブプロパティは表示されません。		
	デフォルト	- メイン項目 *4* - サブ項目 *空欄*	
	変更方法	- メイン項目 変更不可 - サブ項目 ＜ファイル＞ 外部フラッシュメモリ ダイアログ【E1】【E20】による選択 このプロパティを選択すると欄内右端に表示される [...] ボタンをクリックすることで外部フラッシュメモリ ダイアログがオープンします。 ＜開始アドレス＞ 変更不可 ＜終了アドレス＞ 変更不可 ＜ダウンロード前に外部フラッシュ ROM を消去する＞ ドロップダウン・リストによる選択 ただし，デバッグ・ツールと切断中の場合のみ変更可	
	指定可能値	はい	ダウンロード前に，外部フラッシュ ROM を消去します。
		いいえ	ダウンロード前に，外部フラッシュ ROM を消去しません。
	表示内容	- メイン項目 外部フラッシュ定義ファイル数 - サブ項目 外部フラッシュ定義ファイルの“＋”マークをクリックすると，インデックス（[1] ～ [4]）毎に定義ファイル名，開始／終了アドレスおよびダウンロード前に外部フラッシュ ROM を消去するか否かの選択状況を表示します。	

注　マイクロコントローラの動作モードがシングルチップモードの場合は，設定した内容は反映されません。

(9)　［周辺機能シミュレーション］【シミュレータ】

周辺機能シミュレーションに関する詳細情報の表示，および設定の変更を行います。

周辺機能シミュレーションモジュール	使用可能な周辺機能シミュレーションモジュールを表示し，使用するか否かを指定します。		
	デフォルト	- メイン項目 *＜使用可能な周辺機能シミュレーションモジュール数＞* - サブ項目 *＜周辺機能シミュレーションモジュール名＞* 使用しない	
	変更方法	- メイン項目 変更不可 - サブ項目 周辺機能シミュレーションモジュール名は変更不可 周辺機能シミュレーションモジュールを使用するか否かは，ドロップダウン・リストによる選択 ただし，デバッグ・ツールと切断中の場合のみ変更可	
	指定可能値	使用する	周辺機能シミュレーションモジュールを使用します。
		使用しない	周辺機能シミュレーションモジュールを使用しません。
	表示内容	- メイン項目 使用可能な周辺機能モジュール数 - サブ項目 周辺機能シミュレーションモジュール名 周辺機能シミュレーションモジュールの使用状況	
周辺クロックレート	周辺クロックと内部クロックの比（周辺 1 クロックが内部クロックいくつに相当するか）を指定します。		
	デフォルト	*1*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	- ドロップダウン・リストによる次のいずれか *1，2，3，4，6，8，12，16，24，32，64*	

［デバッグ・ツール設定］タブ

［デバッグ・ツール設定］タブでは，次に示すカテゴリごとに詳細情報の表示，および設定の変更を行います。

(1) ［メモリ］
(2) ［実行中のメモリ・アクセス］
(3) ［レジスタ］
(4) ［ブレーク］【E1】【E20】【EZ Emulator】
(5) ［システム］【E1】【E20】【EZ Emulator】
(6) ［トレース］
(7) ［タイマ］【E1】【E20】【EZ Emulator】
(8) ［カバレッジ］【シミュレータ】
(9) ［ストリーム入出力］【シミュレータ】
(10) ［実行モード］【シミュレータ】
(11) ［命令デコードキャッシュ］【シミュレータ】
(12) ［カバレッジ］【E20【RX71M, RX64M グループ】】

図 A.11 プロパティ パネル：［デバッグ・ツール設定］タブ【E1(Serial)【RX600 シリーズ】】

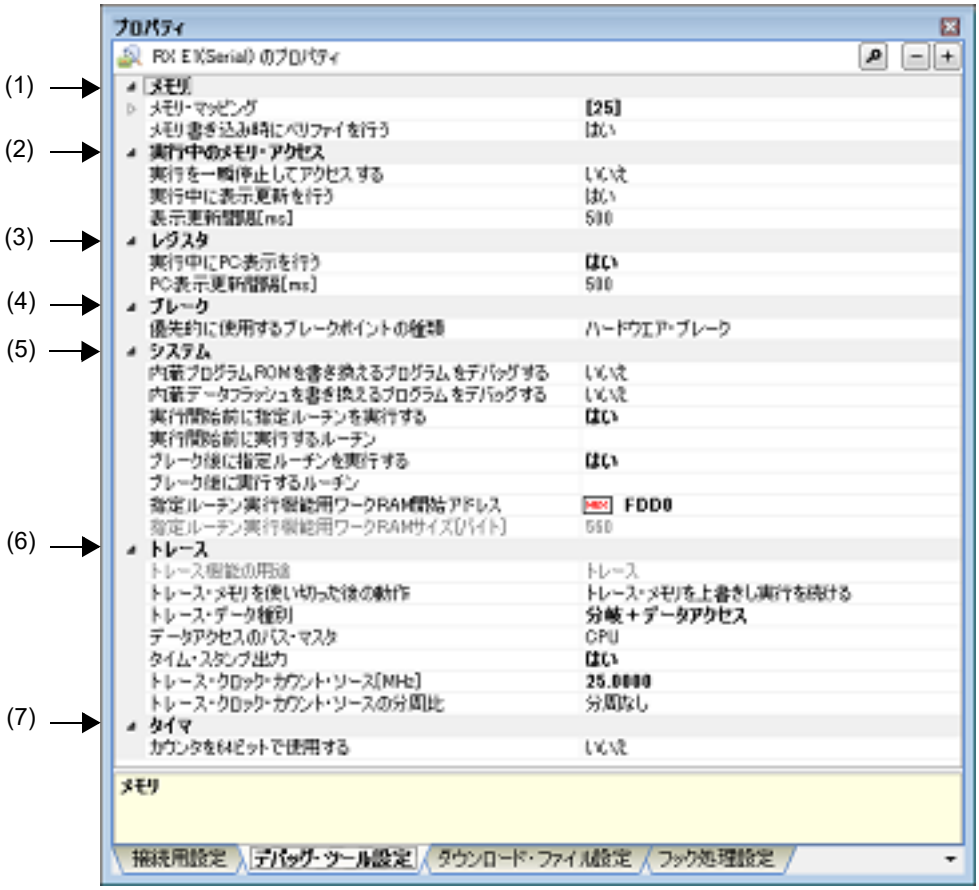


図 A.12 プロパティ パネル：［デバッグ・ツール設定］タブ【E1(Serial)【RX100, RX200 シリーズ】】

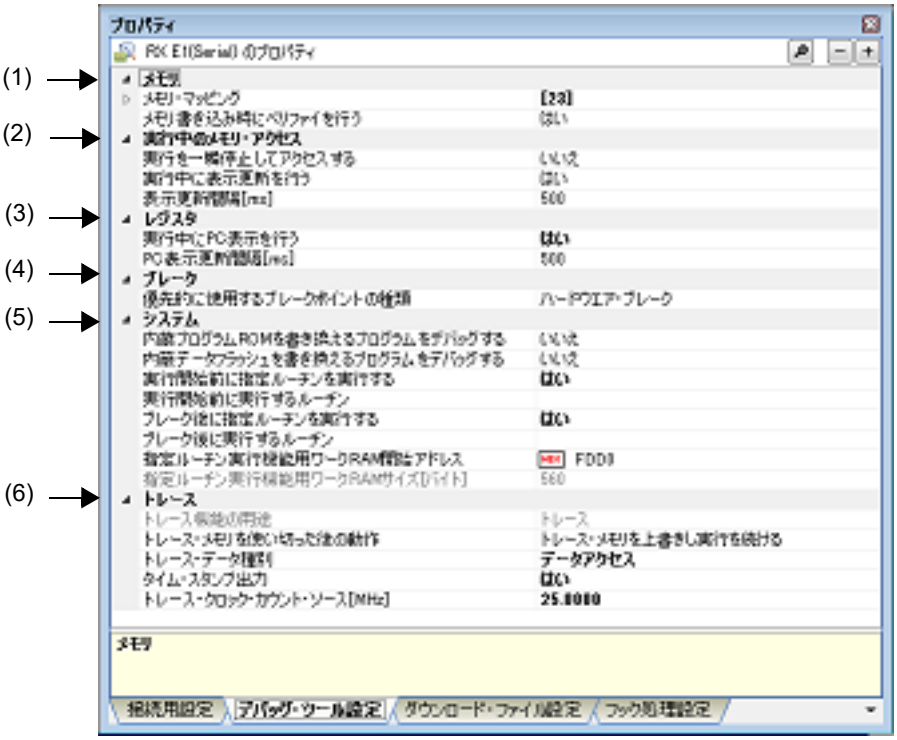


図 A.13 プロパティ パネル：［デバッグ・ツール設定］タブ【E1(JTAG)【RX600 シリーズ】】

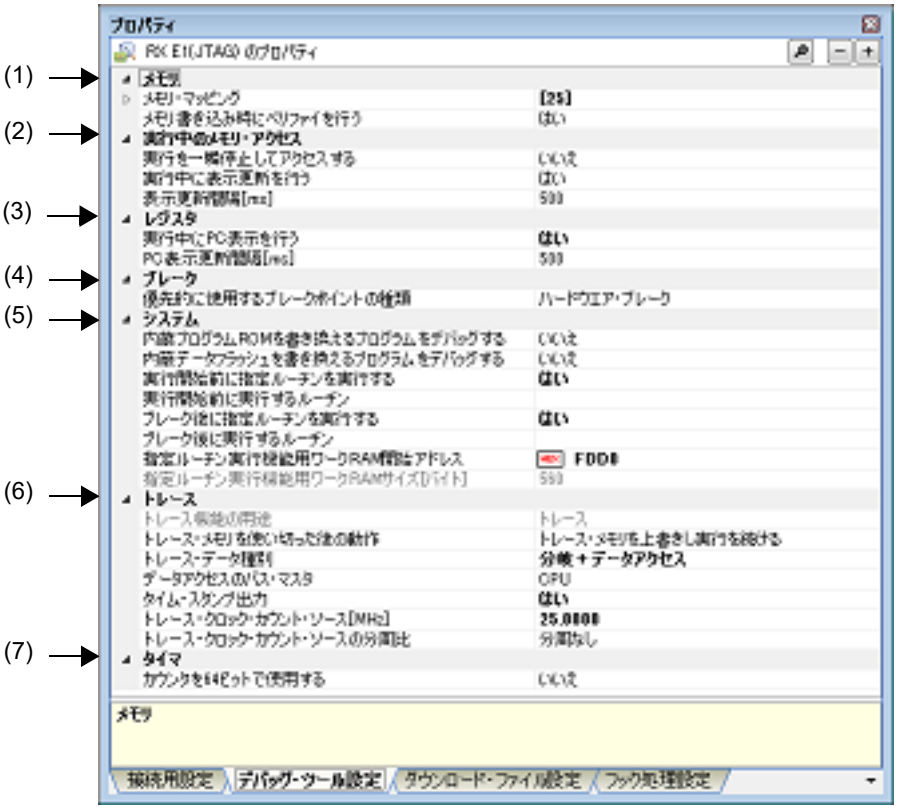


図 A.14 プロパティ パネル：［デバッグ・ツール設定］タブ【E20(Serial)【RX600 シリーズ】】

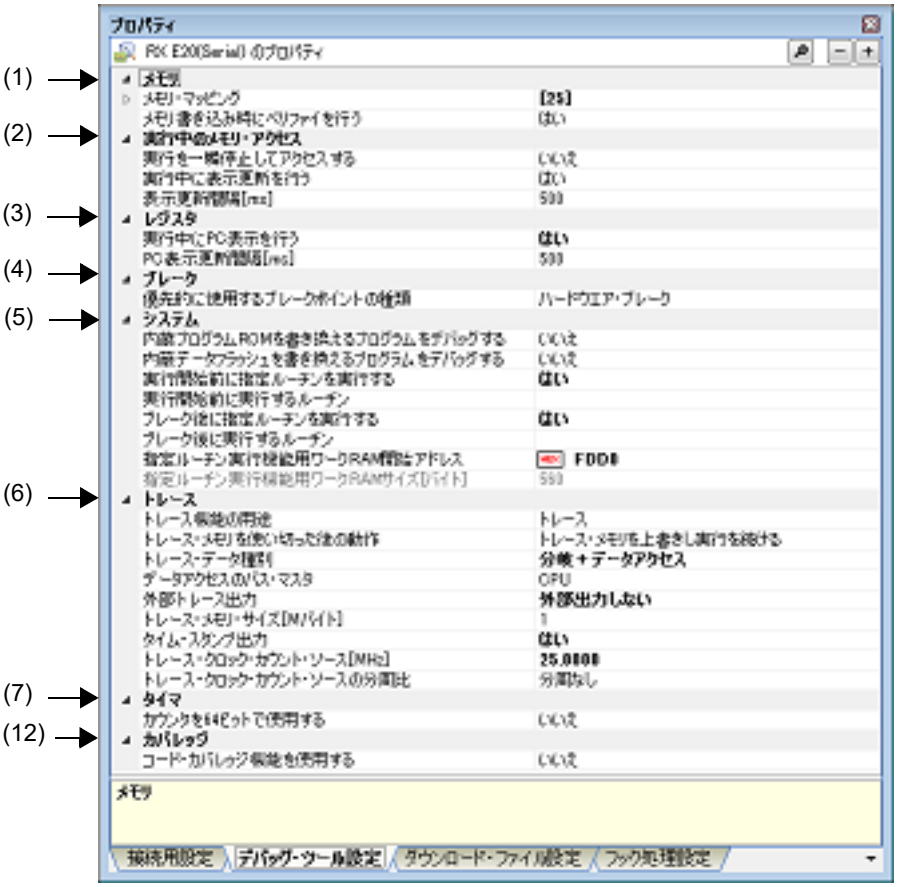


図 A.15 プロパティ パネル：［デバッグ・ツール設定］タブ【E20(Serial)【RX100, RX200 シリーズ】】

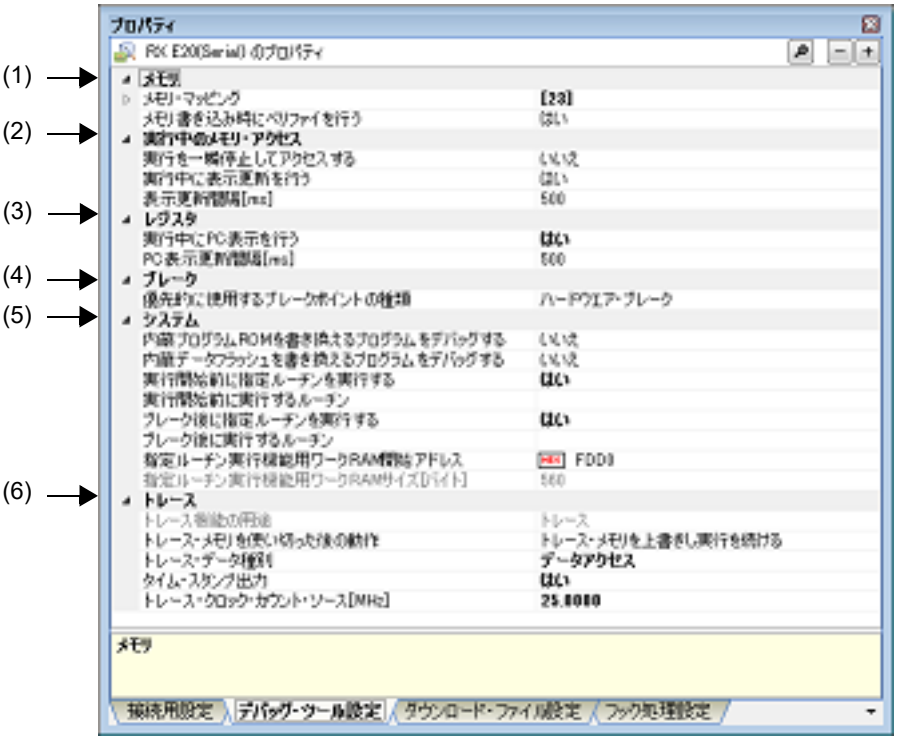


図 A.16 プロパティ パネル：［デバッグ・ツール設定］タブ【E20(JTAG)【RX600 シリーズ】】

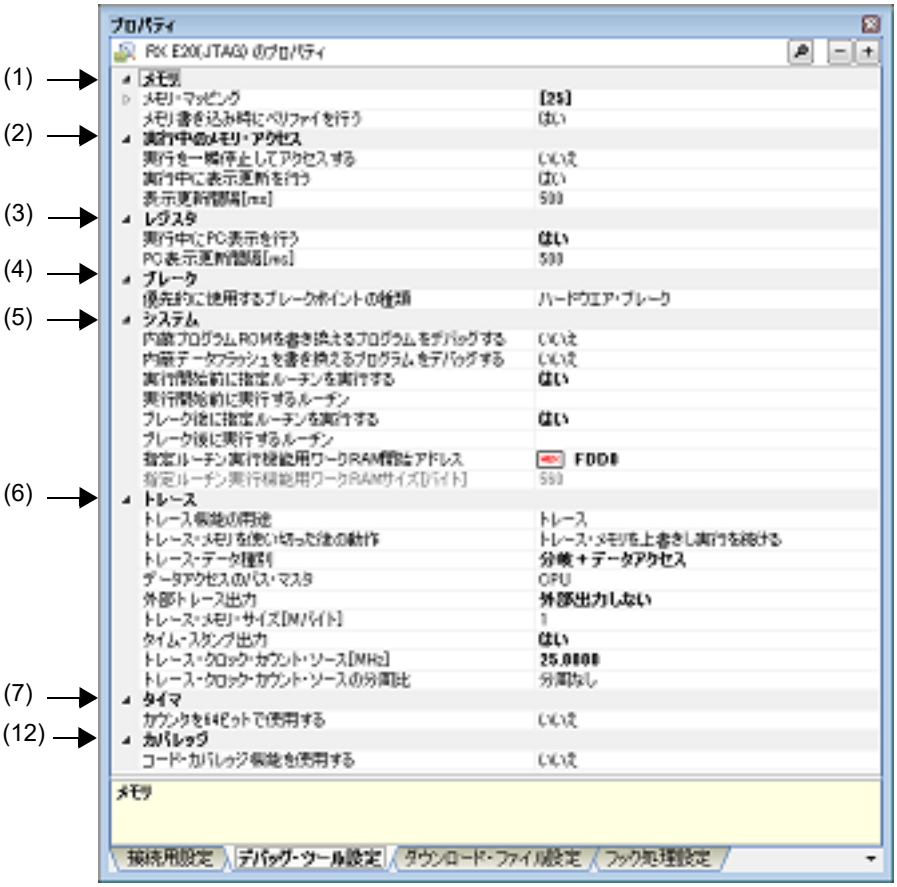


図 A.17 プロパティ パネル：［デバッグ・ツール設定］タブ【EZ Emulator【RX600 シリーズ】】

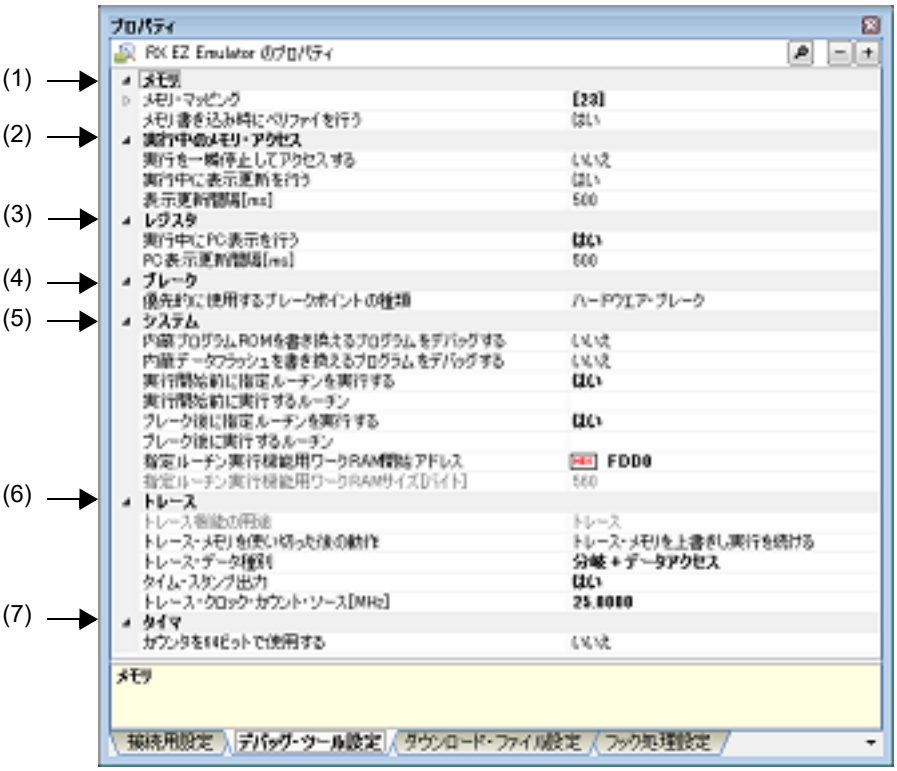


図 A.18 プロパティ パネル：［デバッグ・ツール設定］タブ【EZ Emulator【RX100, RX200 シリーズ】】

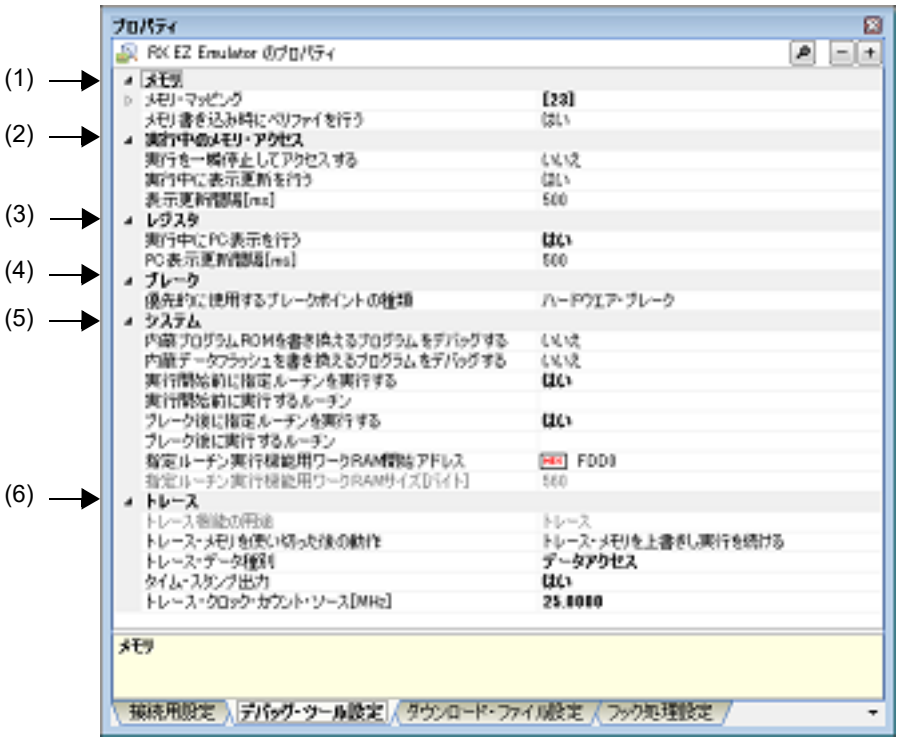


図 A.19 プロパティ パネル：［デバッグ・ツール設定］タブ【シミュレータ【RX600 シリーズ】】

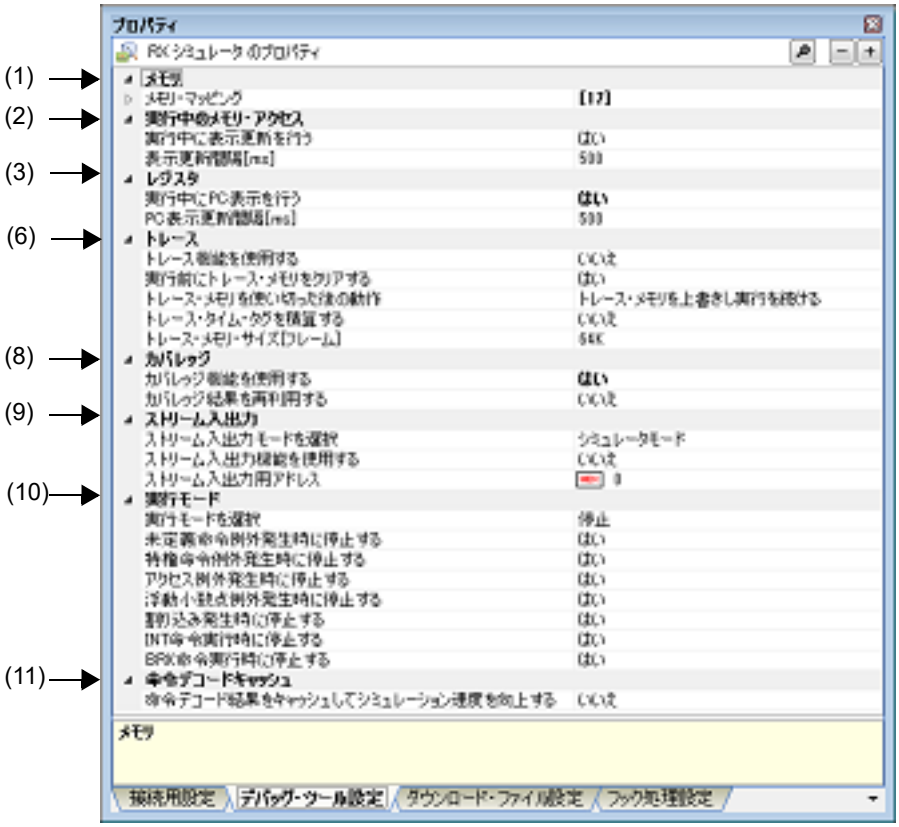
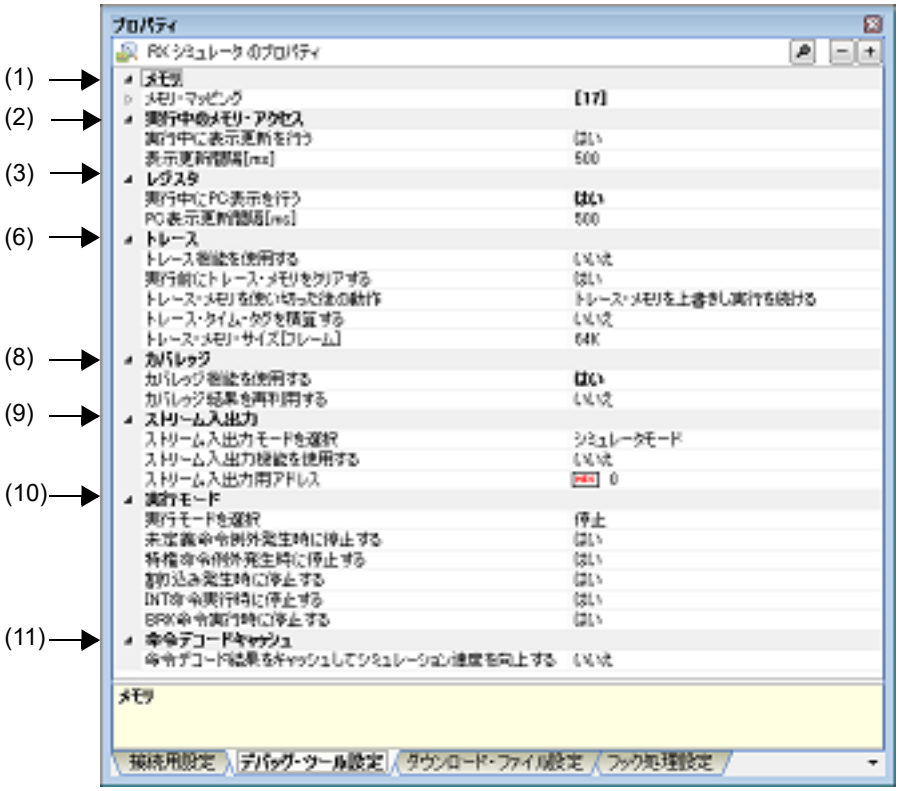


図 A.20 プロパティ パネル：［デバッグ・ツール設定］タブ【シミュレータ【RX100, RX200 シリーズ】】



［各カテゴリの説明］

(1) ［メモリ］

メモリに関する詳細情報の表示，および設定の変更を行います。
なお，表示されるメモリ種別についての詳細は，メモリ・マッピング ダイアログの項を参照してください。

メモリ・マッピング	現在のメモリ・マッピングの状況をメモリ領域の種別[注1]ごとに表示します。		
	デフォルト	*[マイクロコントローラ固有のメモリ・マッピング領域種別の合計]*	
	変更方法	メモリ・マッピング ダイアログによる指定 メモリ・マッピング ダイアログは，このプロパティを選択すると右端に表示される［...］ボタンをクリックすることでオープンします（このパネル上でマッピング値を変更することはできません）。	
	表示内容	メモリ・マッピングの状況をメモリ領域の種別ごとに表示します。 なお，各メモリ種別の “+” マークをクリックすると，次の詳細情報を表示します。 - メモリ種別 - 開始アドレス - 終了アドレス - アクセス幅 [ビット]【E1】【E20】【EZ Emulator】[注2] - エンディアン[注2 注3]	
メモリ書き込み時にベリファイを行う 【E1】【E20】 【EZ Emulator】	メモリ値の初期化を行う際に，ベリファイを行うか否かを指定します。		
	デフォルト	*はい*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	ベリファイを行います。
		いいえ	ベリファイを行いません。

注 1. デバイス・ファイルに登録されているメモリ・マッピング領域の種別です。

注 2. アクセス幅およびエンディアンは，メモリ種別が外部領域においてデバッグ・ツールが切断中のみ変更可能です。また，指定可能値は選択したマイクロコントローラによって異なります。

注 3. エンディアン情報は，メモリ種別によって表示内容，および指定可能値が異なります。

- 外部領域【E1】【E20】【EZ Emulator】
  ドロップダウン・リストより次のいずれかを指定します。
  MCU のエンディアンと同じ，MCU のエンディアンと異なる
- I/O レジスタ領域
  次のいずれかを表示します。
  Little-endian データ，Big-endian データ
- エミュレーション ROM 領域，エミュレーション RAM 領域【シミュレータ】
  次のいずれかを指定します。
  Little-endian データ，Big-endian データ
- それ以外の領域
  エンディアン情報は表示されません。

(2) ［実行中のメモリ・アクセス］
プログラム実行中のメモリ・アクセス（リアルタイム表示更新機能（「2.11.1.4 プログラム実行中にメモリの内容を表示／変更する」参照））に関する詳細情報の表示，および設定の変更を行います。

実行を一瞬停止してアクセスする【E1】【E20】【EZ Emulator】	プログラム実行中にはアクセスできないメモリ領域に対して，アクセスを許可するか否かを指定します。		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	実行を一旦停止し，読み込み／書き込みを行います。
		いいえ	実行中にアクセスは行いません。
実行中に表示更新を行う	プログラム実行中に，ウォッチ パネル／メモリ パネルの表示内容を更新するか否かを指定します。		
	デフォルト	*はい*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	実行中に表示を更新します。
		いいえ	実行中に表示を更新しません。
表示更新間隔 [ms]	プログラム実行中に，ウォッチ パネル／メモリ パネルの表示内容を更新する間隔を 100 ms 単位で指定します。 なお，このプロパティは，［実行中に表示更新を行う］プロパティにおいて［はい］を指定した場合のみ表示されます。		
	デフォルト	*500*	
	変更方法	キーボードからの直接入力	
	指定可能値	100 ～ 65500 の整数（単位：100 ms 未満の端数切り上げ）	

リアルタイム表示更新を自動設定する【E20(JTAG)【RX600 シリーズ】】	RRM 領域を自動的に設定するか否かを指定します。 なお，このプロパティは，[トレース機能の用途] プロパティにおいて [リアルタイム RAM モニタ] を指定し，且つ［実行中に表示更新を行う］プロパティにおいて［はい］を指定した場合のみ表示されます。		
	デフォルト	*はい*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	自動的にリアルタイム表示更新の設定を行います。
		いいえ	リアルタイム表示更新設定はしません。

(3) ［レジスタ］
プログラム実行中におけるステータス・バーへの PC 値表示，および表示更新間隔の変更を行います。

実行中に PC 表示を行う	プログラム実行中に PC 値をステータス・バーへ表示[注]するか否かを指定します。		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	実行中にステータス・バーへ PC 値を表示します。
		いいえ	実行中にステータス・バーへ PC 値を表示しません。 ステータス・バーには，「実行中」と表示されます。
PC 表示更新間隔 [ms]	プログラム実行中にステータス・バーへ表示される PC 値の更新間隔を 100ms 単位で指定します。 なお，このプロパティは［実行中に PC 表示を行う］プロパティにおいて，［はい］を指定した場合のみ表示されます。		
	デフォルト	*500*	
	変更方法	キーボードからの直接入力	
	指定可能値	100 ～ 65500 の整数（単位：100 ms 未満の端数切り上げ）	

注 【E1/E20/EZ Emulator【RX100 シリーズ】】
プログラム実行中の PC 表示機能をサポートしていないため，PC 値は表示されません。

(4) ［ブレーク］【E1】【E20】【EZ Emulator】
ブレーク機能に関する詳細情報の表示，および設定の変更を行います。

優先的に使用するブレークポイントの種類	エディタ パネル／逆アセンブル パネルにおいて，ソース行，または実行アドレスに対してマウスのワンクリック操作でブレークポイントを設定する際に，優先的に使用するブレークポイントの種別を指定します。		
	デフォルト	*ハードウエア・ブレーク*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	ソフトウエア・ブレーク	ソフトウエア・ブレークポイントを優先的に設定します。
		ハードウエア・ブレーク	ハードウエア・ブレークポイントを優先的に設定します。

(5) ［システム］【E1】【E20】【EZ Emulator】
エミュレーションシステムに関する詳細情報の表示，および設定の変更を行います。

内蔵プログラム ROM を書き換えるプログラムをデバッグする	内蔵プログラム ROM の書き換えを伴うプログラム（ROM P/E モードを使用したプログラムなど）をデバッグするか否かを指定します。注 1, 2		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択 ただし，デバッグ・ツールと切断中のみ変更可	
	指定可能値	はい	内蔵プログラム ROM の書き換えを伴うプログラムをデバッグします。
		いいえ	内蔵プログラム ROM の書き換えを伴うプログラムをデバッグしません。
内蔵データフラッシュを書き換えるプログラムをデバッグする	内蔵データフラッシュの書き換えを伴うプログラム（データフラッシュ P/E モードを使用したプログラムなど）をデバッグするか否かを指定します。注 1, 2, 4		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択 ただし，デバッグ・ツールと切断中のみ変更可	
	指定可能値	はい	内蔵データフラッシュの書き換えを伴うプログラムをデバッグします。
		いいえ	内蔵データフラッシュの書き換えを伴うプログラムをデバッグしません。
実行開始前に指定ルーチンを実行する	プログラム実行直前に指定ルーチンを実行するか否かを指定します。		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	はい	プログラム実行直前に指定ルーチンを実行します。
		いいえ	プログラム実行直前に指定ルーチンを実行しません。
実行開始前に実行するルーチン	プログラム実行直前に実行するアドレスを指定します。 このプロパティは，[実行開始前に指定ルーチンを実行する] プロパティにおいて [はい] を指定した場合のみ表示します。		
	デフォルト	*空欄*	
	変更方法	キーボードからの直接入力 ただし，プログラム実行停止中のみ変更可	
	指定可能値	選択するマイクロコントローラに依存	
ブレーク後に指定ルーチンを実行する	プログラムブレーク直後に指定ルーチンを実行するか否かを指定します		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	はい	プログラムブレーク直後に指定ルーチンを実行します。
		いいえ	プログラムブレーク直後に指定ルーチンを実行しません。

ブレーク後に実行するルーチン	プログラムブレーク直後に実行するアドレスを指定します。 このプロパティは，[ブレーク後に指定ルーチンを実行する] プロパティにおいて [はい] を指定した場合のみ表示します。	
	デフォルト	*空欄*
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可
	指定可能値	選択するマイクロコントローラに依存
指定ルーチン実行機能用ワーク RAM 開始アドレス	指定ルーチン実行機能用のワーク RAM 配置アドレスを指定します。 アドレス値は 4 バイト単位の値を指定してください。（入力値が 4 バイト単位の値でなかった場合は，自動的に補正されます。） なお，このプロパティは，[実行開始前に指定ルーチンを実行する] プロパティまたは，[ブレーク後に指定ルーチンを実行する] において [はい] を指定した場合のみ表示します。	
	デフォルト	*0x1000* 注 3
	変更方法	キーボードからの直接入力 ただし，プログラム実行停止中のみ変更可
	指定可能値	選択しているマイクロコントローラの内蔵 RAM 領域に合わせたアドレス値
指定ルーチン実行機能用ワーク RAM サイズ [バイト]	指定ルーチン実行機能用のワーク RAM のサイズを表示します。 このプロパティは，[実行開始前に指定ルーチンを実行する] プロパティまたは，[ブレーク後に指定ルーチンを実行する] において [はい] を指定した場合のみ表示します。	
	デフォルト	*選択しているマイクロコントローラに依存*
	変更方法	変更不可

注 1. 【E1】【E20】【EZ Emulator】
内蔵フラッシュ ROM（プログラム ROM 領域）をユーザ・プログラムで書き換えた後に，メモリパネル等で書き換えた内容を参照する場合は，[内蔵プログラム ROM を書き換えるプログラムをデバッグする] プロパティを [はい] に設定してください。
[内蔵プログラム ROM を書き換えるプログラムをデバッグする] プロパティを [いいえ] に設定している場合は，書き換えた内容を参照することができません。
同様にデータフラッシュ領域では，[内蔵データフラッシュを書き換えるプログラムをデバッグする] プロパティを [はい] に設定してください。

注 2. 【E1】【E20】【EZ Emulator】
ユーザ・プログラム実行中にユーザ・プログラムで書き換えた内蔵フラッシュ ROM（プログラム領域，データフラッシュ領域）の内容をメモリ パネル等で参照しないでください。
ユーザ・プログラム実行中に，書き換えた領域を参照した場合，正しく値が読み出せません。

注 3. 選択しているマイクロコントローラによって，値が異なる場合があります。

注 4. 【RX71M, RX64M グループ】
RX71M, RX64M グループのオプション設定メモリを書き換えるプログラムをデバッグする場合，[内蔵データフラッシュを書き換えるプログラムをデバッグする] プロパティを [はい] に設定してください。

(6) ［トレース］
トレース機能に関する詳細情報の表示，および設定の変更を行います。

注意 【E20(JTAG)【RX600 シリーズ】】
トレース機能／リアルタイム RAM モニタ機能（RRM 機能）は一部排他使用の機能です。

トレース機能の用途 【E1】【E20】 【EZ Eulator】	トレース機能をリアルタイム RAM モニタ機能（RRM 機能）として使用するか否かを指定します。		
	デフォルト	*トレース*	
	変更方法	-【E1(Serial)】【E1(JTAG)】【E20(Serial)】【EZ Emulator】 変更不可 -【E20(JTAG)】 ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	トレース	トレース機能を優先的に使用します。
		リアルタイム RAM モニタ	リアルタイム RAM モニタ機能（RRM 機能）を優先的に使用します。
トレース機能を使用する 【シミュレータ】	トレース機能をするか否かを指定します。注 1		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	はい	トレース機能を使用します。
		いいえ	トレース機能を使用しません。
実行前にトレース・メモリをクリアする 【シミュレータ】	実行前にトレース・メモリをクリアするか否かを指定します。		
	デフォルト	*はい*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	実行前にトレース・メモリをクリアします。
		いいえ	実行前にトレース・メモリをクリアしません。
トレース・メモリを使い切った後の動作	トレースの取得モードを指定します。		
	デフォルト	*トレース・メモリを上書きし実行を続ける*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	トレース・メモリを上書きし実行を続ける	トレース・メモリを使い切った後もトレース・データを上書きします。
		トレースを停止する【E1】【E20】【EZ Emulator】	トレース・メモリを使い切った時点でトレース・データの書き込みを停止します。
		停止する	トレース・メモリを使い切った時点でプログラムの実行とトレース・データの書き込みを停止します。

トレース・データ種別 【E1】【E20】 【EZ Emulator】	トレース取得するデータ種別を指定します。		
	デフォルト	*分岐*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	ドロップダウン・リストによる次のいずれか - RX65N-2M, RX651-2M グループ【E20】 *分岐，分岐＋データアクセス，データアクセス，分岐＋データアクセス (アクセス・データなし)，データアクセス (アクセス・データなし)* - RX600 シリーズ *分岐，分岐＋データアクセス，データアクセス* - RX100, RX200 シリーズ *分岐，データアクセス*	
アクセス・データを取得しない領域の開始アドレス 【E20】【RX65N-2M, RX651-2M グループ】	アクセス・データを取得しない領域の開始アドレスを指定します。		
	デフォルト	*内蔵 RAM 領域の開始アドレス*	
	変更方法	キーボードによる直接入力	
	指定可能値	0x0 ～ 0xFFFFFFFF の 16 進数	
アクセス・データを取得しない領域の終了アドレス 【E20】【RX65N-2M, RX651-2M グループ】	アクセス・データを取得しない領域の終了アドレスを指定します。		
	デフォルト	*内蔵 RAM 領域の終了アドレス*	
	変更方法	キーボードによる直接入力	
	指定可能値	0x0 ～ 0xFFFFFFFF の 16 進数	
データアクセスのバス・マスタ 【E1/E20【RX71M, RX64M グループ】】	データアクセスを発生させたバス・マスタを指定します。 このプロパティは，[トレース・データ種別] プロパティにおいて，[分岐＋データアクセス]，または [データアクセス] を指定した場合のみ選択可能です。 このプロパティは，[トレース機能の用途] プロパティにおいて，[リアルタイム RAM モニタ] を指定した場合，[CPU] 固定となります。		
	デフォルト	*CPU*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	CPU	CPU からのデータアクセス結果を，トレースパネルに出力します。
		DMAC/DTC	DMAC/DTC からのデータアクセス結果を，トレースパネルに出力します。

外部トレース出力【E20(JTAG)【RX600 シリーズ】】	トレースの外部出力方法を指定します。		
	デフォルト	*CPU 実行優先*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	CPU 実行優先	トレース出力より CPU 実行を優先します。
		トレース出力優先	CPU 実行よりトレース出力を優先します。
		外部出力しない	トレース情報は出力されません。
トレース・メモリ・サイズ [M バイト]【E20(JTAG)【RX600 シリーズ】】	トレース・メモリのサイズを指定します。		
	デフォルト	*1*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	- ドロップダウン・リストよる次のいずれか *1，2，4，8，16，32*	
タイム・スタンプ出力【E1】【E20】【EZ Emulator】	収集するトレース・データにタイム・スタンプ情報を付加するか否かを指定します。注 3 このプロパティは，[トレース機能の用途] プロパティにおいて，[トレース] を指定した場合のみ選択可能です。		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	はい	タイム・スタンプ情報を付加します。
		いいえ	タイム・スタンプ情報を付加しません。
トレース・クロック・カウント・ソース [MHz]【E1】【E20】【EZ Emurator】	カウント値からタイム・スタンプを算出するためのカウント・ソースを入力します。 ただし，[タイム・スタンプ出力] プロパティにおいて，[はい] を指定した場合のみ表示されます。		
	デフォルト	*空欄*	
	変更方法	キーボードによる直接入力	
	指定可能値	0.0001 ～ 999.999	

トレース・クロック・カウント・ソースの分周比 【E1/E20【RX71M, RX64M グループ】】	トレースのタイム・スタンプ用カウント・ソースの分周比を指定します。 このプロパティは，[タイム・スタンプ出力] プロパティにおいて，[はい] を指定した場合のみ選択可能です。 [トレース・クロック・カウント・ソース [MHz]] プロパティで指定した周波数をさらに指定した値で分周（周波数を 1/n 倍に変換）した値が実際のタイム・スタンプの単位（1 カウントあたりの周波数）になります。		
	デフォルト	*分周なし*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	分周なし	タイム・スタンプ用カウント・ソースの周波数をそのままの値で使用する。
		16 分周	タイム・スタンプ用カウント・ソースの周波数を 1/16 倍に変換して使用する。
		256 分周	タイム・スタンプ用カウント・ソースの周波数を 1/256 倍に変換して使用する。
		4096 分周	タイム・スタンプ用カウント・ソースの周波数を 1/4096 倍に変換して使用する。
トレース・タイム・タグを積算する 【シミュレータ】	トレース パネルに表示するトレース時間の表示方法を指定します。		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	はい	トレースの時間表示を積算値で表示します。
		いいえ	トレースの時間表示を差分値で表示します。
トレース・メモリ・サイズ [フレーム] 【シミュレータ】	トレース・データを保持するメモリ・サイズをトレース・フレーム[注 2] 数で指定します。		
	デフォルト	*64K*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	- ドロップダウン・リストにより次のいずれか *64K，128K，256K，512K，1M，2M，3M*	

注 1. エディタ パネル／逆アセンブル パネルにおいて，コンテキスト・メニュー→［トレース開始の設定］／［トレース終了の設定］を選択した場合，このプロパティは自動的に［はい］に変更されます。

注 2. トレース・フレームはトレース・データの一単位を表します。
フェッチ／ライト／リードなどで，それぞれ 1 つのトレース・フレームを使用します。

注 3. RX100 シリーズでは，このプロパティは [いいえ] 固定となり，トレース・データにタイム・スタンプ情報を付加することはできません。

(7) ［タイマ］【E1】【E20】【EZ Emulator】
タイマ機能に関する詳細情報の表示，および設定の変更を行います。

カウンタを 64 ビットで使用する 【RX600 シリーズ】	64 ビットの測定カウンタを使用するか否かを指定します。 ただし，[はい] を指定した場合，計測区間は 1 区間となります。		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	64 ビット ×1 本の計測カウンタを使用します。
		いいえ	32 ビット ×2 本の計測カウンタを使用します。

注意 RX100 シリーズでは，タイマ機能は未サポートとなります。

(8) ［カバレッジ］【シミュレータ】

カバレッジ機能を利用する	カバレッジ機能を使用するか否かを指定します，		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	はい	カバレッジ機能を使用します。
		いいえ	カバレッジ機能を使用しません。
カバレッジ結果を再利用する	デバッグ・ツールと接続時／切断時に，カバレッジ測定結果のロード／セーブを行うか否かを指定します。 このプロパティは，［カバレッジ機能を利用する］プロパティにおいて［はい］を指定した場合のみ表示されます。		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	カバレッジ計測結果のロード／セーブを行います。
		いいえ	カバレッジ計測結果のロード／セーブを行いません。

(9) ［ストリーム入出力］【シミュレータ】

ストリーム入出力モードを選択	ストリーム入出力モードを指定します。		
	デフォルト	*シミュレータモード*	
	変更方法	ドロップダウン・リストによる選択 ただし，デバッグ・ツールと切断中のみ変更可	
	指定可能値	シミュレータモード	シミュレータ固有の入出力方式を使用します。 標準入出力およびファイル入出力が使用できます。
		エミュレータモード	エミュレータと同様の入出力方式を使用します。 標準入出力のみ使用できます。

ストリーム入出力機能を使用する	ストリーム入出力機能を使用するか否かを指定します。 このプロパティは，［ストリーム入出力モードを選択］プロパティにおいて［シミュレータモード］を指定した場合にのみ表示されます。		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	はい	ストリーム入出力機能を使用します。
		いいえ	ストリーム入出力機能を使用しません
ストリーム入出力用アドレス	ユーザプログラムから標準入出力またはファイル入出力を行うためのストリーム入出力の開始位置を指定します。 このプロパティは，［ストリーム入出力モードを選択］プロパティにおいて［シミュレータモード］を指定した場合にのみ表示されます。		
	デフォルト	*0*	
	変更方法	キーボードからの直接入力 ただし，プログラム実行停止中のみ変更可	
	指定可能値	0x0 ～ 0xFFFFFFFF の 16 進数	

(10) ［実行モード］【シミュレータ】

実行モードを選択	シミュレーションエラー，または例外が発生した場合に，ユーザプログラムの実行を停止するか否かを指定します。		
	デフォルト	*停止*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	停止	シミュレーションを停止します。
		続行	シミュレーションを続行します。
未定義命令例外処理発生時に停止する	未定義命令例外発生時にユーザプログラムの実行を停止するか否かを指定します。 このプロパティは，［実行モードを選択］プロパティにおいて［停止］を指定した場合のみ表示されます。		
	デフォルト	*はい*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	はい	シミュレーションを停止します。
		いいえ	シミュレーションを停止しません。

特権命令例外発生時に停止する	特権命令例外発生時にユーザプログラムの実行を停止するか否かを指定します。 このプロパティは，［実行モードを選択］プロパティにおいて［停止］を指定した場合のみ表示されます。		
	デフォルト	*はい*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	はい	シミュレーションを停止します。
		いいえ	シミュレーションを停止しません。
アクセス例外発生時に停止する	アクセス例外発生時にユーザプログラムの実行を停止するか否かを指定します。 このプロパティは，［周辺機能シミュレーションモジュール］プロパティに MPU モジュールが存在し，［実行モードを選択］プロパティにおいて［停止］を指定した場合のみ表示されます。		
	デフォルト	*はい*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	はい	シミュレーションを停止します。
		いいえ	シミュレーションを停止しません。
浮動小数点例外発生時に停止する 【FPU 搭載品】	浮動小数点例外発生時にユーザプログラムの実行を停止するか否かを指定します。 このプロパティは，［実行モードを選択］プロパティにおいて［停止］を指定した場合のみ表示されます。		
	デフォルト	*はい*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	はい	シミュレーションを停止します。
		いいえ	シミュレーションを停止しません。
割り込み発生時に停止する	割り込み発生時にユーザプログラムの実行を停止するか否かを指定します。 このプロパティは，［実行モードを選択］プロパティにおいて［停止］を指定した場合のみ表示されます。		
	デフォルト	*はい*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	はい	シミュレーションを停止します。
		いいえ	シミュレーションを停止しません。

INT 命令実行時に停止する	INT 命令実行時にユーザプログラムの実行を停止するか否かを指定します。 このプロパティは，［実行モードを選択］プロパティにおいて［停止］を指定した場合のみ表示されます。		
	デフォルト	*はい*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	はい	シミュレーションを停止します。
		いいえ	シミュレーションを停止しません。
BRK 命令実行時に停止する	BRK 命令実行時にユーザプログラムの実行を停止するか否かを指定します。 このプロパティは，［実行モードを選択］プロパティにおいて［停止］を指定した場合のみ表示されます。		
	デフォルト	*はい*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	はい	シミュレーションを停止します。
		いいえ	シミュレーションを停止しません。

(11) ［命令デコードキャッシュ］【シミュレータ】

命令デコード結果をキャッシュしてシミュレーション速度を向上する	命令デコードキャッシュ機能を使用するか否かを指定します。		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	はい	デコードキャッシュ機能を使用します。
		いいえ	デコードキャッシュ機能を使用しません。

(12) ［カバレッジ］【E20【RX71M, RX64M グループ】】

コード・カバレッジ機能を利用する	コード・カバレッジ機能を使用するか否かを指定します，		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択 ただし，デバッグ・ツールと切断中のみ変更可	
	指定可能値	はい	コード・カバレッジ機能を使用します。
		いいえ	コード・カバレッジ機能を使用しません。

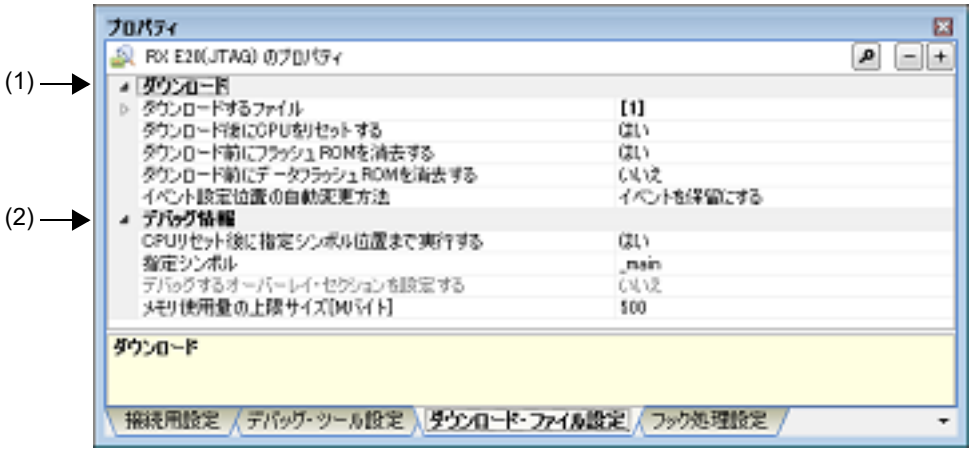
カバレッジ計測方法	カバレッジデータの計測方法を指定します。 このプロパティは，［コード・カバレッジ機能を利用する］プロパティにおいて［はい］を指定した場合のみ表示されます。		
	デフォルト	*CPU 実行優先*	
	変更方法	ドロップダウン・リストによる選択 ただし，プログラム実行停止中のみ変更可	
	指定可能値	CPU 実行優先	CPU 実行を優先します。データ出力時のロストによりカバレッジデータが失われる可能性があります。
		カバレッジ計測優先	カバレッジ計測を優先します。カバレッジデータ出力のために CPU 実行が停止し，プログラム実行のリアルタイム性が損なわれる可能性があります。
カバレッジ計測範囲	カバレッジを計測する範囲を指定します。 メインプロパティには，登録可能なカバレッジ計測範囲の数が表示されます。 サブプロパティには，開始アドレス，カバレッジ計測領域サイズ［M バイト］が下段に展開表示されます。 開始アドレスは，4M バイト単位の値を指定してください（入力値が 4M バイト単位の値でなかった場合は，自動的に補正されます）。 このプロパティは，［コード・カバレッジ機能を利用する］プロパティにおいて［はい］を指定した場合のみ表示されます。		
	デフォルト	- メイン項目 *4* - サブ項目 ＜開始アドレス＞ *空欄* ＜カバレッジ計測領域サイズ［M バイト］＞ *4*	
	変更方法	- メイン項目 変更不可 - サブ項目 ＜開始アドレス＞ キーボードからの直接入力 ＜カバレッジ計測領域サイズ［M バイト］＞ 変更不可 ただし，デバッグ・ツールと切断中のみ変更可	
	指定可能値	0x0 ～ 0xFFFFFFFF の 16 進数	
カバレッジ結果を再利用する	デバッグ・ツールと接続時／切断時に，カバレッジ測定結果のロード／セーブを行うか否かを指定します。 このプロパティは，［コード・カバレッジ機能を利用する］プロパティにおいて［はい］を指定した場合のみ表示されます。		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	カバレッジ計測結果のロード／セーブを行います。
		いいえ	カバレッジ計測結果のロード／セーブを行いません。

［ダウンロード・ファイル設定］タブ

［ダウンロード・ファイル設定］タブでは，次に示すカテゴリごとに詳細情報の表示，および設定の変更を行います。
なお，ダウンロード方法については，「2.5 ダウンロード／アップロード」を参照してください。

(1) ［ダウンロード］
(2) ［デバッグ情報］

図 A.21 プロパティ パネル：［ダウンロード・ファイル設定］タブ



［各カテゴリの説明］

(1) ［ダウンロード］
ダウンロードに関する詳細情報の表示，および設定の変更を行います。

ダウンロードするファイル	ダウンロードするファイルを指定します[注1]。 サブプロパティとして，ダウンロードするファイル名，およびダウンロード条件を下段に展開表示します。		
	デフォルト	*［ダウンロードするファイルの数］*	
	変更方法	ダウンロード・ファイル ダイアログによる選択 ダウンロード・ファイル ダイアログは，このプロパティを選択すると欄内右端に表示される［...］ボタンをクリックすることでオープンします（プロパティ パネル上でダウンロード・ファイルを指定することはできません）。	
ダウンロード後に CPU をリセットする	ダウンロード後に CPU をリセットするか否かを指定します。		
	デフォルト	*はい*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	ダウンロード後に CPU をリセットします。
		いいえ	ダウンロード後に CPU をリセットしません。
ダウンロード前にフラッシュ ROM を消去する 【E1】【E20】 【EZ Emulator】	ダウンロード前にフラッシュ ROM（プログラム ROM）を消去するか否かを指定します。		
	デフォルト	*はい*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	ダウンロード前にフラッシュ ROM を消去します。
		いいえ	ダウンロード前にフラッシュ ROM を消去しません。

ダウンロード前にデータフラッシュ ROM を消去する 【E1】【E20】 【EZ Emulator】	ダウンロード前にデータフラッシュ ROM を消去するか否かを指定します。		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	ダウンロード前にデータフラッシュ ROM を消去します。
		いいえ	ダウンロード前にデータフラッシュ ROM を消去しません。
イベント設定位置の自動変更方法	再ダウンロードすることにより，現在設定されているイベントの設定位置（アドレス）が命令の途中になる場合の再設定方法を指定します[注 2]。		
	デフォルト	*イベントを保留にする*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	命令の先頭に移動する	命令の先頭アドレスに対象イベントを再設定します。
		イベントを保留にする	対象イベントを保留状態にします。

注 1. メイン・プロジェクト／サブプロジェクトでビルド対象に指定しているファイルは，ダウンロードの対象ファイルから削除することはできません（デフォルトで自動的にダウンロード・ファイルとして登録されます）。
なお，ダウンロード可能なファイル形式については，「表 2.2 ダウンロード可能なファイル形式」を参照してください。

注 2. デバッグ情報がないイベント設定位置のみが対象となります。デバッグ情報がある場合のイベント設定位置は，常にソース・テキスト行の先頭に移動します。

(2) ［デバッグ情報］
デバッグ情報に関する詳細情報の表示，および設定の変更を行います。

CPU リセット後に指定シンボル位置まで実行する	CPU リセット後に，プログラムを指定シンボル位置まで実行するか否かを指定します。		
	デフォルト	*はい*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	プログラムを指定シンボル位置まで実行します。
		いいえ	CPU リセット後にプログラムを実行しません。
指定シンボル	CPU リセット後に，プログラムを実行して停止する位置を指定します。 なお，このプロパティは，［CPU リセット後に指定シンボル位置まで実行する］プロパティにおいて［はい］を選択した場合のみ表示されます。		
	デフォルト	*_main*	
	変更方法	キーボードからの直接入力	
	指定可能値	0 ～ “ *アドレス空間の終了アドレス* ” のアドレス式	
デバッグするオーバーレイ・セクションを設定する	ダウンロードしたロード・モジュール内にオーバーレイ・セクションが存在する場合に，デバッグ対象とするセクションを選択します。		
	デフォルト	- ロードモジュール内にオーバーレイ・セクションが存在しない場合 *いいえ* - ロードモジュール内にオーバーレイ・セクションが存在する場合 *はい*	
	変更方法	変更不可	

オーバーレイ・セクション	オーバーレイ・セクションが存在するアドレスグループを表示します。各アドレスグループに定義されたオーバーレイ・セクションの中から，デバッグするセクション（優先セクション）を選択します。 なお，このプロパティは，[デバッグするオーバーレイ・セクションを設定する] プロパティ項目において [はい] と表示された場合のみ表示されます。	
	デフォルト	*なし*
	変更方法	- [ファイル], [開始アドレス], [終了アドレス] 項目 変更不可 - [優先セクション] 項目 ドロップダウン・リストによる選択
	表示内容	各アドレスグループ番号の “+” マークをクリックすると，次の詳細情報が表示されます。 [優先セクション] 項目にて選択されているセクション名がデバッグ対象のセクションになります。 - ファイル - 開始アドレス - 終了アドレス - 優先セクション
メモリ使用量の上限サイズ [M バイト]	デバッグ情報の読み込みで使用するメモリ量の上限サイズ [M バイト] を指定します。 メモリ不足が発生する場合，上限サイズを小さくすることで改善される可能性があります。ただし，上限サイズを小さくするとデバッグ・ツールの応答性が低下する場合があります。	
	デフォルト	*500*
	変更方法	キーボードからの直接入力
	指定可能値	100 ～ 1000 の整数

注意 1. オーバーレイ・セクションに関する設定内容は，プロジェクトファイルに保存しません。ロード・モジュールのダウンロード後は，デバッグ対象のセクションを再設定してください。

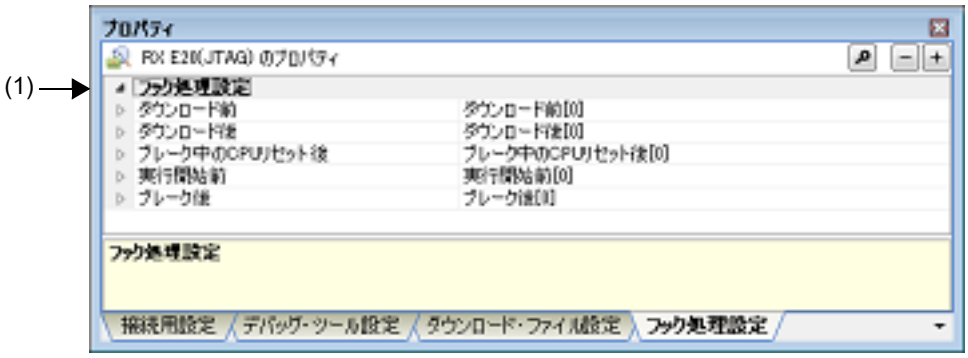
注意 2. [優先セクション] 項目の変更によって切り替わる情報は，デバッグ情報のみです。デバッガは対象セクションのデータのコピーは行いません。

［フック処理設定］タブ

［フック処理設定］タブでは，次に示すカテゴリごとに詳細情報の表示，および設定の変更を行います。
なお，フック処理の設定方法については，「2.18 フック処理を設定する」を参照してください。

(1) ［フック処理設定］

図 A.22 プロパティ パネル：［フック処理設定］タブ



［各カテゴリの説明］

(1) ［フック処理設定］
フック処理に関する詳細情報の表示，および設定の変更を行います。
なお，このタブ上のプロパティの設定は，各プロパティを選択すると右端に表示される [...] ボタンをクリックすることでオープンするテキスト編集 ダイアログで行います（このパネル上で処理を指定することはできません）。

注意 1 処理につき 64 文字まで入力可能で，各プロパティごとに 128 個までの処理を指定することができます（テキスト編集 ダイアログ上の［テキスト］エリア内の 1 行が 1 処理に相当）。

ダウンロード前	ロード・モジュール・ファイルをダウンロードする直前に行う処理を指定します。	
	デフォルト	*ダウンロード前 [0]*（“[ ]” 内は現在の指定処理数を示す）
	変更方法	テキスト編集 ダイアログによる指定
	指定可能値	次のいずれか - *I/O レジスタ名* + 半角スペース + *数値* 【処理】*I/O レジスタの内容を数値に自動的に書き換えます。* - *CPU レジスタ名* + 半角スペース + *数値* 【処理】*CPU レジスタの内容を数値に自動的に書き換えます。* - *Source*+ 半角スペース +*Python スクリプト・パス* 【処理】*Python スクリプト・パス*で指定したスクリプト・ファイルを実行します。
ダウンロード後	ロード・モジュール・ファイルをダウンロードした直後に行う処理を指定します。	
	デフォルト	*ダウンロード後 [0]*（“[ ]” 内は現在の指定処理数を示す）
	変更方法	テキスト編集 ダイアログによる指定
	指定可能値	次のいずれか - *I/O レジスタ名* + 半角スペース + *数値* 【処理】*I/O レジスタの内容を数値に自動的に書き換えます。* - *CPU レジスタ名* + 半角スペース + *数値* 【処理】*CPU レジスタの内容を数値に自動的に書き換えます。* - *Source*+ 半角スペース +*Python スクリプト・パス* 【処理】*Python スクリプト・パス*で指定したスクリプト・ファイルを実行します。

ブレーク中の CPU リセット後	ブレーク中の CPU リセット直後に行う処理を指定します。	
	デフォルト	*ブレーク中のCPU リセット後 [0]*（“[ ]” 内は現在の指定処理数を示す）
	変更方法	テキスト編集 ダイアログによる指定
	指定可能値	次のいずれか - *I/O レジスタ名* + 半角スペース + *数値* 【処理】*I/O レジスタの内容を数値に自動的に書き換えます。* - *CPU レジスタ名* + 半角スペース + *数値* 【処理】*CPU レジスタの内容を数値に自動的に書き換えます。* - *Source*+ 半角スペース +*Python スクリプト・パス* 【処理】*Python スクリプト・パス*で指定したスクリプト・ファイルを実行します。
実行開始前	プログラムの実行開始直前に行う処理を指定します。	
	デフォルト	*実行開始前 [0]*（“[ ]” 内は現在の指定処理数を示す）
	変更方法	テキスト編集 ダイアログによる指定
	指定可能値	次のいずれか - *I/O レジスタ名* + 半角スペース + *数値* 【処理】*I/O レジスタの内容を数値に自動的に書き換えます。* - *CPU レジスタ名* + 半角スペース + *数値* 【処理】*CPU レジスタの内容を数値に自動的に書き換えます。* - *Source*+ 半角スペース +*Python スクリプト・パス* 【処理】*Python スクリプト・パス*で指定したスクリプト・ファイルを実行します。
ブレーク後	プログラムの実行がブレークした直後に行う処理を指定します。	
	デフォルト	*ブレーク後 [0]*（“[ ]” 内は現在の指定処理数を示す）
	変更方法	テキスト編集 ダイアログによる指定
	指定可能値	次のいずれか - *I/O レジスタ名* + 半角スペース + *数値* 【処理】*I/O レジスタの内容を数値に自動的に書き換えます。* - *CPU レジスタ名* + 半角スペース + *数値* 【処理】*CPU レジスタの内容を数値に自動的に書き換えます。* - *Source*+ 半角スペース +*Python スクリプト・パス* 【処理】*Python スクリプト・パス*で指定したスクリプト・ファイルを実行します。

メモリ パネル

メモリの内容の表示，変更を行います（「2.11.1 メモリを表示／変更する」参照）。
このパネルは，最大 4 個までオープンすることができます。各パネルは，タイトルバーの“メモリ 1”，“メモリ 2”，“メモリ 3”，“メモリ 4”の名称で識別されます。
プログラムの実行後，メモリの値が変化すると表示を自動的に更新します（ステップ実行時には，ステップ実行ごとに表示を逐次更新）。
また，リアルタイム表示更新機能を有効にすることにより，プログラム実行中であっても，値の表示をリアルタイムに更新することも可能です。
なお，このパネルは，デバッグ・ツールと接続時のみオープンすることができます。

備考　ツールバーの [表示] → ボタンをクリックすることによりオープンするスクロール範囲設定 ダイアログにより，このパネルの垂直スクロール・バーのスクロール範囲を設定することができます。

図 A.23　メモリ パネル

ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー（メモリ パネル専用部分)]
- [[編集] メニュー（メモリ パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [メモリ] → [メモリ *1* ～ *4*] を選択

[各エリアの説明]

(1)　表示位置指定エリア
アドレス式を指定することにより，メモリ値の表示開始位置を指定することができます。
次の指定を順次行います。

(a)　アドレス式の指定
表示したいメモリ値のアドレスとなるアドレス式をテキスト・ボックスに直接入力します。最大 1024 文字までの入力式を指定することができ，その計算結果を表示開始位置アドレスとして扱います。
ただし，マイクロコントローラのアドレス空間よりも大きいアドレス式を指定することはできません。

備考 1. このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます（「2.20.2 シンボル名の入力補完機能」参照）。

備考 2. 指定したアドレス式がシンボルを表現し，サイズが判明する場合では，そのシンボルの先頭アドレスから終了アドレスまでを選択状態で表示します。

(b) アドレス式の自動／手動評価の指定
表示開始位置を変更するタイミングは，［停止時に移動］チェック・ボックスの指定，および［移動］ボタンにより決定します。

［停止時に移動］	☑	プログラム停止後，自動的にアドレス式の評価を行い，その計算結果のアドレスにキャレットが移動します。
	☐	プログラム停止後，アドレス式の評価を自動的に行いません。 この場合，［移動］ボタンをクリックすることにより，アドレス式の評価を行います。
［移動］		［停止時に移動］チェック・ボックスのチェックをしなかった場合，このボタンをクリックすることによりアドレス式の評価を行い，その計算結果のアドレスにキャレットが移動します。

(2) アドレス・エリア
メモリのアドレスを表示します（16 進数表記固定）。
デフォルトで，0x0 番地より表示を開始します。
ただし，コンテキスト・メニューの [表示アドレス・オフセット値の設定 ...] を選択することでオープンするアドレス・オフセット設定 ダイアログにより，開始アドレスにオフセット値を設定することができます。
アドレス幅は，プロジェクトで指定しているマイクロコントローラのメモリ空間のアドレス幅となります。
このエリアを編集することはできません。

注意 設定されたオフセット値は，メモリ値エリアの表示桁数の指定に従って自動的に変更されます。

(3) メモリ値エリア
マイクロコントローラにマッピングされているプログラム用のメモリ値を表示／変更します。
メモリ値の表示進数／表示幅／表示桁の指定は，ツールバーのボタン，またはコンテキスト・メニューの［表記］／［サイズ表記］／［表示］の選択により行います（「2.11.1.2 値の表示形式を変更する」参照）。
メモリ値として表示されるマークや色の意味は次のとおりです（表示の際の文字色／背景色はオプション ダイアログにおける［全般 - フォントと色］カテゴリの設定に依存）。

表示例（デフォルト）			説明
00	文字色	青	ユーザにより，値が変更されているメモリ値（[Enter] キーによりターゲット・メモリに書き込まれます）
	背景色	標準色	
00 （下線）	文字色	標準色	シンボルが定義されているアドレスのメモリ値（ウォッチ式の登録を行うことができます）
	背景色	標準色	
00	文字色	茶色	プログラムの実行により，値が変化したメモリ値[注 1] ツールバーの ボタンをクリックすると，強調表示をリセットします。
	背景色	クリーム	
00	文字色	ピンク	リアルタイム表示更新機能を行っているメモリ値
	背景色	標準色	

表示例（デフォルト）			説明	
00	文字色	標準色	リード／フェッチ	リアルタイム表示更新機能を行っている場合，現在のメモリ値のアクセス状態
	背景色	薄緑		
00	文字色	標準色	ライト	
	背景色	オレンジ		
00	文字色	標準色	リードとライト	
	背景色	薄青		
00	文字色	白	ロスト	
	背景色	ライトグレー		
00	文字色	グレー	リード不可の領域のメモリ値	
	背景色	標準色		
??	文字色	グレー	メモリがマッピングされていない領域，書き換え不可能領域（IO レジスタ領域 /I/O 保護領域注 2 など），またはメモリ値の取得に失敗した場合	
	背景色	標準色		
**	文字色	標準色	プログラム実行中に，リアルタイム表示更新領域以外の領域を表示指定した場合，またはメモリ値の取得に失敗した場合	
	背景色	標準色		

注 1. プログラム実行直前において，メモリ パネルで表示されていたアドレス範囲のメモリ値のみが対象となります。また，プログラムの実行前後での値の比較であるため，実行結果が同一値となった場合は強調表示を行いません。

注 2. 【E1】【E20】【EZ Emulator】
I/O 保護領域は，“00” と表示されます。

注意 このエリアの表示桁数は，コンテキスト・メニューの［サイズ表記］の指定に従って自動的に変更します。

このエリアは，次の機能を備えています。

(a) ポップアップ表示
メモリ値にマウス・カーソルを重ねることにより，マウス・カーソルが指しているアドレスに対して前方向に存在する一番近いシンボルを基準にして，次の内容をポップアップ表示します。
ただし，シンボル情報が存在しない場合（下線が非表示の場合）はポップアップ表示は行いません。

```
variable    + 0x14
 |            |
シンボル名    オフセット値
```

シンボル名	シンボル名を表示します。
オフセット値	アドレスにシンボルが定義されていない場合は，前方向に存在する一番近いシンボルからのオフセット値を表示します（16 進数表示固定）。

(b) リアルタイム表示更新機能
リアルタイム表示更新機能を使用することにより，プログラムが停止している状態の時だけでなく，実行中の状態であっても，メモリ値の表示／変更を行うことができます。
リアルタイム表示更新機能についての詳細は，「2.11.1.4 プログラム実行中にメモリの内容を表示／変更する」を参照してください。

(c) メモリ値の変更
メモリ値の変更は，対象メモリ値にキャレットを移動したのち，直接キーボードより編集することで行います。

メモリ値を編集すると変更箇所の表示色が変化し，この状態で［Enter］キーを押下することにより変更した値がターゲット・メモリに書き込まれます（［Enter］キーの押下前に［Esc］キーを押下すると編集をキャンセルします）。
メモリ値の変更方法についての詳細は，「2.11.1.3 メモリの内容を変更する」を参照してください。

(d) メモリ値の検索／初期化
コンテキスト・メニューの［検索 ...］を選択することにより，指定したアドレス範囲のメモリ内容を検索するためのメモリ検索 ダイアログをオープンします（「2.11.1.5 メモリの内容を検索する」参照）。
また，コンテキスト・メニューの［初期化 ...］を選択することにより，指定したアドレス範囲のメモリ内容を一括して変更するためのメモリ初期化 ダイアログをオープンします（「2.11.1.6 メモリの内容を一括して変更（初期化）する」参照）。

(e) コピー／貼り付け
メモリ値をマウスにより範囲選択することで，その箇所の内容を文字列としてクリップ・ボードにコピーすることができ，その内容をキャレット位置に貼り付けることができます。
これらの操作は，コンテキスト・メニューの選択，または［編集］メニューの選択により行います。
ただし，貼り付け操作は，貼り付け対象の文字列とそのエリアの表示形式（表示進数／ビット幅）が一致する場合のみ可能です（表示形式が一致しない場合は，メッセージを表示します）。
なお，このエリアで扱うことができる文字コードと文字列は次のとおりです（これ以外の文字列を貼り付けた場合は，メッセージを表示します）。

文字コード	ASCII
文字列	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, A, B, C, D, E, F

(f) ウォッチ式の登録
シンボルが定義されているアドレスでは，メモリ値に下線が表示され，ウォッチ式として登録可能であることを示します。
このメモリ値を選択，またはメモリ値のいずれかにキャレットを置いた状態で，コンテキスト・メニューの［ウォッチ 1 に登録］を選択することにより，指定したアドレスのシンボル名がウォッチ式としてウォッチ パネル（ウォッチ 1）に登録されます。

注意 下線表示のないメモリ値をウォッチ式に登録することはできません。

(g) メモリ値の保存
［ファイル］メニュー→［名前を付けてメモリ・データを保存 ...］を選択することにより，データ保存 ダイアログをオープンし，このパネルの内容をテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存することができます。
メモリ値の保存方法についての詳細は，「2.11.1.7 メモリの表示内容を保存する」を参照してください。

(h) 表示の拡大／縮小
メモリ パネルにフォーカスがある状態で，メイン・ウインドウのツールバー上にある拡大率のドロップダウン・リストを変更することにより，メモリ パネルの表示を拡大／縮小することができます。
また，［Ctrl］キーとマウス・ホイールを組み合わせることでも変更可能です。

- ［Ctrl］キーを押下しながらマウス・ホイールを前方に動かすと，メモリ パネルの表示を拡大し，表示が大きく見やすくなります（最大 300%）。
- ［Ctrl］キーを押下しながらマウス・ホイールを後方に動かすと，メモリ パネルの表示を縮小し，表示が小さくなります（最小 50%）。

なお，拡大率を変更した状態のパネルをクローズした場合，拡大率は保持されます（再度パネルをオープンする際に，変更した拡大率でオープンします）。

(4) 文字列エリア
メモリの値を文字コードに変換して表示します。
文字コードの指定は，ツールバーのボタンのクリック，またはコンテキスト・メニューの［エンコード］の選択により行います（デフォルトでは ASCII コードで表示します）。
また，このエリアでは，メモリ値を浮動小数点数値に変換して文字列として表示することができます。この場合は，コンテキスト・メニューの［エンコード］より次の項目の選択を行ってください。

項目	表記		サイズ
Half-Precision Float	半精度浮動小数点数		16 ビット
	数値	*＜符号＞＜仮数部＞* e *＜符号＞＜指数部＞*	
	無限大	Inf，および -Inf	
	非数	NaN	
	例	+ 1.234e+1	
Float	単精度浮動小数点数		32 ビット
	数値	*＜符号＞＜仮数部＞* e *＜符号＞＜指数部＞*	
	無限大	Inf，および -Inf	
	非数	NaN	
	例	+1.234567e+123	
Double	倍精度浮動小数点数		64 ビット
	数値	*＜符号＞＜仮数部＞* e *＜符号＞＜指数部＞*	
	無限大	Inf，および -Inf	
	非数	NaN	
	例	+1.2345678901234e+123	
Float Complex	単精度浮動小数点数の複素数		64 ビット
	＜単精度浮動小数点数＞＜単精度浮動少数点数＞ *1		
Double Complex	倍精度浮動小数点数の複素数		128 ビット
	＜倍精度浮動小数点数＞＜倍精度浮動少数点数＞ *1		
Float Imaginary	単精度浮動少数点の虚数		32 ビット
	＜単精度浮動小数点数＞ *1		
Double Imaginary	倍精度浮動小数点数の虚数		64 ビット
	＜倍精度浮動少数点数＞ *1		

注意 指定されている文字コード，または浮動小数点数値の最小サイズが " 表示バイト数 × 表示桁数バイト数 " より大きい場合，このエリアには何も表示されません。

このエリアは，次の機能を備えています。

(a) 文字列の変更
現在，文字コードとして［ASCII］が指定されている場合のみ，文字列を変更することができます。
文字列の変更は，対象文字列にキャレットを移動したのち，直接キーボードより編集することで行います。
文字列を編集すると変更箇所の表示色が変化し，この状態で［Enter］キーの押下することにより変更した値がターゲット・メモリに書き込まれます（［Enter］キーの押下前に［Esc］キーを押下すると編集をキャンセルします）。

注意 浮動少数点数値表示している文字列を変更することはできません。

(b) 文字列の検索
コンテキスト・メニューの［検索 ...］を選択することにより，文字列を検索する為の メモリ検索 ダイアログをオープンします（「2.11.1.5 メモリの内容を検索する」参照）。

(c) コピー／貼り付け
文字列をマウスにより範囲選択することで，その箇所の内容を文字列としてクリップ・ボードにコピーすることができ，その内容をキャレット位置に貼り付けることができます。
これらの操作は，コンテキスト・メニューの選択，または［編集］メニューの選択により行います。
ただし，貼り付け操作は，文字コードとして ASCII が指定されている場合のみ可能です（ASCII 以外が指定されている場合は，メッセージを表示します）。

［ツールバー］

	デバッグ・ツールから最新の情報を取得し，表示を更新します。
	プログラム実行により値が変化した箇所を示す強調表示をリセットします。 ただし，プログラム実行中は無効となります。
表記	メモリ値の表示形式を変更する次のボタンを表示します。 ただし，プログラム実行中は無効となります。
16 進数	メモリ値を 16 進数で表示します（デフォルト）。
符号付き 10 進数	メモリ値を符号付き 10 進数で表示します。
符号無し 10 進数	メモリ値を符号なし 10 進数で表示します。
8 進数	メモリ値を 8 進数で表示します。
2 進数	メモリ値を 2 進数で表示します。
サイズ表記	メモリ値のサイズの表示形式を変更する次のボタンを表示します。 ただし，プログラム実行中は無効となります。
4 ビット	メモリ値を 4 ビット幅で表示します。
1 バイト	メモリ値を 8 ビット幅で表示します（デフォルト）。
2 バイト	メモリ値を 16 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。
4 バイト	メモリ値を 32 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。
8 バイト	メモリ値を 64 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。

エンコード	文字列のエンコードを変更する次のボタンを表示します。 ただし，プログラム実行中は無効となります。
ASCII	文字列を ASCII コードで表示します（デフォルト）。
Shift_JIS	文字列を Shift_JIS コードで表示します。
EUC-JP	文字列を EUC-JP コードで表示します。
UTF-8	文字列を UTF-8 コードで表示します。
UTF-16	文字列を UTF-16 コードで表示します。
Half-Precision Float	文字列を半精度浮動小数点数値で表示します。
Float	文字列を単精度浮動小数点数値で表示します。
Double	文字列を倍精度浮動小数点数値で表示します。
Float Complex	文字列を単精度浮動少数点数の複素数で表示します。
Double Complex	文字列を倍精度浮動少数点数の複素数で表示します。
Float Imaginary	文字列を単精度浮動小数点数の虚数で表示します。
Double Imaginary	文字列を倍精度浮動小数点数の虚数で表示します。
表示	表示形式を変更する次のボタンを表示します。
スクロール範囲を設定 ...	スクロール範囲を設定するためのスクロール範囲設定 ダイアログがオープンします。
表示桁数を設定 ...	メモリ値エリアの表示桁数を変更するため，表示桁数設定 ダイアログがオープンします。
表示アドレス・オフセット値を設定 ...	アドレス・エリアに表示するアドレスおよびオフセット値を設定するため，アドレス・オフセット設定 ダイアログがオープンします。

[[ファイル] メニュー（メモリ パネル専用部分)]

メモリ パネル専用の［ファイル］メニューは次のとおりです（その他の項目は共通）。
ただし，プログラム実行中はすべて無効となります。

メモリ・データを保存	メモリの内容を前回保存したテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存します（「(g) メモリ値の保存」参照）。 なお，起動後に初めてこの項目を選択した場合は，［名前を付けてメモリ・データを保存 ...］の選択と同等の動作となります。
名前を付けてメモリ・データを保存 ...	メモリの内容を指定したテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存するために，データ保存 ダイアログをオープンします（「(g) メモリ値の保存」参照）。

[[編集] メニュー（メモリ パネル専用部分)]

メモリ パネル専用の［編集］メニューは次のとおりです（その他の項目はすべて無効）。
ただし，プログラム実行中はすべて無効となります。

コピー	選択している範囲を文字列としてクリップ・ボードにコピーします。
貼り付け	クリップ・ボードにコピーされている文字列をキャレット位置に貼り付けます。 - メモリ値エリアに貼り付ける場合： 「(e) コピー／貼り付け」参照 - 文字列エリアに貼り付ける場合： 「(c) コピー／貼り付け」参照

検索 ...	メモリ検索 ダイアログをオープンします。 検索対象となる箇所は，メモリ値エリアと文字列エリアのうち，キャレットのあるエリア内となります。

［コンテキスト・メニュー］

ウォッチ 1 に登録	キャレット位置のシンボルをウォッチ パネル（ウォッチ 1）に登録します。 ウォッチ式として登録される際は変数名として登録されるため，スコープにより表示されるシンボル名は変化します。 ただし，キャレット位置のメモリ値に対応するアドレスにシンボルが定義されていない場合は無効となります（「(f) ウォッチ式の登録」参照）。
検索 ...	メモリ検索 ダイアログをオープンします。 検索対象となる箇所は，メモリ値エリアと文字列エリア（浮動小数点数値表示を選択している場合を除く）のうち，キャレットのあるエリア内となります。 ただし，プログラム実行中は無効となります。
初期化 ...	メモリ初期化 ダイアログをオープンします。
最新の情報に更新	デバッグ・ツールから最新の情報を取得し，表示を更新します。
コピー	選択している範囲を文字列としてクリップ・ボードにコピーします。 ただし，プログラム実行中は無効となります。
貼り付け	クリップ・ボードにコピーされている文字列を キャレット位置に貼り付けます。 ただし，プログラム実行中は無効となります。 - メモリ値エリアに貼り付ける場合： 「(e) コピー／貼り付け」参照 - 文字列エリアに貼り付ける場合： 「(c) コピー／貼り付け」参照
表記	メモリ値エリアの表示進数を指定するため，次のカスケード・メニューを表示します。 ただし，プログラム実行中は無効となります。
16 進数	メモリ値を 16 進数で表示します（デフォルト）。
符号付き 10 進数	メモリ値を符号付き 10 進数で表示します。
符号なし 10 進数	メモリ値を符号なし 10 進数で表示します。
8 進数	メモリ値を 8 進数で表示します。
2 進数	メモリ値を 2 進数で表示します。
サイズ表記	メモリ値エリアの表示幅を指定するため，次のカスケード・メニューを表示します。 ただし，プログラム実行中は無効となります。
4 ビット	メモリ値を 4 ビット幅で表示します。
1 バイト	メモリ値を 8 ビット幅で表示します（デフォルト）。
2 バイト	メモリ値を 16 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。
4 バイト	メモリ値を 32 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。
8 バイト	メモリ値を 64 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。

エンコード	文字列エリアの表示形式を指定するため，次のカスケード・メニューを表示します。 ただし，プログラム実行中は無効となります。
ASCII	文字列を ASCII コードで表示します（デフォルト）。
Shift_JIS	文字列を Shift_JIS コードで表示します。
EUC-JP	文字列を EUC-JP コードで表示します。
UTF-8	文字列を UTF-8 コードで表示します。
UTF-16	文字列を UTF-16 コードで表示します。
Half-Precision Float	文字列を半精度浮動小数点数値で表示します。
Float	文字列を単精度浮動小数点数値で表示します。
Double	文字列を倍精度浮動少数点数値で表示します。
Float Complex	文字列を単精度浮動小数点数の複素数で表示します。
Double Complex	文字列を倍精度浮動小数点数の複素数で表示します。
Float Imaginary	文字列を単精度浮動少数点数の虚数で表示します。
Double Imaginary	文字列を倍精度浮動小数点数の虚数で表示します。
表示	表示形式を変更するため，次のカスケード・メニューを表示します。
スクロール範囲を設定 ...	スクロール範囲を設定するためのスクロール範囲設定 ダイアログがオープンします。
表示桁数を設定	メモリ値エリアの表示桁数を設定するため，表示桁数設定 ダイアログがオープンします。
表示アドレス・オフセット値を設定	アドレス・エリアに表示するアドレスのオフセット値を設定するため，アドレス・オフセット設定 ダイアログがオープンします。
強調表示	チェックすることにより，プログラムの実行により値が変更されたメモリ値を強調表示します（デフォルト）。 ただし，プログラム実行中は無効となります。
リアルタイム表示更新設定	リアルタイム表示更新設定のため，次のカスケード・メニューを表示します（「(b) リアルタイム表示更新機能」参照）。
リアルタイム表示更新全体設定	リアルタイム表示更新機能の全般設定を行うため，プロパティ パネルをオープンします。

逆アセンブル パネル

メモリ内容を逆アセンブルした結果（逆アセンブル・テキスト）の表示，ライン・アセンブル（「2.6.4 ライン・アセンブルを行う」参照），命令レベル・デバッグ（「2.9.3 プログラムをステップ実行する」参照），およびコード・カバレッジ測定結果の表示【シミュレータ】【E20【RX71M, RX64M グループ】】（「2.15 カバレッジの測定【シミュレータ】【E20【RX71M, RX64M グループ】】」参照）を行います。

このパネルは，最大 4 個までオープンすることができます。各パネルは，タイトルバーの“逆アセンブル 1”，“逆アセンブル 2”，“逆アセンブル 3”，“逆アセンブル 4”の名称で識別されます。

混合表示モードにすることにより，コード・データに対応するソース・ファイル中のソース・テキストも表示することができます（デフォルト）。

なお，このパネルは，デバッグ・ツールと接続時のみオープンすることができます。

注意 このパネルにフォーカスがある状態でプログラムをステップ実行した場合，実行単位は命令レベル単位となります（「2.9.3 プログラムをステップ実行する」参照）。

備考 1. ツールバーの [表示] → ボタンをクリックすることでオープンするスクロール範囲設定 ダイアログにより，このパネルの垂直スクロール・バーのスクロール範囲を設定することができます。

備考 2. [ファイル] メニュー→ [印刷 ...] を選択することにより，現在このパネルで表示している画像イメージを印刷することができます。

図 A.24 逆アセンブル パネル（混合表示モードの場合）

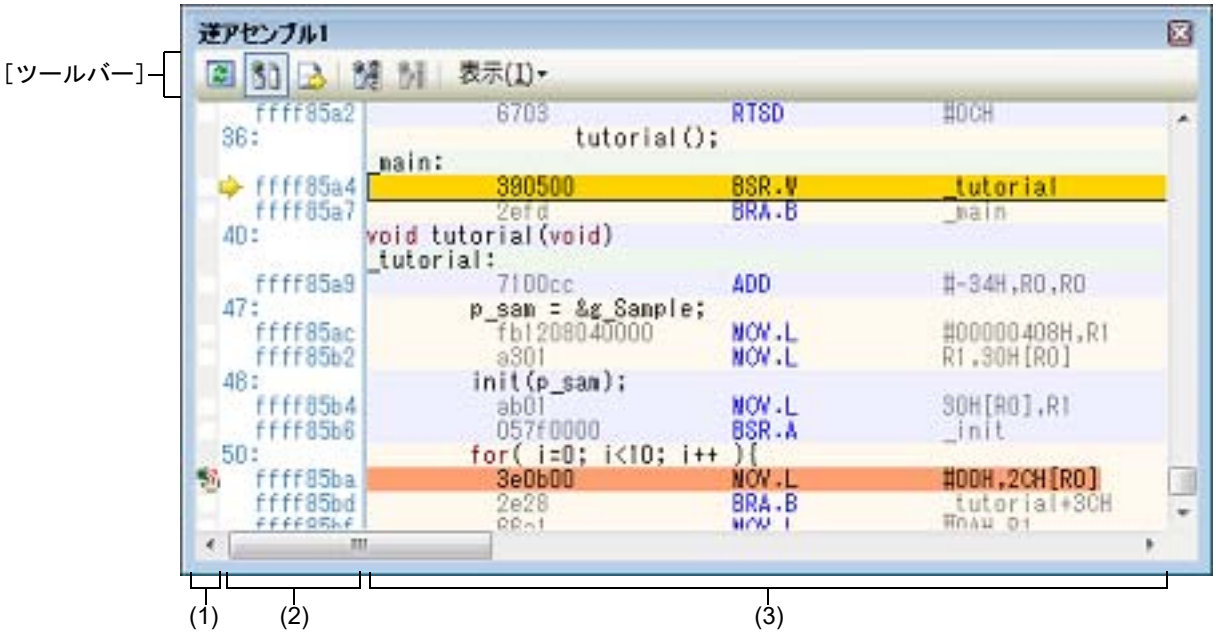


図 A.25 逆アセンブル パネル（混合表示モードを解除した場合）

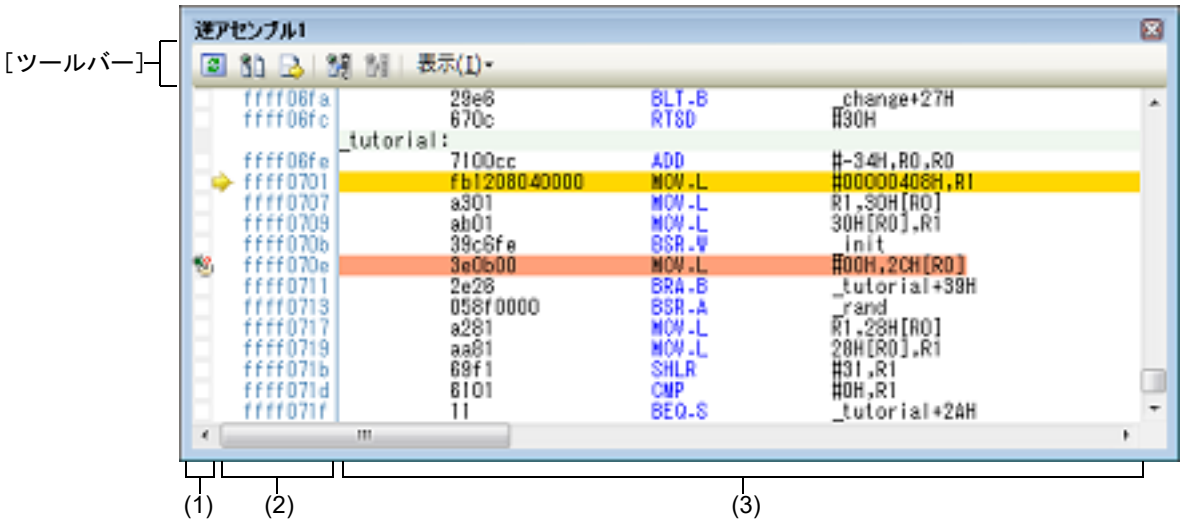
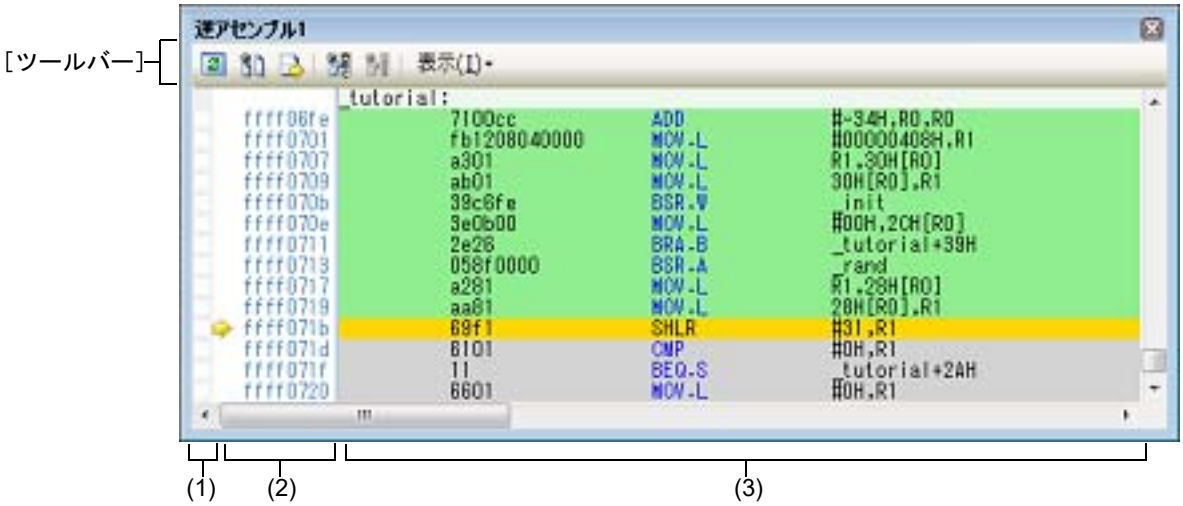


図 A.26 逆アセンブル パネル（コード・カバレッジ測定結果を表示した場合）【シミュレータ】【E20【RX71M, RX64M】】



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー（逆アセンブル パネル専用部分）]
- [[編集] メニュー（逆アセンブル パネル専用部分）]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [逆アセンブル] → [逆アセンブル *1* ～ *4*] を選択

[各エリアの説明]

(1) イベント・エリア
イベントの設定が可能な行は，背景色を白色で表示します（背景色がグレー表示の行は，イベントの設定が不可能であることを示すします）。
また，現在設定しているイベントがある場合，そのイベント設定行に，イベント種別を示すイベント・マークを表示します。
このエリアは，次の機能を備えています。

(a) ブレークポイントの設定／削除
ブレークポイントを設定したい箇所をマウスでクリックすることにより，容易にブレークポイントを設定することができます。
ブレークポイントは，クリックした行位置に対応する先頭アドレスの命令に対して設定されます。
ブレークポイントを設定すると，設定した行にイベント・マークが表示されます。また，設定したブレークポイントの詳細情報がイベント パネルに反映されます。
なお，すでにいずれかのイベント・マークが表示されている箇所において，この操作を行った場合は，そのイベントを削除し，ブレークポイントの設定は行いません。
ただし，イベントの設定は，背景色が白色で表示されている行に対してのみ行うことができます。
ブレークポイントの設定方法についての詳細は，「2.10.2 任意の場所で停止する（ブレークポイント）」を参照してください。

(b) 各種イベントの状態変更
各種イベント・マークを右クリックすることにより，次のメニューが表示され，選択したイベントの状態の変更を行うことができます。

有効化	選択しているイベントを有効状態にします。 指定されている条件の成立で，対象となるイベントが発生します。 なお，複数のイベントが設定されていることを示すイベント・マーク（ ）を選択している場合は，設定されているすべてのイベントを有効状態にします。
無効化	選択しているイベントを無効状態にします。 指定されている条件が成立しても，対象となるイベントは発生しません。 なお，複数のイベントが設定されていることを示すイベント・マーク（ ）を選択している場合は，設定されているすべてのイベントを無効状態にします。
イベント削除	選択しているイベントを削除します。 なお，複数のイベントが設定されていることを示すイベント・マーク（ ）を選択している場合は，設定されているすべてのイベントを削除します。
イベント パネルで詳細を表示	選択しているイベントの詳細情報を表示するイベント パネルをオープンします。

(c) ポップアップ表示
イベント・マークにマウス・カーソルを重ねることにより，そのイベントのイベント名／詳細情報／イベントに付加されたコメントをポップアップ表示します。
なお，該当箇所に複数のイベントが設定されている場合，最大 3 つまで，各イベントの情報を列挙して表示します。

(2) アドレス・エリア
行ごとの逆アセンブル開始アドレスを表示します（16 進数表記固定）。
また，カレント PC 位置（PC レジスタ値）を示すカレント PC マーク（ ）を表示します。
アドレス幅は，プロジェクトで指定しているマイクロコントローラのメモリ空間のアドレス幅となります。
なお，混合表示モード時におけるソース・テキスト行に対しては，開始アドレスに対応するソース・ファイル中の行番号（*xxx* :）を表示します。
このエリアは，次の機能を備えています。

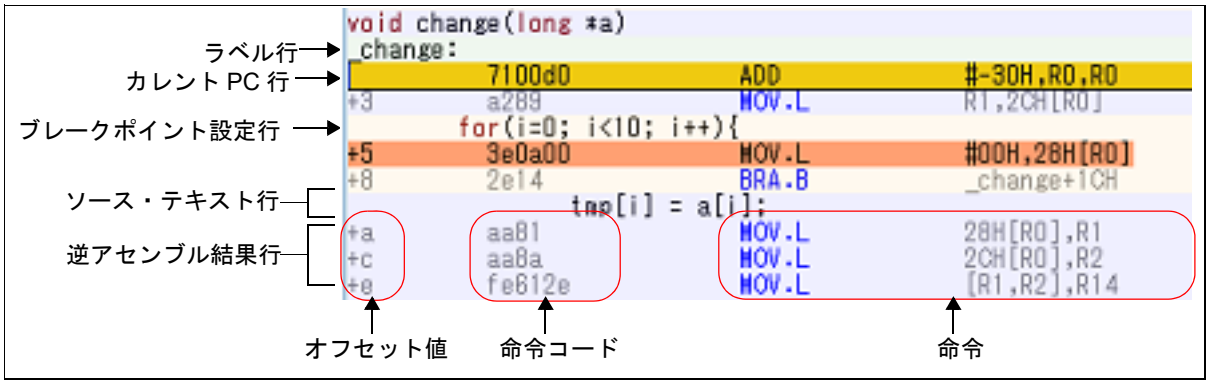
(a) ポップアップ表示
アドレス／ソース行番号にマウス・カーソルを重ねることにより，次の情報をポップアップ表示します。

アドレス	形式： ＜*ロード・モジュール名*＞[注] $ ＜*ラベル 名*＞ + ＜*オフセット値*＞ 例 1： test1.out$main + 0x10 例 2： subfunction + 0x20
ソース行番号	形式： ＜*ロード・モジュール名*＞[注] $ ＜*ファイル名*＞ # ＜*行番号*＞ 例 1： test1.out$main.c#40 例 2： main.c#100

注 ロード・モジュール名は，ロード・モジュール・ファイルが複数ダウンロードされている場合のみ表示します。

(3) 逆アセンブル・エリア
対象となるソース・テキスト行に続き，逆アセンブル結果行を次のように表示します。

図 A.27 逆アセンブル・エリアの表示内容（混合表示モードの場合）



ラベル行		アドレスにラベルが定義されている場合は，ラベル名を表示し，行全体を薄緑色で強調表示します。
カレント PC 行		カレント PC 位置（PC レジスタ値）のアドレスと対応する行を強調表示[注 1] します。
ブレークポイント設定行		ブレークポイントが設定されている行を強調表示[注 1] します。
ソース・テキスト行		コード・データに対応するソース・テキストを表示します[注 2]。
逆アセンブル結果行	オフセット値	アドレスにラベルが定義されていない場合は，一番近いラベルからのオフセット値を表示します[注 3]。
	命令コード	逆アセンブルの対象となったコードを 16 進数で表示します。
	命令	逆アセンブル結果として命令を表示します。ニモニックは青色で強調表示します。

注 1. 強調色は，オプション ダイアログにおける［全般 - フォントと色］カテゴリの設定に依存します。

注 2. ツールバーの ボタン（トグル）のクリック，またはコンテキスト・メニューの［混合表示モード］のチェックを外すことにより，ソース・テキストを非表示にすることができます（デフォルトでチェックされています）。

注 3. オフセット値はデフォルトでは表示されません。表示する場合は，ツール・バーの ボタンのクリック，またはコンテキスト・メニューの［ラベルのオフセットを表示］を選択してください。

このエリアは，次の機能を備えています。

(a) ライン・アセンブル
表示されている命令／コードは，編集（ライン・アセンブル）することができます。
操作方法についての詳細は，「2.6.4 ライン・アセンブルを行う」を参照してください。

(b) 命令レベルでのプログラム実行
このパネルにフォーカスがある状態でプログラムをステップ実行することにより，命令レベル単位で実行を制御することができます。
操作方法についての詳細は，「2.9.3 プログラムをステップ実行する」を参照してください。

(c) 各種イベントの設定
コンテキスト・メニューの［ブレークの設定］／［トレース設定］／［タイマ設定］を選択することにより，現在キャレットのあるアドレス／行に各種イベントを設定することができます。
イベントを設定することにより，対応するイベント・マークがイベント・エリアに表示されます。また，設定したイベントの詳細情報がイベント パネルに反映されます。
ただし，イベントの設定は，イベント・エリアにおいて，背景色が白色で表示されている行に対してのみ行うことができます。
イベントの設定方法についての詳細は次を参照してください。

- 「2.10.3 任意の場所で停止する（ブレーク・イベント）【E1】【E20】【EZ Emulator】」
- 「2.10.4 変数 /I/O レジスタへのアクセスで停止する」
- 「2.13.3 任意区間の実行履歴を収集する」
- 「2.13.4 条件を満たしたときのみの実行履歴を収集する」
- 「2.14.3 任意区間の実行時間を計測する」

備考　ブレークポイントの設定／削除は，イベント・エリアにおいても簡単に行うことができます（「(a) ブレークポイントの設定／削除」参照）。

(d) ウォッチ式の登録
表示されている C 言語変数 /CPU レジスタ /IO レジスタ／アセンブラ・シンボルをウォッチ式としてウォッチ パネルに登録することができます。
操作方法についての詳細は，「2.11.6.1 ウォッチ式を登録する」を参照してください。

(e) シンボル定義箇所への移動
シンボルを参照している命令にキャレットを移動した状態で，ツールバーの ボタンをクリック，またはコンテキスト・メニューの［シンボルへ移動］を選択することにより，キャレット位置のシンボルが定義されているアドレスにキャレット位置を移動します。
また，この操作に続き，ツールバーの ボタンをクリック，またはコンテキスト・メニューの［*アドレス*に戻る］を選択すると，キャレット移動前のシンボルを参照している命令にキャレット位置を戻します（*アドレス*はシンボルを参照している命令のアドレス値を表示）。

(f) ソース行／メモリ値へのジャンプ
コンテキスト・メニューの［ソースへジャンプ］を選択することにより，現在のキャレット位置のアドレスに対応するソース行にキャレットを移動した状態でエディタ パネルがオープンします（すでにオープンしている場合は，エディタ パネルにジャンプ）。
また，同様に［メモリへジャンプ］を選択することにより，現在のキャレット位置のアドレスに対応するメモリ値にキャレットを移動した状態でメモリ パネル（メモリ 1）がオープンします（すでにオープンしている場合は，メモリ パネル（メモリ 1）にジャンプ）。

(g) コード・カバレッジ測定結果の表示【シミュレータ】【E20【RX71M, RX64M グループ】】
カバレッジ機能を有効としている場合，プログラムの実行により取得したコード・カバレッジ測定結果を基に，カバレッジ測定対象領域に相当する行を強調表示します。
カバレッジ測定についての詳細は，「2.15 カバレッジの測定【シミュレータ】【E20【RX71M, RX64M グループ】】」を参照してください。

(h) 逆アセンブル・データの保存
［ファイル］メニュー→［名前を付けて逆アセンブル・データを保存 ...］を選択することにより，データ保存ダイアログをオープンし，このパネルの内容をテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存することができます。
逆アセンブル・データの保存方法についての詳細は，「2.6.2.5 逆アセンブル結果の表示内容を保存する」を参照してください。

(i) 表示の拡大／縮小
逆アセンブル パネルにフォーカスがある状態で，メイン・ウインドウのツールバーある拡大率のドロップダウン・リストを変更することにより，逆アセンブル パネルの表示を拡大／縮小することができます。
また，［Ctrl］キーとマウス・ホイールを組み合わせることでも変更可能です。

- ［Ctrl］キーを押下しながらマウス・ホイールを前方に動かすと，逆アセンブル パネルの表示を拡大し，表示が大きく見やすくなります（最大 300%）。
- ［Ctrl］キーを押下しながらマウス・ホイールを後方に動かすと，逆アセンブル パネルの表示を縮小し，表示が小さくなります（最小 25%）。

備考　オプション ダイアログの設定により，次の項目をカスタマイズすることができます。

- 表示フォント
- 予約語／コメントの色分け

[ツールバー]

	デバッグ・ツールから最新の情報を取得し，表示を更新します。
	逆アセンブル結果とソース・テキストとの対応を表示する，混合表示モードに設定します（デフォルト）。
	キャレット位置をカレント PC 値に追従するように指定します。
	選択しているシンボルの定義位置へキャレットを移動します。
	ボタンで移動する直前の位置（*アドレス*）へ移動します。
表示	逆アセンブル・エリアの表示形式を変更する次のボタンを表示します。
ラベルのオフセットを表示	ラベルのオフセット値を表示します。アドレスにラベルが定義されていない場合，一番近いラベルからのオフセット値を表示します。
アドレス値をシンボルで表示	アドレス値を“シンボル＋オフセット値”で表示します（デフォルト）。ただし，アドレス値にシンボルが定義されている場合は，シンボルのみを表示します。
	スクロール範囲を設定するためのスクロール範囲設定 ダイアログがオープンします。

[[ファイル] メニュー（逆アセンブル パネル専用部分）]

逆アセンブル パネル専用の［ファイル］メニューは次のとおりです（その他の項目は共通）。
ただし，プログラム実行中はすべて無効となります。

逆アセンブル・データを保存	逆アセンブルの内容を前回保存したテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存します（「(h) 逆アセンブル・データの保存」参照）。 なお，起動後に初めてこの項目を選択した場合は，［名前を付けて逆アセンブル・データを保存 ...］の選択と同等の動作となります。
名前を付けて逆アセンブル・データを保存 ...	逆アセンブルの内容を指定したテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存するために，データ保存 ダイアログをオープンします（「(h) 逆アセンブル・データの保存」参照）。
印刷 ...	このパネルの内容を印刷するために，印刷アドレス範囲設定 ダイアログをオープンします。

[[編集] メニュー（逆アセンブル パネル専用部分）]

逆アセンブル パネル専用の［編集］メニューは次のとおりです（その他の項目はすべて無効）。

コピー	行を選択している場合，選択している行の内容を文字列としてクリップ・ボードにコピーします。 編集モードの場合，選択している文字列をクリップ・ボードにコピーします。
名前の変更	キャレット位置の命令／命令コードを編集するために，編集モードに移行します（「2.6.4 ライン・アセンブルを行う」参照）。 ただし，プログラム実行中は無効となります。
検索 ...	検索・置換 ダイアログを［一括検索］タブが選択状態でオープンします。
置換 ...	検索・置換 ダイアログを［一括置換］タブが選択状態でオープンします。
移動 ...	指定したアドレスへキャレットを移動するため，指定位置へ移動 ダイアログをオープンします。

［コンテキスト・メニュー］

【逆アセンブル・エリア／アドレス・エリア】

ウォッチ 1 に登録	選択している文字列，またはキャレット位置の単語をウォッチ式としてウォッチパネル（ウォッチ 1）に登録します（単語の判断は現在のビルド・ツールに依存）。 ウォッチ式として登録する際は変数名として登録されるため，スコープにより表示されるシンボル名は変化します。
アクション・イベントの登録 ...	キャレット位置のアドレスにアクション・イベントを設定するため，アクション・イベント ダイアログをオープンします。
ここまで実行	カレント PC 値で示されるアドレスから，キャレット位置の行に対応するアドレスまでプログラムを実行します。 ただし，プログラム実行中，または［ビルド & デバッグ・ツールへダウンロード］実行中は無効となります。
PC をここに設定	カレント PC 値を現在キャレットのある行のアドレスに変更します。 ただし，プログラム実行中，または［ビルド & デバッグ・ツールへダウンロード］実行中は無効となります。
移動 ...	指定したアドレスへキャレットを移動するため，指定位置へ移動 ダイアログをオープンします。
シンボルへ移動	選択しているシンボルの定義位置へキャレットを移動します。
*アドレス*へ戻る	［シンボルへ移動］で移動する直前の位置（*アドレス*）へ移動します。 ただし，*アドレス*にシンボル名が表示されていない場合は無効となります。

ブレークの設定	ブレーク関連のイベントを設定するために，次のカスケード・メニューを表示します。 なお，イベントは，イベントの設定が可能な行のみ設定することができます（「(1) イベント・エリア」参照）。
ハード・ブレークの設定	キャレット位置のアドレスにブレークポイント（ハードウエア・ブレーク・イベント）を設定します（「2.10.2 任意の場所で停止する（ブレークポイント）」参照）。
ソフト・ブレークの設定 【E1】【E20】【EZ Emulator】	キャレット位置のアドレスにブレークポイント（ソフトウエア・ブレーク・イベント）を設定します（「2.10.2 任意の場所で停止する（ブレークポイント）」参照）。
読み込みブレークを設定 【シミュレータ】	キャレット位置，または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）/I/O レジスタに，リード・アクセスのブレーク・イベントを設定します（「2.10.4.1 変数 /I/O レジスタへのブレーク・イベント（アクセス系）を設定する」参照）。
書き込みブレークを設定 【シミュレータ】	キャレット位置，または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）/I/O レジスタに，ライト・アクセスのブレーク・イベントを設定します（「2.10.4.1 変数 /I/O レジスタへのブレーク・イベント（アクセス系）を設定する」参照）。
読み書きブレークを設定 【シミュレータ】	キャレット位置，または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）/I/O レジスタに，リード／ライト・アクセスのブレーク・イベントを設定します（「2.10.4.1 変数 /I/O レジスタへのブレーク・イベント（アクセス系）を設定する」参照）。
組み合わせブレークを設定 【E1】【E20】【EZ Emulator】	キャレット位置のアドレス／キャレット位置，または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）/I/O レジスタでのブレークを，組み合わせブレークイベントの条件として設定します（「2.10.3.1 ブレーク・イベント（実行系）を設定する」参照）。
読み込み組み合わせブレークを設定 【E1】【E20】【EZ Emulator】	組み合わせブレークの条件として，キャレット位置，または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）/I/O レジスタのリード・アクセスによるブレーク・イベントを設定します（「2.10.4.1 変数 /I/O レジスタへのブレーク・イベント（アクセス系）を設定する」参照）。
書き込み組み合わせブレークを設定 【E1】【E20】【EZ Emulator】	組み合わせブレークの条件として，キャレット位置，または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）/I/O レジスタのライト・アクセスによるブレーク・イベントを設定します（「2.10.4.1 変数 /I/O レジスタへのブレーク・イベント（アクセス系）を設定する」参照）。
読み書き組み合わせブレークを設定 【E1】【E20】【EZ Emulator】	組み合わせブレークの条件として，キャレット位置，または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）/I/O レジスタのリード／ライト・アクセスによるブレーク・イベントを設定します（「2.10.4.1 変数 /I/O レジスタへのブレーク・イベント（アクセス系）を設定する」参照）。
ブレーク動作の設定	プロパティ パネルをオープンし，ブレーク機能の設定を行います。

トレース設定	トレース関連のイベントを設定するために，次のカスケード・メニューを表示します。 なお，イベントは，イベントの設定が可能な行のみ設定することができます（「(1) イベント・エリア」参照）。
トレース開始の設定	キャレット位置のアドレスの命令が実行された際に，プログラムの実行履歴を示すトレース・データの収集を開始するトレース開始イベントを設定します（「2.13.3.1 トレース開始イベント／トレース終了イベントを設定する」参照）。 【シミュレータ】 プロパティ パネル上の［トレース］カテゴリ内［トレース機能を使用する］プロパティの設定を自動的に［はい］にします。
トレース終了の設定	キャレット位置のアドレスの命令が実行された際に，プログラムの実行履歴を示すトレース・データの収集を終了するトレース終了イベントを設定します（「2.13.3.1 トレース開始イベント／トレース終了イベントを設定する」参照）。 【シミュレータ】 プロパティ パネル上の［トレース］カテゴリ内［トレース機能を使用する］プロパティの設定を自動的に［はい］にします。
値をトレースに記録（読み込み時）	キャレット位置，または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）/I/O レジスタにリード・アクセスした際に，その値をトレース・メモリに記録するポイント・トレース・イベントを設定します（「(1) 変数 /I/O レジスタへのアクセスが発生したとき」参照）。
値をトレースに記録（書き込み時）	キャレット位置，または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）/I/O レジスタにライト・アクセスした際に，その値をトレース・メモリに記録するポイント・トレース・イベントを設定します（「(1) 変数 /I/O レジスタへのアクセスが発生したとき」参照）。
値をトレースに記録（読み書き時）	キャレット位置，または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）/I/O レジスタにリード／ライト・アクセスした際に，その値をトレース・メモリに記録するポイント・トレース・イベントを設定します（「(1) 変数 /I/O レジスタへのアクセスが発生したとき」参照）。
読み書き時にトレース開始の設定	キャレット位置，または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）/I/O レジスタにリード／ライト・アクセスした際に，トレースの記録を開始するトレース・イベントを設定します（「2.13.3.1 トレース開始イベント／トレース終了イベントを設定する」参照）。
読み書き時にトレース終了の設定	キャレット位置，または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）/I/O レジスタにリード／ライト・アクセスした際に，トレースの記録を終了するトレース・イベントを設定します（「2.13.3.1 トレース開始イベント／トレース終了イベントを設定する」参照）。
トレース結果の表示	トレース パネルをオープンし，取得したトレース・データを表示します。
トレース動作の設定	プロパティ パネルをオープンし，トレース機能の設定を行います。 ただし，トレーサ動作中は無効となります。

タイマ設定[注1]	タイマ関連のイベントを設定するために，次のカスケード・メニューを表示します（「2.14.3 任意区間の実行時間を計測する」参照）。 なお，イベントは，イベントの設定が可能な行のみ設定することができます（「(1) イベント・エリア」参照）。
実行時にタイマ開始	キャレット位置のアドレスの命令が実行された際に，プログラムの実行時間の計測を開始するタイマ開始イベントを設定します（「(1) タイマ開始イベントの設定方法」参照）。
実行時にタイマ終了	キャレット位置のアドレスの命令が実行された際に，プログラムの実行時間の計測を終了するタイマ終了イベントを設定します（「(2) タイマ終了イベントの設定方法」参照）。
読み書き時にタイマ開始	キャレット位置，または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）/I/O レジスタにリード／ライト・アクセスした際に，プログラムの実行時間の計測を開始するタイマ開始イベントを設定します（「(1) タイマ開始イベントの設定方法」参照）。
タイマ <N> に設定 【E1】【E20】【EZ Emulator】	タイマ開始イベントを設定するチャネル[注2]を指定します。
読み書き時にタイマ終了	キャレット位置，または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）/I/O レジスタにリード／ライト・アクセスした際に，プログラムの実行時間の計測を終了するタイマ終了イベントを設定します（「(2) タイマ終了イベントの設定方法」参照）。
タイマ <N> に設定 【E1】【E20】【EZ Emulator】	タイマ終了イベントを設定するチャネル[注2]を指定します。
タイマ結果の表示	イベント パネルをオープンし，タイマ関連のイベントのみ表示します。
カバレッジ情報をクリア 【シミュレータ】【E20【RX71M, RX64M グループ】】	現在デバッグ・ツールが保持しているコード・カバレッジ測定結果をすべてクリアします。 ただし，使用するデバッグ・ツールがカバレッジ機能をサポートしていない場合，この項目は表示されません。
命令の編集	キャレット位置の行の命令を編集するために，編集モードに移行します（「2.6.4 ライン・アセンブルを行う」参照）。 ただし，プログラム実行中は無効となります。
コードの編集	キャレット位置の行の命令コードを編集するために，編集モードに移行します（「2.6.4 ライン・アセンブルを行う」参照）。 ただし，プログラム実行中は無効となります。
表示	逆アセンブル・エリアの表示内容を設定するために，次のカスケード・メニューを表示します。
ラベルのオフセットを表示	ラベルのオフセット値を表示します。アドレスにラベルが定義されていない場合，一番近いラベルからのオフセット値を表示します。
アドレス値をシンボルで表示	アドレス値を“シンボル＋オフセット値”で表示します（デフォルト）。 ただし，アドレス値にシンボルが定義されている場合は，シンボルのみを表示します。
スクロール範囲を設定 ...	スクロール範囲を設定するためのスクロール範囲設定 ダイアログがオープンします。
混合表示	逆アセンブル結果とソース・テキストとの対応を表示する，混合表示モードに設定します（デフォルト）。
ソースへジャンプ	キャレット位置のアドレスに対応するソース行にキャレットを移動した状態で，エディタ パネルがオープンします。
メモリへジャンプ	キャレット位置のアドレスに対応するメモリ値にキャレットを移動した状態で，メモリ パネル（メモリ 1）がオープンします。

注 1. 【E1】【E20】【EZ Emulator】
RX100 シリーズでは，タイマ機能は未サポートとなります。

注 2. 指定可能なチャネル数については，以下のとおり RX600 シリーズと RX200 シリーズで異なります。
RX600 シリーズ：2（32bit×2）または 1（64bit×1）
RX200 シリーズ：1（24bit×1）

【イベント・エリア】（【E1】【E20】【EZ Emulator】）

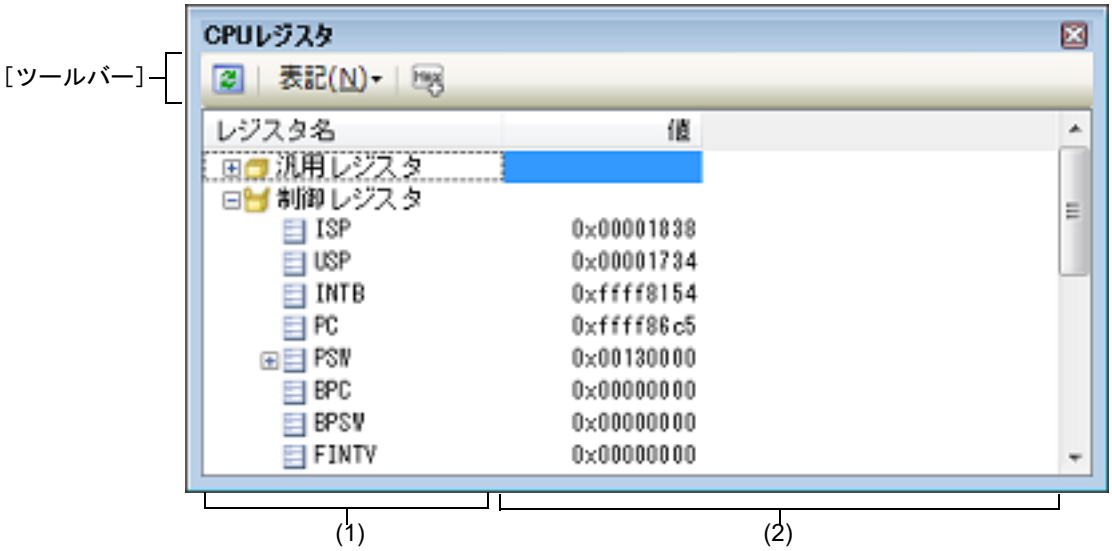
ハードウエア・ブレークを優先する	マウスのワンクリック操作で設定できるブレークの種類をハードウエア・ブレークポイントとします（プロパティ パネル上の［ブレーク］【E1】【E20】【EZ Emulator】カテゴリ内［優先的に使用するブレークポイントの種類］プロパティの設定に反映されます）。
ソフトウエア・ブレークを優先する	マウスのワンクリック操作で設定できるブレークの種類をソフトウエア・ブレークポイントとします（プロパティ パネル上の［ブレーク］【E1】【E20】【EZ Emulator】カテゴリ内［優先的に使用するブレークポイントの種類］プロパティの設定に反映されます）。

CPU レジスタ パネル

CPU レジスタ（汎用レジスタ／制御レジスタ）の内容の表示，および値の変更を行います（「2.11.2 CPU レジスタを表示／変更する」参照）。
なお，このパネルは，デバッグ・ツールと接続時のみオープンすることができます。

備考　パネル上の各エリアの区切り線をダブルクリックすることにより，該当エリアの内容を省略することなく表示可能な最小幅に変更することができます。

図 A.28　CPU レジスタ パネル



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー（CPU レジスタ パネル専用部分）]
- [[編集] メニュー（CPU レジスタ パネル専用部分）]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [CPU レジスタ] を選択

[各エリアの説明]

(1)　[レジスタ名] エリア
レジスタの種別をカテゴリ（フォルダ）として分類し，各レジスタ名を一覧表示します。
表示される各アイコンの意味は次のとおりです。
なお，カテゴリ名／レジスタ名を編集／削除することはできません。

	このカテゴリに属するレジスタ名を表示している状態を示します。アイコンをダブルクリック，または “-” マークをクリックすると，カテゴリを閉じてレジスタ名を非表示にします。
	このカテゴリに属するレジスタ名が非表示の状態を示します。アイコンをダブルクリック，または “+” マークをクリックすると，カテゴリを開いてレジスタ名を表示します。
	レジスタ名を示します。アイコンをダブルクリック，または “+” ／ “-” マークをクリックすると，下階層のレジスタ名（レジスタの部分を表す名称）を表示／非表示します。

	レジスタ名（レジスタの部分を表す名称）を示します。

表示されるカテゴリ名／レジスタ名は次のとおりです（各レジスタ名の先頭の “+” マークの数は，表示される際の階層の深さを示します）。

表 A.2 ［汎用レジスタ］カテゴリとレジスタ名【RX】

レジスタ名（別名）	ビット幅	レジスタ名（別名）	ビット幅
+ R0	32	+ R8	32
+ R1	32	+ R9	32
+ R2	32	+ R10	32
+ R3	32	+ R11	32
+ R4	32	+ R12	32
+ R5	32	+ R13	32
+ R6	32	+ R14	32
+ R7	32	+ R15	32

表 A.3 ［制御レジスタ］カテゴリとレジスタ名【RX】

レジスタ名	ビット幅	レジスタ名	ビット幅
+ ISP	32	+ FPSW 注 2	32
+ USP	32	++ FS	1
+ INTB	32	++ FX	1
+ PC	32	++ FU	1
+ PSW	32	++ FZ	1
++ IPL	4 注 1	++ FO	1
++ PM	1	++ FV	1
++ U	1	++ EX	1
++ I	1	++ EU	1
++ O	1	++ EZ	1
++ S	1	++ EO	1
++ Z	1	++ EV	1
++ C	1	++ DN	1
+ BPC	32	++ CE	1
+ BPSW	32	++ CX	1
+ FINTV	32	++ CU	1

レジスタ名	ビット幅	レジスタ名	ビット幅
		++ CZ	1
		++ CO	1
		++ CY	1
		++ RM	2
		+ ACC 注 3	64
		+ ACC0 注 4	72
		+ ACC1 注 4	72
		+ EXTB 注 4	32

注 1. RX610 グループでは，3 ビットとなります。

注 2. FPU を搭載していないマイクロコントローラでは，FPSW レジスタをサポートしていません。

注 3. RX71M, RX64M グループでは，ACC レジスタをサポートしていません。

注 4. ACC0，ACC1，EXTB レジスタは，RX71M, RX64M グループのみでサポートしています。

このエリアは，次の機能を備えています。

(a) ウォッチ式の登録
CPU レジスタ／カテゴリをウォッチ式としてウォッチ パネルに登録することができます。
操作方法についての詳細は，「2.11.6.1 ウォッチ式を登録する」を参照してください。

備考 1. カテゴリを対象としてウォッチ式の登録を行った場合，そのカテゴリに属するすべての CPU レジスタがウォッチ式として登録されます。

備考 2. 登録したウォッチ式には，自動的にスコープ指定が付与されます。

注意 【RX71M, RX64M グループ】
ACC0，ACC1 レジスタをウォッチ式としてウォッチ パネル**に登録して，値の表示形式として** [Float] または [Double] が選択された場合は、[16 進数] が選択された場合と同じ値表示になります。

(2) ［値］エリア
各 CPU レジスタの値を表示／変更します。
表示進数は，ツールバーのボタン，またはコンテキスト・メニューより選択することができます。また，常に 16 進数表示を併記する表示形式を選択することもできます。
CPU レジスタの値として表示されるマークや色の意味は次のとおりです（表示の際の文字色／背景色はオプション ダイアログにおける［全般 - フォントと色］カテゴリの設定に依存）。

表示例（デフォルト）			説明
0x0	文字色	青色	ユーザにより，値が変更されている CPU レジスタの値（[Enter］キーによりターゲット・メモリに書き込まれます）
	背景色	標準色	
0x0	文字色	茶色	プログラムの実行により，値が変化した CPU レジスタの値 プログラムを再実行させることにより，強調色をリセットします。
	背景色	クリーム	

このエリアは，次の機能を備えています。

(a) CPU レジスタ値の変更
CPU レジスタ値の変更は，対象 CPU レジスタ値を選択したのち，再度クリックし，キーボードからの直接入力により行います（[Esc］キーの押下で編集モードをキャンセルします）。
CPU レジスタ値を編集したのち，［Enter］キーの押下，または編集領域以外へのフォーカスの移動により，デバッグ・ツールのレジスタに書き込まれます。

(b) CPU レジスタ値の保存

［ファイル］メニュー→［名前を付けて CPU レジスタ・データを保存 ...］を選択することにより，名前を付けて保存 ダイアログをオープンし，このパネルのすべての内容をテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存することができます。
CPU レジスタ値の保存方法についての詳細は，「2.11.2.3 CPU レジスタの表示内容を保存する」を参照してください。

(c) 表示の拡大／縮小
CPU レジスタ パネルにフォーカスがある状態で，メイン・ウインドウのツールバー上にある拡大率のドロップダウン・リストを変更することにより，CPU レジスタ パネルの表示を拡大／縮小することができます。
また，［Ctrl］キーとマウス・ホイールを組み合わせることでも変更可能です。

- ［Ctrl］キーを押下しながらマウス・ホイールを前方に動かすと，CPU レジスタ パネルの表示を拡大し，表示が大きく見やすくなります（最大 300%）。
- ［Ctrl］キーを押下しながらマウス・ホイールを後方に動かすと，CPU レジスタ パネルの表示を縮小し，表示が小さくなります（最小 50%）。

なお，拡大率を変更した状態のパネルをクローズした場合，拡大率は保持されます（再度パネルをオープンする際に，変更した拡大率でオープンします）。

［ツールバー］

	デバッグ・ツールから最新の情報を取得し，表示を更新します。 ただし，プログラム実行中は無効となります。
表記	値の表示形式を変更する次のボタンを表示します。
自動	選択している項目（下位項目を含む）の値を規定値で表示します（デフォルト）。
16 進数	選択している項目（下位項目を含む）の値を 16 進数で表示します。
符号付き 10 進数	選択している項目（下位項目を含む）の値を符号付き 10 進数で表示します。
符号なし 10 進数	選択している項目（下位項目を含む）の値を符号なし 10 進数で表示します。
8 進数	選択している項目（下位項目を含む）の値を 8 進数で表示します。
2 進数	選択している項目（下位項目を含む）の値を 2 進数で表示します。
ASCII	選択している項目（下位項目を含む）の文字列を ASCII コードで表示します。対象が 2 バイト以上ある場合は，1 バイトずつの文字を並べて表示します。
Float	選択している項目を Float で表示します。 ただし，4 バイト・データ以外の場合は，規定値で表示します。
Double	選択している項目を Double で表示します。 ただし，8 バイト・データ以外の場合は，規定値で表示します。
	値表示の末尾に，その値の 16 進数表記を“（）”で囲んで併記します。

［［ファイル］メニュー（CPU レジスタ パネル専用部分）］

CPU レジスタ パネル専用の［ファイル］メニューは次のとおりです（その他の項目は共通）。
ただし，プログラム実行中はすべて無効となります。

CPU レジスタ・データを保存	このパネルの内容を前回保存したテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存します（「(b) CPU レジスタ値の保存」参照）。 なお，起動後に初めてこの項目を選択した場合は，［名前を付けて CPU レジスタ・データを保存 ...］の選択と同等の動作となります。
名前を付けて CPU レジスタ・データを保存 ...	このパネルの内容を指定したテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存するために，名前を付けて保存 ダイアログをオープンします（「(b) CPU レジスタ値の保存」参照）。

[[編集] メニュー（CPU レジスタ パネル専用部分)]

CPU レジスタ パネル専用の［編集］メニューは次のとおりです（その他の項目はすべて無効)。

切り取り	選択範囲の文字列を切り取り，クリップ・ボードにコピーします。 ただし，文字列を編集中の場合のみ有効となります。
コピー	編集中の場合，選択している文字列をクリップ・ボードにコピーします。 行を選択している場合，レジスタ／カテゴリをクリップ・ボードにコピーします。 なお，コピーした項目は，ウォッチ パネルに貼り付け可能です。
貼り付け	クリップ・ボードにコピーされている文字列をキャレット位置に貼り付けます。 ただし，文字列を編集中の場合のみ有効となります。
すべて選択	すべての項目を選択状態にします。
検索 ...	検索・置換 ダイアログを［一括検索］タブが選択状態でオープンします。
置換 ...	検索・置換 ダイアログを［一括置換］タブが選択状態でオープンします。

[コンテキスト・メニュー]

ウォッチ 1 に登録	選択しているレジスタ名／カテゴリをウォッチ パネル（ウォッチ 1）に登録します。
コピー	編集中の場合，選択している文字列をクリップ・ボードにコピーします。 行選択している場合，レジスタ項目／カテゴリをクリップ・ボードにコピーします。 なお，コピーした項目は，ウォッチ パネルに貼り付け可能です。
表記	表示形式を指定するため，次のカスケード・メニューを表示します。
自動	選択している項目（下位項目を含む）の値を規定値で表示します（デフォルト)。
16 進数	選択している項目（下位項目を含む）の値を 16 進数で表示します。
符号付き 10 進数	選択している項目（下位項目を含む）の値を符号付き 10 進数で表示します。
符号なし 10 進数	選択している項目（下位項目を含む）の値を符号なし 10 進数で表示します。
8 進数	選択している項目（下位項目を含む）の値を 8 進数で表示します。
2 進数	選択している項目（下位項目を含む）の値を 2 進数で表示します。
ASCII	選択している項目（下位項目を含む）の文字列を ASCII コードで表示します。 対象が 2 バイト以上ある場合は，1 バイトずつの文字を並べて表示します。
Float	選択している項目を Float で表示します。 ただし，4 バイト・データ以外の場合は，規定値で表示します。
Double	選択している項目を Double で表示します。 ただし，8 バイト・データ以外の場合は，規定値で表示します。
16 進数値を併記	値表示の末尾に，その値の 16 進数表記を“（）”で囲んで併記します。

注意 【RX71M, RX64M グループ】
ACC0，ACC1 レジスタに対して [Float] または [Double] が選択された場合は，[16 進数] が選択された場合と同じ値表示になります。

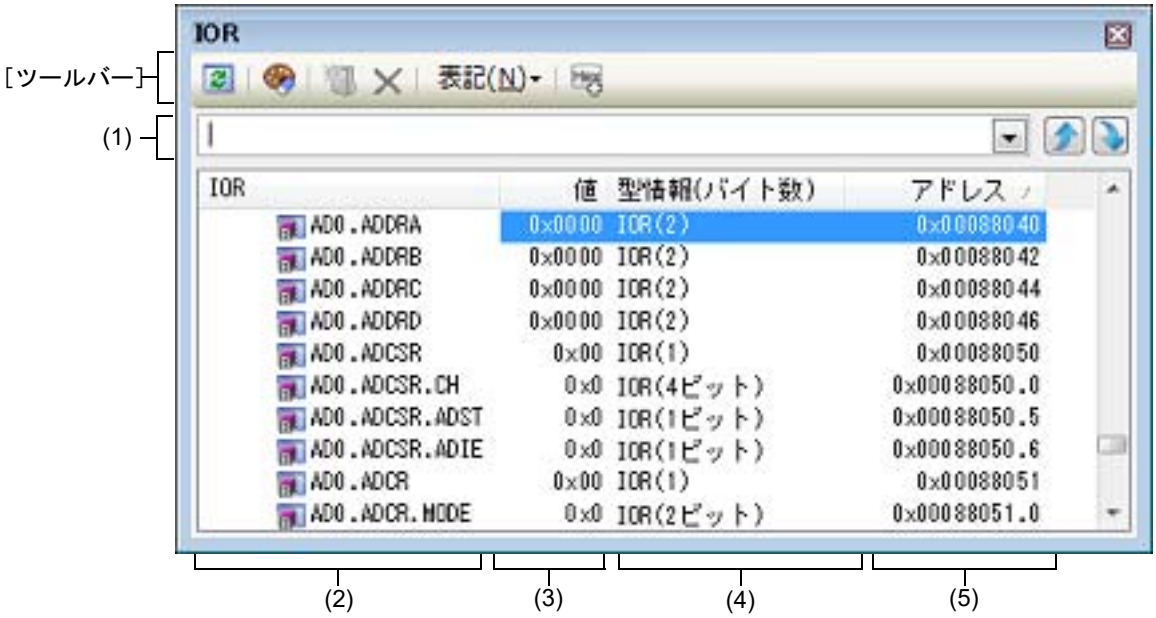
IOR パネル

I/O レジスタの内容の表示，および値の変更を行います（「2.11.3 I/O レジスタを表示／変更する」参照）。
なお，このパネルは，デバッグ・ツールと接続時のみオープンすることができます。

注意 読み込み動作によってマイクロコントローラが動作してしまう I/O レジスタは，読み込み保護対象となるため，値の読み込みは行いません（[値] に "?" を表示）。
読み込み保護対象の I/O レジスタ の内容を取得したい場合は，コンテキスト・メニューの［値を強制読み込み］を選択してください。

備考 パネル上の各エリアの区切り線をダブルクリックすることにより，該当エリアの内容を省略することなく表示可能な最小幅に変更することができます。

図 A.29 IOR パネル



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］
- ［ツールバー］
- ［［ファイル］メニュー（IOR パネル専用部分）］
- ［［編集］メニュー（IOR パネル専用部分）］
- ［コンテキスト・メニュー］

［オープン方法］

- ［表示］メニュー→［IOR］を選択

［各エリアの説明］

(1) 検索エリア
I/O レジスタ名の検索を行います。

	検索対象の文字列を指定します（大文字／小文字不問）。 キーボードより文字列を直接入力するか（最大指定文字数：512 文字），ドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。

	テキスト・ボックスで指定している文字列を含む I/O レジスタ名を上方向に検索し，検索結果を選択状態にします。
	テキスト・ボックスで指定している文字列を含む I/O レジスタ名を下方向に検索し，検索結果を選択状態にします。

備考 1. カテゴリ（フォルダ）により分類されて非表示の状態の I/O レジスタ名も検索します（展開して選択状態となります）。

備考 2. 検索対象の文字列入力後，[Enter] キーを押下することにより，ボタンのクリックと同等の動作を行い，[Shift]+[Enter] キーを押下することにより，ボタンのクリックと同等の動作を行います。

(2) ［IOR］エリア

I/O レジスタの種別をカテゴリ（フォルダ）として分類し，各 I/O レジスタ名を一覧表示します。
表示される各アイコンの意味は次のとおりです。

	このカテゴリに属する I/O レジスタ名を表示している状態を示します。アイコンをダブルクリック，または “-” マークをクリックすると，カテゴリを閉じ I/O レジスタ名を非表示にします。 なお，カテゴリはデフォルトでは存在しません。必要な場合は，カテゴリを新規作成したのち，ツリーの編集を行ってください。
	I/O レジスタ名が非表示の状態を示します。アイコンをダブルクリック，または “+” マークをクリックすると，カテゴリを開き I/O レジスタ名を表示します。 なお，カテゴリはデフォルトでは存在しません。必要な場合は，カテゴリを新規作成したのち，ツリーの編集を行ってください。
	I/O レジスタ名を示します。

備考 このエリアのヘッダ部をクリックすることにより，カテゴリ名を文字コード順でソートします（カテゴリ内 I/O レジスタ名も同様にソートします）。

このエリアは，次の機能を備えています。

(a) ツリーの編集

各 I/O レジスタを任意のカテゴリ（フォルダ）で分類し，ツリー形式を編集することができます。
カテゴリを新規に作成する場合は，作成したい I/O レジスタ名にキャレットを移動したのち，ツールバーのボタンのクリック，またはコンテキスト・メニューの［カテゴリを作成］を選択し，任意にカテゴリ名称を入力することにより行います（最大指定文字数：1024 文字）。
なお，カテゴリを削除する場合は，削除したいカテゴリを選択したのち，ツールバーのボタンのクリック，またはコンテキスト・メニューの［削除］を選択します。ただし，削除できるカテゴリは，空のカテゴリのみです。
また，カテゴリ名を編集する場合は，編集したいカテゴリ名を選択したのち，次のいずれかの操作により行います。

- 再度クリック後，キーボードよりカテゴリ名を直接編集
- ［編集］メニュー→［名前の変更］を選択後，キーボードよりカテゴリ名を直接編集
- ［F2］キーを押下後，キーボードよりカテゴリ名を直接編集

カテゴリを作成したのち，I/O レジスタ名をカテゴリ内に直接ドラッグ・アンド・ドロップすることにより，各 I/O レジスタをカテゴリで分類したツリー形式で表示します。
同様に，カテゴリと I/O レジスタ名の表示の順番（上下位置）も，ドラッグ・アンド・ドロップ操作により自由に変更することができます。

注意 1. カテゴリ内にカテゴリを作成することはできません。

注意 2. I/O レジスタの追加／削除はできません。

(b) ウォッチ式の登録

I/O レジスタ／カテゴリをウォッチ式としてウォッチ パネルに登録することができます。
操作方法についての詳細は，「2.11.6.1 ウォッチ式を登録する」を参照してください。

備考 1. カテゴリを対象としてウォッチ式の登録を行った場合，そのカテゴリに属するすべての I/O レジスタがウォッチ式として登録されます。

備考 2. 登録したウォッチ式には，自動的にスコープ指定が付与されます。

(c) 表示の拡大／縮小
IOR パネルにフォーカスがある状態で，メイン・ウインドウのツールバー上にある拡大率のドロップダウン・リストを変更することにより，IOR パネルの表示を拡大／縮小することができます。
また，［Ctrl］キーとマウス・ホイールを組み合わせることでも変更可能です。

- ［Ctrl］キーを押下しながらマウス・ホイールを前方に動かすと，IOR パネルの表示を拡大し，表示が大きく見やすくなります（最大 300%）。
- ［Ctrl］キーを押下しながらマウス・ホイールを後方に動かすと，IOR パネルの表示を縮小し，表示が小さくなります（最小 50%）。

なお，拡大率を変更した状態のパネルをクローズした場合，拡大率は保持されます（再度パネルをオープンする際に，変更した拡大率でオープンします）。

(3) ［値］エリア
I/O レジスタの値を表示／変更します。
表示進数は，ツールバーのボタン，またはコンテキスト・メニューより選択することができます。また，常に 16 進数表示を併記する表示形式を選択することもできます。
I/O レジスタの値として表示されるマークや色の意味は次のとおりです（表示の際の文字色／背景色はオプション ダイアログにおける［全般 - フォントと色］カテゴリの設定に依存）。

表示例（デフォルト）			説明
0x0	文字色	青色	ユーザにより，値が変更されている I/O レジスタの値（［Enter］キーによりターゲット・メモリに書き込まれます）
	背景色	標準色	
0x0	文字色	茶色	プログラムの実行により，値が変化した I/O レジスタの値 ツールバーの ボタン，またはコンテキスト・メニューの［表示色をリセット］を選択することにより，強調表示をリセットします。
	背景色	クリーム	
?	文字色	グレー	読み込み保護対象の I/O レジスタ注の値
	背景色	標準色	

注 読み込み動作によってマイクロコントローラが動作してしまう I/O レジスタを示します。
読み込み保護対象の I/O レジスタの内容を取得する場合は，コンテキスト・メニューの［値を強制読み込み］を選択することにより行ってください。

注意 1 バイト /2 バイト I/O レジスタと，1 バイト /2 バイト I/O レジスタに割り付けられている 1 ビット I/O レジスタでは，値を取得するタイミングが異なります。このため，同一の I/O レジスタの値を表示していても値が異なる場合があります。

備考 このエリアのヘッダ部をクリックすることにより，値を数値の昇順でソートします。

このエリアは，次の機能を備えています。

(a) I/O レジスタ値の変更
I/O レジスタの値の変更は，対象 I/O レジスタ値を選択したのち，再度クリックし，キーボードからの直接入力により行います（［Esc］キーの押下で編集モードをキャンセルします）。
I/O レジスタ値を編集したのち，［Enter］キーの押下，または編集領域以外へのフォーカスの移動により，デバッグ・ツールのターゲット・メモリに書き込まれます。
I/O レジスタ値の変更方法についての詳細は，「2.11.3.4 I/O レジスタの内容を変更する」を参照してください。

(b) I/O レジスタ値の保存
［ファイル］メニュー→［名前を付けて *IOR* データを保存 ...］を選択することにより，名前を付けて保存 ダイアログをオープンし，I/O レジスタのすべての内容をテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存することができます。
I/O レジスタ値の保存方法についての詳細は，「2.11.3.6 I/O レジスタの表示内容を保存する」を参照してください。

(4) ［型情報（バイト数）］エリア
各 I/O レジスタの型情報を次の形式で表示します。

- ＜*I/O レジスタの種類*＞ [＜*アクセス属性*＞ ＜*すべてのアクセス可能サイズ*＞] (＜*サイズ*＞)

アクセス属性	アクセス属性として，次のいずれかをを表示します。	
	R	リードのみ可能
	W	ライトのみ可能
	R/W	リード／ライト可能
すべてのアクセス可能サイズ	アクセス可能なサイズをビット単位で小さい順に “,” で区切り列挙します。	
サイズ	I/O レジスタのサイズを表示します。 バイト単位で表示可能な場合はバイト単位で，ビット単位でのみ表示可能な場合はビット単位で単位を付与して表示します。	

例 1. 「IOR [R/W 1.8] (1 バイト)」の場合
リード／ライト可能，1 ビット・アクセス /8 ビット・アクセス可能，サイズが 1 バイトの I/O レジスタ

例 2. 「IOR [R/W 1] (1 ビット)」の場合
リード／ライト可能，1 ビット・アクセス可能，サイズが 1 ビットの I/O レジスタ

備考 このエリアのヘッダ部をクリックすることにより，型情報を文字コード順でソートします。

(5) ［アドレス］エリア
各 I/O レジスタがマッピングされているアドレスを表示します（16 進数表記固定）。
ただし，ビット・レジスタの場合は，次の例のようにビット・オフセット値を付与して表示します。

例 1. 「0xFF40」の場合
アドレス “0xFF40” に割り当てられている

例 2. 「0xFF40.4」の場合
アドレス “0xFF40” のビット 4 に割り当てられている（ビット・レジスタ）

備考 このエリアのヘッダ部をクリックすることにより，アドレスを数値の昇順でソートします。

［ツールバー］

	デバッグ・ツールから最新の情報を取得し，表示を更新します。 読み込み保護対象の I/O レジスタの再読み込みは行いません。 ただし，プログラム実行中は無効となります。
	選択している I/O レジスタに対して，プログラム実行により値が変化したことを示す強調表示をリセットします。 ただし，プログラム実行中は無効となります。
	新規カテゴリ（フォルダ）を追加します。テキスト・ボックスに直接カテゴリ名を入力します。 なお，新規に作成できるカテゴリの数に制限はありませが，カテゴリ内にカテゴリを作成することはできません。 ただし，プログラム実行中は無効となります。
	選択している範囲の文字列を削除します。 空のカテゴリが選択状態の場合は，そのカテゴリを削除します（I/O レジスタの削除不可）。
表記	値の表示形式を変更する次のボタンを表示します。

16 進数	選択している項目の値を 16 進数で表示します（デフォルト）。
符号付き 10 進数	選択している項目の値を符号付き 10 進数で表示します。
符号なし 10 進数	選択している項目の値を符号なし 10 進数で表示します。
8 進数	選択している項目の値を 8 進数で表示します。
2 進数	選択している項目の値を 2 進数で表示します。
ASCII	選択している項目の値を ASCII コードで表示します。
	選択している項目の値表示の末尾に，その値の 16 進数表記を“（）”で囲んで併記します。

[[ファイル］メニュー（IOR パネル専用部分）]

IOR パネル専用の［ファイル］メニューは次のとおりです（その他の項目は共通）。
ただし，プログラム実行中はすべて無効となります。

IOR データを保存	このパネルの内容を前回保存したテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存します（「(b) I/O レジスタ値の保存」参照）。 なお，起動後に初めてこの項目を選択した場合は，［名前を付けて IOR データを保存 ...］の選択と同等の動作となります。
名前を付けて IOR データを保存 ...	このパネルの内容を指定したテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存するために，名前を付けて保存 ダイアログをオープンします（「(b) I/O レジスタ値の保存」参照）。

[[編集］メニュー（IOR パネル専用部分）]

IOR パネル専用の［編集］メニューは次のとおりです（その他の項目はすべて無効）。

切り取り	選択している範囲の文字列を切り取ってクリップ・ボードに移動します（I/O レジスタ／カテゴリの切り取り不可）。
コピー	選択している範囲の文字列をクリップ・ボードにコピーします。 I/O レジスタ／カテゴリが選択状態の場合は，その項目をコピーします。 なお，コピーした項目は，ウォッチ パネルに貼り付け可能です。
貼り付け	テキストが編集状態の場合，クリップ・ボードの内容をキャレット位置に挿入します（I/O レジスタ／カテゴリの貼り付け不可）。
削除	選択している範囲の文字列を削除します。 空のカテゴリが選択状態の場合は，その項目を削除します（I/O レジスタの削除不可）。
すべて選択	テキストが編集状態の場合，すべての文字列を選択します。 テキストが編集状態以外の場合，すべての I/O レジスタ／カテゴリを選択状態にします。
名前の変更	選択しているカテゴリの名称を編集します。
検索 ...	検索エリアのテキスト・ボックスにフォーカスを移動します。
移動 ...	指定した I/O レジスタへキャレットを移動するため，指定位置へ移動 ダイアログをオープンします。

［コンテキスト・メニュー］

ウォッチ 1 に登録	選択している I/O レジスタ／カテゴリをウォッチ パネル（ウォッチ 1）に登録します。
最新の情報に更新	デバッグ・ツールから最新の情報を取得し，表示を更新します。 読み込み保護対象の I/O レジスタの再読み込みは行いません。 ただし，プログラム実行中は無効となります。
値を強制読み込み	読み込み保護対象の I/O レジスタの値を 1 回強制的に読み込みます。
移動 ...	指定位置へ移動 ダイアログをオープンします。
カテゴリを作成	新規カテゴリ（フォルダ）を追加します。テキスト・ボックスに直接カテゴリ名を入力します。 なお，新規に作成できるカテゴリの数に制限はありませが，カテゴリ内にカテゴリを作成することはできません。 ただし，プログラム実行中は無効となります。
コピー	選択している範囲の文字列をクリップ・ボードにコピーします。 I/O レジスタ／カテゴリが選択状態の場合は，その項目をコピーします。 なお，コピーした項目は，ウォッチ パネルに貼り付け可能です。
削除	選択している範囲の文字列を削除します。 空のカテゴリが選択状態の場合は，その項目を削除します（I/O レジスタの削除不可）。
表記	表示形式を指定するため，次のカスケード・メニューを表示します。
16 進数	選択している項目の値を 16 進数で表示します（デフォルト）。
符号付き 10 進数	選択している項目の値を符号付き 10 進数で表示します。
符号なし 10 進数	選択している項目の値を符号なし 10 進数で表示します。
8 進数	選択している項目の値を 8 進数で表示します。
2 進数	選択している項目の値を 2 進数で表示します。
ASCII	選択している項目の値を ASCII コードで表示します。
16 進数値を併記	選択している項目の値表示の末尾に，その値の 16 進数表記を“（）”で囲んで併記します。
表示色をリセット	選択している I/O レジスタに対して，プログラム実行により値が変化したことを示す強調表示をリセットします。

ローカル変数 パネル

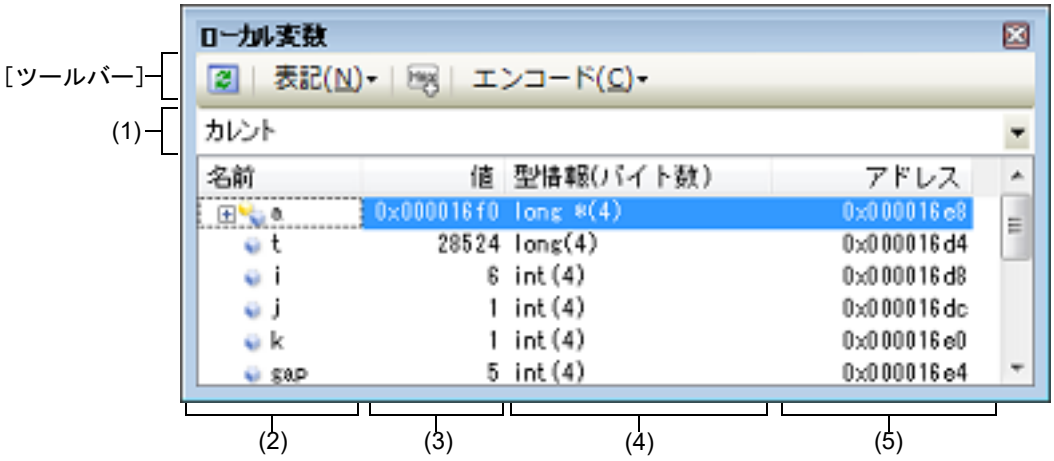
ローカル変数の内容の表示，および値の変更を行います（「2.11.5 ローカル変数を表示／変更する」参照）。
なお，このパネルは，デバッグ・ツールと接続時のみオープンすることができます。

注意 1. プログラム実行中は，このパネルには何も表示されません。プログラムの実行が停止したタイミングで，各エリアの表示を行います。

注意 2. コンパイラによる最適化のため，対象となる変数を使用していないブロックでは変数データがスタック／レジスタに存在しない場合があります。この場合は対象となる変数は表示されません。

備考 パネル上の各エリアの区切り線をダブルクリックすることにより，該当エリアの内容を省略することなく表示可能な最小幅に変更することができます。

図 A.30 ローカル変数 パネル



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］
- ［ツールバー］
- ［［ファイル］メニュー（ローカル変数 パネル専用部分）］
- ［［編集］メニュー（ローカル変数 パネル専用部分）］
- ［コンテキスト・メニュー］

［オープン方法］

- ［表示］メニュー→［ローカル変数］を選択

［各エリアの説明］

(1) スコープ・エリア
表示するローカル変数のスコープをドロップダウン・リストにより選択します。
選択できる項目は次のとおりです。

項目	動作
カレント	カレント PC 値のスコープでのローカル変数を表示します。
＜*深さ*＞ ＜ *関数名* ()[*ファイル名* # *行番号*]＞[注]	呼び出し元の関数のスコープでのローカル変数を表示します。 プログラム実行後，選択したスコープが存在するかぎり，ここで選択したスコープを保ちます。

注 コール・スタック パネルで表示している関数呼び出し元を表示します。

(2) ［名前］エリア
ローカル変数名や関数名を表示します。関数の引数もローカル変数として表示します。
また，配列，ポインタ型変数，構造体／共用体は，階層構造をツリー形式で表示します。
このエリアを編集することはできません。
表示される各アイコンの意味は次のとおりです。

	変数を示します。 Auto 変数，内部スタティック変数，Register 変数の表示も行います注。 配列，ポインタ型変数，構造体／共用体は，階層構造をツリー形式で表示します。 先頭に “+” マークがある場合は，これをクリックすることにより次を展開表示します（展開後 “-” マークに変化）。	
	配列	配列中の全要素
	ポインタ型変数	ポインタが指し示す先の変数 なお，ポインタが指し示す先がポインタの場合は，さらに “+” マークを付与し，これをクリックすることにより参照先を表示します。 ただし，ポインタの指す値が不明な場合は，“?” を表示します。
	構造体／共用体	構造体／共用体の全メンバ
	引数を示します。	
	関数を示します。	

注 Auto 変数を表示する場合，関数のプロローグ（関数の “{”）やエピローグ（関数の “}”）ではローカル変数の正確な値を表示することができません（Auto 変数のアドレスは，スタック・ポインタ（SP）からの相対アドレスとなり，関数内で SP の値が確定するまで確定しません。プロローグやエピローグでは SP の操作が行われており，SP の値が確定していません。このため，プロローグやエピローグでは正確な値の表示ができません）。

このエリアは，次の機能を備えています。

(a) ウォッチ式の登録
C 言語変数をウォッチ式としてウォッチ パネルに登録することができます。
操作方法についての詳細は，「2.11.6.1 ウォッチ式を登録する」を参照してください。

備考 登録したウォッチ式には，自動的にスコープ指定が付与されます。

(b) メモリへのジャンプ
コンテキスト・メニューの［メモリへジャンプ］を選択することにより，選択しているローカル変数が配置されているアドレスにキャレットを移動した状態でメモリ パネル（メモリ 1）がオープンします（すでにオープンしている場合はメモリ パネル（メモリ 1）にジャンプ）。

(c) 表示の拡大／縮小
ローカル変数 パネルにフォーカスがある状態で，メイン・ウインドウのツールバー上にある拡大率のドロップダウン・リストを変更することにより，ローカル変数 パネルの表示を拡大／縮小することができます。
また，［Ctrl］キーとマウス・ホイールを組み合わせることでも変更可能です。

- ［Ctrl］キーを押下しながらマウス・ホイールを前方に動かすと，ローカル変数 パネルの表示を拡大し，表示が大きく見やすくなります（最大 300%）。
- ［Ctrl］キーを押下しながらマウス・ホイールを後方に動かすと，ローカル変数 パネルの表示を縮小し，表示が小さくなります（最小 50%）。

なお，拡大率を変更した状態のパネルをクローズした場合，拡大率は保持されます（再度パネルをオープンする際に，変更した拡大率でオープンします）。

(3) ［値］エリア
ローカル変数の値を表示／変更します。
表示進数や文字列のエンコードは，ツールバーのボタン，またはコンテキスト・メニューより選択することができます。また，常に 16 進数表示を併記する表示形式を選択することもできます。
ローカル変数の値として表示されるマークや色の意味は次のとおりです（表示の際の文字色／背景色はオプション ダイアログにおける［全般 - フォントと色］カテゴリの設定に依存）。

表示例（デフォルト）			説明
0x0	文字色	青色	ユーザにより，値が変更されているローカル変数値（[Enter] キーによりターゲット・メモリに書き込まれます）
	背景色	標準色	
0x0	文字色	茶色	プログラムの実行により，値が変化したローカル変数値注 プログラムを再び実行することにより，強調色がリセットされます。
	背景色	クリーム	
?	文字色	グレー	ローカル変数の値を取得できない場合
	背景色	標準色	

注　プログラムの実行開始位置からブレークした位置で同じ変数名を表示していて，かつ，その変数値が変化している場合が対象となります。

このエリアは，次の機能を備えています。

(a) ローカル変数値／引数値の変更
ローカル変数値，および引数値の変更は，対象ローカル変数値を選択したのち，再度クリックし，キーボードからの直接入力により行います（[Esc] キーの押下で編集モードをキャンセルします）。
ローカル変数値，および引数値を編集したのち，[Enter] キーの押下，または編集領域以外へのフォーカスの移動により，デバッグ・ツールのターゲット・メモリに書き込まれます。
ローカル変数値／引数値の変更方法についての詳細は，「2.11.5.2 ローカル変数の内容を変更する」を参照してください。

(b) ローカル変数値の保存
[ファイル] メニュー→ [名前を付けてローカル変数データを保存 ...] を選択することにより，名前を付けて保存 ダイアログをオープンし，このパネルのすべての内容をテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存することができます。
ローカル変数値の保存方法についての詳細は，「2.11.5.3 ローカル変数の表示内容を保存する」を参照してください。

(4) [型情報（バイト数）] エリア
ローカル変数の型名を表示します。表記は C 言語の記述に従います。
配列の場合は “[ ]” 内に要素数を，関数の場合は “()” 内にサイズ（バイト数）を付与して表示します。
なお，このエリアを編集することはできません。

(5) [アドレス] エリア
ローカル変数のアドレスを表示します。変数がレジスタに割り当てれられている場合は，レジスタ名を表示します。
このエリアを編集することはできません。

[ツールバー]

ツールバーの各ボタンは，プログラム実行中は無効となります。

	デバッグ・ツールから最新の情報を取得し，表示を更新します。
表記	値の表示形式を変更する次のボタンを表示します。

自動	このパネル上の値の表記を変数ごとの規定値で表示します。
16 進数	このパネル上の値を 16 進数で表示します。
10 進数	このパネル上の値を 10 進数で表示します。
8 進数	このパネル上の値を 8 進数で表示します。
2 進数	このパネル上の値を 2 進数で表示します。
配列のインデックスを 10 進数表記	このパネル上の配列のインデックスを 10 進数で表示します（デフォルト）。
配列のインデックスを 16 進数表記	このパネル上の配列のインデックスを 16 進数で表示します。
Float	このパネル上の値を Float で表示します。 ただし，4 バイト・データ以外，または型情報を持つ場合は，規定値で表示します。
Double	このパネル上の値を Double で表示します。 ただし，8 バイト・データ以外，または型情報を持つ場合は，規定値で表示します。
	値表示の末尾に，その値の 16 進数表記を“（）”で囲んで併記します。
エンコード	文字列変数のエンコードを変更する次のボタンを表示します。
ASCII	文字列変数を ASCII コードで表示します。
Shift_JIS	文字列変数を Shift_JIS コードで表示します（デフォルト）。
EUC-JP	文字列変数を EUC-JP コードで表示します。
UTF-8	文字列変数を UTF-8 コードで表示します。
UTF-16	文字列変数を UTF-16 コードで表示します。

[[ファイル] メニュー（ローカル変数 パネル専用部分）]

ローカル変数 パネル専用の［ファイル］メニューは次のとおりです（その他の項目は共通）。
ただし，プログラム実行中はすべて無効となります。

ローカル変数データを保存	このパネルの内容を前回保存したテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存します（「(b) ローカル変数値の保存」参照）。 なお，起動後に初めてこの項目を選択した場合は，［名前を付けてローカル変数データを保存 ...］の選択と同等の動作となります。
名前を付けてローカル変数データを保存 ...	このパネルの内容を指定したテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存するために，名前を付けて保存 ダイアログをオープンします（「(b) ローカル変数値の保存」参照）。

[[編集] メニュー（ローカル変数 パネル専用部分）]

ローカル変数 パネル専用の［編集］メニューは次のとおりです（その他の項目はすべて無効）。

コピー	選択している行の内容，または文字列をクリップ・ボードにコピーします。
すべて選択	項目をすべて選択状態にします。
名前の変更	選択しているローカル変数の値を変更するために，編集モードに移行します（「2.11.5.2 ローカル変数の内容を変更する」参照）。 ただし，プログラム実行中は無効となります。
検索 ...	検索・置換 ダイアログを［一括検索］タブが選択状態でオープンします。

置換 ...	検索・置換 ダイアログを ［一括置換］タブが選択状態でオープンします。

［コンテキスト・メニュー］

コンテキスト・メニューの各項目は，プログラム実行中は無効となります。

ウォッチ 1 に登録	選択しているローカル変数をウォッチ パネル（ウォッチ 1）に登録します。
コピー	選択している行の内容，または文字列をクリップ・ボードにコピーします。
表記	表示形式を指定するために，次のカスケード・メニューを表示します。
自動	このパネル上の値の表記を変数ごとの規定値で表示します（デフォルト）。
16 進数	このパネル上の値を 16 進数で表示します。
10 進数	このパネル上の値を 10 進数で表示します。
8 進数	このパネル上の値を 8 進数で表示します。
2 進数	このパネル上の値を 2 進数で表示します。
配列のインデックスを 10 進表記	このパネル上の配列のインデックスを 10 進数で表示します（デフォルト）。
配列のインデックスを 16 進表記	このパネル上の配列のインデックスを 16 進数で表示します。
Float	このパネル上の値を Float で表示します。 ただし，4 バイト・データ以外，または型情報を持つ場合は，規定値で表示します。
Double	このパネル上の値を Double で表示します。 ただし，8 バイト・データ以外，または型情報を持つ場合は，規定値で表示します。
16 進数値を併記	値表示の末尾に，その値の 16 進数表記を“（）”で囲んで併記します。
エンコード	文字コードを指定するため，次のカスケード・メニューを表示します。
ASCII	文字列変数を ASCII コードで表示します。
Shift_JIS	文字列変数を Shift_JIS コードで表示します（デフォルト）。
EUC-JP	文字列変数を EUC-JP コードで表示します。
UTF-8	文字列変数を UTF-8 コードで表示します。
UTF-16	文字列変数を UTF-16 コードで表示します。
メモリへジャンプ	選択している行が示すアドレスにキャレットを移動した状態で，メモリ パネル（メモリ 1）がオープンします。

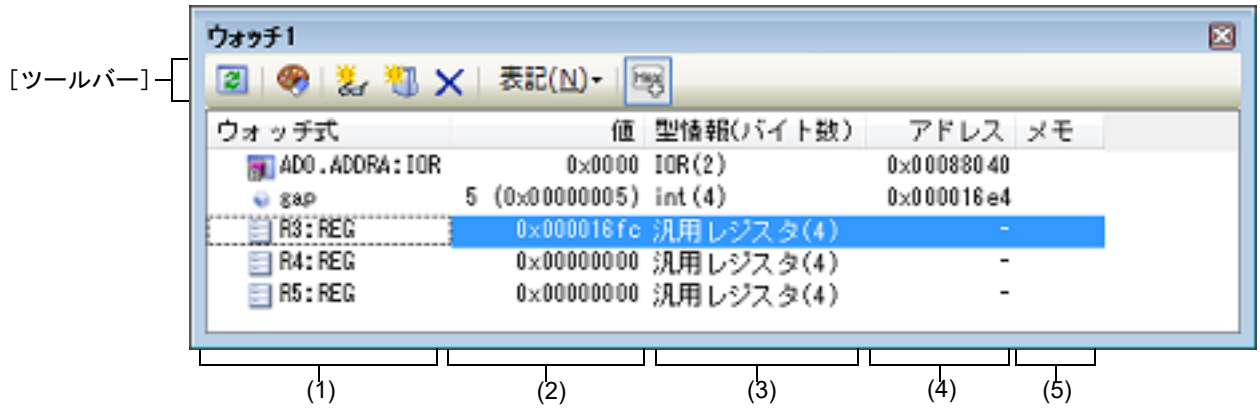
ウォッチ パネル

登録したウォッチ式の内容の表示，および値の変更を行います（「2.11.6 ウォッチ式を表示／変更する」参照）。
このパネルは，最大 4 個までオープンすることができます。各パネルは，タイトルバーの“ウォッチ 1”，“ウォッチ 2”，“ウォッチ 3”，“ウォッチ 4”の名称で識別され，それぞれ個別にウォッチ式の登録／削除／移動を行うことができます。
ウォッチ式の登録はこのパネル上から行えますが，エディタ パネル／逆アセンブル パネル／メモリ パネル／ CPU レジスタ パネル／ローカル変数 パネル／ IOR パネルより行うことも可能です。
ウォッチ式が登録されている状態のパネルをクローズした場合，そのパネルは非表示となりますが，登録されていたウォッチ式の情報は保持されます（再度そのパネルをオープンした際に，ウォッチ式が登録されている状態でオープンします）。
プログラムの実行後，ウォッチ式の値が変化すると表示を自動的に更新します（ステップ実行時には，ステップ実行ごとに表示を逐次更新）。
また，リアルタイム表示更新機能を有効にすることにより，プログラム実行中であっても，値の表示をリアルタイムに更新することも可能です。
なお，このパネルは，デバッグ・ツールと接続時のみオープンすることができます。

注意 プログラム実行中に CPU レジスタの内容を表示／変更することはできません。

備考 パネル上の各エリアの区切り線をダブルクリックすることにより，該当エリアの内容を省略することなく表示可能な最小幅に変更することができます。

図 A.31 ウォッチ パネル



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー（ウォッチ パネル専用部分）]
- [[編集] メニュー（ウォッチ パネル専用部分）]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [ウォッチ] → [ウォッチ *1* ～ *4*] を選択

[各エリアの説明]

(1) [ウォッチ式] エリア
登録しているウォッチ式を一覧で表示します。
このエリアの表タイトル部をクリックすることにより，一覧内のウォッチ式をアルファベット順でソートすることができます。
また，カテゴリ（フォルダ）を自由に作成してウォッチ式を分類し，ツリー形式で表示することができます（「(a) ツリーの編集」参照）。

表示される各アイコンの意味は次のとおりです。

	このカテゴリに属するウォッチ式を表示している状態を示します。アイコンをダブルクリック，または “-” マークをクリックすると，カテゴリを閉じウォッチ式を非表示にします。
	このカテゴリに属するウォッチ式が非表示の状態を示します。アイコンをダブルクリック，または “+” マークをクリックすると，カテゴリを開きウォッチ式を表示します。
	ウォッチ式が変数であることを示します。 配列，ポインタ型変数，構造体／共用体を示すウォッチ式の先頭には，“+” ／ “-” マークを表示し，これをクリックすることにより展開／折りたたみ表示します。
	ウォッチ式が関数であることを示します。
	ウォッチ式が即値であることを示します。
	ウォッチ式が式であることを示します。
	ウォッチ式が I/O レジスタであることを示します。
	ウォッチ式が CPU レジスタであることを示します。 下階層のレジスタ（レジスタの部分）を持つウォッチ式の先頭には，“+” ／ “-” マークを表示し，これをクリックすることにより展開／折りたたみ表示します。

このエリアは，次の機能を備えています。

(a) ツリーの編集

ウォッチ式をカテゴリ（フォルダ）で分類し，ツリー形式で表示することができます。
カテゴリを新規に作成する場合は，作成したい位置にキャレットを移動したのち，ツールバーの ボタンのクリック，またはコンテキスト・メニューの［カテゴリを作成］を選択し，任意にカテゴリ名称を入力することにより行います。
なお，カテゴリを削除する場合は，削除したいカテゴリを選択したのち，ツールバーの ボタンのクリック，またはコンテキスト・メニューの［削除］を選択します。
また，作成したカテゴリ名を編集する場合は，編集したいカテゴリ名を選択したのち，次のいずれかの操作により行います。

- 再度クリック後，キーボードよりカテゴリ名を直接編集
- ［編集］メニュー→［名前の変更］を選択後，キーボードよりカテゴリ名を直接編集
- ［F2］キーを押下後，キーボードよりカテゴリ名を直接編集

カテゴリを作成したのち，登録済みのウォッチ式をカテゴリ内に直接ドラッグ・アンド・ドロップすることにより，各ウォッチ式をカテゴリで分類したツリー形式で表示します。
同様に，カテゴリとウォッチ式の表示の順番（上下位置）も，ドラッグ・アンド・ドロップ操作により自由に変更することができます。

注意 1. カテゴリ内にカテゴリを作成することはできません。

注意 2. 1 つのウォッチ パネルにおいて，カテゴリは 64 個まで作成することができます（上限値を越えて作成しようとした場合，メッセージを表示します）。

備考 ウォッチ式／カテゴリを他のウォッチ パネル（ウォッチ 1 ～ウォッチ 4）にドラッグ・アンド・ドロップすると，ドロップ先のウォッチ パネルにウォッチ式／カテゴリがコピーされます。

(b) 展開／折りたたみ表示

配列，ポインタ型変数，構造体／共用体／クラス，レジスタ（部分を表す名前がついているもののみ）を示すウォッチ式の先頭には，“+” マークを表示し，これをクリックすることにより次を展開表示します（展開後 “-” マークに変化）。

ウォッチ式	展開表示の際の内容
配列	配列中の全要素 コンテキスト・メニューの［表記］→［ASCII］を選択することにより，文字列として表示可能です（最大表示文字数：256 文字）。 ただし，エンコードの種類により表示不可能な場合は，“.” または “?” を表示します。
ポインタ型変数	ポインタが指し示す先の変数

ウォッチ式	展開表示の際の内容
構造体／共用体／クラス	構造体／共用体／クラスの全メンバ
レジスタ	レジスタを構成するビット／ビット列の名称 例）PSW レジスタの場合 IPL レジスタ，PM レジスタ，U レジスタ，I レジスタ O レジスタ，S レジスタ，Z レジスタ，C レジスタ

(c) 新規ウォッチ式の登録
新規にウォッチ式を登録する方法には，次の 3 通りがあります。

<1> 他のパネルからのウォッチ式の登録
他のパネル上において，ウォッチ式として登録したい対象に対して，次のいずれかの操作を行います。

- 対象文字列を選択したのち，任意のウォッチ パネル（ウォッチ 1 ～ウォッチ 4）上のこのエリアに直接ドラッグ・アンド・ドロップ（エディタ パネルを除く）
- 対象文字列を選択したのち，または対象文字列のいずれかにキャレットを移動したのち（対象は自動的に決定されます），コンテキスト・メニューの［ウォッチ 1 に登録］を選択
- 対象文字列を［編集］メニュー→［コピー］したのち，任意のウォッチ パネル（ウォッチ 1 ～ウォッチ 4）上のこのエリアで［編集］メニュー→［貼り付け］を選択

なお，この操作が可能なパネルとウォッチ式として登録可能な対象との関係は次のとおりです。

表 A.4 各パネルとウォッチ式として登録可能な対象の関係

パネル名	ウォッチ式として登録可能な対象
エディタ パネル	C/C++ 言語変数注 1/CPU レジスタ /I/O レジスタ／アセンブラ・シンボル
逆アセンブル パネル	C/C++ 言語変数注 1/CPU レジスタ /I/O レジスタ／アセンブラ・シンボル
CPU レジスタ パネル	CPU レジスタ注 2
ローカル変数 パネル	C/C++ 言語変数注 1（ローカル変数）
IOR パネル	I/O レジスタ注 2

注 1. C89，C99，C++ 言語の変数を表します。

注 2. 自動的にスコープ指定がウォッチ式に付与されます。

<2> ウォッチ パネル上での直接登録
任意のウォッチ パネル（ウォッチ 1 ～ウォッチ 4）において，ツールバーの ボタンをクリック，またはコンテキスト・メニューの［新規ウォッチ式を追加］を選択することにより，このエリアの最下段に新規ウォッチ式用のエントリ・ボックスが表示されます。
エントリ・ボックスの［ウォッチ式］エリアにおいて，キーボードより直接ウォッチ式を入力したのち，［Enter］キーを押下します。
この際のウォッチ式の入力形式は次のとおりです。

表 A.5 ウォッチ式の入力形式

ウォッチ式	表示する値
C/C++ 言語変数名注 1	C/C++ 言語の変数の値
ウォッチ式[*ウォッチ式*]	配列の要素値
ウォッチ式. メンバ名注 2	構造体／共用体／クラスのメンバ値
ウォッチ式-> メンバ名注 2	ポインタの指し示す構造体／共用体／クラスのメンバ値
ウォッチ式 .* *キャスト式*	メンバ変数へのポインタの値
ウォッチ式 ->* *キャスト式*	メンバ変数へのポインタの値

ウォッチ式	表示する値
* *ウォッチ式*	ポインタの変数の値
& *ウォッチ式*	配置アドレス
(型名) *ウォッチ式*	指定された型へキャストした値
CPU レジスタ名	CPU レジスタの値
I/O レジスタ名	I/O レジスタの値
ラベル名[注 3]，EQU シンボル名[注 3]，[即値]	ラベルの値，EQU シンボルの値，即値アドレスの値
整定数	整数の定数値
浮動定数	浮動小数点の定数値
文字定数	文字定数値

注 1. C89，C99，C++ 言語の変数を表します。

注 2. 基底クラスのメンバ変数を指定する場合，メンバ名の前にスコープを指定してください。（例：variable.BaseClass::member）

注 3. ラベル名または，EQU シンボル名に “$” が含まれている場合，名前を “{}” で囲んでください。（例：{$Label}）
虚数の値には，大文字の “I” を掛けてください（例：1.0 + 2.0*I）。“I” はキーワードとなるため，CPU レジスタの “I” を指定する場合は “:REG” を付加してください（例：I:REG）。

また，ウォッチ式は，スコープを指定して登録することができます。スコープ指定してウォッチ式を登録した場合の扱いは次のとおりです。

表 A.6 C/C++ 言語関数をスコープ指定してウォッチ登録した場合の扱い

スコープ指定	ロード・モジュール名[注 1]	ソース・ファイル名[注 1]	関数名[注 2]	検索対象
prog$file#func	prog	file	func	スタティック関数
prog$func	prog	グローバル	func	グローバル関数
file#func	カレント	file	func	スタティック関数
func	カレント	カレント	func	すべて[注 2]

注 1. ロード・モジュール名またはファイル名に空白や以下の記号が含まれている場合，名前をダブルクォーテーション “ ” で囲んでください（例：“c: ¥ folder ¥ prog.abs”$file.c#func）。
$，#，(，)，[，]，&，^，~，%，+，-，*，/，:，?，’，|，¥，<，>，！

注 2. カレント PC 値のスコープからスタティック関数，グローバル関数の順で検索します。スコープ範囲外のスタティック関数は検索対象外になります。
名前空間内に定義されている関数を指定する場合は，スコープを記入しないでください（例：Scope::func）。
また，同名の関数が存在する場合はパラメータの型名を明記してください（例：func(int, int)）。

表 A.7 C 言語変数をスコープ指定してウォッチ登録した場合の扱い

スコープ指定	ロード・モジュール名 注 1	ソース・ファイル名 注 1	関数名 注 2	変数名 注 2	検索対象
prog$file#func#var	prog	file	func	var	スタティック関数内スタティック変数[注 2,3]
prog$file#var	prog	file	グローバル	var	ファイル内スタティック変数

スコープ指定	ロード・モジュール名 注 1	ソース・ファイル名 注 1	関数名 注 2	変数名 注 2	検索対象
prog$var	prog	グローバル	グローバル	var	グローバル変数
file#func#var	カレント	file	func	var	スタティック関数内スタティック変数注 2,3
file#var	カレント	file	グローバル	var	ファイル内スタティック変数
var	カレント	カレント	カレント	var	すべて注 4

注 1. ロード・モジュール名またはファイル名に空白や以下の記号が含まれている場合，名前をダブルクォーテーション“ ”で囲んでください（例：“c:¥folder¥prog.abs”$file.c#func#var）。
$，#，(，)，[，]，&，^，~，%，+，-，*，/，:，?，’，|，¥，<，>，！

注 2. 名前空間内に定義されている関数および変数を指定する場合は，スコープを記入してください（例：Scope::func）。
また，同名の関数が存在する場合はパラメータの型名を明記してください（例：func(int, int)）。

注 3. カレント PC 値が指定関数内にある場合は，スタティック宣言されていないローカル変数も検索対象となります。

注 4. カレント PC 値のスコープからローカル変数，ファイル内スタティック変数，グローバル変数の順で検索します。スコープ範囲外のローカル変数およびファイル内スタティック変数は，検索対象外となります。

表 A.8 CPU レジスタをスコープ指定してウォッチ登録した場合の扱い

スコープ指定	レジスタ・バンク	CPU レジスタ名
R10:REG	（なし）	R10

表 A.9 I/O レジスタをスコープ指定してウォッチ登録した場合の扱い

スコープ指定	I/O レジスタ名
AD0.ADCR:IOR	AD0.ADCR
AD0.ADCR	AD0.ADCR

備考 1. このエリアで [Ctrl]+[Space] キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます。（「2.20.2 シンボル名の入力補完機能」参照）。

備考 2. 即値は数値として扱われます。また，即値に演算子を使用することができます。

備考 3. ウォッチ式として，シンボルを使用した演算式を指定することができます。

備考 4. 同名の C/C++ 言語変数 /CPU レジスタ /I/O レジスタが存在する際に，スコープ指定せずにそれらを登録した場合，次の順にシンボルを解決し，値を表示します。
C/C++ 言語変数＞ CPU レジスタ＞ I/O レジスタ

備考 5. 同名のローカル変数とグローバル変数が存在する際に，スコープを指定せずにシンボル名のみ登録した場合，カレント PC 値のスコープを元にシンボルを解決し，値を表示します。

備考 6. IOR パネル /CPU レジスタ パネルよりウォッチ式を登録した場合，ウォッチ式には自動的にスコープ指定が付与されます。

備考 7. ウォッチ式として単に “I” と指定した場合，虚数のキーワードとして解釈します。レジスタ “I” の値を取得する場合は，レジスタの後ろに “:REG” を付加してください。

<3> 他のアプリケーションからの登録
外部エディタなどから，C/C++ 言語変数 /CPU レジスタ /I/O レジスタ／アセンブラ・シンボルの文字列を選択し，次のいずれかの操作を行います。

- 対象文字列を，任意のウォッチ パネル（ウォッチ 1 ～ウォッチ 4）上のこのエリアに直接ドラッグ・アンド・ドロップ

- 対象文字列をクリップ・ボードにコピーしたのち，任意のウォッチ パネル（ウォッチ 1 ～ウォッチ 4）上のこのエリアで［編集］メニュー→［貼り付け］を選択

注意 1. 1 つのウォッチ パネルにおいて，ウォッチ式は 128 個まで登録することができます（上限値を越えて登録しようとした場合，メッセージを表示します）。

注意 2. コンパイラによる最適化のため，対象となる変数を使用していないブロックでは変数データがスタック／レジスタに存在しない場合があります。この場合，対象となる変数をウオッチ式として登録しても値の表示は “?” のままとなります。

備考 1. 各ウォッチ パネル（ウォッチ 1 ～ウォッチ 4）上で登録したウォッチ式は，それぞれ個別に管理され，プロジェクトのユーザ情報として保存されます。

備考 2. ウォッチ式は，同名を複数登録することができます。

(d) ウォッチ式の編集
登録済みのウォッチ式の編集は，対象ウォッチ式をダブルクリックし，対象ウォッチ式を編集モードにしたのち，キーボードから編集内容を直接入力して行います（[Esc］キーの押下で編集モードをキャンセルします）。
ウォッチ式を編集したのち，［Enter］キーを押下すると編集を完了します。

(e) ウォッチ式の削除
ツールバーの ボタンのクリック，またはコンテキスト・メニューの［削除］を選択することにより，選択しているウォッチ式を削除します。

(f) 各種イベントの設定
コンテキスト・メニューの［アクセス・ブレークの設定］／［トレース出力］を選択することにより，選択しているウォッチ式に各種イベントを設定することができます。
アクセス系のブレーク・イベントが設定された場合，ウォッチ式のアイコンが変化します（ウォッチ式のアイコンの下にブレーク・イベントのイベント・マークを重ねて表示）。トレース・イベントの場合は，ウォッチ式のマークに変化はありません。
イベントを設定することにより，設定したイベントの詳細情報がイベント パネルに反映されます。
ただし，イベントの設定は，対象となるウォッチ式がグローバル変数／関数内スタティック変数／ファイル内スタティック変数 /I/O レジスタの場合にのみ行うことができます。
イベントの設定方法についての詳細は次を参照してください。

- 「2.10.4 変数 /I/O レジスタへのアクセスで停止する」
- 「2.13.4 条件を満たしたときのみの実行履歴を収集する」
- 「2.14.3 任意区間の実行時間を計測する」

(g) メモリ定義アドレスへのジャンプ
コンテキスト・メニューの［メモリへジャンプ］を選択することにより，選択しているウォッチ式が定義されているアドレスにキャレットを移動した状態でメモリ パネル（メモリ 1）がオープンします（すでにオープンしている場合は，メモリ パネル（メモリ 1）にジャンプ）。
ただし，同時に複数のウォッチ式を選択している場合，または I/O レジスタ /CPU レジスタを選択している場合は，無効となります。

(h) 表示の拡大／縮小
ウォッチ パネルにフォーカスがある状態で，メイン・ウインドウのツールバー上にある拡大率のドロップダウン・リストを変更することにより，ウォッチ パネルの表示を拡大／縮小することができます。
また，［Ctrl］キーとマウス・ホイールを組み合わせることでも変更可能です。

- ［Ctrl］キーを押下しながらマウス・ホイールを前方に動かすと，ウォッチ パネルの表示を拡大し，表示が大きく見やすくなります（最大 300%）。
- ［Ctrl］キーを押下しながらマウス・ホイールを後方に動かすと，ウォッチ パネルの表示を縮小し，表示が小さくなります（最小 50%）。

なお，拡大率を変更した状態のパネルをクローズした場合，拡大率は保持されます（再度パネルをオープンする際に，変更した拡大率でオープンします）。

(2) ［値］エリア
登録しているウォッチ式の値を表示／変更します。
なお，ウォッチ式が関数ポインタの場合は，関数名を表示します。
表示進数やエンコードは，ツールバーのボタン，またはコンテキスト・メニューより選択することができます。
また，常に 16 進数値を併記する表示形式を選択することもできます。
なお，デフォルトの表示形式は，ウォッチ式の型に依存して，次のように自動的に決定されます。

表 A.10 ウォッチ式の表示形式（デフォルト）

ウォッチ式の型	表示形式
char，signed char，unsigned char	ASCII 文字に続き“（）”内に 16 進数値を併記
short，signed short，short int，signed short int，int，signed，signed int，long，signed long，long int，signed long int	符号付き 10 進数値に続き“（）”内に 16 進数値を併記
unsigned short，unsigned short int，unsigned，unsigned int，unsigned long，unsigned long int	符号なし 10 進数値に続き“（）”内に 16 進数値を併記
float，float _Complex，float _Imaginary	Float（サイズが 4 バイトの場合）値注
double，long double，double _Complex，long double _Complex，double _Imaginary，long double _Imaginary	Double（サイズが 8 バイトの場合）値注
char，signed char，unsigned char へのポインタ	文字列 エンコード：Shift_JIS
char，signed char，unsigned char 以外へのポインタ	16 進数
char，signed char，unsigned char 型の配列	文字列 エンコード：Shift_JIS
bit，boolean，_boolean	符号なし 10 進数値に続き“（）”内に 16 進数値を併記
列挙型	列挙定数値に続き“（）”内に 16 進数値を併記
ラベル，即値アドレス，EQU シンボル	符号付き 10 進数値に続き“（）”内に 16 進数値を併記
ビット・シンボル	符号なし 10 進数値に続き“（）”内に 16 進数値を併記
その他	16 進数

注 浮動小数点型および複素数型の値は，近似値（round to nearest）で丸めたものを表示します。

また，ウォッチ式の値として表示されるマークや色の意味は次のとおりです（表示の際の文字色／背景色はオプション ダイアログにおける［全般 - フォントと色］カテゴリの設定に依存）。

表示例（デフォルト）			説明
0x0	文字色	青色	ユーザにより，値が変更されているウォッチ式の値（［Enter］キーによりターゲット・メモリに書き込まれます）
	背景色	標準色	
0x0	文字色	ピンク	リアルタイム表示更新機能を行っているウォッチ式の値
	背景色	標準色	
0x0	文字色	茶色	プログラムの実行により，値が変化したウォッチ式の値 ツールバーの ボタン，またはコンテキスト・メニューの［表示色をリセット］を選択することにより，強調表示をリセットします。
	背景色	クリーム	
?	文字色	グレー	存在しない変数をウォッチ式として登録した場合，またはウォッチ式の値を取得できなかった場合（変数がスコープを外れた場合など）
	背景色	標準色	

備考 1. 読み込み動作によってマイクロコントローラが動作してしまう I/O レジスタは，読み込み保護対象となり，値の読み込みは行いません。読み込み保護対象の I/O レジスタ の内容を読み込みたい場合には，コンテキスト・メニューの［値を強制読み込み］を選択してください。

備考 2. 各ウォッチ式は，登録された順序で値の取得を行います。
このため，同一の I/O レジスタを複数登録した場合，値を取得するタイミングに差が生じるため，表示される値が異なる場合があります。

備考 3. 16 進数値を併記している場合では，指定表記の値と 16 進数の値を個別に読み出します。
このため，値を取得するタイミングに差を生じるため，指定表記値と 16 進数値が異なる場合があります。

このエリアは，次の機能を備えています。

(a) リアルタイム表示更新機能
リアルタイム表示更新機能を使用することにより，プログラムが停止している状態の時だけでなく，実行中の状態であっても，登録したウォッチ式の値の表示／変更を行うことができます。
リアルタイム表示更新機能についての詳細は，「2.11.1.4 プログラム実行中にメモリの内容を表示／変更する」を参照してください。

(b) ウォッチ式の値の変更
ウォッチ式の値の変更は，対象ウォッチ式の値を選択したのち，再度クリックし，キーボードからの直接入力により行います（[Esc] キーの押下で編集モードをキャンセルします）。
ウォッチ式の値を編集したのち，[Enter] キーの押下，または編集領域以外へのフォーカスの移動により，ターゲット・メモリに書き込まれます。
ウォッチ式の値の変更方法についての詳細は，「2.11.6.6 ウォッチ式の内容を変更する」を参照してください。

(c) ウォッチ式の値の保存
[ファイル] メニュー→ [名前を付けてウォッチ・データを保存 ...] を選択することにより，名前を付けて保存 ダイアログをオープンし，このパネルのすべての内容をテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存することができます。
また，コンテキスト・メニューの [ウォッチ・データを展開保存 ...] を選択することにより，名前を付けて保存 ダイアログをオープンし，選択したウォッチ式の内容をテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存することができます。
ウォッチ式の値の保存方法については，「2.11.6.9 ウォッチ式の表示内容を保存する」を参照してください。

(3) [型情報（バイト数）] エリア
ウォッチ式に対して，次の形式で型情報を表示します。

ウォッチ式	表示形式	
単独の CPU レジスタ	*＜CPU レジスタの種類＞*（*＜サイズ*[注 1]*＞*）	
単独の I/O レジスタ	*＜I/O レジスタの種類＞*（*＜アクセス属性＞ ＜アクセス・タイプ＞＜サイズ*[注 1]*＞*）	
	アクセス属性	R : 読み出しのみ可能 W : 書き込みのみ可能 R/W : 読み出し／書き込み可能
	アクセス・タイプ	1 : 1 ビット・アクセス可能 8 : バイト・アクセス可能 16 : ハーフ・ワード・アクセス可能 32 : ワード・アクセス可能
判別不能	?	
上記以外	*＜C コンパイラの判定に従ったウォッチ式の型*[注 2]*＞*（*＜サイズ*[注 1]*＞*）	

注 1. ウォッチ式のサイズをバイト単位で示します。
ただし，ビット IOR/C 言語ビット・フィールドについては，ビット単位で表示し，数値の末尾に“ビット”表記を付与します。

注 2. ウォッチ式をコンパイルする際に，どの型として扱われるかを示します。
double，long double 型は，CC-RX のプロパティの [共通オプション] タブ - [CPU] カテゴリ内にある「double 型，および long double 型の精度」に従った型名を出力します。“単精度として扱う”が設定された場合は float 型，“倍精度として扱う”が設定された場合は double 型の型名とサイズを出力します。

(4) [アドレス] エリア
各ウォッチ式がマッピングされているアドレスを表示します（16 進数表記固定）。
ただし，ウォッチ式が，単独の CPU レジスタの場合は “-” を，また判別不能の場合では，“?” を表示します。

備考 ウォッチ式が I/O レジスタで，ビット・レジスタの場合は，次のようにビット・オフセット値を付与して表示します。

例 アドレス “0xFF40” のビット 4 に割り当てられている（ビット・レジスタ）の場合
表示内容：0xFF40.4

(5) ［メモ］エリア
ウォッチ式／カテゴリに対して，ユーザが自由にコメントを入力することができます。
このエリアに入力したコメントの内容は，各ウォッチ式／カテゴリに対して個別に保持され，プロジェクトのユーザ情報として保存されます。したがって，ウォッチ式／カテゴリを削除すると，対応するメモの内容も破棄されます。
ただし，配列，レジスタなどを展開表示している場合，各展開要素に対してコメントを入力することはできません。
コメントを編集する場合は，編集したい項目をダブルクリックすることにより，選択した項目が編集モードとなります（［Esc］キーの押下で編集モードをキャンセルします）。最大 256 文字までの文字列をキーボードより直接入力することができます（改行コードは無効）。
文字列編集後，［Enter］キーの押下，または編集領域以外へのフォーカスの移動により，文字列編集を完了します。

［ツールバー］

	登録しているウォッチ式のすべての値を再取得し，表示を更新します。 ただし，読み込み保護対象の I/O レジスタの再読み込みは行いません。
	選択しているウォッチ式に対して，プログラムの実行により値が変化したことを示す強調表示をリセットします。 ただし，プログラム実行中は無効となります。
	新規ウォッチ式を登録します。テキスト・ボックスに直接ウォッチ式を入力します（「(c) 新規ウォッチ式の登録」参照）。 なお，1 つのウォッチ パネルに登録可能なウォッチ式数は，最大 128 個までです。
	新規カテゴリ（フォルダ）を追加します。テキスト・ボックスに直接カテゴリ名を入力します。 なお，1 つのウォッチ パネルに作成可能なカテゴリ数は，最大 64 個までです（カテゴリ内のカテゴリ作成は不可）。
	選択している範囲の文字列を削除します。 ウォッチ式／カテゴリが選択状態の場合は，その項目を削除します。 ただし，ウォッチ式の展開項目を選択している場合は無効となります。
表記	値の表示形式を変更する次のボタンを表示します。
自動	選択しているウォッチ式の値の表記を変数ごとの規定値（「表 A.10 ウォッチ式の表示形式（デフォルト）」参照）で表示します（デフォルト）。
16 進数	選択している項目の値を 16 進数で表示します。
符号付き 10 進数	選択している項目の値を符号付き 10 進数で表示します。
符号無し 10 進数	選択している項目の値を符号なし 10 進数で表示します。
8 進数	選択している項目の値を 8 進数で表示します。
2 進数	選択している項目の値を 2 進数で表示します。
ASCII	選択している項目の値を ASCII コードで表示します。
Float	選択している項目の値を Float で表示します。 ただし，選択しているウォッチ式が 4 バイト・データの場合のみ有効となります。
Double	選択している項目の値を Double で表示します。 ただし，選択しているウォッチ式が 8 バイト・データの場合のみ有効となります。
	選択している項目の値表示の末尾に，その値の 16 進数表記を“（）”で囲んで併記します。 ただし，16 進数表記をしている場合は併記しません。

[[ファイル] メニュー（ウォッチ パネル専用部分)]

ウォッチ パネル専用の［ファイル］メニューは次のとおりです（その他の項目は共通)。
ただし，プログラム実行中はすべて無効となります。

ウォッチ・データを保存	このパネルの内容を前回保存したテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存します（「(c) ウォッチ式の値の保存」参照)。 なお，起動後に初めてこの項目を選択した場合は，［名前を付けてウォッチ・データを保存 ...］の選択と同等の動作となります。
名前を付けてウォッチ・データを保存 ...	このパネルの内容を指定したテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存するために，名前を付けて保存 ダイアログをオープンします（「(c) ウォッチ式の値の保存」参照)。

[[編集] メニュー（ウォッチ パネル専用部分)]

ウォッチ パネル専用の［編集］メニューは次のとおりです（その他の項目はすべて無効)。

切り取り	選択範囲の文字列を切り取り，クリップ・ボードにコピーします。 ウォッチ式／カテゴリが選択状態の場合は，その項目を切り取ります。 ただし，ウォッチ式の展開項目を選択している場合は無効となります。
コピー	選択している範囲を文字列としてクリップ・ボードにコピー します。 ウォッチ式／カテゴリが選択状態の場合は，その項目をコピーします。 ただし，ウォッチ式の展開項目を選択している場合は無効となります。
貼り付け	テキストが編集状態の場合，クリップ・ボードの内容をキャレット位置に挿入します。 テキストが編集状態以外の場合で，ウォッチ式がクリップ・ボードにコピーされている場合は，コピーされているウォッチ式をキャレット位置に登録します。
削除	選択している範囲の文字列を削除します。 ウォッチ式／カテゴリが選択状態の場合は，その項目を削除します。 ただし，ウォッチ式の展開項目を選択している場合は無効となります。
すべて選択	テキストが編集状態の場合，すべての文字列を選択します。 テキストが編集状態以外の場合，すべてのウォッチ式／カテゴリを選択状態にします。
名前の変更	選択しているウォッチ式，またはカテゴリの名称を編集します。
検索 ...	検索・置換 ダイアログを ［一括検索］タブが選択状態でオープンします。
置換 ...	検索・置換 ダイアログを ［一括置換］タブが選択状態でオープンします。

［コンテキスト・メニュー］

アクセス・ブレークの設定	この項目は，選択しているウォッチ式がグローバル変数／関数内スタティック変数／ファイル内スタティック変数，および I/O レジスタの場合のみ有効です（複数選択不可）。 アクセス系のブレーク・イベントを設定するために，次のカスケード・メニューを表示します（「2.10.4.1 変数 /I/O レジスタへのブレーク・イベント（アクセス系）を設定する」参照）。
読み込みブレークを設定【シミュレータ】	選択しているウォッチ式に，リード・アクセスのブレーク・イベントを設定します。
書き込みブレークを設定【シミュレータ】	選択しているウォッチ式に，ライト・アクセスのブレーク・イベントを設定します。
読み書きブレークを設定【シミュレータ】	選択しているウォッチ式に，リード／ライト・アクセスのブレーク・イベントを設定します。
読み込み組み合わせブレークを設定【E1】【E20】【EZ Emulator】	選択しているウォッチ式に，リード・アクセスのブレーク・イベントを，組み合わせブレーク・イベントの条件として設定します。
書き込み組み合わせブレークを設定【E1】【E20】【EZ Emulator】	選択しているウォッチ式に，ライト・アクセスのブレーク・イベントを，組み合わせブレーク・イベントの条件として設定します。
読み書き組み合わせブレークを設定【E1】【E20】【EZ Emulator】	選択しているウォッチ式に，リード／ライト・アクセスのブレーク・イベントを，組み合わせブレーク・イベントの条件として設定します。
トレース出力	この項目は，選択しているウォッチ式がグローバル変数／関数内スタティック変数／ファイル内スタティック変数，および I/O レジスタの場合のみ有効です（複数選択不可）。 トレース関連のイベントを設定するために，次のカスケード・メニューを表示します（「2.13.3 任意区間の実行履歴を収集する」および「2.13.4 条件を満たしたときのみの実行履歴を収集する」参照）。 イベント設定後，対象となるウォッチ式の先頭にイベント・マークが表示されます。
値をトレースに記録（読み込み時）	選択しているウォッチ式にリード・アクセスした際に，その値をトレース・メモリに記録するポイント・トレース・イベントを設定します。（「(1) 変数 /I/O レジスタへのアクセスが発生したとき」参照）。
値をトレースに記録（書き込み時）	選択しているウォッチ式にライト・アクセスした際に，その値をトレース・メモリに記録するポイント・トレース・イベントを設定します。（「(1) 変数 /I/O レジスタへのアクセスが発生したとき」参照）。
値をトレースに記録（読み書き時）	選択しているウォッチ式にリード／ライト・アクセスした際に，その値をトレース・メモリに記録するポイント・トレース・イベントを設定します。（「(1) 変数 /I/O レジスタへのアクセスが発生したとき」参照）。
読み書き時にトレース開始の設定	選択しているウォッチ式に対して，リード／ライト・アクセスした際にトレースの記録を開始するトレース・イベントを設定します（「2.13.3.1 トレース開始イベント／トレース終了イベントを設定する」参照）。
読み書き時にトレース終了の設定	選択しているウォッチ式に対して，リード／ライト・アクセスした際にトレースの記録を終了するトレース・イベントを設定します（「2.13.3.1 トレース開始イベント／トレース終了イベントを設定する」参照）。
トレース	トレース パネルをオープンし，取得したトレース・データを表示します。

タイマ設定[注1]	この項目は，選択しているウォッチ式がグローバル変数／関数内スタティック変数／ファイル内スタティック変数，および I/O レジスタの場合のみ有効です（複数選択不可）。 タイマ関連のイベントを設定するために，次のカスケード・メニューを表示します（「2.14.3 任意区間の実行時間を計測する」参照）。 イベント設定後，対象となるウォッチ式の先頭にイベント・マークが表示されます。
読み書き時にタイマ開始	選択しているウォッチ式に対して，リード／ライト・アクセスした際にタイマを開始するイベントを設定します。（「(1) タイマ開始イベントの設定方法」参照）。
タイマ <N> に設定【E1】【E20】【EZ Emulator】	タイマ開始イベントを設定するチャネル[注2]を指定します。
読み書き時にタイマ終了	選択しているウォッチ式に対して，リード／ライト・アクセスした際にタイマを終了するイベントを設定します。（「(2) タイマ終了イベントの設定方法」参照）。
タイマ <N> に設定【E1】【E20】【EZ Emulator】	タイマ終了イベントを設定するチャネル[注2]を指定します。
リアルタイム表示更新設定	リアルタイム表示更新設定のため，次のカスケード・メニューを表示します（「(a) リアルタイム表示更新機能」参照）。
リアルタイム表示更新全体設定	リアルタイム表示更新機能の全般設定を行うため，プロパティ パネルをオープンします。
最新の情報に更新	登録しているウォッチ式のすべての値を再取得し，表示を更新します。 ただし，読み込み保護対象の I/O レジスタの再読み込みは行いません。
値を強制読み込み	読み込み保護対象の I/O レジスタの値を強制的に一度読み込みます。 ただし，プログラム実行中は無効となります。
新規ウォッチ式を追加	新規ウォッチ式を登録します。テキスト・ボックスに直接ウォッチ式を入力します（「(c) 新規ウォッチ式の登録」参照）。 なお，1 つのウォッチ パネルに登録可能なウォッチ式数は，最大 128 個までです。
カテゴリを作成	新規カテゴリ（フォルダ）を追加します。テキスト・ボックスに直接カテゴリ名を入力します。 なお，1 つのウォッチ パネルに作成可能なカテゴリ数は，最大 64 個までです（カテゴリ内のカテゴリ作成は不可）。
削除	選択している範囲の文字列を削除します。 ウォッチ式／カテゴリが選択状態の場合は，その項目を削除します。 ただし，ウォッチ式の展開項目を選択している場合は無効となります。
切り取り	選択している範囲の文字列を切り取ってクリップ・ボードに移動します。 ウォッチ式／カテゴリが選択状態の場合は，その項目を切り取ります。 ただし，ウォッチ式の展開項目を選択している場合は無効となります。
コピー	選択している範囲の文字列をクリップ・ボードにコピーします。 ウォッチ式／カテゴリが選択状態の場合は，その項目をコピーします。
貼り付け	テキストが編集状態の場合，クリップ・ボードの内容をキャレット位置に挿入します。 テキストが編集状態以外の場合で，ウォッチ式がクリップ・ボードにコピーされている場合は，コピーされているウォッチ式をキャレット位置に登録します。 ただし，ウォッチ式の展開項目を選択している場合は無効となります。
名前の変更	選択しているウォッチ式，またはカテゴリの名称を編集します。
ウォッチ式をインポート	ウォッチ式をインポートするために，ウォッチ式データ・ファイルを開く ダイアログをオープンします（「2.11.6.8 ウォッチ式をエクスポート／インポートする」参照）。

表記	表示形式を指定するため，次のカスケード・メニューを表示します。
自動	選択している項目の表記を変数ごとの規定値（「表 A.10 ウォッチ式の表示形式（デフォルト）」参照）で表示します（デフォルト）。
16 進数	選択している項目を 16 進数で表示します。
符号付き 10 進数	選択している項目を符号付き 10 進数で表示します。
符号なし 10 進数	選択している項目を符号なし 10 進数で表示します。
8 進数	選択している項目を 8 進数で表示します。
2 進数	選択している項目を 2 進数で表示します。
ASCII	選択している項目を ASCII コードで表示します。
16 進数値を併記	選択している項目の値表示の末尾に，その値の 16 進数表記を“（）”で囲んで併記します。 ただし，16 進数表記をしている場合は併記しません。
Float	選択している項目を Float で表示します。 ただし，選択しているウォッチ式が 4 バイト・データ以外，または型情報を持つ場合は，規定値（「表 A.10 ウォッチ式の表示形式（デフォルト）」参照）で表示します。
Double	選択している項目を Double で表示します。 ただし，選択しているウォッチ式が 8 バイト・データ以外，または型情報を持つ場合は，規定値（「表 A.10 ウォッチ式の表示形式（デフォルト）」参照）で表示します。
配列のインデックスを 10 進表記	すべての配列のインデックスを 10 進数で表示します。
配列のインデックスを 16 進表記	すべての配列のインデックスを 16 進数で表示します。
エンコード	文字コードを指定するため，次のカスケード・メニューを表示します。
ASCII	選択している項目を ASCII コードで表示します。
Shift_JIS	選択している項目を Shift_JIS コードで表示します（デフォルト）。
EUC-JP	選択している項目を EUC-JP コードで表示します。
UTF-8	選択している項目を UTF-8 コードで表示します。
UTF-16	選択している項目を UTF-16 コードで表示します。
サイズ表記	サイズを指定するため，次のカスケード・メニューを表示します。
1 バイト	選択している項目を 8 ビット・データとして表示します。
2 バイト	選択している項目を 16 ビット・データとして表示します。
4 バイト	選択している項目を 32 ビット・データとして表示します。
8 バイト	選択している項目を 64 ビット・データとして表示します。
メモリへジャンプ	選択しているウォッチ式が定義されているアドレスへキャレットを移動した状態でメモリ パネル（メモリ 1）をオープンします（「(g) メモリ定義アドレスへのジャンプ」参照）。
表示色をリセット	選択しているウォッチ式に対して，プログラムの実行により値が変化したことを示す強調表示をリセットします。 ただし，プログラム実行中は無効となります。

ウォッチ・データを展開保存 ...	選択しているウォッチ式の内容を指定したテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存するために，名前を付けて保存 ダイアログをオープンします（「(c) ウォッチ式の値の保存」参照）。

注 1. 【E1】【E20】【EZ Emulator】
RX100 シリーズでは，タイマ機能は未サポートとなります。

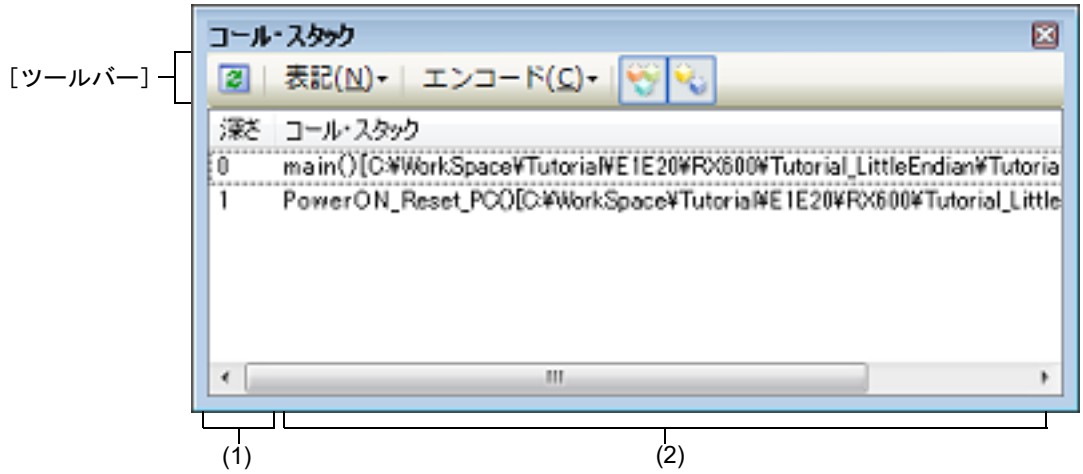
注 2. 指定可能なチャネル数については，以下のとおり RX600 シリーズと RX200 シリーズで異なります。
RX600 シリーズ：2（32bit×2）または 1（64bit×1）
RX200 シリーズ：1（24bit×1）

コール・スタック パネル

関数呼び出しのコール・スタック情報の表示を行います（「2.12.1 コール・スタック情報を表示する」参照）。
なお，このパネルは，デバッグ・ツールと接続時のみオープンすることができます。

注意 プログラム実行中は，このパネルには何も表示されません。
プログラムの実行が停止したタイミングで，各エリアの表示を行います。

図 A.32 コール・スタック パネル



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー（コール・スタック パネル専用部分）]
- [[編集] メニュー（コール・スタック パネル専用部分）]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [コール・スタック] を選択

[各エリアの説明]

(1) [深さ] エリア
呼び出しの深さを表示します。
カレント PC 位置を表示している行を 0 とし，呼び出し元に 1 から順に番号を付与します。

(2) [コール・スタック] エリア
現在のソース位置とスタックに積まれているコール・スタック情報（関数呼び出し元位置／各関数の引数など）を表示します。
ツールバーの ／ ボタン，またはコンテキスト・メニューの [引数表示] ／ [モジュール・ファイル名表示] の選択による状態により，このエリアに表示する表示形式は次のように異なります。

状態	表示形式
- 引数を表示する - モジュール・ファイル名を表示する	*<関数>* (*<引数>* = *<引数値*[注]*>* ,...)[*<モジュール・ファイル名>* $ *<ファイル名>* # *<行番号>*] （デフォルト）

状態	表示形式
- 引数を表示する - モジュール・ファイル名を表示しない	＜*関数*＞ (＜*引数*＞ = ＜*引数値*注＞ ,...)[＜*ファイル名*＞ # ＜*行番号*＞]
- 引数を表示しない - モジュール・ファイル名を表示する	＜*関数*＞ ()[＜*モジュール・ファイル名*＞ $ ＜*ファイル名*＞ # ＜*行番号*＞]
- 引数を表示しない - モジュール・ファイル名を表示しない	＜*関数*＞ ()[＜*ファイル名*＞ # ＜*行番号*＞]

注　引数値が文字列の場合，最大 20 文字まで表示します。

備考　配列の引数は，配列としてではなくポインタとして渡されます（C 言語仕様）。そのため，引数が配列の場合，ポインタ扱いとして表示します。

このエリアは，次の機能を備えています。

(a) ソース行／逆アセンブル行へのジャンプ
コンテキスト・メニューの［ソースへジャンプ］を選択することにより，現在のキャレット位置の関数呼び出し元のソース行にキャレットを移動した状態でエディタ パネルがオープンします（すでにオープンしている場合は，エディタ パネルにジャンプ）。
また，同様に［逆アセンブルへジャンプ］を選択することにより，現在のキャレット位置の関数呼び出し元のアドレスを示す逆アセンブル行にキャレットを移動した状態で逆アセンブル パネル（逆アセンブル 1）がオープンします（すでにオープンしている場合は，逆アセンブル パネル（逆アセンブル 1）にジャンプ）。

備考　行をダブルクリックすることでも，対象ソース行へジャンプすることができます。

(b) コール・スタック情報の保存
［ファイル］メニュー→［名前を付けてコール・スタック・データを保存 ...］を選択することにより，名前を付けて保存 ダイアログをオープンし，このパネルのすべての内容をテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存することができます。
コール・スタック情報の保存方法についての詳細は，「2.12.1.4 コール・スタック情報の表示内容を保存する」を参照してください。

(c) 表示の拡大／縮小
コール・スタック パネルにフォーカスがある状態で，メイン・ウインドウのツールバー上にある拡大率のドロップダウン・リストを変更することにより，コール・スタック パネルの表示を拡大／縮小することができます。
また，［Ctrl］キーとマウス・ホイールを組み合わせることでも変更可能です。

- ［Ctrl］キーを押下しながらマウス・ホイールを前方に動かすと，コール・スタック パネルの表示を拡大し，表示が大きく見やすくなります（最大 300%）。
- ［Ctrl］キーを押下しながらマウス・ホイールを後方に動かすと，コール・スタック パネルの表示を縮小し，表示が小さくなります（最小 50%）。

なお，拡大率を変更した状態のパネルをクローズした場合，拡大率は保持されます（再度パネルをオープンする際に，変更した拡大率でオープンします）。

［ツールバー］

ツールバーの各ボタンは，プログラム実行中は無効となります。

	デバッグ・ツールから最新の情報を取得し，表示を更新します。
表記	値の表示形式を変更する次のボタンを表示します。

自動	このパネル上の値の表記を変数ごとの規定値で表示します（デフォルト）。
16 進数	このパネル上の値を 16 進数で表示します。
10 進数	このパネル上の値を 10 進数で表示します。
8 進数	このパネル上の値を 8 進数で表示します。
2 進数	このパネル上の値を 2 進数で表示します。
エンコード	文字列変数のエンコードを変更する次のボタンを表示します。
ASCII	このパネル上の文字列変数を ASCII コードで表示します（デフォルト）。
Shift_JIS	このパネル上の文字列変数を Shift_JIS コードで表示します。
EUC-JP	このパネル上の文字列変数を EUC-JP コードで表示します。
UTF-8	このパネル上の文字列変数を UTF-8 コードで表示します。
UTF-16	このパネル上の文字列変数を UTF-16 コードで表示します。
	モジュール・ファイル名を付加して表示します（デフォルト）。
	関数呼び出しのパラメータ（引数）を付加して表示します（デフォルト）。

[[ファイル] メニュー（コール・スタック パネル専用部分)]

コール・スタック パネル専用の［ファイル］メニューは次のとおりです（その他の項目は共通）。
ただし，プログラム実行中はすべて無効となります。

コール・スタック・データを保存	このパネルの内容を前回保存したテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存します（「(b) コール・スタック情報の保存」参照）。 なお，起動後に初めてこの項目を選択した場合は，［名前を付けてコール・スタック・データを保存 ...］の選択と同等の動作となります。
名前を付けてコール・スタック・データを保存 ...	このパネルの内容を指定したテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存するために，名前を付けて保存 ダイアログをオープンします（「(b) コール・スタック情報の保存」参照）。

[[編集] メニュー（コール・スタック パネル専用部分)]

コール・スタック パネル専用の［編集］メニューは次のとおりです（その他の項目はすべて無効）。

コピー	選択している行の内容を文字列としてクリップ・ボードにコピーします。
すべて選択	項目をすべて選択状態にします。
検索 ...	検索・置換 ダイアログを［一括検索］タブが選択状態でオープンします。
置換 ...	検索・置換 ダイアログを［一括置換］タブが選択状態でオープンします。

[コンテキスト・メニュー]

コンテキスト・メニューの各項目は，プログラム実行中は無効となります。

コピー	選択している行の内容を文字列としてクリップ・ボードにコピーします。
モジュール・ファイル名表示	モジュール・ファイル名を付加して表示します（デフォルト）。
引数表示	関数呼び出しのパラメータ（引数）を付加して表示します（デフォルト）。

表記	表示形式を指定するために，次のカスケード・メニューを表示します。
自動	このパネル上の値の表記を変数ごとの規定値で表示します（デフォルト）。
16 進数	このパネル上の値を 16 進数で表示します。
10 進数	このパネル上の値を 10 進数で表示します。
8 進数	このパネル上の値を 8 進数で表示します。
2 進数	このパネル上の値を 2 進数で表示します。
エンコード	文字コードを指定するため，次のカスケード・メニューを表示します。
ASCII	文字列変数を ASCII コードで表示します（デフォルト）。
Shift_JIS	文字列変数を Shift_JIS コードで表示します。
EUC-JP	文字列変数を EUC-JP コードで表示します。
UTF-8	文字列変数を UTF-8 コードで表示します。
UTF-16	文字列変数を UTF-16 コードで表示します。
逆アセンブルへジャンプ	選択している行が示す関数呼び出し元のアドレスにキャレットを移動した状態で，逆アセンブル パネル（逆アセンブル 1）がオープンします。
ソースへジャンプ	選択している行が示す関数呼び出し元のソース行にキャレットを移動した状態で，エディタ パネルがオープンします。
このときのローカル変数を表示	選択している行が示す関数のローカル変数を表示するローカル変数 パネルをオープンします。

トレース パネル

プログラムの実行履歴を記録したトレース・データの表示を行います（「2.13 実行履歴の収集」参照）。
トレース・データは，デフォルトで逆アセンブル・テキストとソース・テキストを混合して表示しますが，表示モードを選択することにより，そのどちらか一方のみを表示させることもできます。
プログラムの実行停止後，最新のトレース・データが表示されるよう表示位置を自動更新します。
なお，このパネルは，デバッグ・ツールと接続時のみオープンすることができます。

注意 【E20(JTAG)【RX600 シリーズ】】
トレース機能／リアルタイム RAM モニタ機能（RRM 機能）は一部排他使用の機能です。

備考 パネル上の各エリアの区切り線をダブルクリックすることにより，該当エリアの内容を省略することなく表示可能な最小幅に変更することができます。

図 A.33 トレース パネル【E1/E20/EZ Emulator【RX200, RX600 シリーズ】】

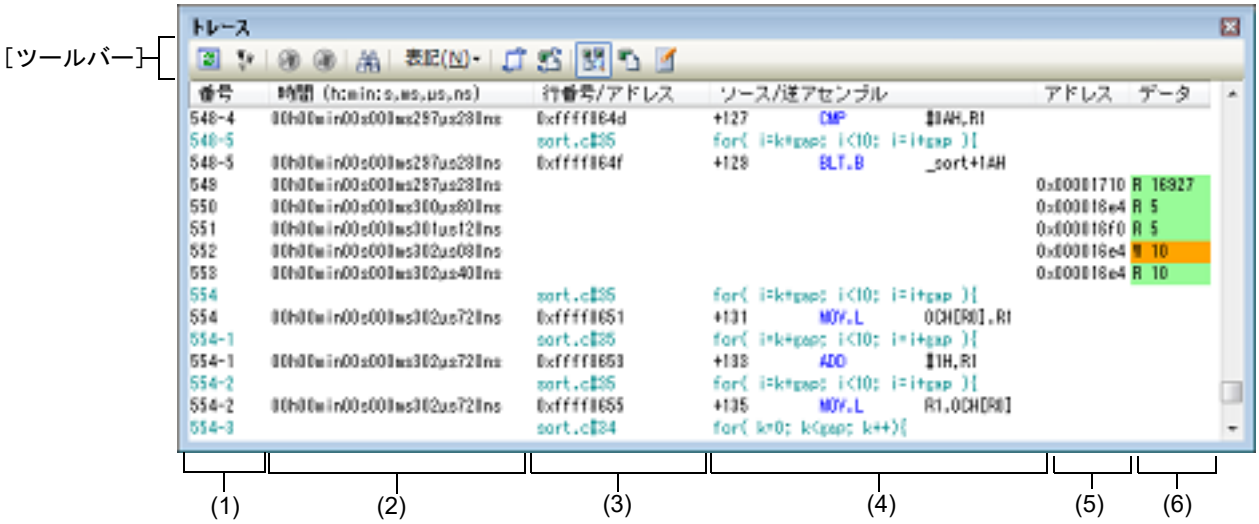


図 A.34 トレース パネル【E1/E20/EZ Emulator【RX100 シリーズ】】

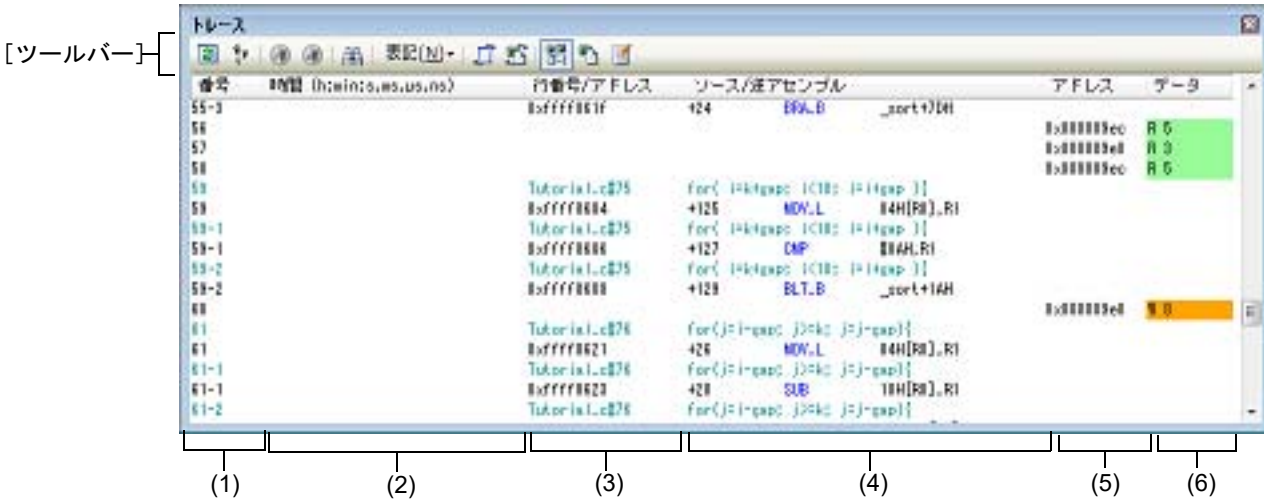
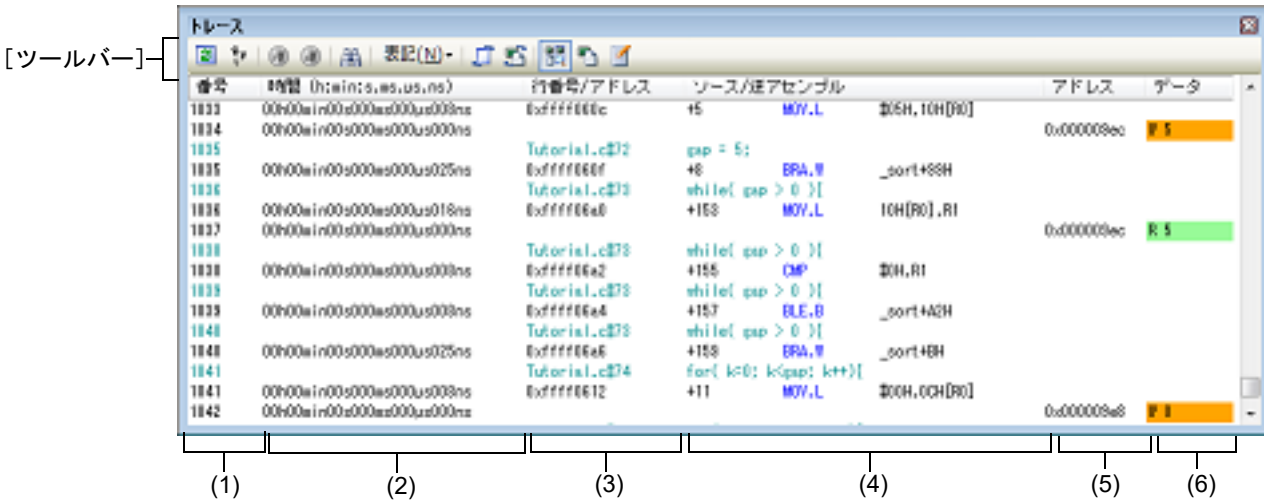


図 A.35 トレース パネル【シミュレータ】



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー（トレース パネル専用部分)]
- [[編集] メニュー（トレース パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [トレース] を選択
- エディタ パネル／逆アセンブル パネルにおいて，コンテキスト・メニューの [トレース設定] → [トレース結果の表示] を選択

[各エリアの説明]

(1) [番号] エリア
トレース・フレームに対応するトレース番号を表示します。

(2) [時間 (h:min:s,ms,µs,ns)] エリア
プログラムの実行開始から，各フレームの命令実行，またはメモリ・アクセスの要因が発生するまでに要した時間を“時間，分，秒，ミリ秒，マイクロ秒，ナノ秒”の単位で表示します。

備考 【シミュレータ】
時間表示を積算値とするか差分値とするかは，プロパティ パネルの [デバッグ・ツール設定] タブ上の [トレース] カテゴリ内 [トレース・タイム・タグを積算する] プロパティの設定に依存します。

注意 1. 【E1】【E20】【EZ Emulator】
タイム・スタンプ情報をトレース・データに付加するか否かは，プロパティ パネルの [デバッグ・ツール設定] タブ上の [トレース] カテゴリ内 [タイム・スタンプ出力] プロパティで指定します（「2.13.1 トレース動作の設定をする」参照)。なお，RX100 シリーズでは [タイム・スタンプ出力] プロパティの項目を変更することはできず，タイム・スタンプ情報をトレース・データに付加することはできません。

注意 2. 【E1】【E20】【EZ Emulator】
タイム・スタンプ情報は，プログラム実行開始からの時間ではなく，先頭サイクルを基点とした相対時間を表示します。

注意 3. 【E1】【E20】【EZ Emulator】
タイム・スタンプ情報は，プロパティ パネルの [ダウンロード・ファイル設定] タブ上の [ト

レース］カテゴリ内［トレース・クロック・カウント・ソース [MHz]］プロパティの設定値とマイクロコントローラごとのカウント・ソースでカウントした値によって算出されます。
マイクロコントローラのカウンタの幅は , 次のとおりです。
RX600 シリーズ : 20 ビット
RX200 シリーズ : 24 ビット
なお , マイクロコントローラのカウント値がオーバフローした場合 , オーバフローの内容を補正したタイム・スタンプ情報が表示されます。
ただし , 連続するトレース・フレーム間で 2 回以上オーバフローが発生した場合 , 以降のトレース・データには正しいタイム・スタンプ情報が表示されません。

注意 4. 【E1(Serial) / E20(Serial) / EZ Emulator【RX200 シリーズ】】
トレース・パネルのタイム・スタンプ情報は , タイマ計測用カウンタを用いています。そのため , イベント パネルにてタイマ計測イベントを設定した場合，期待したタイム・スタンプ情報が表示されません。

注意 5. トレースのタイム・スタンプ情報のカウント周波数 (トレース・クロック・カウント・ソース [MHz]) は，マイクロコントローラにより下記のように異なります。

【RX610, RX621, RX62N, RX62T, RX62G グループ】
(1)EXTAL 周波数 × 8<= 100 MHz の場合
トレース・クロック・カウント・ソース [MHz] ＝ EXTAL 周波数 × 8
(2)EXTAL 周波数 × 8> 100 MHz の場合
トレース・クロック・カウント・ソース [MHz] ＝ EXTAL 周波数 × 4

【RX71M, RX64M, RX630, RX631, RX63N, RX63T グループ】
(1)EXTAL または PLL をシステムクロック (ICLK) のクロックソースに使用する場合 (注)
(a)SCKCR.ICK を 1 分周に設定した場合
トレース・クロック・カウント・ソース [MHz] ＝ 選択クロックソースの 1 分周
(b)SCKCR.ICK を 1 分周以外に設定した場合
トレース・クロック・カウント・ソース [MHz] ＝ 選択クロックソースの 2 分周
(2)"EXTAL または PLL をシステムクロック (ICLK) のクロックソースに使用しない場合
トレース・クロック・カウント・ソース [MHz] ＝ 選択クロックソースの 1 分周

【RX210, RX21A, RX220 グループ】
トレース・クロック・カウント・ソース [MHz] ＝ ICLK（パフォーマンスカウンタを使用）

注 : ユーザプログラム動作中にシステムクロック (ICLK) のクロックソースを切り替えた場合，トレースのタイム・スタンプ情報のカウントソースも切り替わります。この場合，タイム・スタンプ値の変化分に単純に周波数を乗じても絶対時間にはなりません。

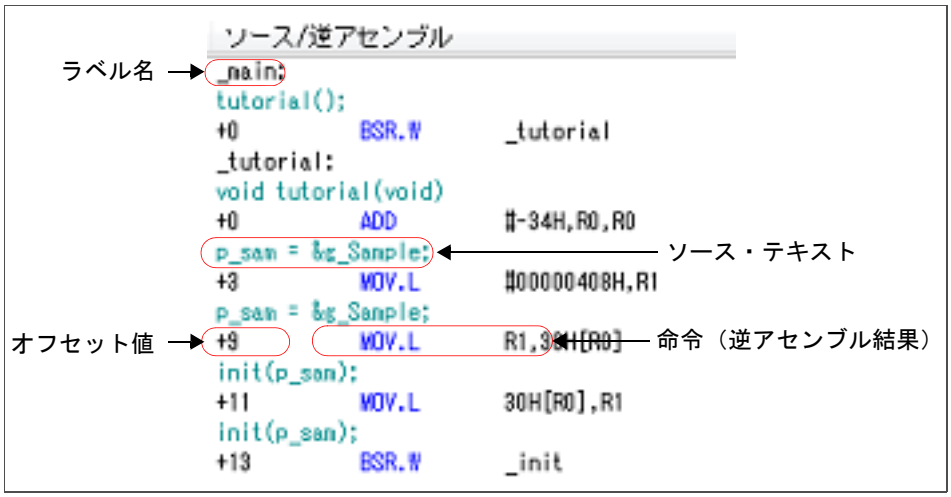
(3) ［行番号 / アドレス］エリア
アセンブル命令のアドレス，またはソース・ファイルの行番号を表示します。
表示進数や文字列のエンコードは，ツールバーのボタン，またはコンテキスト・メニューより選択することができます。
表示形式は次のとおりです。

表示行の種類	表示形式
命令（逆アセンブル）	*＜アドレス＞*
ソース・テキスト	*＜ファイル名＞ # ＜行番号＞*
ラベル	―
ポイント・トレース結果	―

備考 次の実行履歴を表示しないため，行番号は連番にはなりません。
- CPU レジスタ・アクセス
- 無効フェッチ

(4) ［ソース / 逆アセンブル］エリア
収集したトレース・データを次のように表示します。
なお，表示モードの選択により，このエリアに表示される項目は異なります（「(a) 表示モード」参照）。

図 A.36 ［ソース / 逆アセンブル］エリアの表示内容（**デフォルト**）



ラベル名	アドレスにラベルが定義されている場合は，ラベル名を表示します。
オフセット値	アドレスにラベルが定義されていない場合は，一番近いラベルからのオフセット値を表示します。
ソース・テキスト	混合表示モード／ソース表示モードを選択している場合，対応するソース・テキストを表示します。 ただし，デバッグ情報が存在しない箇所を実行した場合は，“ デバッグ情報なし ” と表示します。
命令（逆アセンブル結果）	混合表示モード／逆アセンブル表示モードを選択している場合，対応する命令（逆アセンブル結果）を表示します注。ニモニックは強調表示されます。

注 トレース・データの取りこぼしがあった場合は，“(LOST)” を表示し，該当行全体をエラー色で表示します（エラー色はオプション ダイアログにおける［全般 - フォントと色］カテゴリの設定に依存）。

このエリアは，次の機能を備えています。

(a) 表示モード
ツールバーのボタン，またはコンテキスト・メニューの選択により，次の 3 つの表示モードを選択することができます。

表示モード	表示内容
混合表示モード	命令（逆アセンブル）／ラベル名／ソース・テキスト（対応するソース行）／データアクセス情報を表示します（デフォルト）。
逆アセンブル表示モード	命令（逆アセンブル）／ラベル名／データアクセス情報を表示します。
ソース表示モード	ソース・テキスト（対応するソース行）を表示します。 ただし，デバッグ情報が存在しない箇所を実行した場合は，“ デバッグ情報なし ” と表示します。

(b) ソース行／逆アセンブル行へのジャンプ
コンテキスト・メニューの［ソースへジャンプ］を選択することにより，現在のキャレット位置の行に対応するソース行にキャレットを移動した状態でエディタ パネルがオープンします（すでにオープンしている場合は，エディタ パネルにジャンプ）。
また，同様に［逆アセンブルへジャンプ］を選択することにより，現在のキャレット位置の行のフェッチ・アドレスにキャレットを移動した状態で逆アセンブル パネル（逆アセンブル 1）がオープンします（すでにオープンしている場合は，逆アセンブル パネル（逆アセンブル 1）にジャンプ）。

(c) 他のパネルとの連動
ツールバーの ／ ボタン，またはコンテキスト・メニューの［ウインドウ連動］→［ソースと連動］／［逆アセンブルと連動］を選択することにより，このパネル上のキャレット位置のアドレスをポインタとして，エディタ パネル／逆アセンブル パネルで対応箇所を連動して表示させることができます（フォーカスの移動は行いません）。

(d) ポップアップ表示
マウス・カーソルを行に重ねることにより，その行に対応するすべてのエリア（項目）のデータを縦並びにポップアップ表示します。

(e) トレース・データの保存
［ファイル］メニュー→［名前を付けてトレース・データを保存 ...］を選択することにより，データ保存 ダイアログをオープンし，このパネルの内容をテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存することができます。
トレース・データの保存方法についての詳細は，「2.13.9 実行履歴の表示内容を保存する」を参照してください。

(f) 表示の拡大／縮小
トレース パネルにフォーカスがある状態で，メイン・ウインドウのツールバー上にある拡大率のドロップダウン・リストを変更することにより，トレース パネルの表示を拡大／縮小することができます。
また，［Ctrl］キーとマウス・ホイールを組み合わせることでも変更可能です。

- ［Ctrl］キーを押下しながらマウス・ホイールを前方に動かすと，トレース パネルの表示を拡大し，表示が大きく見やすくなります（最大 300%）。
- ［Ctrl］キーを押下しながらマウス・ホイールを後方に動かすと，トレース パネルの表示を縮小し，表示が小さくなります（最小 50%）。

なお，拡大率を変更した状態のパネルをクローズした場合，拡大率は保持されます（再度パネルをオープンする際に，変更した拡大率でオープンします）。

(5) ［アドレス］エリア
メモリ・アクセスの対象アドレスを表示します。
ただし，I/O レジスタへのアクセスの場合は，アドレスの代わりに I/O レジスタ名を表示します（アクセスが複数ある場合は次の行に表示）。
表示進数は，ツールバーのボタン，またはコンテキスト・メニューより選択することができます。

(6) ［データ］エリア
アクセスしたデータ値，およびその際のアクセス種別を表示します。
ただし，CPU レジスタ・アクセスは表示しません。
表示進数や文字列のエンコードは，ツールバーのボタン，またはコンテキスト・メニューより選択することができます。
データ値，およびアクセス種別の表示形式は次のとおりです（表示の際の文字色／背景色はオプション ダイアログにおける［全般 - フォントと色］カテゴリの設定に依存）。

表示例（デフォルト）			メモリ・アクセス種別
R *データ値*	文字色	標準色	リード・アクセス
	背景色	薄緑	
W *データ値*	文字色	標準色	ライト・アクセス
	背景色	オレンジ	
VECT *データ値* 【シミュレータ】	文字色	標準色	ベクタ・リード・アクセス
	背景色	薄緑	

［ツールバー］

	デバッグ・ツールから最新の情報を取得し，表示を更新します。 ただし，プログラム実行中は無効となります。
	トレース・メモリをクリア（初期化）し，このパネルの表示もクリアします。 ただし，プログラム実行中は無効となります。
	プログラム実行中に停止したトレース機能を再開します。 ただし，プログラム停止中，またはトレーサ動作中は無効となります。
	プログラム実行中にトレース機能を一時的に停止します。 ただし，プログラム停止中，またはトレーサ停止中は無効となります。

	トレース検索 ダイアログをオープンします。
表記	値の表示形式を変更する次のボタンを表示します。 ただし，プログラム実行中は無効となります。
	このパネル上の値を 16 進数で表示します（デフォルト）。
	このパネル上の値を 10 進数で表示します。
	このパネル上の値を 8 進数で表示します。
	このパネル上の値を 2 進数で表示します。
	選択している行に連動してエディタ パネルをスクロールします。
	選択している行に連動して逆アセンブル パネルをスクロールします。
	表示モードを混合表示モードにします（デフォルト）。 ただし，プログラム実行中は無効となります。
	表示モードを逆アセンブル表示モードにします。 ただし，プログラム実行中は無効となります。
	表示モードをソース表示モードにします。 ただし，プログラム実行中は無効となります。

[[ファイル] メニュー（トレース パネル専用部分）]

トレース パネル専用の［ファイル］メニューは次のとおりです（その他の項目は共通）。
ただし，プログラム実行中はすべて無効となります。

トレース・データを保存	トレース・データの内容を前回保存したテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存します（「(e) トレース・データの保存」参照）。 なお，起動後に初めてこの項目を選択した場合は，［名前を付けてトレース・データを保存 ...］の選択と同等の動作となります。
名前を付けてトレース・データを保存 ...	トレース・データの内容を指定したテキスト・ファイル（*.txt）/CSV ファイル（*.csv）に保存するために，データ保存 ダイアログをオープンします（「(e) トレース・データの保存」参照）。

[[編集] メニュー（トレース パネル専用部分）]

トレース パネル専用の［編集］メニューは次のとおりです（その他の項目はすべて無効）。
ただし，プログラム実行中はすべて無効となります。

コピー	選択している行の内容を文字列としてクリップ・ボードにコピーします（複数行選択不可）。
検索 ...	トレース検索 ダイアログをオープンします。

[コンテキスト・メニュー]

トレース・クリア	トレース・メモリをクリア（初期化）し，このパネルの表示もクリアします。 ただし，プログラム実行中は無効となります。
トレース開始	プログラム実行中に停止したトレース機能を再開します。 ただし，プログラム停止中，またはトレーサ動作中は無効となります。
トレース停止	プログラム実行中にトレース機能を一時的に停止します。 ただし，プログラム停止中，またはトレーサ停止中は無効となります。
検索 ...	トレース検索 ダイアログをオープンします。 ただし，プログラム実行中は無効となります。

コピー	選択している行の内容を文字列としてクリップ・ボードにコピーします（複数行選択不可）。 ただし，プログラム実行中は無効となります。
混合表示	表示モードを混合表示モードにします。 ただし，プログラム実行中は無効となります。
逆アセンブル表示	表示モードを逆アセンブル表示モードにします。 ただし，プログラム実行中は無効となります。
ソース表示	表示モードをソース表示モードにします。 ただし，プログラム実行中は無効となります。
表記	表示進数を指定するために，次のカスケード・メニューを表示します。 ただし，プログラム実行中は無効となります。
16 進数	このパネル上の値を 16 進数で表示します（デフォルト）。
10 進数	このパネル上の値を 10 進数で表示します。
8 進数	このパネル上の値を 8 進数で表示します。
2 進数	このパネル上の値を 2 進数で表示します。
ウインドウ連動	他のパネルとの連動を行うために，次のカスケード・メニューを表示します。
ソースと連動	選択している行に連動してエディタ パネルをスクロールします。
逆アセンブルと連動	選択している行に連動して逆アセンブル パネルをスクロールします。
逆アセンブルへジャンプ	選択している行のフェッチ・アドレスにキャレットを移動した状態で，逆アセンブル パネル（逆アセンブル 1）がオープンします。
ソースへジャンプ	選択している行に対応するソース行にキャレットを移動した状態で，エディタ パネルがオープンします。
メモリへジャンプ	選択している行に対応するメモリ値にキャレットを移動した状態で，メモリ パネルがオープンします。

イベント パネル

エディタ パネル／逆アセンブル パネル／ウォッチ パネル上で設定したイベントの詳細情報の表示，設定状態の有効／無効の切り替え，および削除などを行います（「2.17 イベントの管理」参照）。
なお，このパネルは，デバッグ・ツールと接続時のみオープンすることができます。

備考 1. イベントの設定に関しては，「2.17.7 イベント設定に関する留意事項」を参照してください。

備考 2. 解析ツールの関数 パネル／変数 パネルで設定したイベントもこのパネルで管理されます。

備考 3. パネル上の各エリアの区切り線をダブルクリックすることにより，該当エリアの内容を省略することなく表示可能な最小幅に変更することができます。

図 A.37 イベント パネル【E1】【E20】【シミュレータ】

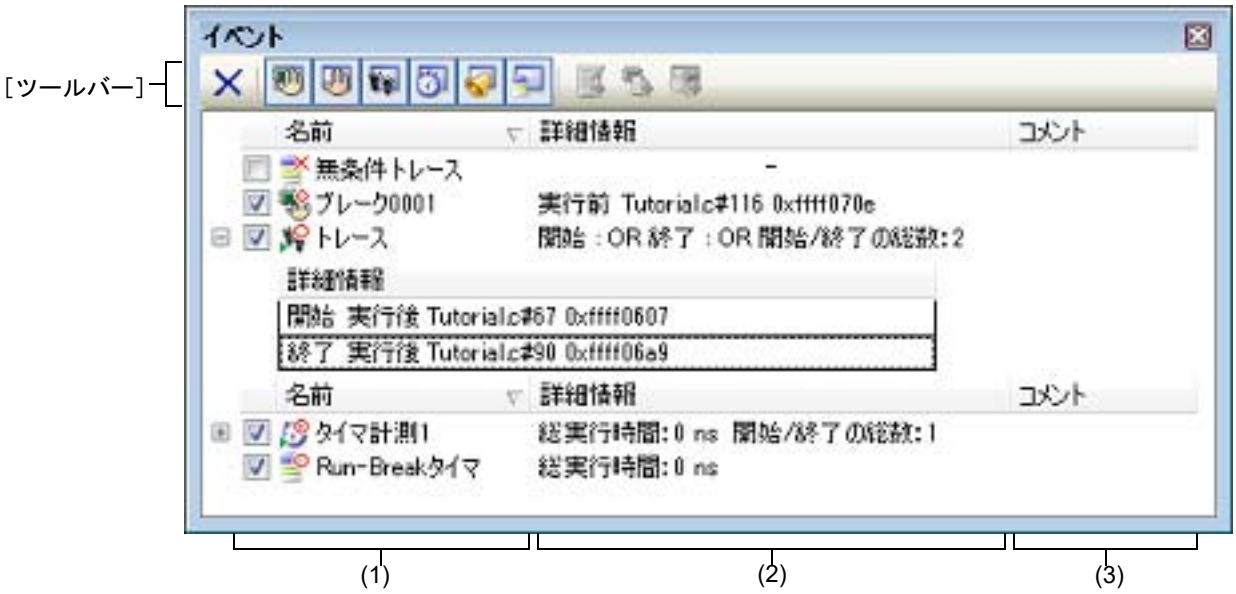
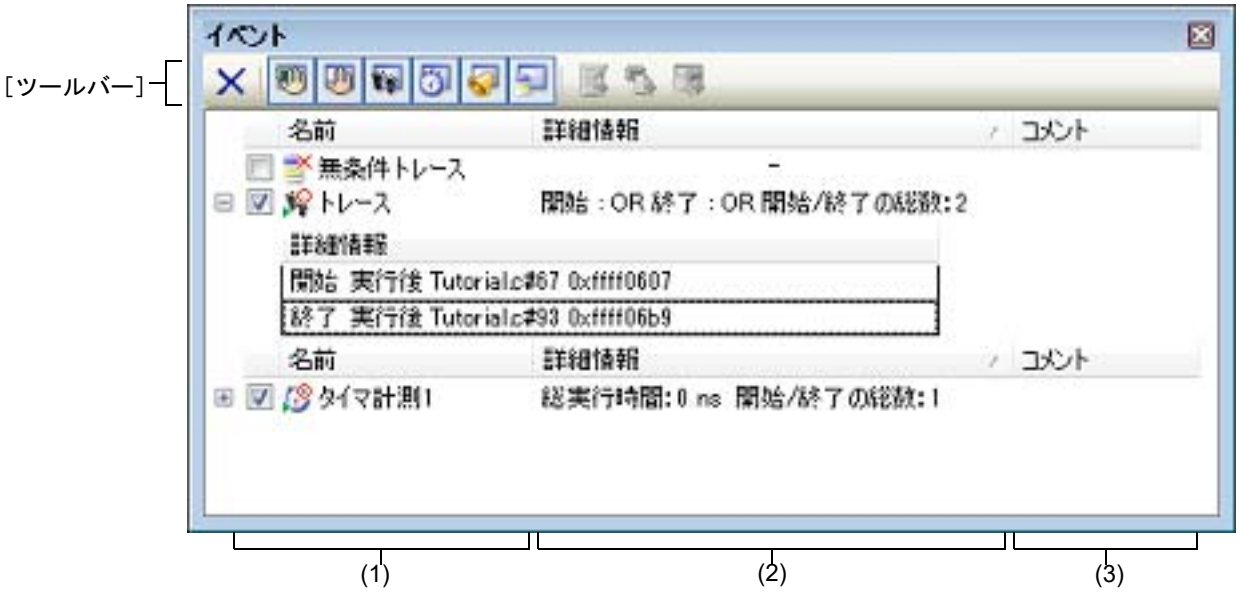


図 A.38 イベント パネル【EZ Emulator】



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］
- ［ツールバー］

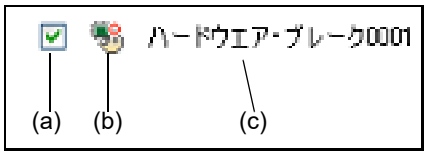
- [[編集] メニュー（イベント パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [イベント] を選択
- エディタ パネル／逆アセンブル パネルにおいて，コンテキスト・メニューの [タイマ設定] → [タイマ結果の表示] を選択
- エディタ パネル／逆アセンブル パネルにおいて，イベント・マークにキャレットを移動したのち，コンテキスト・メニューの [イベント パネルで詳細を表示] を選択

[各エリアの説明]

(1) [名前] エリア
現在設定されているイベント名を次の形式で一覧表示します。



備考 ツールバーのボタンの選択により，表示するイベント種別を限定することができます（「[ツールバー]」参照)。

(a) チェック・ボックス
イベントの設定状態を表示／変更します。
なお，イベントの設定状態を変更すると，対応してイベント・マークも変化します。

	有効状態	指定されている条件の成立で，対象となるイベントが発生します。 チェックを外すことにより，イベントを無効状態にすることができます。
	無効状態	指定されている条件が成立しても，対象となるイベントは発生しません。 チェックすることにより，イベントを有効状態にすることができます。
	保留状態	指定されている条件が，デバッグ対象のプログラムでは設定することができません。 チェック・ボックスを操作することはできません。

備考 1. Run-Break タイマ・イベント（EZ Emulator では未サポート）を無効状態／保留状態にすることはできません。

備考 2. 無条件トレース・イベントとトレース・イベントにおける有効／無効状態の設定は，排他制御となります。このため，ビルトイン・イベントである無条件トレース・イベントは，デフォルトで有効状態で設定されていますが，トレース開始イベント／トレース終了イベントのいずれかが設定されると同時に自動的に無効状態に変更され，トレース・イベント（トレース開始イベント／トレース終了イベントを 1 つにまとめたイベント）が有効状態になります。
また逆に，設定されているトレース・イベントを無効状態にすると，自動的に無条件トレース・イベントが有効状態となります。

注意 【シミュレータ】
タイマ計測イベントを有効状態にするためには，タイマ開始イベントとタイマ終了イベントを両方設定する必要があります。

(b) イベント・マーク
イベント・マークは，イベントの種別を示すとともに，現在の設定状態を示します。
表示されるイベント・マークとその意味は次のとおりです。

表 A.11 イベント・マーク

イベント種別	有効状態	無効状態	保留状態	備考
ハードウエア・ブレーク				–

イベント種別	有効状態	無効状態	保留状態	備考
ソフトウエア・ブレーク 【E1】【E20】【EZ Emulator】				–
組み合わせブレーク 【E1】【E20】【EZ Emulator】				–
無条件トレース			–	–
Run-Break タイマ		–	–	–
トレース				イベント パネルでのみ表示
トレース開始				エディタ パネル／逆アセンブル パネルでのみ表示
トレース終了				
タイマ計測				イベント パネルでのみ表示
タイマ開始				エディタ パネル／逆アセンブル パネルでのみ表示
タイマ終了				
ポイント・トレース				–
Printf イベント				–
割り込みイベント 【シミュレータ】				–
上記イベントの複数設定	注 1	注 2	注 3	エディタ パネル／逆アセンブル パネルでのみ表示

注 1. 複数のイベントの中で，1 つでも有効状態のイベントがある場合。

注 2. 複数のイベントの中で，有効状態のイベントがなく，1 つでも無効状態のイベントがある場合。

注 3. 複数のイベントのすべてが保留状態の場合。

(c) イベント名

イベント名として，イベント種別と ID 番号を表示します。
ID 番号は，イベント種別ごとに 0001 からの番号が自動的に付与されます（一度設定したイベントを削除した場合でも ID 番号の振り直しは行いません）。
表示されるイベント種別は次のとおりです。

表 A.12 イベント種別

イベント種別	説明
ハードウエア・ブレーク （ブレーク注 1）	デバッグ・ツールが，プログラム実行中にブレーク条件を逐次確認し，条件を満たした際にプログラムをブレークさせるイベントです。 →「2.10.2 任意の場所で停止する（ブレークポイント）」参照 →「2.10.4 変数 /I/O レジスタへのアクセスで停止する」参照
ソフトウエア・ブレーク （ブレーク注 1） 【E1】【E20】【EZ Emulator】	ブレークさせるアドレスの命令コードをブレーク用の命令に書き換え，その命令を実行する際にプログラムをブレークさせるイベントです。 →「2.10.2 任意の場所で停止する（ブレークポイント）」参照
組み合わせブレーク 【E1】【E20】【EZ Emulator】	デバッグ・ツールが，プログラム実行中に複数のブレーク条件を逐次確認し，組み合わせ条件を満たした際にプログラムをブレークさせるイベントです。 →「2.10.5 複数のブレーク・イベントを組み合わせる（組み合わせブレーク）【E1】【E20】【EZ Emulator】」参照

イベント種別	説明
無条件トレース	プログラムの実行開始と同時に自動的にトレース・データを収集し，実行停止とともにトレース・データの収集を停止します。 このイベントは，ビルトイン・イベント注 2 であるため，削除することはできません（デフォルトで有効状態で設定されています）。 →「2.13.2 実行停止までの実行履歴を収集する」参照
Run-Break タイマ	プログラムの実行開始と同時に自動的にプログラムの実行時間の計測を開始し，実行停止とともに実行時間の計測を終了します。このイベントは，ビルトイン・イベント注 2 であるため，削除することはできません（デフォルトで有効状態で設定されています）。 →「2.14.2 実行開始から停止までの実行時間を計測する」参照
トレース	トレース開始イベント，およびトレース終了イベントにより設定された条件を満たした際に，トレース・データの収集を開始／終了するイベントです（トレース開始イベント／トレース終了イベントのいずれかが設定されると表示されます）。 →「2.13.3 任意区間の実行履歴を収集する」参照
タイマ計測注 3	タイマ開始イベント，およびタイマ終了イベントにより設定された条件を満たした際に，プログラムの実行時間の計測を開始／終了するイベントです（タイマ開始イベント／タイマ終了イベントのいずれかが設定されると表示されます）。 →「2.14.3 任意区間の実行時間を計測する」参照
ポイント・トレース	プログラムの実行により，指定した変数 /I/O レジスタにアクセスした際に，その情報をトレース・メモリに記録するイベントです。 →「2.13.4 条件を満たしたときのみの実行履歴を収集する」参照
Printf イベント	プログラムの実行を任意の箇所で一瞬停止させたのち，ソフトウエア処理により printf コマンドを実行させるイベントです（アクション・イベント）。 →「2.16.1 printf を挿入する」参照
割り込みイベント【シミュレータ】	プログラム実行中に任意の箇所で割り込みを発生させるイベントです。（アクション・イベント） →「2.16.2 割り込みイベントを挿入する【シミュレータ】」参照

注 1. マウスのワンクリック操作により設定されたブレークポイント（「2.10.2.2 ブレークポイントを設定する」参照）は，“ ブレーク ” と表示します。

注 2. デバッグ・ツールにデフォルトで設定されているイベントです。

注 3. 【E1】【E20】【EZ Emulator】
タイマ計測区間の番号としてイベント種別にチャネル番号が付与されます（例：タイマ計測 1）。タイマ計測可能な区間数は，マイクロコントローラによって異なります。なお，RX100 シリーズでは，タイマ機能は未サポートとなります。

- RX600 シリーズ：2 区間
- RX200 シリーズ：1 区間

【シミュレータ】
タイマ計測可能な区間数は，1 区間です。

備考 上記のイベント種別のほか，解析ツールの関数 パネル／変数 パネルでイベント（ブレークポイント／ブレーク・イベント）を設定した場合，次のイベント種別を表示します。

- 関数へのブレークポイントの場合： “ 関数への先頭へのブレーク ”
- 変数へのブレーク・イベントの場合： “ 変数のアクセス・ブレーク ”

(d) 表示の拡大／縮小
イベント パネルにフォーカスがある状態で，メイン・ウインドウのツールバー上にある拡大率のドロップダウン・リストを変更することにより，イベント パネルの表示を拡大／縮小することができます。
また，［Ctrl］キーとマウス・ホイールを組み合わせることでも変更可能です。

- ［Ctrl］キーを押下しながらマウス・ホイールを前方に動かすと，イベント パネルの表示を拡大し，表示が大きく見やすくなります（最大 300%）。
- ［Ctrl］キーを押下しながらマウス・ホイールを後方に動かすと，イベント パネルの表示を縮小し，表示が小さくなります（最小 50%）。

なお，拡大率を変更した状態のパネルをクローズした場合，拡大率は保持されます（再度パネルをオープンする際に，変更した拡大率でオープンします）。

(2) ［詳細情報］エリア

各イベントに関する詳細情報を表示します。
表示される情報の内容は，イベント種別によって異なります。
イベント種別ごとの詳細情報の見方は次のとおりです。

表 A.13 イベント種別ごとの詳細情報

イベント種別	表示内容[注1]	
ハードウエア・ブレーク（発生条件：実行系）	表示形式 1	*＜発生条件＞ ＜ファイル名＃ 行番号＞ ＜アドレス＞*
	表示例	実行前 main.c#39 0x100
		実行前 ― 0x300
	表示形式 2	*＜発生条件＞ ＜シンボル＋ オフセット＞ ＜アドレス＞*
	表示例	実行前 funcA + 0x10 0x100
		実行前 ― 0x300
ハードウエア・ブレーク（発生条件：アクセス系）	表示形式 1	*＜発生条件＞ ＜ファイル名＃ 変数名＞ ＜アドレス（範囲）＞ ＜比較条件＞ ＜比較値＞*
	表示例	リード main.c#variable1 0x100 - 0x101 == 0x5
		ライト sub.c#variable2 0x200 - 0x200 == 0x7
		リード／ライト sub2.c#variable3 0x300 - 0x303 == 0x8
	表示形式 2	*＜発生条件＞ ＜ファイル名＃ 関数名＃ 変数名＞ ＜アドレス（範囲）＞ ＜比較条件＞ ＜比較値＞*
	表示例	リード main.c#func1#variable1 0x100 - 0x101 == 0x10
	表示形式 3	*＜発生条件＞ ＜変数名＞ ＜アドレス（範囲）＞ ＜比較条件＞ ＜比較値＞*
	表示例	ライト variable1 0x100 - 0x101 == 0x10
ソフトウエア・ブレーク【E1】【E20】【EZ Emulator】	表示形式 1	*＜発生条件＞ ＜ファイル名＃ 行番号＞ ＜アドレス＞*
	表示例	実行前 main.c#40 0x102
		実行前 sub.c#101 0x204
	表示形式 2	*＜発生条件＞ ＜シンボル＋ オフセット＞ ＜アドレス＞*
	表示例	実行前 funcA + 0x12 0x102
組み合わせブレーク（発生条件：実行系，アクセス系）【E1】【E20】【EZ Emulator】	表示形式	＜組み合わせ条件＞ *＜組み合わせブレークの詳細情報＞*
	表示例	OR - 実行後 main.c#100 0x300 - 実行後 funcA + 0x10 0x100 - ライト sub.c#variable2 0x200 - 0x200 == 0x7 - リード／ライト sub2.c#variable3 0x300 - 0x303 == 0x8
無条件トレース	表示形式	―
	表示例	―

イベント種別	表示内容[注1]	
Run-Break タイマ	表示形式	総実行時間：*＜総実行時間＞*
	表示例	総実行時間：1000ms
		総実行時間：OVERFLOW
	表示形式 2	総実行時間：*＜総実行時間＞* 実行サイクル数：*＜総実行サイクル数＞* 実行命令数：*＜総実行命令数＞*【シミュレータ】
	表示例	総実行時間：3300ns 実行サイクル数：330 実行命令数：200
トレース （発生条件：実行系，アクセス系） 【E1】【E20】【EZ Emulator】	表示形式	開始：*＜トレース開始の組み合わせ条件＞* 終了：*＜トレース終了の組み合わせ条件＞* 開始／終了の総数：*＜トレース開始／トレース終了イベントの総数＞*[注2] *＜開始／終了＞ ＜トレース開始／トレース終了の詳細情報＞*
	表示例	開始：OR 終了：OR 開始／終了の総数：6 - 開始　実行後　main.c#100　0x300 - 開始　実行後　funcA + 0x100　0x300 - 開始　ライト　variable1　0x100 - 0x101　==　0x10 - 終了　実行後　main.c#200　0x100 - 終了　実行後　funcA + 0x10　0x100 - 終了　リード　main.c#variable1　0x100 - 0x101　==　0x5
トレース （発生条件：実行系，アクセス系） 【シミュレータ】	表示形式	開始：OR 終了：OR 開始／終了の総数：*＜トレース開始／トレース終了イベントの総数＞*[注2] *＜開始／終了＞ ＜トレース開始／トレース終了の詳細情報＞*
	表示例	開始：OR 終了：OR 開始／終了の総数：6 - 開始　実行前　main.c#100　0x300 - 開始　実行前　funcA + 0x100　0x300 - 開始　ライト　variable1　0x100 - 0x101　==　0x10 - 終了　実行前　main.c#200　0x100 - 終了　実行前　funcA + 0x10　0x100 - 終了　リード　main.c#variable1　0x100 - 0x101　==　0x5
タイマ計測 （発生条件：実行系，アクセス系） 【E1】【E20】【EZ Emulator】	表示形式	総実行時間：*＜総実行時間＞* 開始／終了の総数：*＜タイマ開始／タイマ終了イベントの総数＞*[注2] - *＜総実行時間＞ ＜パス・カウント＞* - *＜開始／終了＞ ＜タイマ開始／タイマ終了の詳細情報＞*
	表示例	総実行時間：10ms　開始／終了の総数：6 - 総実行時間：10ms　パス・カウント：5 - 開始　実行後　main.c#100　0x300 - 開始　実行後　funcA + 0x30　0x100 - 開始　ライト　variable1　0x100 - 0x101　==　0x10 - 終了　実行後　main.c#100　0x300 - 終了　実行後　funcA + 0x50　0x100 - 終了　リード　main.c#variable1　0x100 - 0x101　==　0x5

イベント種別	表示内容[注1]	
タイマ計測 （発生条件：実行系，アクセス系） 【シミュレータ】	表示形式	総実行時間：＜*総実行時間*＞ 開始／終了の総数：＜*タイマ開始／タイマ終了イベントの総数*＞[注2] - ＜*総実行時間*＞ ＜*パス・カウント*＞ ＜*平均実行時間*＞ ＜*最大行時間*＞ ＜*最小実行時間*＞ - ＜*開始／終了*＞ ＜*タイマ開始／タイマ終了の詳細情報*＞
	表示例	総実行時間：10ms 開始／終了の総数：2 - 総実行時間：10ms パス・カウント：5 平均実行時間：2ms 最大実行時間：4ms 最小実行時間：1ms - 開始 ライト variable1 0x100 - 0x101 == 0x10 - 終了 実行前 main.c#100 0x300
ポイント・トレース （発生条件：アクセス系）	表示形式 1	＜*発生条件*＞ ＜*変数名*＞ ＜*変数のアドレス*＞
	表示例	リード variable1 0x100
	表示形式 2	＜*発生条件*＞ ＜*ファイル名＃変数名*＞ ＜*変数のアドレス*＞
	表示例	ライト sub.c#variable2 0x200
	表示形式 3	＜*発生条件*＞ ＜*ファイル名＃関数名＃変数名*＞ ＜*変数のアドレス*＞
	表示例	リード／ライト sub.c#func1#variable3 0x300
Printf イベント （アクション・イベント）	表示形式	＜*発生条件*＞ ＜*ファイル名＃行番号*＞ ＜*アドレス*＞ ＜*Print イベントの設定*＞
	表示例	実行前 main.c#39 0x100 aaa, bbb, ccc
		実行後 sub.c#100 0x200 aaa の結果の表示：aaa
割り込みイベント （アクション・イベント） 【シミュレータ】	表示形式	＜*発生条件*＞ ＜*ファイル名＃行番号*＞ ＜*アドレス*＞ 割り込みベクタ：＜*割り込みベクタ*＞ 優先順位：＜*割り込み優先順位*＞
	表示例	実行前 main.c#39 0x100 割り込みベクタ：1c 優先順位：7

注 1. 表示形式の詳細は次のとおりです。

＜*発生条件*＞	次の条件のいずれか 1 つを表示します。 【E1】【E20】 実行系： 実行前，実行後 アクセス系： リード，ライト，リード／ライト 【シミュレータ】 実行系： 実行前 アクセス系： リード，ライト，リード／ライト
＜*ファイル名＃行番号*＞	ソース・ファイル名とソース・ファイル中の行番号を表示します。表示形式はウォッチ式のスコープ指定式と同等です。 複数のロード・モジュール・ファイルをダウンロードしている場合では，＜*ロード・モジュール名*＄*ファイル名＃行番号*＞を表示します。 なお，逆アセンブル パネルで設定されたイベントでは，次の場合，＜*行番号*＞を＜*シンボル＋オフセット*＞形式で表示します。 - 行情報があり，指定されたイベント設定位置が行情報の先頭でない場合 - 行情報がなく，シンボル情報がある場合 また，次の場合は，*行番号を*“ー”で表示します。 - 行情報がなく，シンボル情報がない場合

＜変数名＞	ソース・ファイル中の変数名を表示します。表示形式はウォッチ式のスコープ指定式と同等です。
＜比較条件＞	比較の条件（==）を表示します。比較値が指定されなかった場合は表示しません。
＜比較値＞	比較値を表示します。比較値が指定されなかった場合は表示しません。
＜アドレス＞	指定された変数の，メモリ領域中の開始アドレス - 終了アドレスを表示します（16 進数表記固定）。
＜組み合わせ条件＞	次の条件のいずれか 1 つを表示します。 OR，AND，シーケンシャル
＜組み合わせブレークの詳細情報＞	組み合わせブレーク・イベントに関する詳細情報を表示します。
＜総実行時間＞	タイマの総実行時間の測定結果を表示します。 単位は，ns/μs,/ms/s/min のいずれか 1 つが表示されます（ただし，“min” の場合は "s" も同時に表示）。 なお，タイマ・オーバフロー発生時（「2.14.4 測定可能時間の範囲」参照），または不正な値の場合は “OVERFLOW” を表示します。
＜トレース開始の組み合わせ条件＞	次の条件のいずれか 1 つを表示します。 OR，AND，シーケンシャル
＜トレース終了の組み合わせ条件＞	OR 条件を表示します。
＜トレース開始／トレース終了イベントの総数＞	トレース開始イベントおよびトレース終了イベントの総数を表示します。
＜開始／終了＞	詳細情報の内容が，開始イベントか終了イベントかを表示します。
＜トレース開始／トレース終了の詳細情報＞	トレース開始イベントおよびトレース終了イベントに関する詳細情報を表示します。
＜パス・カウント＞	タイマのパス・カウントを表示します。 なお，タイマ・オーバフロー発生時（「2.14.4 測定可能時間の範囲」参照），または不正な値の場合は “OVERFLOW” を表示します。
＜平均実行時間＞	タイマの平均実行時間の測定結果を表示します。 単位は，ns/μs,/ms/s/min のいずれか 1 つが表示されます（ただし，“min” の場合は "s" も同時に表示）。
＜最大実行時間＞	タイマの最大実行時間の測定結果を表示します。 単位は，ns/μs,/ms/s/min のいずれか 1 つが表示されます（ただし，“min” の場合は "s" も同時に表示）。
＜最小実行時間＞	タイマの最小実行時間の測定結果を表示します。 単位は，ns/μs,/ms/s/min のいずれか 1 つが表示されます（ただし，“min” の場合は "s" も同時に表示）。
＜総実行サイクル数＞	Run-Break 間の総実行サイクル数の測定結果を表示します。
＜総実行命令数＞	Run-Break 間の総実行命令数の測定結果を表示します。
＜ Print イベントの設定＞	アクション・イベント ダイアログ上で指定した，*＜出力文字列＞*：*＜変数式＞* を表示します。
＜割り込みベクタ＞	アクション・イベント ダイアログ，または割り込みイベント詳細設定ダイアログ【シミュレータ】上で指定した割り込みベクタを表示します。

＜割り込み優先順位＞	アクション・イベント ダイアログ，または割り込みイベント詳細設定ダイアログ【シミュレータ】上で指定した優先順位を表示します。

注 2. この行をクリックすることにより，下行の詳細情報を表示します。

(3) ［コメント］エリア
設定されている各イベントに対して，ユーザが自由にコメントを入力できるエリアです。
コメントの入力は，コメントを入力したいイベントを選択後，このエリアをクリックするか，またはコンテキスト・メニューの［コメントの編集］を選択したのち，任意のテキストをキーボードから直接入力します（[Esc] キーの押下で編集モードをキャンセルします）。
コメントを編集したのち，[Enter] キーの押下，または編集領域以外へのフォーカスの移動により，編集を完了します。
なお，コメントは最大 256 文字まで入力することができ，使用中のユーザの設定として保存されます。

［ツールバー］

	選択しているイベント，およびイベント条件を削除します。 ただし，ビルトイン・イベント（無条件トレース・イベント /Run-Break タイマ・イベント）を削除することはできません。
	ハードウエア・ブレーク関連のイベントを表示します（デフォルト）。
【E1】【E20】【EZ Emulator】	ソフトウエア・ブレーク関連のイベントを表示します（デフォルト）。
	トレース関連のイベントを表示します（デフォルト）。
	タイマ関連のイベントを表示します（デフォルト）。
	アクション・イベント関連（Printf イベント，割り込みイベント）を表示します（デフォルト）。
	ビルトイン・イベント関連（無条件トレース・イベント /Run-Break タイマ・イベント）を表示します（デフォルト）。
	選択しているイベント[注]が設定されているアドレスに対応するソース行にキャレットを移動した状態で，エディタ パネルがオープンします。
	選択しているイベント[注]が設定されているアドレスに対応する逆アセンブル結果にキャレットを移動した状態で，逆アセンブル パネル（逆アセンブル 1）がオープンします。
	選択しているイベント[注]が設定されているアドレスに対応するメモリ値にキャレットを移動した状態で，メモリ パネル（メモリ 1）がオープンします。

注 トレース・イベント／タイマ計測イベント／ビルトイン・イベント（無条件トレース・イベント /Run-Break タイマ・イベント）以外のイベントが対象となります。

［［編集］メニュー（イベント パネル専用部分）］

イベント パネル専用の［編集］メニューは次のとおりです（その他の項目はすべて無効）。

削除	選択しているイベントを削除します。 ただし，ビルトイン・イベント（無条件トレース・イベント /Run-Break タイマ・イベント）を削除することはできません。
すべて選択	このパネルに表示されているすべてのイベントを選択状態にします。
検索 ...	検索・置換 ダイアログを［一括検索］タブが選択状態でオープンします。
置換 ...	検索・置換 ダイアログを［一括置換］タブが選択状態でオープンします。

［コンテキスト・メニュー］

有効化	選択しているイベントを有効状態にします。 ただし，選択しているイベントがすでに有効状態の場合は無効となります。
無効化	選択しているイベントを無効状態にします。 ただし，選択しているイベントがすでに無効状態の場合は無効となります。
削除	選択しているイベント，およびイベント条件を削除します。 ただし，ビルトイン・イベント（無条件トレース・イベント /Run-Break タイマ・イベント）を削除することはできません。
すべて選択	現在表示しているすべてのイベントを選択状態にします。
表示選択	表示するイベント種別を限定するために，次のカスケード・メニューを表示します。 デフォルトでは，すべての項目が選択されています。
ハードウエア・ブレーク	ハードウエア・ブレーク関連のイベントを表示します。
ソフトウエア・ブレーク	ソフトウエア・ブレーク関連のイベントを表示します。
タイマ	タイマ関連のイベントを表示します。
トレース	トレース関連のイベントを表示します。
アクション・イベント	アクション・イベント（Printf イベント）を表示します。
ビルトイン・イベント	ビルトイン・イベント（無条件トレース・イベント /Run-Break タイマ）を表示します。
タイマ設定	タイマ関連の設定をするために，次のカスケード・メニューを表示します。 ただし，タイマ関連のイベントを選択している場合のみ有効です。
タイマの初期化	選択しているイベント（Run-Break タイマ・イベントを除く）で使用するタイマを初期化します。
ナノ秒表示	選択しているイベントのタイマ結果をナノ秒（ns）単位で表示します。
マイクロ秒表示	選択しているイベントのタイマ結果をマイクロ秒（μs）単位で表示します。
ミリ秒表示	選択しているイベントのタイマ結果をミリ秒（ms）単位で表示します。
秒表示	選択しているイベントのタイマ結果を秒（s）単位で表示します。
分表示	選択しているイベントのタイマ結果を分（min）単位で表示します。
メモリへジャンプ	選択しているイベント[注1]が設定されているアドレスに対応するメモリ値にキャレットを移動した状態で，メモリ パネル（メモリ 1）がオープンします。
逆アセンブルへジャンプ	選択しているイベント[注1]が設定されているアドレスに対応する逆アセンブル結果にキャレットを移動した状態で，逆アセンブル パネル（逆アセンブル 1）がオープンします。
ソースへジャンプ	選択しているイベント[注1]が設定されているアドレスに対応するソース行にキャレットを移動した状態で，エディタ パネルがオープンします。
条件の編集 ...	選択しているイベント[注2]に対応した詳細設定ダイアログがオープンします。 トレース・イベントおよび組み合わせブレーク・イベントを選択した場合，組み合わせ条件設定ダイアログ【E1】【E20】【EZ Emulator】がオープンします。 Printf イベントを選択した場合は，アクション・イベント ダイアログをオープンします。
コメントの編集	選択しているイベントのコメントを編集モードにします。 すでにコメントが存在する場合は，その文字列のすべてを選択状態にします。

注 1. トレース・イベント／タイマ計測イベント／ビルトイン・イベント（無条件トレース・イベント /Run-Break タイマ・イベント）以外のイベントが対象となります。

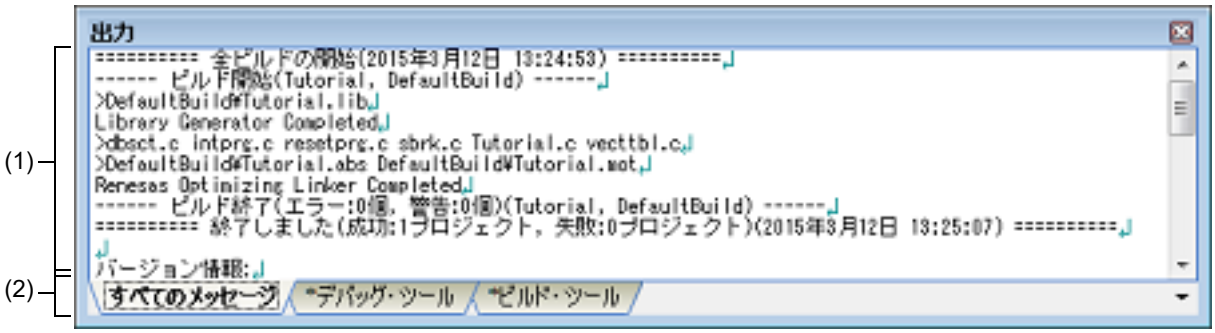
注 2. 次のイベントが対象となります。

- ハードウエア・ブレーク／実行系イベント／アクセス系イベント／タイマ計測イベント【E1】【E20】

出力 パネル

CS+ が提供している各種コンポーネント（デバッグ・ツールを含む，設計ツール／ビルド・ツール／解析ツールなど）から出力されるメッセージの表示，または検索・置換 ダイアログによる一括検索を行った際の結果，および Printf イベント（「2.16.1 printf を挿入する」参照）による出力結果の表示を行います。
メッセージは，出力元のツールごとに分類されたタブ上でそれぞれ個別に表示されます。

図 A.39 出力 パネル



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］
- ［［ファイル］メニュー（出力 パネル専用部分）］
- ［［編集］メニュー（出力 パネル専用部分）］
- ［コンテキスト・メニュー］

［オープン方法］

- ［表示］メニュー→［出力］を選択

［各エリアの説明］

(1) メッセージ・エリア
各ツールから出力されたメッセージ，検索結果，および Printf イベントによる出力結果を表示します。
検索結果（一括検索）の表示では，検索を行うごとに，以前のメッセージをクリアしたのち新しいメッセージを表示します（［すべてのメッセージ］タブを除く）。
なお，メッセージの表示色は，出力メッセージの種別により，次のように異なります（表示の際の文字色／背景色はオプション ダイアログにおける［全般 - フォントと色］カテゴリの設定に依存）。

メッセージ種別	表示例（デフォルト）			説明
通常メッセージ	AaBbCc	文字色	黒	何らかの情報を通知する際に表示されます。
		背景色	白	
警告メッセージ	AaBbCc	文字色	青	操作に対して，何らかの警告を通知する際に表示されます。
		背景色	標準色	
エラー・メッセージ	AaBbCc	文字色	赤	致命的なエラー，または操作ミスにより実行が不可能な場合に表示されます。
		背景色	薄グレー	

このエリアは，次の機能を備えています。

(a) タグ・ジャンプ
出力されたメッセージをダブルクリック，またはメッセージにキャレットを移動したのち［Enter］キーを押下することにより，エディタ パネルをオープンして該当ファイルの該当行番号を表示します。
これにより，ビルド時に出力されたエラー・メッセージなどから，ソース・ファイルの該当するエラー行へジャンプすることができます。

(b) ヘルプの表示
警告メッセージ，またはエラー・メッセージを表示している行にキャレットがある状態で，コンテキスト・メニューの［メッセージに関するヘルプ］を選択するか，または［F1］キーを押下することにより，その行のメッセージに関するヘルプを表示します。

(c) ログの保存
［ファイル］メニュー→［名前を付けて出力 - *タブ名*を保存 ...］を選択することにより，名前を付けて保存 ダイアログをオープンし，現在選択しているタブ上に表示されている全内容をテキスト・ファイル（*.txt）に保存することができます（非選択状態のタブ上のメッセージは保存の対象となりません）。

(d) 表示の拡大／縮小
出力 パネルにフォーカスがある状態で，メイン・ウインドウのツールバー上にある拡大率のドロップダウン・リストを変更することにより，出力 パネルの表示を拡大／縮小することができます。
また，［Ctrl］キーとマウス・ホイールを組み合わせることでも変更可能です。

- ［Ctrl］キーを押下しながらマウス・ホイールを前方に動かすと，出力 パネルの表示を拡大し，表示が大きく見やすくなります（最大 300%）。
- ［Ctrl］キーを押下しながらマウス・ホイールを後方に動かすと，出力 パネルの表示を縮小し，表示が小さくなります（最小 25%）。

なお，拡大率を変更した状態のパネルをクローズした場合，拡大率は保持されます（再度パネルをオープンする際に，変更した拡大率でオープンします）。

(2) タブ選択エリア
メッセージの出力元を示すタブを選択します。
デバッグ・ツールでは，次のタブを使用します。

タブ名	説明
すべてのメッセージ	CS+ が提供している全コンポーネント（デバッグ・ツールを含む，設計ツール／ビルド・ツール／解析ツールなど）から出力されるメッセージを表示します（ラピッド・ビルドの実行によるメッセージを除く）。
デバッグ・ツール	CS+ が提供している各種コンポーネント（デバッグ・ツールを含む，設計ツール／ビルド・ツール／解析ツールなど）から出力されるメッセージのうち，デバッグ・ツールが出力するメッセージを表示します。
検索・置換	検索・置換 ダイアログによる一括検索結果を表示します。

注意 新たなメッセージが非選択状態のタブ上に出力されても，自動的なタブの表示切り替えは行いません。この場合，タブ名の先頭に “*” マークが付加し，新たなメッセージが出力されていることを示します。

［［ファイル］メニュー（出力 パネル専用部分）］

出力 パネル専用の［ファイル］メニューは次のとおりです（その他の項目は共通）。
ただし，プログラム実行中はすべて無効となります。

出力 - *タブ名*を保存	現在選択しているタブ上に表示されている内容を，前回保存したテキスト・ファイル（*.txt）に保存します（「(c) ログの保存」参照）。 なお，起動後に初めてこの項目を選択した場合は，［名前を付けて *タブ名*を保存 ...］の選択と同等の動作となります。 ただし，ビルド実行中は無効となります。
名前を付けて出力 - *タブ名*を保存 ...	現在選択しているタブ上に表示されている内容を，指定したテキスト・ファイル（*.txt）に保存するために，名前を付けて保存 ダイアログをオープンします（「(c) ログの保存」参照）。

[[編集] メニュー（出力 パネル専用部分)]

出力 パネル専用の［編集］メニューは次のとおりです（その他の項目はすべて無効)。

コピー	選択している文字列をクリップ・ボードにコピーします。
すべて選択	現在選択しているタブ上に表示されているすべてのメッセージを選択状態にします。
検索 ...	検索・置換 ダイアログを［クイック検索］タブが選択状態でオープンします。
置換 ...	検索・置換 ダイアログを ［一括置換］タブが選択状態でオープンします。

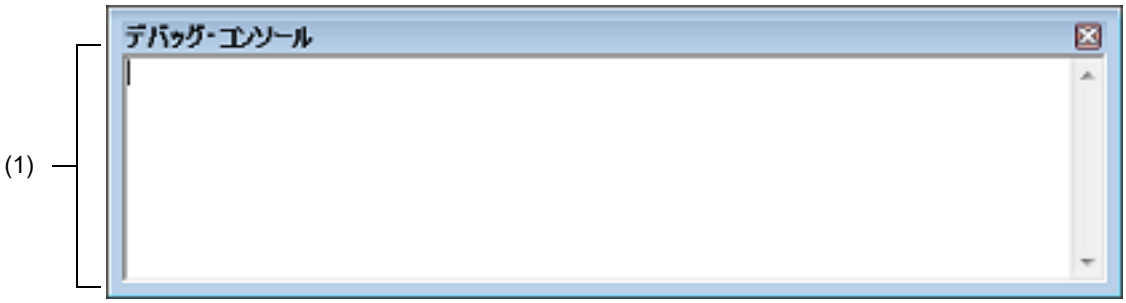
[コンテキスト・メニュー]

コピー	選択している文字列をクリップ・ボードにコピーします。
すべて選択	現在選択しているタブ上に表示されているすべてのメッセージを選択状態にします。
クリア	現在選択しているタブ上に表示されているすべてのメッセージを消去します。
タグ・ジャンプ	エディタ パネルをオープンし，キャレット位置のメッセージに該当するファイルの該当行番号にジャンプします。
検索の中止	現在実行中の検索を中止します。 ただし，検索を実行していない場合は無効となります。
メッセージに関するヘルプ	現在のキャレット位置のメッセージに関するヘルプを表示します。 ただし，警告メッセージ／エラー・メッセージのみが対象となります。

デバッグ・コンソール パネル

コンソールとプログラム間で，データのやり取りを行います。標準ライブラリ関数を実装したプログラムを実行することにより，実現しています。
なお，このパネルは，デバッグ・ツールと接続時のみオープンすることができます。

図 A.40 デバッグ・コンソール パネル



ここでは，以下の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］
- ［［編集］メニュー（デバッグ・コンソール パネル専用部分）］
- ［コンテキスト・メニュー］

［オープン方法］

- ［表示］メニュー→［デバッグ・コンソール］の選択

［各エリアの説明］

(1) 入出力エリア
プログラム内に実装した標準ライブラリ関数，例えば，scanf 関数によるキーボード入力データの読み込み，printf 関数によるデータの出力を行うことができます。
また，パネル上での COM ポートの指定により，対象の COM ポートへプログラムの標準入出力をリダイレクトできます。
この機能を使用するには，プログラム内にデバッグ・ツールが提供する低水準インタフェース・ルーチンの実装が必要です（「2.19 デバッグ・コンソールの使用」参照）。

備考 【シミュレータ】
シミュレータが提供する入出力機能の詳細については，「B. 入出力機能」を参照してください。

(a) 表示の拡大／縮小
デバッグ・コンソール パネルにフォーカスがある状態で，メイン・ウインドウのツールバー上にある拡大率のドロップダウン・リストを変更することにより，デバッグ・コンソール パネルの表示を拡大／縮小することができます。
また，［Ctrl］キーとマウス・ホイールを組み合わせることでも変更可能です。

- ［Ctrl］キーを押下しながらマウス・ホイールを前方に動かすと，デバッグ・コンソール パネルの表示を拡大し，表示が大きく見やすくなります（最大 300%）。
- ［Ctrl］キーを押下しながらマウス・ホイールを後方に動かすと，デバッグ・コンソール パネルの表示を縮小し，表示が小さくなります（最小 25%）。

なお，拡大率を変更した状態のパネルをクローズした場合，拡大率は保持されます（再度パネルをオープンする際に，変更した拡大率でオープンします）。

[[編集] メニュー（デバッグ・コンソール パネル専用部分)]

デバッグ・コンソール パネル専用の［編集］メニューは以下のとおりです（その他の項目は共通です)。

コピー	選択している文字列をクリップ・ボードにコピーします。
貼り付け	クリップ・ボードの内容をキャレット位置に挿入します。
すべて選択	このパネルに表示しているすべての文字列を選択状態にします。

[コンテキスト・メニュー]

コピー	選択している文字列をクリップ・ボードにコピーします。
貼り付け	クリップ・ボードの内容をキャレット位置に挿入します。
クリア	このパネルの表示をクリアします。
デバッグ・コンソール有効 / 無効	デバッグ・コンソール機能の有効（デフォルト）／無効を選択します（トグル)。 なお，デバッグ・コンソール機能を無効化にすると，パネルの背景色がグレーになります。
COM ポートの設定 ...	マイクロコントローラからの通信をリダイレクトするホスト・マシン上の COM ポートを設定するために，COM ポート設定 ダイアログをオープンします。
ログ・ファイル指定 ...	このパネルに表示している内容を指定したログ・ファイル（*.log）に保存するために，ログ・ファイルの指定 ダイアログをオープンします。 ロギングを開始します。
ロギング開始 / 停止	ロギングの開始（デフォルト）／停止を選択します（トグル)。 ただし，ログ・ファイルを指定していない場合，この項目は無効になります。
ローカル・エコー・バック有効 / 無効	ローカル・エコー・バックの有効（デフォルト）／無効を選択します（トグル)。 ローカル・エコー・バックが無効の場合，入力したデータはこの パネル上に出力しません。

メモリ・マッピング ダイアログ

メモリ・マッピングの設定をメモリ種別ごとに行います。

図 A.41 メモリ・マッピング ダイアログ【E1】【E20】【EZ Emulator】

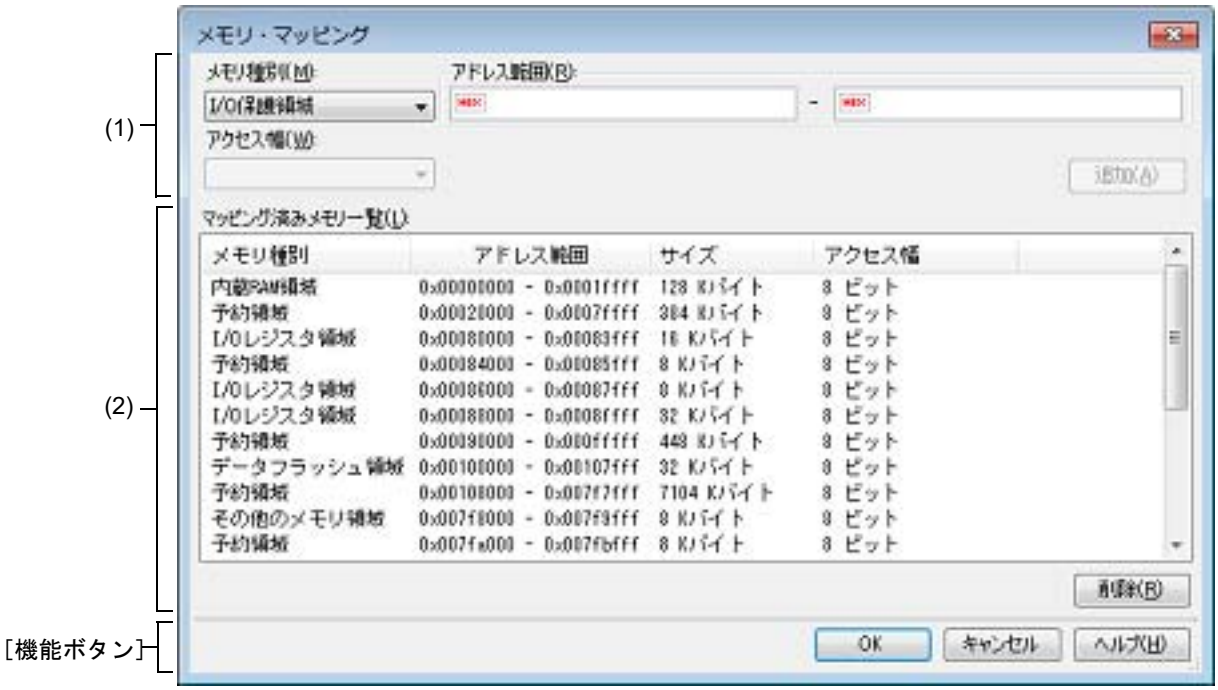
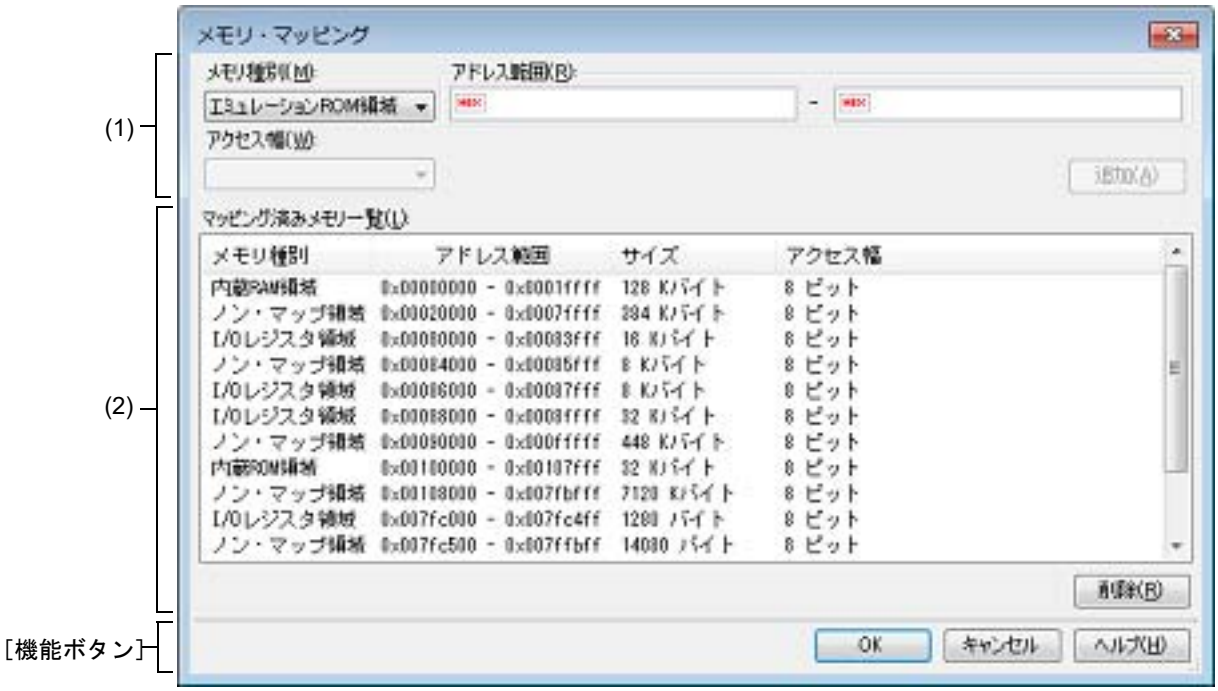


図 A.42 メモリ・マッピング ダイアログ【シミュレータ】



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- プロパティ パネルの［デバッグ・ツール設定］タブにおいて，［メモリ］カテゴリ内［メモリ・マッピング］プロパティを選択することにより表示される［...］ボタンをクリック

注意 1. 【シミュレータ】
プログラム実行中は，このダイアログをオープンすることはできません。

注意 2. 【E1】【E20】【EZ Emulator】
デバッグ・ツールに接続後，このダイアログをオープンすることはできません。

[各エリアの説明]

(1) 追加メモリ・マッピング指定エリア
新たに追加するメモリ・マッピングの情報を指定します。

(a) ［メモリ種別］
追加するメモリ・マッピングのメモリ種別を次のドロップダウン・リストより選択します（デフォルトで選択される項目は使用するデバッグ・ツールに依存）。

エミュレーション ROM 領域 【シミュレータ】	エミュレーション ROM 領域を追加します。 シミュレータ代替 ROM を使用します。
エミュレーション RAM 領域 【シミュレータ】	エミュレーション RAM 領域を追加します。 シミュレータ代替 RAM を使用します。
I/O 保護領域 【E1】【E20】【EZ Emulator】	I/O 保護領域を追加します。 外部領域のうち，デバッガからリードされないようにするアドレス範囲を指定してください。

注意 1. 【シミュレータ】
追加できるメモリ・マッピングは 16 バイト単位です。16 バイト境界以外で設定した場合は，［OK］ボタンを押下後に設定した領域を含む 16 バイト境界の領域に補正します。

注意 2. 【シミュレータ】
追加されたメモリ・マッピングへのアクセスは，読み出し，書き込みともに 1 サイクルで動作します。

注意 3. 【E1】【E20】【EZ Emulator】
追加される I/O 保護領域は，[OK] ボタンを押下後に I/O 保護領域が含まれる外部領域のアクセス幅で補正されます。

(b) ［アドレス範囲］
追加するメモリ・マッピングの開始アドレスと終了アドレスを指定します。それぞれのテキスト・ボックスに，16 進数を直接入力します。
エミュレータの場合，I/O 保護領域は外部領域と重なる領域にのみ追加できます。また，各外部領域の境界をまたぐ指定はできません。
シミュレータの場合，次のメモリ種別と重なる領域にメモリ・マッピングを追加することはできません（このエリアの［追加］ボタンをクリックした際に，メッセージを表示します）。

- ［内蔵 ROM 領域］
- ［内蔵 RAM 領域］
- ［IO レジスタ領域］

(c) ［アクセス幅］
アクセス幅は設定できません。

(d) ボタン

ボタン	機能
追加	このエリアで指定した内容をメモリ・マッピングに追加します。 追加されたメモリ・マッピングは，［マッピング済みメモリ一覧］エリアに表示されます。なお，［OK］ボタンを押下するまでは，変更内容の設定は行われません。

(2) ［マッピング済みメモリ一覧］エリア

(a) 一覧の表示

追加メモリ・マッピング指定エリアで追加したメモリ・マッピングと，マイクロコントローラ内のメモリ・マッピングの情報を表示します。このエリアを編集することはできません。

メモリ種別	次のメモリ種別を表示します。 - 内蔵 ROM 領域 - 内蔵 RAM 領域 - I/O レジスタ領域 - 外部領域（CS7/CS6/.../CS0）【E1】【E20】【EZ Emulator】 - その他のメモリ領域【E1】【E20】【EZ Emulator】 - 予約領域【E1】【E20】【EZ Emulator】 - I/O 保護領域【E1】【E20】【EZ Emulator】 - エミュレーション ROM 領域【シミュレータ】 - エミュレーション RAM 領域【シミュレータ】 - ノン・マップ領域【シミュレータ】
アドレス範囲	アドレス範囲を＜*開始アドレス*＞ - ＜*終了アドレス*＞で表示します。 “0x” を付与した 16 進数表示固定です。
サイズ	サイズを 10 進数で表示します（単位：バイト /K バイト）。
アクセス幅	アクセス幅を表示します（単位：ビット）注。

注 シミュレータではアクセス幅は未サポートのため，固定値（8 ビット）を表示します。
なお，アクセス幅の設定値は，シミュレーションの実行時間には影響しません。

(b) ボタン

ボタン	機能
削除	このエリアで選択しているメモリ・マッピングを削除します。 エミュレータの場合，削除できるメモリ領域は I/O 保護領域のみです。 シミュレータの場合，削除できるメモリ領域は，エミュレーション ROM 領域／エミュレーション RAM 領域のいずれかです（マイクロコントローラ内のメモリ・マッピングを削除することはできません）。

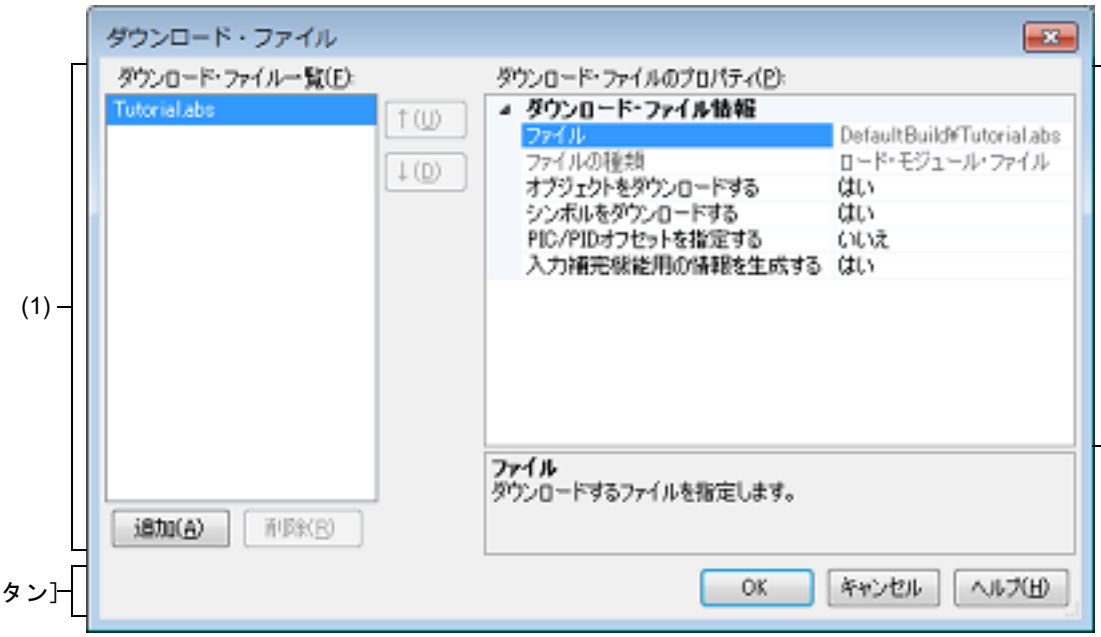
[機能ボタン]

ボタン	機能
OK	現在設定されているメモリ・マッピングをデバッグ・ツールに設定し，このダイアログをクローズします。
キャンセル	メモリ・マッピングの変更を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

ダウンロード・ファイル ダイアログ

ダウンロードする際のファイルの選択，およびダウンロード条件の設定を行います（「2.5 ダウンロード／アップロード」参照）。
プロジェクト（メイン・プロジェクト／サブプロジェクト）でビルド対象に指定しているファイルは，自動的にダウンロードの対象ファイルとして登録されます（削除不可）。

図 A.43 ダウンロード・ファイル ダイアログ



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- プロパティ パネルの［ダウンロード・ファイル設定］タブにおいて，［ダウンロード］カテゴリ内［ダウンロードするファイル］プロパティを選択することにより表示される［...］ボタンをクリック

注意 プログラム実行中は，このダイアログをオープンすることはできません。

[各エリアの説明]

(1) ［ダウンロード・ファイル一覧］エリア

(a) 一覧の表示
ダウンロードするファイル名の一覧を表示します。デフォルトで，プロジェクト（メイン・プロジェクト／サブプロジェクト）においてビルド対象に指定しているファイル名を表示します（削除不可）。
ここでの表示順序が，ダウンロードの際の実行順序となります。
新規にダウンロード・ファイルを追加する場合は，このエリア内の［追加］ボタンをクリックし，［ダウンロード・ファイルのプロパティ］エリアにおいて，追加するファイルのダウンロード条件を指定することにより行います。

(b) ボタン

ボタン	機能
↑	選択しているファイルを 1 行上に移動します。 ただし，最上部のファイル，またはプロジェクトのビルド対象に指定しているファイルを選択している場合は無効となります。
↓	選択しているファイルを 1 行下に移動します。 ただし，最下部のファイル，またはプロジェクトのビルド対象に指定しているファイルを選択している場合は無効となります。
追加	一覧に空欄の項目（“-”）を 1 つ追加し，選択状態にします。 ［ダウンロード・ファイルのプロパティ］エリアにおいて，追加するファイルのダウンロード条件を指定してください。 ただし，すでに 20 個以上のファイルが登録されている場合は無効となります。
削除	選択しているファイルを一覧から削除します。 ただし，プロジェクトのビルド対象に指定しているファイルは削除することはできません。

備考 1. ファイル名にマウス・カーソルを合わせることにより，対象ファイルのパス情報をポップアップ表示します。

備考 2. ファイル名をマウスでドラッグすることにより，一覧内の表示順序を変更することができます。ただし，プロジェクトでビルド対象に指定しているファイルの表示順序を変更することはできません。

(2) ［ダウンロード・ファイルのプロパティ］エリア

(a) ［ダウンロード・ファイル情報］

［ダウンロード・ファイル一覧］エリアで選択しているファイルに対して，ダウンロード条件の表示／設定変更を行います。
また，［追加］ボタンにより，新規にダウンロード・ファイルを追加する場合は，ここで追加ファイルのダウンロード条件を指定します。

ファイル	ダウンロードするファイルを指定します。	
	デフォルト	*ファイル名*（ただし，新規追加の場合は空欄）
	変更方法	キーボードからの直接入力，またはこの項目を選択すると欄内右端に表示される［...］ボタン[注 1]のクリックによりオープンするダウンロードするファイルを選択 ダイアログによる指定
	指定可能値	「表 2.2 ダウンロード可能なファイル形式」参照 最大指定文字数：259 文字
ファイルの種類	ダウンロードするファイルのファイル形式を指定します。	
	デフォルト	ロード・モジュール・ファイル
	変更方法	ドロップダウン・リストによる選択
	指定可能値	次のいずれか - ロード・モジュール・ファイル - ヘキサ・ファイル - S レコード・ファイル - バイナリ・データ・ファイル
オフセット	この項目は，ダウンロードするファイルがヘキサ・フォーマット，および S レコード・ファイルの場合のみ表示されます。指定したファイルのダウンロードを開始するアドレスからのオフセット値を指定します。	
	デフォルト	0
	変更方法	キーボードからの直接入力
	指定可能値	0x0 ～ 0xFFFFFFFF の 16 進数

開始アドレス	この項目は，ダウンロードするファイルがバイナリ・データ・フォーマットの場合のみ表示されます。指定したファイルをダウンロードする開始アドレスを指定します。		
	デフォルト	0	
	変更方法	キーボードからの直接入力	
	指定可能値	0x0 ～ 0xFFFFFFFF の 16 進数	
オブジェクトをダウンロードする	この項目は，ダウンロードするファイルがロード・モジュール・フォーマットの場合のみ表示されます。指定したファイルからオブジェクト情報をダウンロードするか否かを指定します。		
	デフォルト	はい	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	オブジェクト情報をダウンロードします。
		いいえ	オブジェクト情報をダウンロードしません。
シンボルをダウンロードする	この項目は，ダウンロードするファイルがロード・モジュール・フォーマットの場合のみ表示されます。 指定したファイルからシンボル情報をダウンロードするか否かを指定します注 2。		
	デフォルト	はい	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	シンボル情報をダウンロードします。
		いいえ	シンボル情報をダウンロードしません。
PIC/PID オフセットを指定する	ダウンロードするロード・モジュールの PIC（Position Independent Code）領域，および PID（Position Independent Data）領域の位置をロード・モジュール作成時から変更するか否かを指定します。この項目を " はい " に切り替えると，サブ項目に "PIC オフセット "，"PID オフセット " 項目を表示します。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	PIC/PID オフセットを指定します注 3。
		いいえ	PIC/PID オフセットを指定しません。
PIC オフセット	ロード・モジュール作成時のアドレスからのオフセット値を入力します。 例えば，プログラム・セクションの開始アドレスが 0x1000 番地からだった場合に本項目に "1000" を入力すると，対象のセクションは 0x2000 番地にダウンロードされます。		
	デフォルト	0	
	変更方法	キーボードからの直接入力	
	指定可能値	0x0 ～ 0xFFFFFFFF の 16 進数	
PID オフセット	ロード・モジュール作成時に指定した PID レジスタへ設定するオフセット値を入力します。 例えば，ロード・モジュールの実行時に PID レジスタに 0x200 を設定する場合は，本項目に "200" を入力してください。		
	デフォルト	0	
	変更方法	キーボードからの直接入力	
	指定可能値	0x0 ～ 0xFFFFFFFF の 16 進数	

入力補完機能用の情報を生成する	この項目は，ダウンロードするファイルがロード・モジュール・フォーマットの場合のみ表示されます。 選択されたファイルからシンボル情報をダウンロードする際に，シンボル名の入力補完機能用の情報を生成するか否かを指定します。 ウォッチパネルやメモリパネルなどでシンボル名の入力補完機能を使う場合は“はい”を選択してください。なお，この情報はシンボルのダウンロード中に生成するため，ダウンロードに時間が掛かる場合があります。		
	デフォルト	はい	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	入力補完機能用の情報を生成します。
		いいえ	入力補完機能用の情報を生成しません。

注 1. ［ダウンロード・ファイル一覧］エリアにおいて，プロジェクトのビルド対象のファイルを選択している場合，またはプログラム実行中は，［...］ボタンは表示されません。

注 2. シンボル情報をダウンロードしない場合，ソース・レベル・デバッグを行うことはできません。

注 3. PIC/PID 機能（「2.7 PIC/PID 機能の使用」参照）を使って作成していないロード・モジュールに対して“はい”を選択した時のデバッグ動作は保証できません。

［機能ボタン］

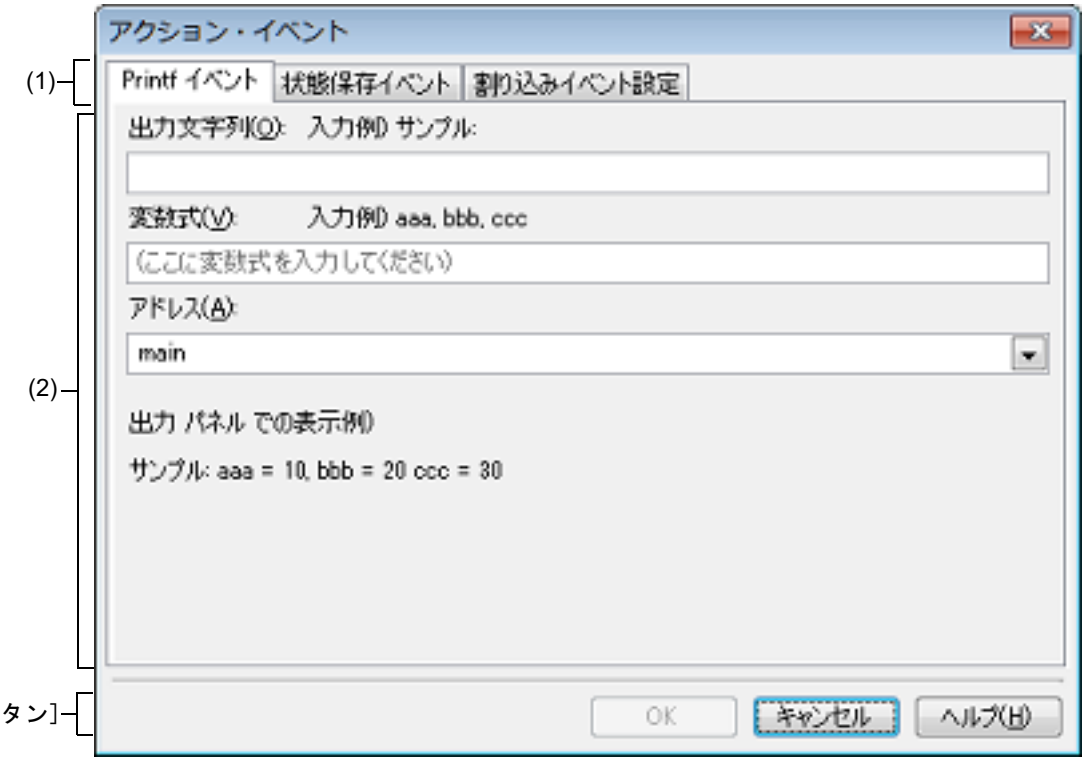
ボタン	機能
OK	ダウンロード・ファイルの設定を終了し，このダイアログをクローズします。
キャンセル	ダウンロード・ファイルの変更を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

アクション・イベント ダイアログ

アクション・イベントの設定を行います（「2.16 プログラム内へのアクションの設定」参照）。
なお，このダイアログは，デバッグ・ツールと接続時のみオープンすることができます。

注意 アクション・イベントの設定に関しては（有効イベント数の制限など），（「2.17.7 イベント設定に関する留意事項」も参照してください。

図 A.44 アクション・イベント ダイアログ



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］
- ［機能ボタン］

［オープン方法］

- エディタ パネルにおいて，アクション・イベントを設定したい行にキャレットを移動したのち，コンテキスト・メニュー→［アクション・イベントの登録 ...］を選択
- 逆アセンブル パネルにおいて，アクション・イベントを設定したいアドレスにキャレットを移動したのち，コンテキスト・メニュー→［アクション・イベントの登録 ...］を選択
- イベント パネルにおいて，アクション・イベントを選択したのち，コンテキストメニュー→［条件の編集 ...］を選択

［各エリアの説明］

(1) タブ選択エリア
タブを選択することにより，登録するアクション・イベントが切り替わります。
このダイアログには，次のタブが存在します。

- ［Printf イベント］タブ
- ［状態保存イベント］タブ

- ［割り込みイベント設定］タブ【シミュレータ】

注意 コンテキスト・メニューの［条件の編集 ...］の選択によりこのダイアログをオープンした場合，このエリアは非表示となります。

(2) イベント条件設定エリア
アクション・イベントの詳細条件を設定します。
設定方法についての詳細は，該当するタブの項を参照してください。

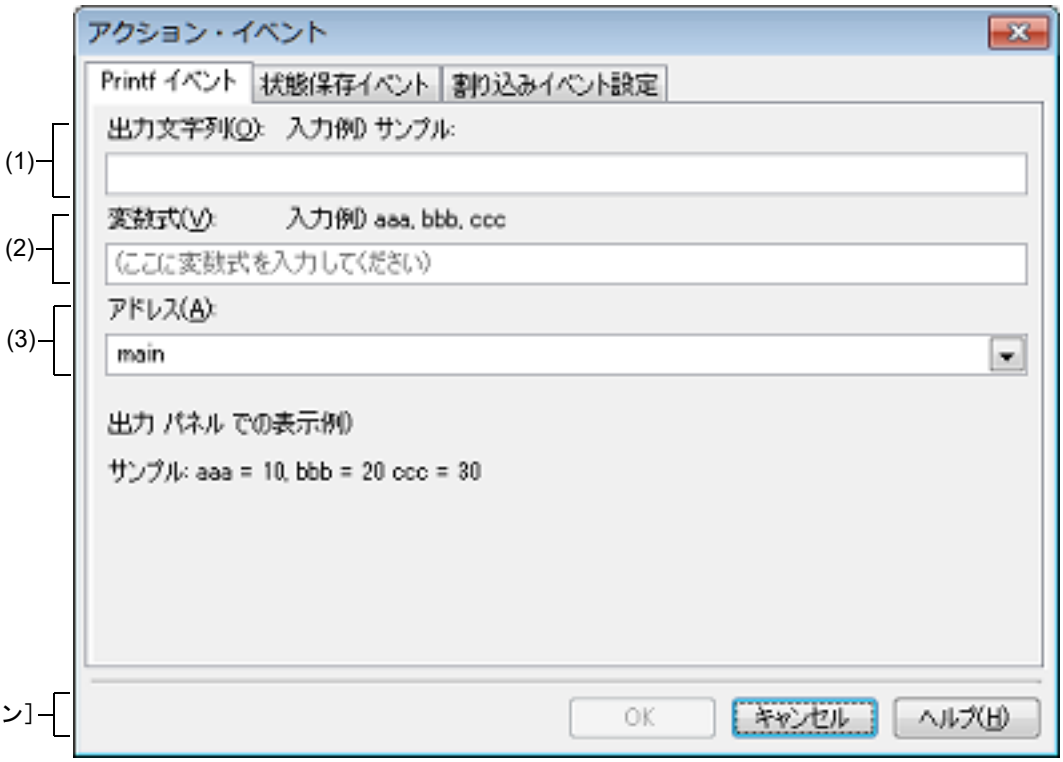
［機能ボタン］

ボタン	機能
OK	アクション・イベントの設定を終了し，指定したアクション・イベントを指定した位置に設定します。
キャンセル	アクション・イベントの設定を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

［Printf イベント］タブ

アクション・イベントとして，Printf イベントの設定を行います（「2.16.1 printf を挿入する」参照）。
Printf イベントとは，プログラムの実行を指定した箇所で一瞬停止させ，ソフトウエア処理によりコマンド（printf）を実行させる機能です。Printf イベントを設定すると，このイベントを設定した箇所の命令実行直前にプログラムが一瞬停止し，このダイアログで指定した変数式の値を出力 パネルに出力します。

図 A.45 アクション・イベント ダイアログ：［Printf イベント］タブ



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］
- ［機能ボタン］

［オープン方法］

- エディタ パネルにおいて，Printf イベントを設定したい行にキャレットを移動したのち，コンテキスト・メニュー→［アクション・イベントの登録 ...］を選択
- 逆アセンブル パネルにおいて，Printf イベントを設定したいアドレスにキャレットを移動したのち，コンテキスト・メニュー→［アクション・イベントの登録 ...］を選択
- イベント パネルにおいて，Printf イベントを選択したのち，コンテキストメニュー→［条件の編集 ...］を選択

［各エリアの説明］

(1) ［出力文字列］エリア
出力 パネルに出力する際に付与する文字列をキーボードより直接入力で指定します（最大指定文字数：1024 文字）。
なお，出力する文字列は，1 行分のみ入力可能です（空白可）。

(2) ［変数式］エリア
Printf イベントの対象となる変数式を指定します。
変数式は，テキスト・ボックスに直接入力で指定します（最大指定文字数：1024 文字）。
“,” で区切ることにより，1 つの Printf イベントとして 10 個までの変数式を指定することができます。

エディタ パネル／逆アセンブル パネルにおいて，変数式を選択した状態でこのダイアログをオープンした場合では，選択している変数式がデフォルトで表示されます。
なお，変数式として指定できる基本入力形式と，その際に Printf イベントとして出力される値は次のとおりです。

表 A.14 変数式と出力される値の関係（Printf イベント）

変数式	出力される値
C/C++ 言語変数名[注 1]	C/C++ 言語の変数の値
変数式 [*変数式*]	配列の要素値
変数式 . メンバ名[注 2]	構造体／共用体／クラスのメンバ値
変数式 -> メンバ名[注 2]	ポインタの指し示す構造体／共用体／クラスのメンバ値
変数式 .* キャスト式	メンバ変数のポインタの値
変数式 ->* キャスト式	メンバ変数のポインタの値
* *変数式*	ポインタの変数の値
& 変数式	配置アドレス
（型名）変数式	指定された型へキャストした値
CPU レジスタ名	CPU レジスタの値
I/O レジスタ名	I/O レジスタの値
ラベル名[注 3] ／ EQU シンボル名[注 3] ／［即値］	ラベルの値／ EQU シンボルの値／即値アドレスの値

注 1. C89，C99，C++ 言語の変数を表します。

注 2. 基底クラスのメンバ変数を指定する場合，メンバ名の前にスコープを指定してください（例：variable.BaseClass::member）。

注 3. ラベル名または EQU シンボル名に “$” が含まれている場合，名前を “{}” で囲んでください（例：{$Label}）。
虚数の値には，大文字の “I” を掛けてください（例：1.0 + 2.0*I）。“I” はキーワードとなるため，CPU レジスタの “I” を指定する場合は “:REG” を付加してください（例：I:REG）。

備考 このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます。（「2.20.2 シンボル名の入力補完機能」参照）

(3) ［アドレス］エリア
Printf イベントを設定するアドレスを指定します。
テキスト・ボックスにアドレス式を直接入力するか（最大指定文字数：1024 文字），またはドロップダウン・リストにより入力履歴項目（最大履歴個数：10 個）を選択します。デフォルトで，現在の指定位置のアドレスを表示します。

備考 このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます。（「2.20.2 シンボル名の入力補完機能」参照）

なお，出力 パネル上における，Printf イベントによる出力結果のフォーマットは次のとおりです。

図 A.46 Printf イベントの出力結果フォーマット

指定された文字列 変数式 1 = 値 1, 変数式 2 = 値 2, 変数式 3 = 値 3, ...

指定された文字列	［出力文字列］で指定した文字列
変数式 1 ～ 10	［変数式］で指定した文字列

値 1 ～ 10	“ *変数式 1 ～ 10*” に対する変数値 値は変数の型に応じた表示形式（「表 A.10 ウォッチ式の表示形式（デフォルト）」参照）で表示します（指定された変数式を取得できない場合は，“?” を表示）。 また，“()” 内に 16 進数値も併記します（表示不能の場合は “ ー ” を表示）。

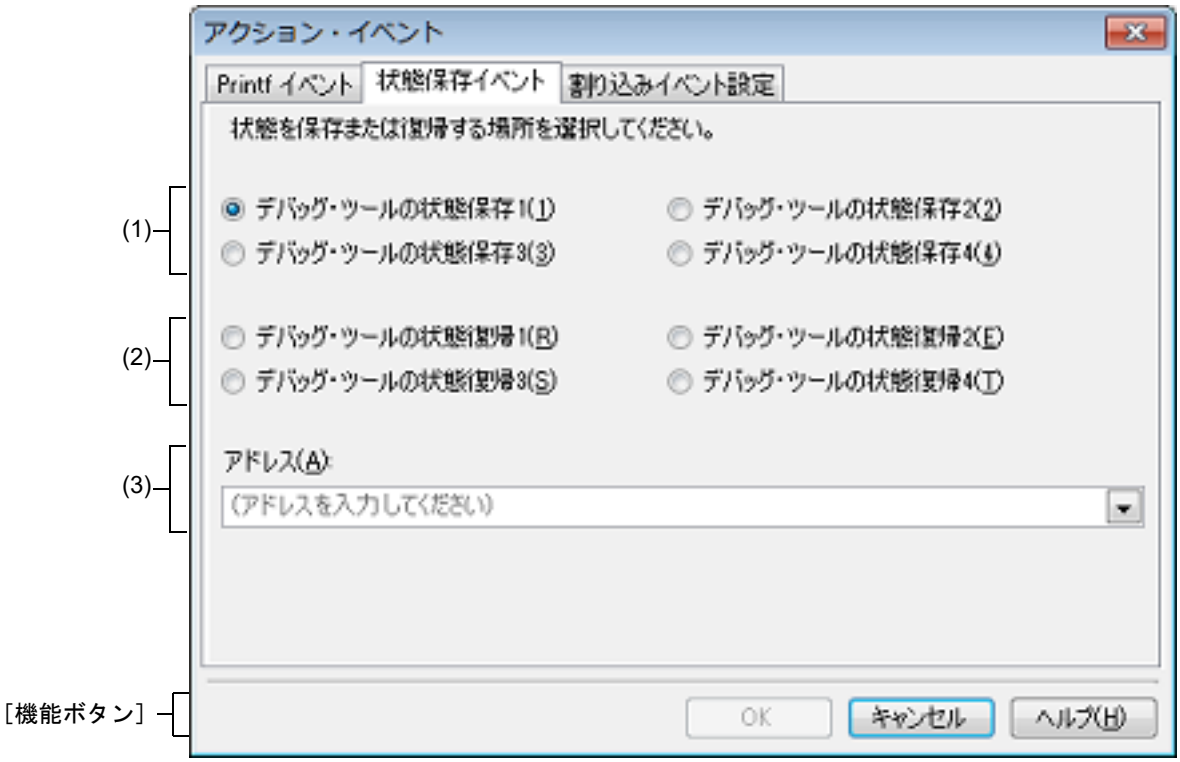
［機能ボタン］

ボタン	機能
OK	Printf イベントの設定を終了し，ここで指定した Printf イベントをエディタ パネル／逆アセンブル パネル上のキャレット位置の行／アドレスに設定します。
キャンセル	Printf イベントの設定を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

［状態保存イベント］タブ

アクション・イベント発生時のデバッグ・ツールの状態の保存／復帰の設定を行います。
なお、復元されるデータは読み書き可能なメモリとレジスタの値に限られます。

図 A.47 アクション・イベント ダイアログ：［状態保存イベント］タブ



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］
- ［機能ボタン］

［オープン方法］

- エディタ パネルにおいて，状態保存イベントを設定したい行にキャレットを移動したのち，コンテキスト・メニュー→［アクション・イベントの登録 ...］を選択
- 逆アセンブル パネルにおいて，状態保存イベントを設定したいアドレスにキャレットを移動したのち，コンテキスト・メニュー→［アクション・イベントの登録 ...］を選択
- イベント パネルにおいて，状態保存イベントを選択したのち，コンテキストメニュー→［条件の編集 ...］を選択

［各エリアの説明］

(1) ［デバッグ・ツールの状態保存 *n*］エリア
アクション・イベント発生時，デバッグ・ツールの状態を *n* 番目のデータとしてファイルに保存します。

(2) ［デバッグ・ツールの状態復帰 *n*］エリア
アクション・イベント発生時，デバッグ・ツールの状態を *n* 番目のデータ・ファイルから復帰します。

(3) ［アドレス］エリア
状態保存イベントを設定するアドレスを指定します。
テキスト・ボックスにアドレス式を直接入力するか（最大指定文字数：1024 文字），またはドロップダウン・リストにより入力履歴項目（最大履歴個数：10 個）を選択します。デフォルトで，現在の指定位置のアドレスを表示します。

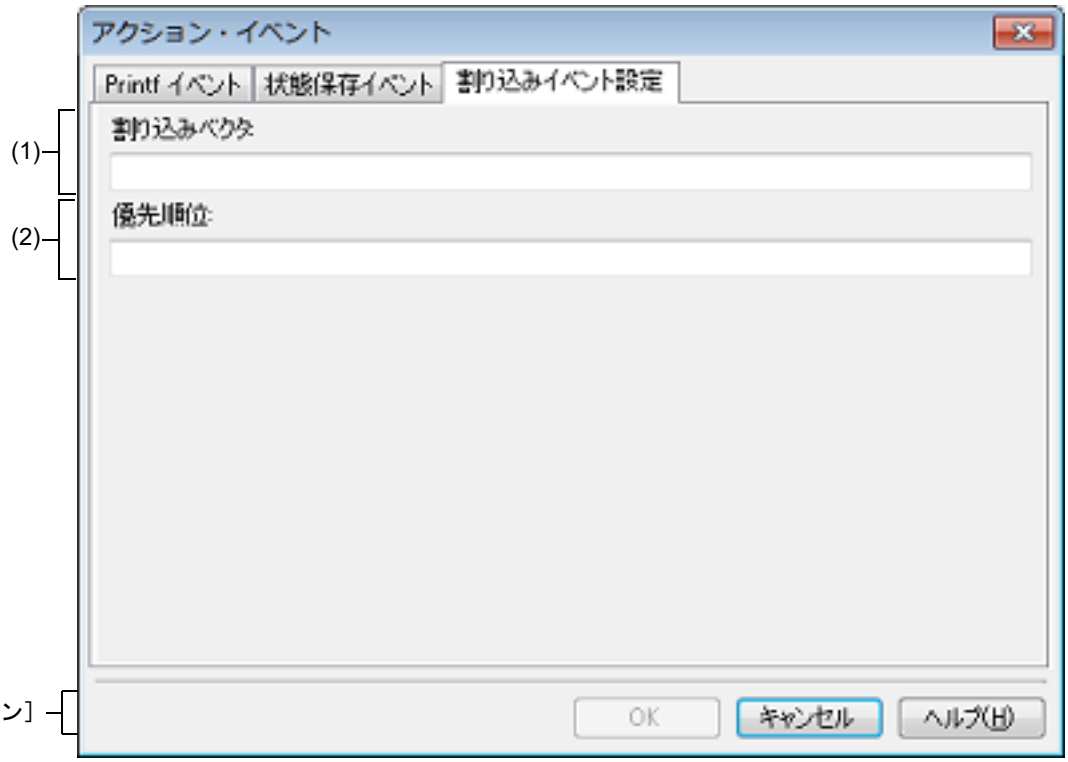
［機能ボタン］

ボタン	機能
OK	状態保存イベントの設定を終了し，ここで指定した状態保存イベントをエディタパネル／逆アセンブル パネル上のキャレット位置の行／アドレスに設定します。
キャンセル	状態保存イベントの設定を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

［割り込みイベント設定］タブ【シミュレータ】

割り込みイベントの設定を行います（「2.16.2 割り込みイベントを挿入する【シミュレータ】」参照）。
割り込みイベントとは，プログラムの実行を指定した箇所で割り込み要求を発生させる機能です。割り込みイベントを設定すると，このイベントを設定した箇所の命令実行直前に割り込み要求が発生し，CPU が割り込み要求を受け付けると割り込み例外処理を実行します。

図 A.48 アクション・イベント ダイアログ：［割り込みイベント設定］タブ



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］
- ［機能ボタン］

［オープン方法］

- エディタ パネルにおいて，割り込みイベントを設定したい行にキャレットを移動したのち，コンテキスト・メニュー→［アクション・イベントの登録 ...］を選択
- 逆アセンブル パネルにおいて，割り込みイベントを設定したいアドレスにキャレットを移動したのち，コンテキスト・メニュー→［アクション・イベントの登録 ...］を選択
- イベント パネルにおいて，割り込みイベントを選択したのち，コンテキストメニュー→［条件の編集 ...］を選択

［各エリアの説明］

(1) ［割り込みベクタ］エリア
割り込みベクタ番号を指定します。
0 ～ 255 の範囲で直接入力により指定してください。

(2) ［優先順位］エリア
優先順位を指定します。

- 【RX610 グループ】
  1 ～ 8 の範囲で直接入力により指定してください。

8 を指定した場合は，高速割り込みとして動作します。

-【RX610 グループ以外】
1 ～ 16 の範囲で直接入力により指定してください。
16 を指定した場合は，高速割り込みとして動作します。

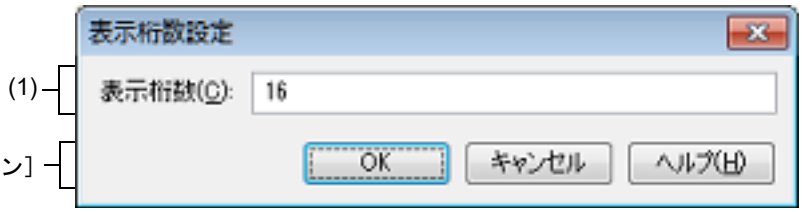
［機能ボタン］

ボタン	機能
OK	割り込みイベントの設定を終了し，ここで指定した割り込みイベントをエディタパネル／逆アセンブル パネル上のキャレット位置の行／アドレスに設定します。
キャンセル	割り込みイベントの設定を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

表示桁数設定 ダイアログ

メモリ パネルにおいて，メモリ値の表示桁数の設定を行います。

図 A.49 表示桁数設定 ダイアログ



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- メモリ パネルにおいて，コンテキスト・メニューの［表示］→［表示桁数を設定 ...］を選択

[各エリアの説明]

(1) ［表示桁数］エリア
表示する桁数を 10 進数で直接入力により指定します。
指定可能な値の範囲は，現在のメモリ パネルにおける［サイズ表記］の設定により，次のように異なります。

サイズ表記	指定可能な範囲
4 ビット	2 ～ 512 注
1 バイト	1 ～ 256
2 バイト	1 ～ 128
4 バイト	1 ～ 64
8 バイト	1 ～ 32

注 偶数値でのみ指定できます（奇数値が入力された場合，1 つ大きな値に変更されます）。

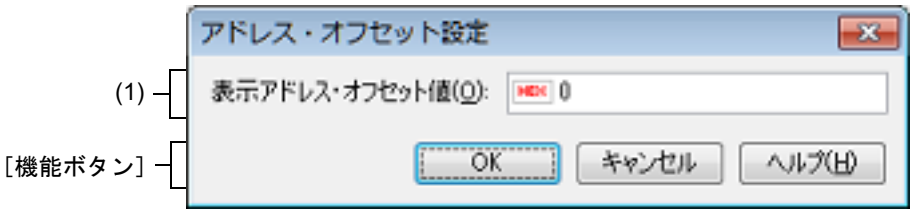
[機能ボタン]

ボタン	機能
OK	指定した桁数でメモリ値を表示します。
キャンセル	設定を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

アドレス・オフセット設定 ダイアログ

メモリ パネルのアドレス・エリアにおいて，開始アドレスのオフセット値を設定します。

図 A.50 アドレス・オフセット設定 ダイアログ



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- メモリ パネルにおいて，コンテキスト・メニューの［表示］→［表示アドレス・オフセット値を設定 ...］を選択

[各エリアの説明]

(1) ［表示アドレス・オフセット値］エリア
アドレス表示のオフセット値を 16 進数で直接入力により指定します。
指定可能な値の範囲は，現在のメモリ パネルにおいて 1 行に表示されているメモリのバイト数により，次のように異なります。

- 指定可能な範囲： 0x0 ～ (*［サイズ表記］の設定* × *表示桁数*) － 1

例 ［サイズ表記］: 1 バイト／表示桁数：16 桁の場合

オフセット値	アドレス・エリアの表示内容
0x0（デフォルト）	0000 0010 0020
0x1	0001 0011 0021
0x2	0002 0012 0022

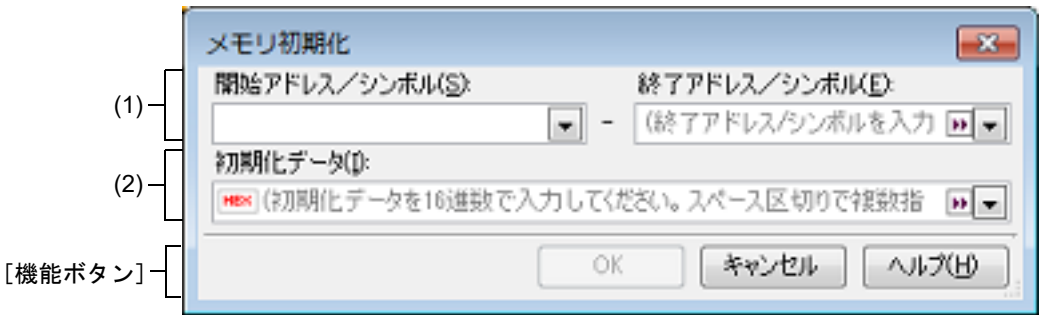
[機能ボタン]

ボタン	機能
OK	指定したオフセット値でメモリのアドレス表示を行います。
キャンセル	設定を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

メモリ初期化 ダイアログ

メモリ値の初期化を行います（「2.11.1.6 メモリの内容を一括して変更（初期化）する」参照）。
指定したアドレス範囲のメモリ領域に，指定した初期化データのパターンを繰り返し書き込みます。

図 A.51 メモリ初期化 ダイアログ



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- メモリ パネルにおいて，コンテキスト・メニュー→［初期化 ...］を選択

[各エリアの説明]

(1) 範囲指定エリア
メモリ値を初期化するアドレス範囲を［開始アドレス / シンボル］と［終了アドレス / シンボル］に指定します。それぞれのテキスト・ボックスにアドレス式を直接入力するか（最大指定文字数：1024 文字），またはドロップダウン・リストにより入力履歴項目（最大履歴個数：10 個）を選択します。
入力したアドレス式の計算結果を，それぞれ開始アドレス／終了アドレスとして扱います。
なお，マイクロコントローラのアドレス空間よりも大きいアドレス値を指定することはできません。

注意 エンディアンの異なる領域をまたいだアドレス範囲を指定することはできません。

備考 このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます（「2.20.2 シンボル名の入力補完機能」参照）。

(2) ［初期化データ］エリア
メモリに書き込む初期化データを指定します。
初期化データの指定は，16 進数の数値をテキスト・ボックスに直接入力するか，またはドロップダウン・リストにより入力履歴項目（最大履歴個数：10 個）を選択することにより行います。
初期化データを複数指定する場合は，1 個 4 バイト（8 文字）までのデータを最大 16 個まで，半角スペースで区切り指定します。
個々の初期化データは，文字列終端より 2 文字単位で 1 バイトと解釈され，奇数文字数の場合は先頭 1 文字で 1 バイトと解釈されます。
なお，バイト数が 2 バイト以上の場合は，次のようにプロジェクトのエンディアンに合わせたバイト列に変換してターゲット・メモリへの書き込み処理を行います。

入力文字列（初期化データ）	書き込みイメージ（バイト単位）	
	リトル・エンディアン	ビック・エンディアン
1	01	01
0 12	00 12	00 12
00 012 345	00 12 00 45 03	00 00 12 03 45

入力文字列（初期化データ）	書き込みイメージ（バイト単位）	
	リトル・エンディアン	ビック・エンディアン
000 12 000345	00 00 12 45 03 00	00 00 12 00 03 45

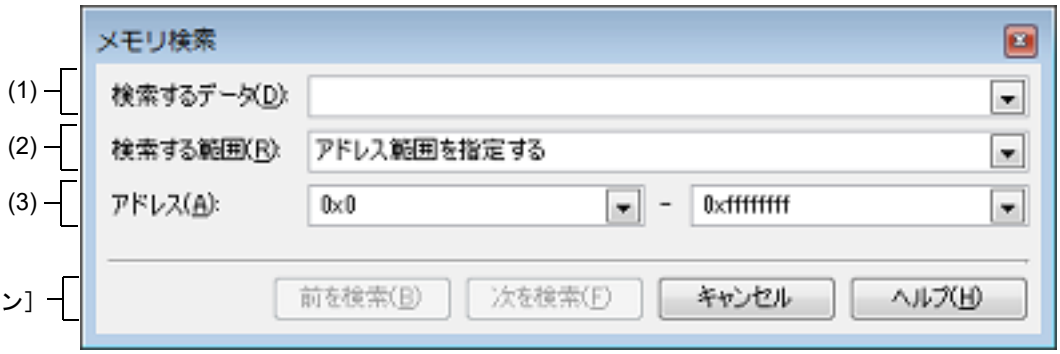
［機能ボタン］

ボタン	機能
OK	指定したアドレス範囲のメモリ領域に，指定した初期化データのパターンを繰り返し書き込みます（パターンの途中で終了アドレスに達した場合は書き込みを終了します）。
キャンセル	メモリ値の初期化の設定を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

メモリ検索 ダイアログ

メモリ値の検索を行います（「2.11.1.5 メモリの内容を検索する」参照）。
このダイアログをオープンする直前にメモリ パネル上でキャレットが存在した，メモリ値エリア／文字列エリアのどちらかが検索の対象となります。

図 A.52 メモリ検索 ダイアログ



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- メモリ パネルにおいて，コンテキスト・メニュー→［検索 ...］を選択

[各エリアの説明]

(1) ［検索するデータ］エリア
検索するデータを指定します。
テキスト・ボックスに直接入力するか（最大指定バイト数：256 バイト），またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
検索の対象がメモリ パネル上のメモリ値エリアの場合，そのエリアと同じ表示形式（表示進数／サイズ）でデータを入力する必要があります。
また，検索の対象が文字列エリアの場合では，検索するデータとして，文字列を指定する必要があります。指定した文字列は，そのエリアで表示しているエンコード形式でデータに変換され検索されます。
なお，このダイアログをオープンする直前にメモリ値を選択していた場合は，デフォルトでその値が表示されます。

(2) ［検索する範囲］エリア
検索する範囲を次のドロップダウン・リストより選択します。

アドレス範囲を指定する	［アドレス］エリアで指定したアドレス範囲内で検索を行います。
メモリ・マッピング	選択したメモリ・マッピング範囲内で検索を行います。 このリスト項目は，メモリ・マッピング ダイアログで表示しているメモリ・マッピングを個々に表示します（ノン・マップ領域を除く）。 表示形式：*＜メモリ種別＞ ＜アドレス範囲＞ ＜サイズ＞*

(3) ［アドレス］エリア
この項目は，［検索する範囲］エリアで［アドレス範囲を指定する］を選択した場合のみ有効となります。
メモリ値検索の対象となるアドレス範囲を“開始アドレス”と“終了アドレス”で指定します。それぞれのテキスト・ボックスにアドレス式を直接入力するか（最大指定文字数：1024 文字），またはドロップダウン・リストにより入力履歴項目（最大履歴個数：10 個）を選択します。
入力したアドレス式の計算結果を，それぞれ開始アドレス／終了アドレスとして扱います。
ただし，検索可能なアドレスの上限値は，プログラム空間の上限アドレス（0xFFFFFFFF）です。
また，32 ビットで表現できる値より大きいアドレス値を指定することはできません。

備考 1. このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます（「2.20.2 シンボル名の入力補完機能」参照）。

備考 2. “ 開始アドレス ” が空欄の場合は，“0x0” の指定として扱われます。

備考 3. “ 終了アドレス ” が空欄の場合は，マイクロコントローラのアドレス空間の上限値の指定として扱われます。

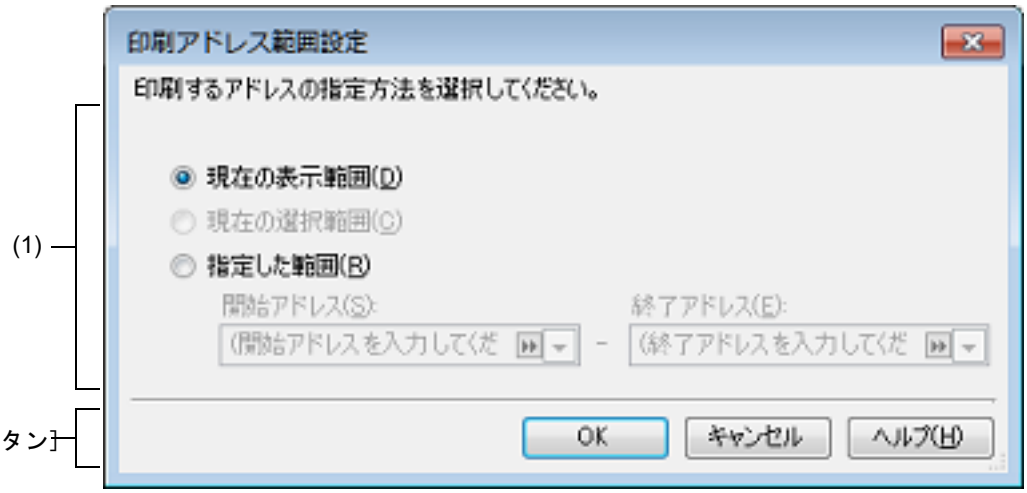
［機能ボタン］

ボタン	機能
前を検索	［検索する範囲］エリア／［アドレス］エリアで指定した範囲内で，アドレスの小さい方向に検索を行います。検索結果箇所をメモリ パネル上で選択状態にします。 ただし，不正な値を指定している場合，またはプログラム実行中は，メッセージを表示し，メモリ値の検索は行いません。 また，メモリ パネルが非表示の場合，または他のパネルにフォーカスがある状態からこのダイアログへフォーカスを移動した場合，このボタンは無効となります。
次を検索	［検索する範囲］エリア／［アドレス］エリアで指定した範囲内で，アドレスの大きい方向に検索を行います。検索結果箇所をメモリ パネル上で選択状態にします。 ただし，不正な値を指定している場合，またはプログラム実行中は，メッセージを表示し，メモリ値の検索は行いません。 また，メモリ パネルが非表示の場合，または他のパネルにフォーカスがある状態からこのダイアログへフォーカスを移動した場合，このボタンは無効となります。
キャンセル	メモリ値の検索の設定を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

印刷アドレス範囲設定 ダイアログ

逆アセンブル パネルの内容を印刷する際に，対象となるアドレス範囲の指定を行います。

図 A.53 印刷アドレス範囲設定 ダイアログ



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- 逆アセンブル パネルにおいて，[ファイル] メニュー→ [印刷 ...] を選択

[各エリアの説明]

(1) 範囲指定エリア
印刷する範囲を指定するために，次のオプション・ボタンのいずれか 1 つを選択します。

(a) [現在の表示範囲]（デフォルト）
逆アセンブル パネルで現在表示している範囲のみを印刷します。

(b) [現在の選択範囲]
逆アセンブル パネルで現在選択している範囲のみを印刷します。
ただし，逆アセンブル パネルにおいて，何も選択していない場合は無効となります。

(c) [指定した範囲]
印刷の対象となるアドレス範囲を [開始アドレス] と [終了アドレス] で指定します。
それぞれのテキスト・ボックスにアドレス式を直接入力するか（最大指定文字数：1024 文字），またはドロップダウン・リストにより入力履歴項目（最大履歴個数：10 個）を選択します。

備考 このテキスト・ボックスで [Ctrl] + [Space] キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます（「2.20.2 シンボル名の入力補完機能」参照）。

[機能ボタン]

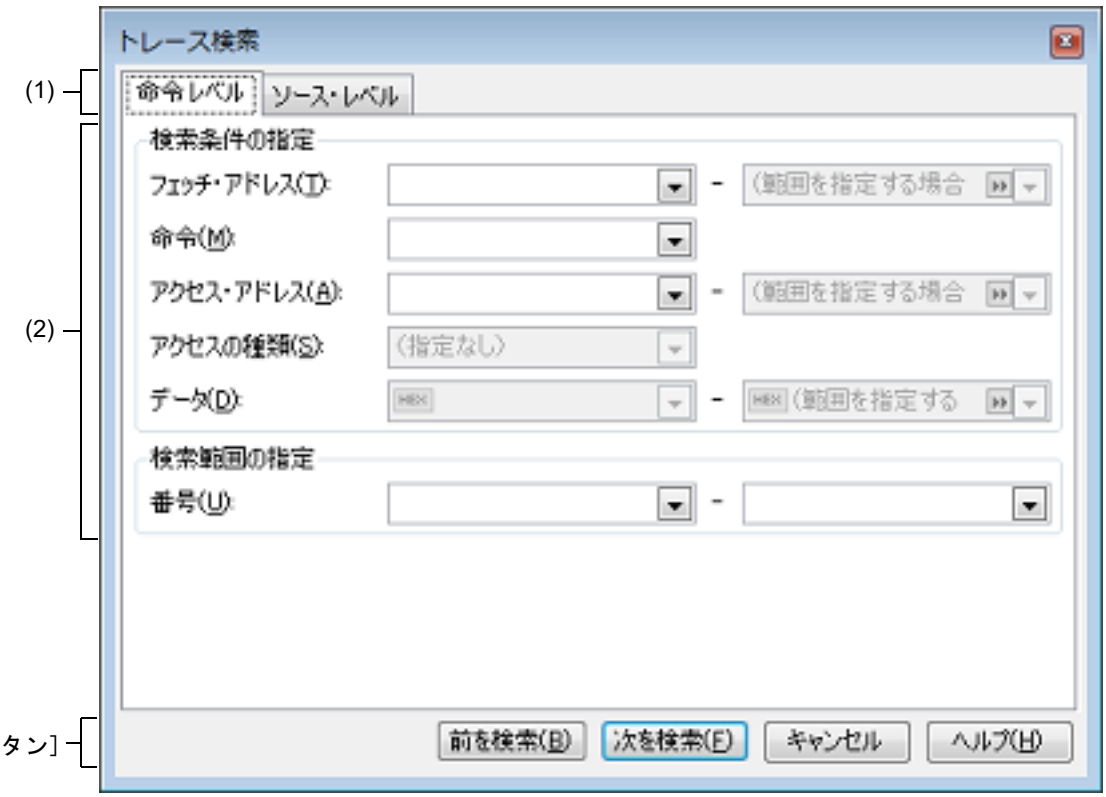
ボタン	機能
OK	指定した範囲で逆アセンブル パネルの内容を印刷するために，このダイアログをクローズして Windows の印刷用 ダイアログをオープンします。
キャンセル	範囲選択の設定を無視し，このダイアログをクローズします。

ボタン	機能
ヘルプ	このダイアログのヘルプを表示します。

トレース検索 ダイアログ

トレース・データの検索を行います（「2.13.8 トレース・データを検索する」参照）。
命令レベル／ソース・レベルを選択して検索することができます。

図 A.54 トレース検索 ダイアログ



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］
- ［機能ボタン］

［オープン方法］

- トレース パネルにおいて，ツールバーの ボタンをクリック
- トレース パネルにおいて，コンテキスト・メニュー→［検索 ...］を選択

［各エリアの説明］

(1) タブ選択エリア
タブを選択することにより，検索するレベルが切り替わります。
このダイアログには，次のタブが存在します。

- ［命令レベル］タブ
- ［ソース・レベル］タブ

(2) 検索条件設定エリア
検索する際の詳細条件を設定します。
表示内容／設定方法についての詳細は，該当するタブの項を参照してください。

［機能ボタン］

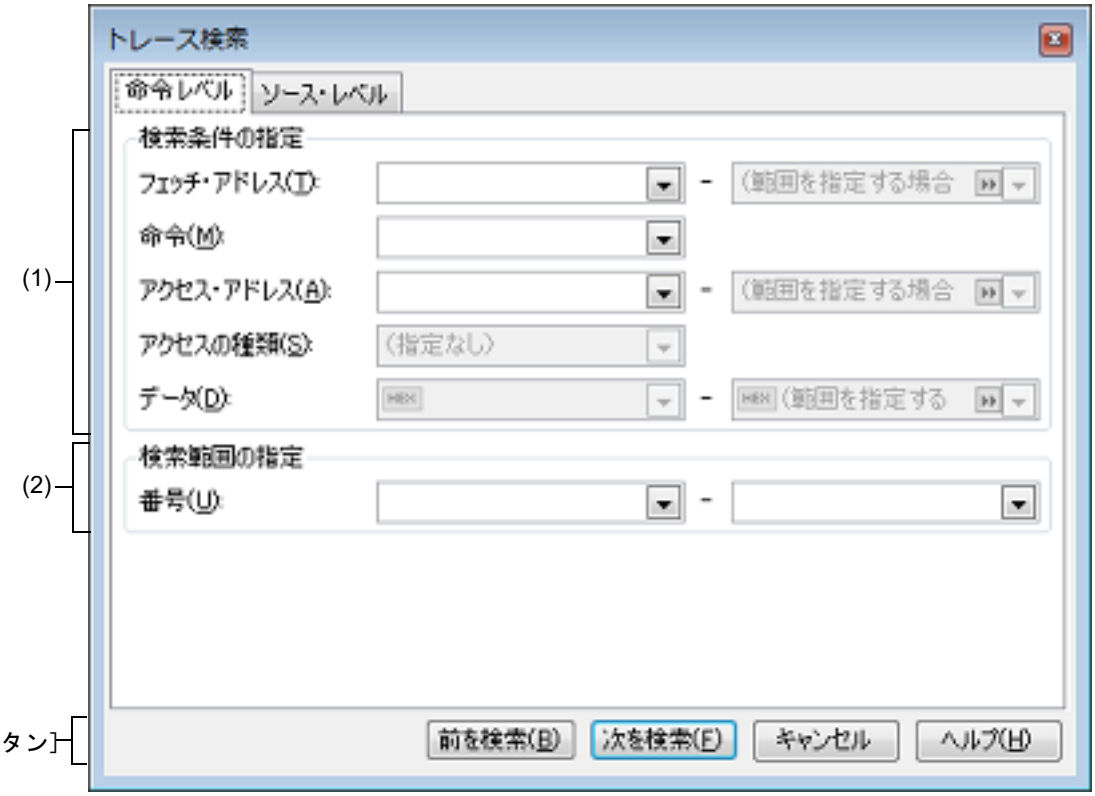
ボタン	機能
前を検索	指定した範囲内で，番号の小さい方向に検索を行います。 検索結果箇所をトレース パネル上で選択状態にします。 ただし，不正な値を指定している場合，またはプログラム実行中は，メッセージを表示し，トレース・データの検索は行いません。 また，トレース パネルが非表示の場合，または他のパネルにフォーカスがある状態からこのダイアログへフォーカスを移動した場合，このボタンは無効となります。
次を検索	指定した範囲内で，番号の大きい方向に検索を行います。 検索結果箇所をトレース パネル上で選択状態にします。 ただし，不正な値を指定している場合，またはプログラム実行中は，メッセージを表示し，トレース・データの検索は行いません。 また，トレース パネルが非表示の場合，または他のパネルにフォーカスがある状態からこのダイアログへフォーカスを移動した場合，このボタンは無効となります。
キャンセル	トレース・データの検索の設定を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

［命令レベル］タブ

取得したトレース・データを命令レベルで検索します。

注意 トレース パネルをソース表示モードで表示している場合，このタブで命令レベルの検索を行っても対象を正しく検索することはできません。
命令レベルの検索を行う際は，混合表示モード，または逆アセンブル表示モードで表示を行ってください。

図 A.55 トレース検索 ダイアログ：［命令レベル］タブ



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］
- ［機能ボタン］

［オープン方法］

- トレース パネルにおいて，ツールバーの ボタンをクリック
- トレース パネルにおいて，コンテキスト・メニュー→［検索 ...］を選択

［各エリアの説明］

(1) ［検索条件の指定］エリア

(a) ［フェッチ・アドレス］
検索条件として必要な場合，フェッチ・アドレスを指定します。
アドレス式をテキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
フェッチ・アドレスの指定は範囲で指定することができます。この場合は，左右両方のテキスト・ボックスにアドレス式を指定することにより範囲を指定します。
右側のテキスト・ボックスが空欄，または“（範囲を指定する場合に入力）”の場合は，左側のテキスト・ボックスに指定された固定アドレスで検索を行います。

なお，マイクロコントローラのアドレス空間よりも大きいアドレス値が指定された場合は，上位のアドレス値をマスクして扱います。
また，32 ビットで表現できる値より大きいアドレス値を指定することはできません。

(b) ［命令］
検索条件として必要な場合，命令の文字列を指定します。
ここで指定した文字列をトレース パネルの［ソース / 逆アセンブル］エリア内より検索します。
命令をテキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
なお，検索の際は，大文字／小文字は区別せず，部分一致も検索の対象とします。

(c) ［アクセス・アドレス］
検索条件として必要な場合，アクセス・アドレスを指定します。
アドレス式をテキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
アクセス・アドレスの指定は範囲で指定することができます。この場合は，左右両方のテキスト・ボックスにアドレス式を指定することにより範囲を指定します。
右側のテキスト・ボックスが空欄，または“（範囲を指定する場合に入力）”の場合は，左側のテキスト・ボックスに指定された固定アドレスで検索を行います。
なお，マイクロコントローラのアドレス空間よりも大きいアドレス値が指定された場合は，上位のアドレス値をマスクして扱います。
また，32 ビットで表現できる値より大きいアドレス値を指定することはできません。

(d) ［アクセスの種類］
この項目は［アクセス・アドレス］が指定された場合のみ有効となります。
アクセスの種類を次のドロップダウン・リストより選択します。
アクセスの種類を限定しない場合は，“（指定なし）”を選択してください。

（指定なし）
リード／ライト
リード
ライト
ベクタ・リード【シミュレータ】
DMA

注意 【E1】【E20】【EZ Emulator】
トレース パネル上で収集可能なアクセスの種類は，リードおよびライトです。そのため，ドロップダウン・リストよりリード／ライト，ベクタ・リードおよび DMA を選択しないでください。
【シミュレータ】
トレース パネル上で収集可能なアクセスの種類は，リード，ライトおよびベクタ・リードです。そのため，ドロップダウン・リストよりリード／ライトおよび DMA を選択しないでください。

(e) ［データ］
この項目は［アクセス・アドレス］が指定された場合のみ有効となります。
アクセスした数値を指定します。
16 進数値をテキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
数値の指定は範囲で指定することができます。この場合は，左右両方のテキスト・ボックスにデータを指定することにより範囲を指定します。
右側のテキスト・ボックスが空欄，または“（範囲を指定する場合に入力）”の場合は，左側のテキスト・ボックスに指定された固定数値で検索を行います。

(2) ［検索範囲の指定］エリア

(a) ［番号］
検索するトレース・データの範囲を，トレース パネルの［番号］エリアに表示されている番号で指定します。
左右のテキスト・ボックスに，それぞれ開始番号と終了番号を指定します（デフォルトでは，“0”～“*最終番号*”が指定されます）。
10 進数で番号をテキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
左側のテキスト・ボックスが空欄の場合は，“0”の指定として扱われます。

右側のテキスト・ボックスが空欄の場合は，最終番号の指定として扱われます。

[機能ボタン]

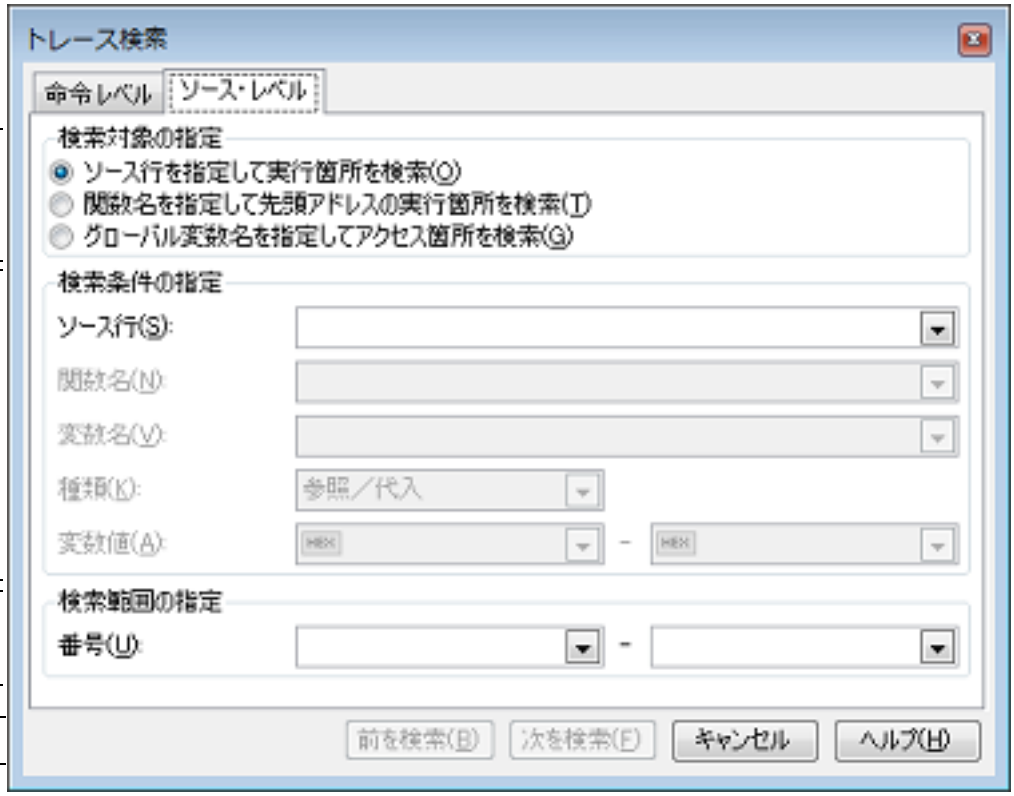
ボタン	機能
前を検索	指定した範囲内で，番号の小さい方向に検索を行います。 検索結果箇所をトレース パネル上で選択状態にします。 ただし，不正な値を指定している場合はメッセージを表示し，トレース・データの検索は行いません。また，トレース パネルが非表示の場合，または他のパネルにフォーカスがある状態からこのダイアログへフォーカスを移動した場合，このボタンは無効となります。
次を検索	指定した範囲内で，番号の大きい方向に検索を行います。 検索結果箇所をトレース パネル上で選択状態にします。 ただし，不正な値を指定している場合はメッセージを表示し，トレース・データの検索は行いません。また，トレース パネルが非表示の場合，または他のパネルにフォーカスがある状態からこのダイアログへフォーカスを移動した場合，このボタンは無効となります。
キャンセル	トレース・データの検索の設定を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

［ソース・レベル］タブ

取得したトレース・データをソース・レベルで検索します。

注意 トレース パネルを逆アセンブル表示モードで表示している場合，このタブでソース・レベルの検索を行っても対象を正しく検索することはできません。
ソース・レベルの検索を行う際は，混合表示モード，またはソース表示モードで表示を行ってください。

図 A.56 トレース検索 ダイアログ：［ソース・レベル］タブ



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］
- ［機能ボタン］

［オープン方法］

- トレース パネルにおいて，ツールバーの ボタンをクリック
- トレース パネルにおいて，コンテキスト・メニュー→［検索 ...］を選択

［各エリアの説明］

(1) ［検索対象の指定］エリア
検索する対象を次のオプション・ボタンの中から選択します。

ソース行を指定して実行箇所を検索	指定したソースの実行箇所を検索します（デフォルト）。 検索条件として［ソース行］の指定のみが有効となります。
関数名を指定して先頭アドレスの実行箇所を検索	指定した関数の実行箇所を検索します。 検索条件として［関数名］の指定のみが有効となります。

グローバル変数名を指定してアクセス箇所を検索	指定したグローバル変数をアクセスした箇所を検索します。 検索条件として［変数名］／［種類］／［変数値］の指定のみが有効となります。

(2) ［検索条件の指定］エリア

(a) ［ソース行］
この項目は“ソース行を指定して実行箇所を検索”が選択された場合のみ有効となります。
ここで指定した文字列をトレース パネルの［行番号 / アドレス］エリア内より検索します。検索するソース行に含まれる文字列を，テキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
なお，検索の際は，大文字／小文字は区別せず，部分一致も検索の対象とします。

例 1. main.c#40

例 2. main.c

例 3. main

(b) ［関数名］
この項目は“関数名を指定して先頭アドレスの実行箇所を検索”が選択された場合のみ有効となります。
検索する関数名を，テキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
なお，検索の際は，大文字／小文字を区別し，完全一致のみを検索の対象とします。

(c) ［変数名］
この項目は“グローバル変数名を指定してアクセス箇所を検索”が選択された場合のみ有効となります。
検索する変数名を，テキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
なお，検索の際は，大文字／小文字を区別し，完全一致のみを検索の対象とします。

(d) ［種類］
この項目は“グローバル変数名を指定してアクセス箇所を検索”が選択された場合のみ有効となります。
アクセスの種類（参照／代入（デフォルト），参照，代入）をドロップダウン・リストより選択します。

(e) ［変数値］
この項目は“グローバル変数名を指定してアクセス箇所を検索”が選択された場合のみ有効となります。
アクセスした変数値を 16 進数で指定します。
変数値をテキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
変数値の指定は範囲で指定することができます。この場合は，左右両方のテキスト・ボックスに変数値を指定することにより範囲を指定します。
右側のテキスト・ボックスが空欄の場合は，左側のテキスト・ボックスに指定された固定変数値でアクセス箇所を検索を行います。

(3) ［検索範囲の指定］エリア

(a) ［番号］
検索するトレース・データの範囲を，トレース パネルの［番号］エリアに表示されている番号で指定します。
左右のテキスト・ボックスに，それぞれ開始番号と終了番号を指定します（デフォルトでは，“0”～“*最終番号*”が指定されます）。
10 進数で番号をテキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
左側のテキスト・ボックスが空欄の場合は，“0”の指定として扱われます。
右側のテキスト・ボックスが空欄の場合は，最終番号の指定として扱われます。

［機能ボタン］

ボタン	機能
前を検索	指定した範囲内で，番号の小さい方向に検索を行います。 検索結果箇所をトレース パネル上で選択状態にします。 ただし，不正な値を指定している場合メッセージを表示し，トレース・データの検索は行いません。また，トレース パネルが非表示の場合，または他のパネルにフォーカスがある状態からこのダイアログへフォーカスを移動した場合，このボタンは無効となります。

ボタン	機能
次を検索	指定した範囲内で，番号の大きい方向に検索を行います。 検索結果箇所をトレース パネル上で選択状態にします。 ただし，不正な値を指定している場合メッセージを表示し，トレース・データの検索は行いません。また，トレース パネルが非表示の場合，または他のパネルにフォーカスがある状態からこのダイアログへフォーカスを移動した場合，このボタンは無効となります。
キャンセル	トレース・データの検索の設定を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

組み合わせ条件設定ダイアログ【E1】【E20】【EZ Emulator】

イベント パネル上で選択した組み合わせブレーク，またはトレース・イベントの組み合わせ条件として詳細情報の変更を行います。

備考 イベントの設定に関しては，「2.17.7 イベント設定に関する留意事項」を参照してください。

図 A.57 組み合わせ条件設定ダイアログ（組み合わせブレーク）

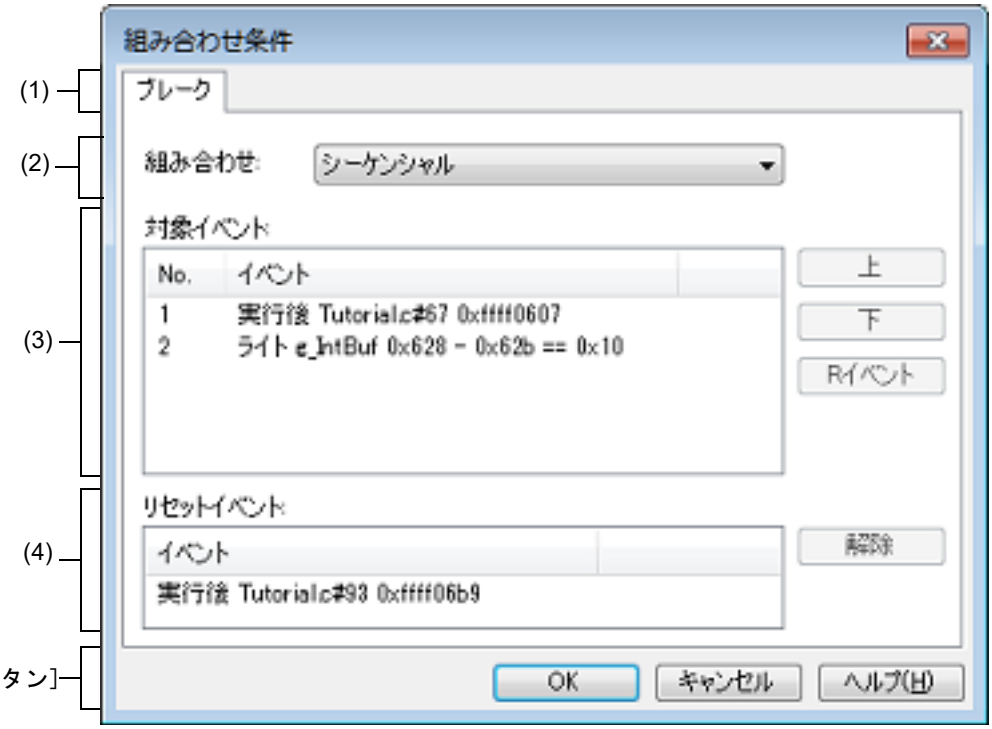
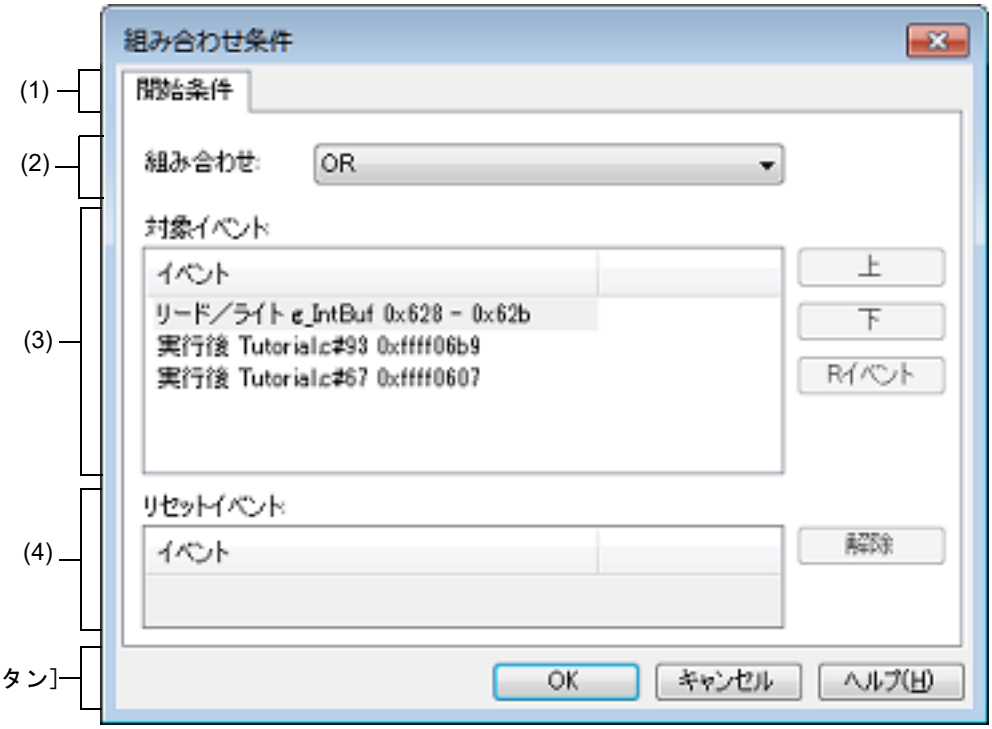


図 A.58 組み合わせ条件設定ダイアログ（トレース・イベント）



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］
- ［機能ボタン］

［オープン方法］

- イベント パネル上において，詳細情報を編集したい組み合わせブレークにキャレットを移動したのち，コンテキスト・メニュー→［条件の編集 ...］を選択
- イベント パネル上において，詳細情報を編集したいトレースにキャレットを移動したのち，コンテキスト・メニュー→［条件の編集 ...］を選択

［各エリアの説明］

(1) タブ選択エリア
タブを選択することにより，条件に合った組み合わせ条件の詳細情報を表示します。
イベント パネル上で選択するイベントによって次の異なるタブが表示されます。

組み合わせブレーク	[ブレーク] タブのみが存在します。
トレース	[開始条件] タブ，および [終了条件] タブが存在します。 タブを切り替えることでそれぞれの組み合わせ条件を設定できます。

(2) 組み合わせ条件選択エリア

(a) [組み合わせ]
組み合わせ条件を次のドロップダウン・リストより指定します。

条件	機能
OR	設定されたイベントのうちいずれか 1 つが成立した場合，条件が成立します。
AND	時間軸に関係なく設定された全てのイベントが成立した場合，条件が成立します。
シーケンシャル	設定されたイベントが指定された順序で成立した場合，条件が成立します。

注意 イベントの組み合わせに関しては，「(1) 組み合わせ条件の編集」の注意も参照してください。

(3) 対象イベント条件表示エリア

(a) リストの表示
組み合わせの対象となるイベントを一覧でリストに表示します。

No.	リストの上から順に番号が割り振られます。 組み合わせ条件で [シーケンシャル] を選択した場合のみ表示され，この番号が条件成立の順序になります。
イベント	イベント条件の詳細情報を表示します。 イベント パネル上で表示される詳細情報と同じ表示内容です。

(b) ボタン

ボタン	機能
↑上	対象イベントリスト内で選択したイベントの順序が 1 つ上げます。 このボタンは，組み合わせ条件で [シーケンシャル] を設定したときのみ有効です。
↓下	対象イベントリスト内で選択したイベントの順序が 1 つ下げます。 このボタンは，組み合わせ条件で [シーケンシャル] を設定したときのみ有効です。
R イベント	対象イベントリスト内で選択したイベントをリセットイベントリストに移動します。 このボタンは，組み合わせ条件で [シーケンシャル] を設定したときのみ有効です。

(4) リセットイベント条件表示／解除エリア

(a) リストの表示
リセットイベントとして登録されたイベントの詳細条件を表示します。
このリストに表示されたリセットイベントが成立した場合，その時点まで成立していた他のイベント条件は全てクリアされます。

注意 リセットイベントに関しては，「(3) リセットイベントの編集」の注意も参照してください。

(b) ボタン

ボタン	機能
解除	リセットイベントリスト内のイベントを対象イベントリストに移動します。 このボタンは，組み合わせ条件で [シーケンシャル] を設定したときのみ有効です。

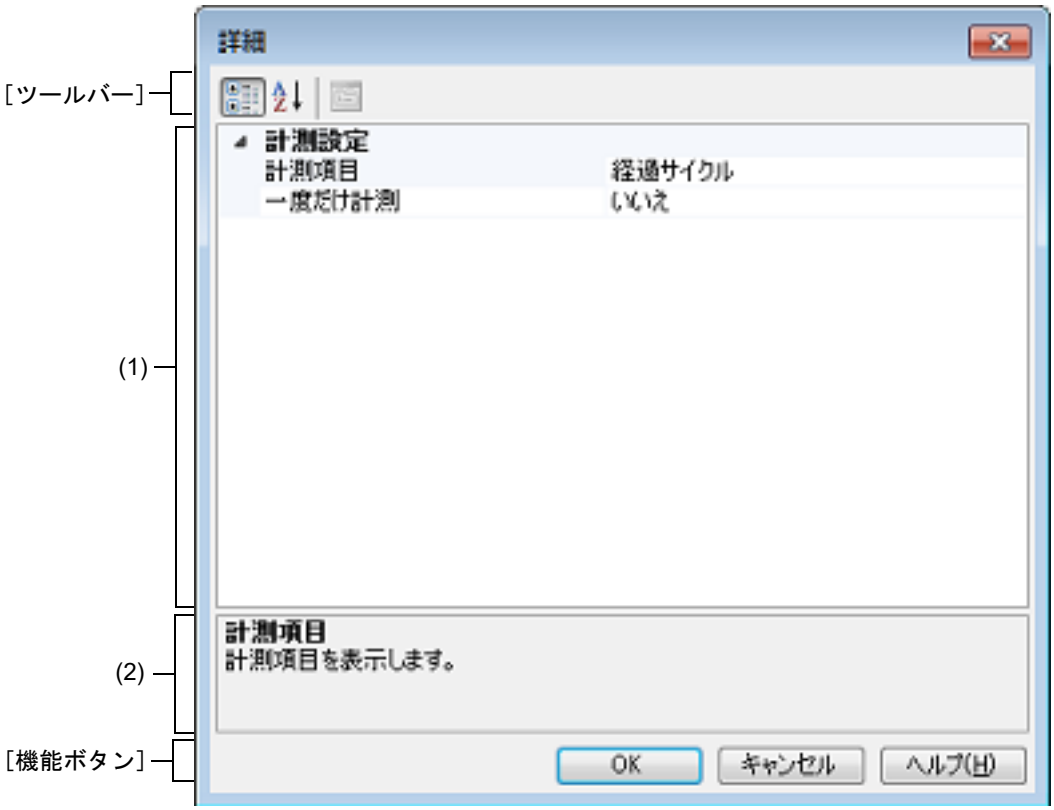
[機能ボタン]

ボタン	機能
OK	ダイアログで指定した詳細設定を組み合わせブレーク，またはトレースに設定し，このダイアログをクローズします。
キャンセル	設定を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

タイマ計測詳細設定ダイアログ【E1】【E20】【EZ Emulator】

イベント パネル上で選択したタイマ・イベントの詳細情報の表示，および変更を行います。
このダイアログではタイマ・イベントのアドレス値を変更することはできません。変更する場合は，タイマ・イベントを削除して新たに作成し直してください。タイマ・イベントの設定に関しては，「2.14 実行時間の計測」を参照してください。

図 A.59 タイマ計測詳細設定ダイアログ



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [カテゴリの説明]
- [機能ボタン]

[オープン方法]

- イベント パネル上において，詳細情報を編集したいタイマ・イベントにキャレットを移動したのち，コンテキスト・メニュー→［条件の編集 ...］を選択

[各エリアの説明]

(1) 詳細情報表示／変更エリア
イベント パネル上で選択しているタイマ・イベントの詳細情報を，カテゴリのリスト形式で表示し，設定の変更を直接行うことができるエリアです。

(2) プロパティの説明エリア
詳細情報表示／変更エリアで選択したカテゴリやプロパティの簡単な説明を表示します。

[ツールバー]

	詳細情報表示／変更エリアにてカテゴリを表示にします。
	詳細情報表示／変更エリアにてカテゴリを非表示にしてプロパティ項目のみを昇順に並び替えます。

[カテゴリの説明]

(1) [計測設定]
タイマ計測に関する詳細設定の表示，および変更を行います。

計測項目注	- RX600 シリーズの場合 計測項目を指定します。 - RX200 シリーズの場合 計測項目に経過サイクルが表示されます。		
	デフォルト	経過サイクル	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	ドロップダウン・リストによる次のいずれか 【RX600 シリーズ】 *経過サイクル，経過サイクル（スーパバイザモードサイクル），* 例外・割り込み処理サイクル，例外処理サイクル， 割り込み処理サイクル，有効命令発行回数， 例外・割り込み受付回数，例外受付回数，割り込み受付回数 【RX200 シリーズ】 *経過サイクル*	
一度だけ計測	指定された計測区間を一度だけ計測して終了するか否かを指定します。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	指定された計測区間を一度だけ計測して終了します。
		いいえ	指定された計測区間を通過した分だけ計測値が加算されます。

注 【RX100 シリーズ】
タイマ機能は未サポートとなります。

[機能ボタン]

ボタン	機能
OK	ダイアログで指定した詳細設定をタイマ・イベントに設定し，このダイアログをクローズします。
キャンセル	設定を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

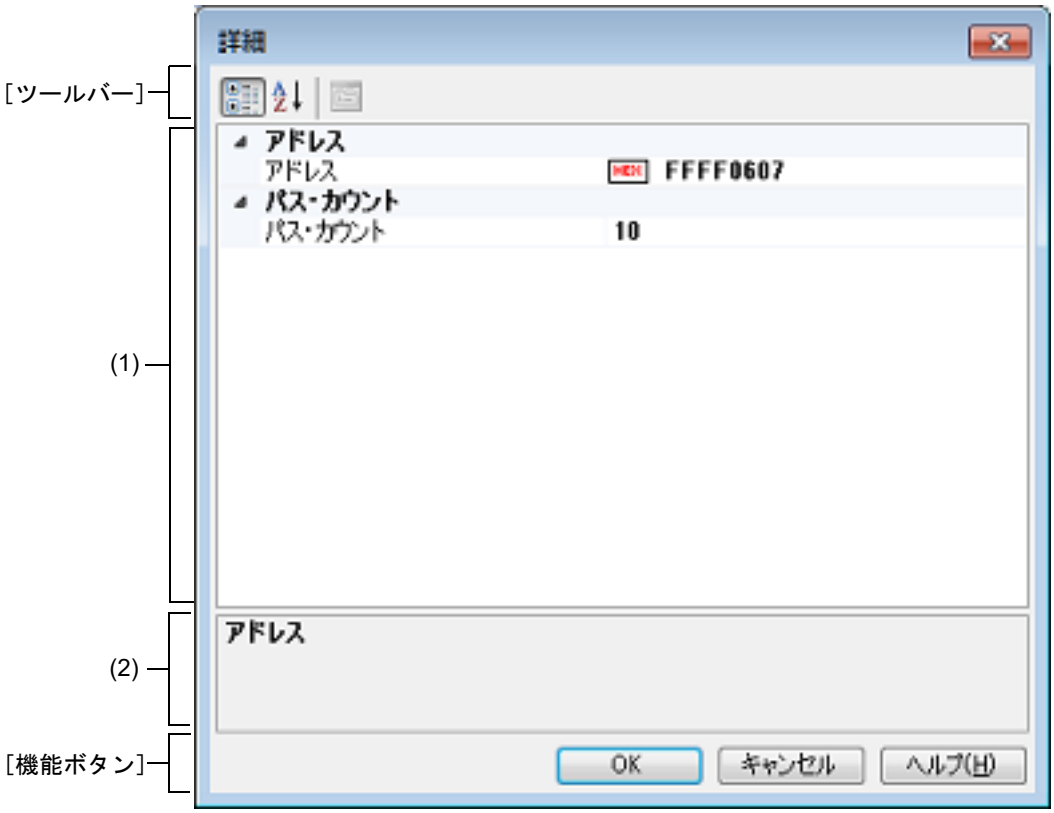
実行イベント詳細設定ダイアログ

イベント パネル上で選択した実行系イベントの詳細情報の表示，および変更を行います。
なお，実行系イベントとは，イベント パネル上において次のイベントが対象です。

- ハードウエア・ブレーク
- 組み合わせブレークの詳細情報内における実行後ブレーク【E1】【E20】【EZ Emulator】
- トレースの詳細情報内における開始，および終了条件の実行後イベント注
- タイマの詳細情報内における開始，および終了条件の実行後イベント注

注 【シミュレータ】
実行前イベントとなります。

図 A.60 実行イベント詳細設定ダイアログ



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [カテゴリの説明]
- [機能ボタン]

[オープン方法]

- イベント パネル上において，詳細情報を編集したいハードウエア・ブレークにキャレットを移動したのち，コンテキスト・メニュー→［条件の編集 ...］を選択
- イベント パネル上において，詳細情報を編集したいトレース内の実行系イベントにキャレットを移動したのち，コンテキスト・メニュー→［条件の編集 ...］を選択

- イベント パネル上において，詳細情報を編集したいタイマ計測内の実行系イベントにキャレットを移動したのち，コンテキスト・メニュー→［条件の編集 ...］を選択

【E1】【E20】【EZ Emulator】

- イベント パネル上において，詳細情報を編集したい組み合わせブレーク内の実行系イベントにキャレットを移動したのち，コンテキスト・メニュー→［条件の編集 ...］を選択

［各エリアの説明］

(1) 詳細情報表示／変更エリア
イベント パネル上で選択している実行系イベントの詳細情報を，カテゴリのリスト形式で表示し，設定の変更を直接行うことができるエリアです。

(2) プロパティの説明エリア
詳細情報表示／変更エリアで選択したカテゴリやプロパティの簡単な説明を表示します。

［ツールバー］

	詳細情報表示／変更エリアにてカテゴリを表示にします。
	詳細情報表示／変更エリアにてカテゴリを非表示にしてプロパティ項目のみを昇順に並び替えます。

［カテゴリの説明］

(1) ［アドレス］

アドレス	実行系イベントが設定されているアドレスを表示，および指定します。	
	デフォルト	*選択したイベントに依存*
	変更方法	キーボードからの直接入力
	指定可能値	0x0 ～ 0xFFFFFFFF

(2) ［パス・カウント］

パス・カウント	パス・カウントを 10 進数で指定します。 パス・カウント分のイベント条件を満たした時点で，イベント成立となります。	
	デフォルト	*1*
	変更方法	キーボードからの直接入力
	指定可能値	-【E1】【E20】【EZ Emulator】 1 ～ 256 -【シミュレータ】 1 ～ 16383

注意 1. 【E1】【E20】【EZ Emulator】
パス・カウントは，全イベント（アクセス系イベント含む）中で 1 つのイベントに対してのみ 1 以外の値を指定可能です。
２つのイベントに 1 以外のパス・カウントを設定した場合，エラーとなります。

注意 2. 【E1(Serial) / E20(Serial) / EZ Emulator【RX100, RX200 シリーズ】】
パス・カウントは，1 以外の値を指定できません。

［機能ボタン］

ボタン	機能
OK	ダイアログで指定した詳細設定を実行系イベントに設定し，このダイアログをクローズします。
キャンセル	設定を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

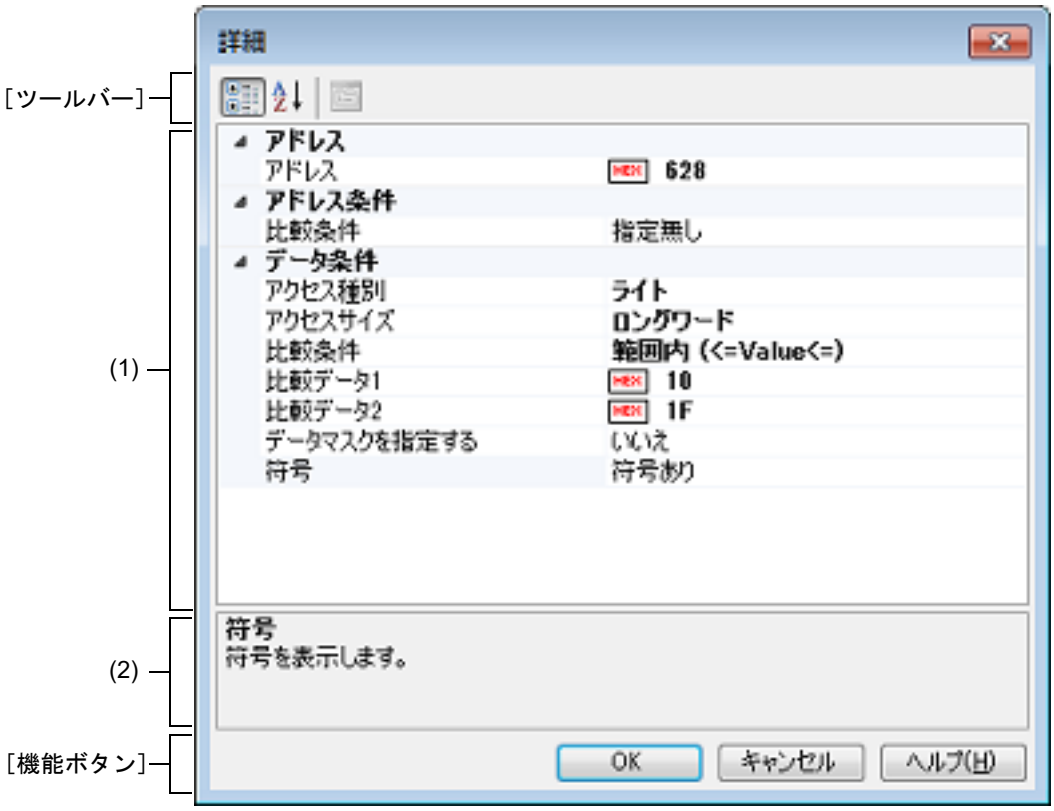
アクセスイベント詳細設定ダイアログ

イベント パネル上で選択したアクセス系イベントの詳細情報の表示，または変更を行います。
なお，アクセス系イベントとはイベント パネル上において次のイベントが対象です。

- 組み合わせブレークの詳細情報内におけるリード，ライト，およびリード／ライト【E1】【E20】【EZ Emulator】
- ハードウエア・ブレークの詳細情報内におけるリード，ライト，およびリード／ライト【シミュレータ】
- トレースの詳細情報内におけるリード，ライト，およびリード／ライト
- ポイント・トレースの詳細情報内におけるリード，ライト，およびリード／ライト
- タイマ計測の詳細情報内におけるリード，ライト，およびリード／ライト

図 A.61 アクセスイベント詳細設定ダイアログ【E1】【E20】【EZ Emulator】

図 A.62 アクセスイベント詳細設定ダイアログ【シミュレータ】



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [カテゴリの説明]
- [機能ボタン]

[オープン方法]

- イベント パネル上において，詳細情報を編集したいトレース内のアクセス系イベントにキャレットを移動したのち，コンテキスト・メニュー→［条件の編集 ...］を選択
- イベント パネル上において，詳細情報を編集したいポイント・トレース内のアクセス系イベントにキャレットを移動したのち，コンテキスト・メニュー→［条件の編集 ...］を選択
- イベント パネル上において，詳細情報を編集したいタイマ計測内のアクセス系イベントにキャレットを移動したのち，コンテキスト・メニュー→［条件の編集 ...］を選択

【E1】【E20】【EZ Emulator】

- イベント パネル上において，詳細情報を編集したい組み合わせブレーク内のアクセス系イベントにキャレットを移動したのち，コンテキスト・メニュー→［条件の編集 ...］を選択

【シミュレータ】

- イベント パネル上において，詳細情報を編集したいアクセス系のハードウエア・ブレークにキャレットを移動したのち，コンテキスト・メニュー→［条件の編集 ...］を選択

[各エリアの説明]

(1) 詳細情報表示／変更エリア
イベント パネル上で選択しているアクセス系イベントの詳細情報を，カテゴリのリスト形式で表示し，設定の変更を直接行うことができるエリアです。

(2) プロパティの説明エリア
詳細情報表示／変更エリアで選択したカテゴリやプロパティの簡単な説明を表示します。

[ツールバー]

	詳細情報表示／変更エリアにてカテゴリを表示にします。
	詳細情報表示／変更エリアにてカテゴリを非表示にしてプロパティ項目のみを昇順に並び替えます。

[カテゴリの説明]

表示および変更可能なプロパティは，使用するデバッグ・ツールにより異なります。

(1) [E1] [E20] [EZ Emulator] の場合
(2) [シミュレータ] の場合

(1) [E1] [E20] [EZ Emulator] の場合

(a) [アドレス]

アドレス	アクセス系イベントが設定されているアドレスを表示，および指定します。	
	デフォルト	*選択したイベントに依存*
	変更方法	キーボードからの直接入力
	指定可能値	0x0 ～ 0xFFFFFFFF

(b) [アドレス条件]

比較条件	アドレス比較条件を指定します。		
	デフォルト	*指定無し*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	指定無し	アドレス比較条件を指定しません。
		アドレス範囲[注1 注2]	アドレス範囲による比較条件を指定します。
		マスク付き比較	マスク付き比較条件を指定します。
範囲条件	アドレス比較範囲条件を指定します。 このプロパティは，[比較条件] プロパティにおいて [アドレス範囲] を選択したときのみ表示されます。		
	デフォルト	*範囲内 (<=Value<=)*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	範囲内 (<=Value<=)	範囲内のアクセスで条件成立します。
		範囲外 !(<=Value<=)	範囲外のアクセスで条件成立します。

終了アドレス	終了アドレスを指定します。 開始アドレスは，[アドレス] プロパティで表示された値となります。 このプロパティは，[比較条件] プロパティにおいて [アドレス範囲] を選択したときのみ表示されます。		
	デフォルト	*空欄*	
	変更方法	キーボードからの直接入力	
	指定可能値	0x0 ～ 0xFFFFFFFF	
マスク値	アドレスマスク値を 16 進数で指定します。注 3 このプロパティは，[比較条件] プロパティにおいて [マスク付き比較] を選択したときのみ表示されます。		
	デフォルト	*空欄*	
	変更方法	キーボードからの直接入力	
	指定可能値	0x0 ～ 0xFFFFFFFF	
比較	比較条件を指定します。 このプロパティは，[比較条件] プロパティにおいて [マスク付き比較] を選択したときのみ表示されます。		
	デフォルト	*指定値 (==)*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	指定値 (==)	アドレス比較条件と一致で成立します。
		指定値以外 (!=)	アドレス比較条件と不一致で成立します。

注 1. 【RX100, RX200 シリーズ】
アドレス範囲による比較条件は未サポートです。

注 2. 【RX600 シリーズ】
設定可能なアドレス範囲による比較条件は，1 点のみです。

注 3. マスクは，条件となるアドレス値に対してビット単位でマスク値 “ 0 ” を Don’t Care とします。
例) 0x1000 ～ 0x1FFF のアドレス条件を設定する場合
アドレス値：0x1000
マスク値：0xF000

(c) [データ条件]

アクセス種別	アクセス種別を指定します。		
	デフォルト	*リード/ ライト*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	リード	アクセス種別をリードアクセスとします。
		ライト	アクセス種別をライトアクセスとします。
		リード / ライト	アクセス種別をリード，およびライトアクセスとします。

アクセスサイズ	アクセスサイズを指定します。		
	デフォルト	*指定無し*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	指定無し	アクセスサイズを指定しません。 全てのアクセスサイズで成立します。
		バイト	アクセスサイズをバイトとします。
		ワード	アクセスサイズをワードとします。
		ロングワード	アクセスサイズをロングワードとします。
比較データ	比較データを 16 進数で指定します。		
	デフォルト	*空欄*	
	変更方法	キーボードからの直接入力	
	指定可能値	0x0 ～ 0xFFFFFFFF	
データマスクを指定する	データマスクを指定するか否かを指定します。		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	データマスクを指定します。
		いいえ	データマスクを指定しません。
マスク値	データマスクを 16 進数で指定します。注 このプロパティは，[データマスクを指定する] プロパティにおいて [はい] を指定した場合のみに表示されます。		
	デフォルト	*空欄*	
	変更方法	キーボードからの直接入力	
	指定可能値	0x0 ～ 0xFFFFFFFF	
比較条件	データ比較条件を指定します。 このプロパティは，[データマスクを指定する] プロパティにおいて [はい] を指定した場合のみに表示されます。		
	デフォルト	*指定値 (==)*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	指定値 (==)	データ比較条件が一致で成立します。
		指定値以外 (!=)	データ比較条件が不一致で成立します。

注 マスクは，条件となるデータ値に対してビット単位でマスク値 “ 0 ” を Don’t Care とします。

(d) [バス・マスタ]【RX71M, RX64M グループ】

バス・マスタ	バス・マスタを指定します。		
	デフォルト	*CPU*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	CPU	指定したデータアクセスが，CPU からのアクセスにより発生した場合に条件成立します。
		DMAC/DTC	指定したデータアクセスが，DMAC/DTC からのアクセスにより発生した場合に条件成立します。

注意 データアクセスのバス・マスタ選択機能が無いマイクロコントローラの場合，このプロパティは表示されません。

(e) [パス・カウント]

パス・カウント	パス・カウントを 10 進数で指定します。 パス・カウント分のイベント条件を満たした時点で，イベント成立となります。	
	デフォルト	*1*
	変更方法	キーボードからの直接入力
	指定可能値	1 ～ 256

注意 1. 【E1】【E20】【EZ Emulator】
パス・カウントは，全イベント（実行系イベント含む）中で 1 つのイベントに対してのみ 1 以外の値を指定可能です。
２つのイベントに 1 以外のパス・カウントを設定した場合，エラーとなります。

注意 2. 【E1(Serial) / E20(Serial) / EZ Emulator【RX100, RX200 シリーズ】】
パス・カウントは，1 以外の値を指定できません。

(2) [シミュレータ] の場合

(a) [アドレス]

アドレス	アクセス系イベントが設定されているアドレスを表示，および指定します。	
	デフォルト	*選択したイベントに依存*
	変更方法	キーボードからの直接入力
	指定可能値	0x0 ～ 0xFFFFFFFF

(b) [アドレス条件]

比較条件	アドレス比較条件を表示します。	
	デフォルト	*指定無し*
	変更方法	変更不可

(c) [データ条件]

アクセス種別	アクセス種別を指定します。		
	デフォルト	*選択したイベントに依存*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	リード	アクセス種別をリードアクセスとします。
		ライト	アクセス種別をライトアクセスとします。
		リード / ライト	アクセス種別をリード，およびライトアクセスとします。
アクセスサイズ	アクセスサイズを指定します。		
	デフォルト	*選択したイベントに依存*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	指定無し	アクセスサイズを指定しません。 全てのアクセスサイズで成立します。
		バイト	アクセスサイズをバイトとします。
		ワード	アクセスサイズをワードとします。
		ロングワード	アクセスサイズをロングワードとします。

比較条件	データ比較条件を指定します。 このプロパティは，［アクセスサイズ］プロパティにおいて［指定無し］を指定した場合は表示されません。		
	デフォルト	*一致 (==)*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	一致 (==)	データが指定値と一致で成立します。
		不一致 (!=)	データが指定値と不一致で成立します。
		符号反転	前回アクセス時のデータと今回アクセス時のデータで符号が反転した場合に成立します。注 1
		差分	前回アクセス時のデータと今回アクセス時のデータの差分が指定値を超えた場合に成立します。注 1
		指定値より大きい (>)	データが指定値より大で成立します。
		指定値より小さい (<)	データが指定値より小で成立します。
		指定値以上 (>=)	データが指定値以上で成立します。
		指定値以下 (<=)	データが指定値以下で成立します。
		範囲内 (<=Value<=)	データが指定値の範囲内で成立します。 （[比較データ 1] <= データ <= [比較データ 2]）
		範囲外 !(<=Value<=)	データが指定値の範囲外で成立します。 （データ < [比較データ 1] || [比較データ 2] < データ）
		指定無し	比較データを指定しません。
比較データ 1	比較データを 16 進数で指定します。 このプロパティは，［比較条件］プロパティにおいて以下を指定した場合に表示されます。 - [一致 (==)] - [不一致 (!=)] - [指定値より大きい (>)] - [指定値より小さい (<)] - [指定値以上 (>=)] - [指定値以下 (<=)] - [範囲内 (<=Value<=)] - [範囲外 !(<=Value<=)]		
	デフォルト	空欄	
	変更方法	キーボードからの直接入力	
	指定可能値	0x0 ～ 0xFFFFFFFF	

比較データ 2	比較データを 16 進数で指定します。 このプロパティは，［比較条件］において以下を指定した場合に表示されます。 - ［範囲内 (<=Value<=)］ - ［範囲外 !(<=Value<=)］		
	デフォルト	*空欄*	
	変更方法	キーボードからの直接入力	
	指定可能値	0x0 ～ 0xFFFFFFFF	
差分	比較データを 16 進数で指定します。 このプロパティは，［比較条件］において［差分］を指定した場合のみ表示されます。		
	デフォルト	*空欄*	
	変更方法	キーボードからの直接入力	
	指定可能値	0x0 ～ 0xFFFFFFFF	
データマスクを指定する	データマスクを指定するか否かを指定します。 このプロパティは，［比較条件］において以下を指定した場合に表示されます。 - ［一致 (==)］ - ［不一致 (!=)］ - ［指定値より大きい (>)］ - ［指定値より小さい (<)］ - ［指定値以上 (>=)］ - ［指定値以下 (<=)］ - ［範囲内 (<=Value<=)］ - ［範囲外 !(<=Value<=)］		
	デフォルト	*いいえ*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	データマスクを指定します。
		いいえ	データマスクを指定しません。
マスク値	データマスクを 16 進数で指定します。注 2 このプロパティは，[データマスクを指定する] プロパティにおいて [はい] を指定した場合のみに表示されます。		
	デフォルト	*空欄*	
	変更方法	キーボードからの直接入力	
	指定可能値	0x0 ～ 0xFFFFFFFF	

符号	比較するデータに符号があるか否かを指定します。 このプロパティは，［比較条件］において以下を指定した場合に表示されます。 - ［差分］ - ［指定値より大きい (>)］ - ［指定値より小さい (<)］ - ［指定値以上 (>=)］ - ［指定値以下 (<=)］ - ［範囲内 (<=Value<=)］ - ［範囲外 !(<=Value<=)］		
	デフォルト	*符号あり*	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	符号あり	符号ありでデータを比較します。
		符号なし	符号なしでデータを比較します。

注 1. ［符号反転］，および［差分］は前回データを比較するため，リセット後，および条件成立 1 回目の判定は常に不成立となります。

注 2. マスクは，条件となるデータ値に対してビット単位でマスク値 “ 0 ” を Don’t Care とします。

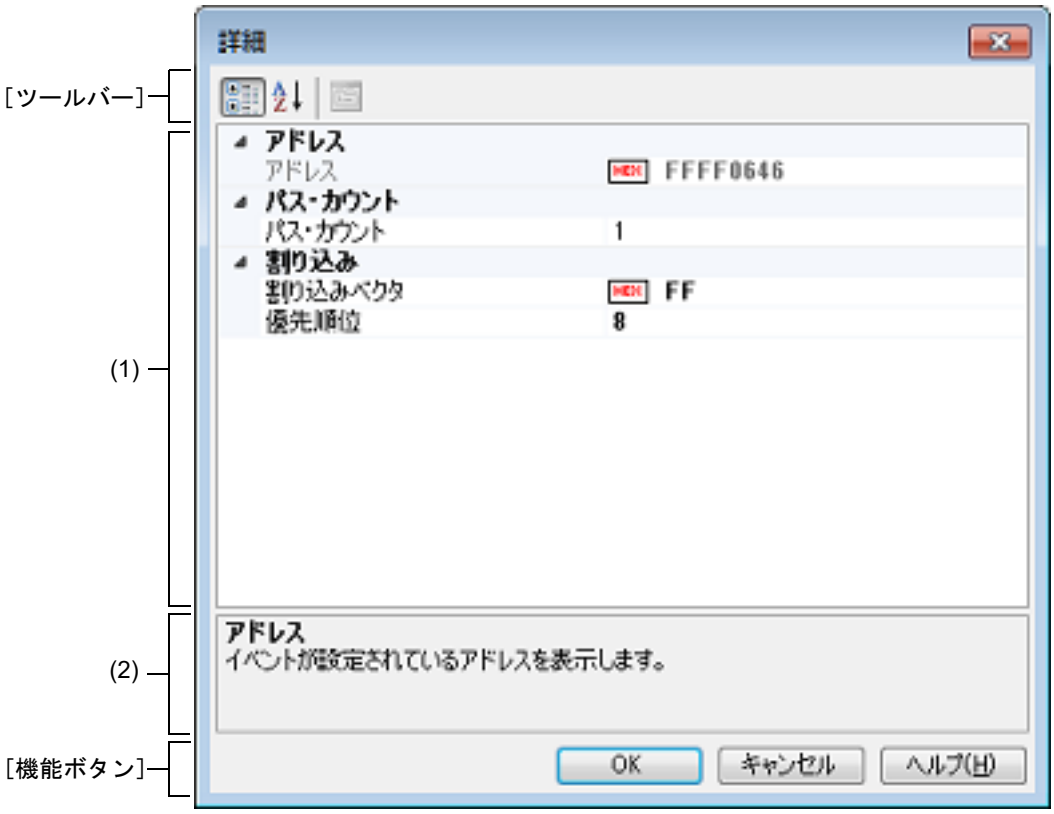
［機能ボタン］

ボタン	機能
OK	ダイアログで指定した詳細設定をアクセス系イベントに設定し，このダイアログをクローズします。
キャンセル	設定を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

割り込みイベント詳細設定ダイアログ【シミュレータ】

イベント パネル上で選択した割り込みイベントの詳細情報の表示，および変更を行います。

図 A.63 割り込みイベント詳細設定ダイアログ【シミュレータ】



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [カテゴリの説明]
- [機能ボタン]

[オープン方法]

- イベント パネル上において，詳細情報を編集したい割り込みイベントにキャレットを移動したのち，コンテキスト・メニュー→［条件の編集 ...］を選択

[各エリアの説明]

(1) 詳細情報表示／変更エリア
イベント パネル上で選択している割り込みイベントの詳細情報を，カテゴリのリスト形式で表示し，設定の変更を直接行うことができるエリアです。

(2) プロパティの説明エリア
詳細情報表示／変更エリアで選択したカテゴリやプロパティの簡単な説明を表示します。

[ツールバー]

	詳細情報表示／変更エリアにてカテゴリを表示にします。
	詳細情報表示／変更エリアにてカテゴリを非表示にしてプロパティ項目のみを昇順に並び替えます。

[カテゴリの説明]

(1) [アドレス]

アドレス	割り込みイベントが設定されているアドレスを表示します。 ここでは，アドレス値を指定することはできません。	
	デフォルト	*選択したイベントに依存*
	変更方法	変更不可

(2) [パス・カウント]

パス・カウント	パス・カウントを 10 進数で指定します。 パス・カウント分のイベント条件を満たした時点で，イベント成立となります。	
	デフォルト	*1*
	変更方法	キーボードからの直接入力
	指定可能値	1 ～ 16383

(3) [割り込み]

割り込みベクタ	割り込みベクタを指定します。	
	デフォルト	*選択したイベントに依存*
	変更方法	キーボードからの直接入力
	指定可能値	0x00 ～ 0xFF
優先順位	割り込み優先順位を指定します。	
	デフォルト	*選択したイベントに依存*
	変更方法	キーボードからの直接入力
	指定可能値	-【RX610 グループ】 1 ～ 8 8 を指定した場合は，高速割り込みとして動作します。 -【RX610 グループ以外】 1 ～ 16 16 を指定した場合は，高速割り込みとして動作します。

［機能ボタン］

ボタン	機能
OK	ダイアログで指定した詳細設定を割り込みイベントに設定し，このダイアログをクローズします。
キャンセル	設定を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

COM ポート設定 ダイアログ

マイクロコントローラからの通信をリダイレクトするホスト・マシン上の COM ポートを設定します。

図 A.64 COM ポート設定 ダイアログ

ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- デバッグ・コンソール パネルにおいて，コンテキスト・メニュー→［COM ポートの設定 ...］を選択

[各エリアの説明]

(1) ［COM ポート］エリア
リダイレクト先の COM ポートを選択します。ホスト・マシン上に利用可能な COM ポートがない場合，このドロップダウン・リストは空になります。

(2) ［ボーレート］エリア
リダイレクト先の COM ポートとのボーレートを選択します。
ドロップダウン・リストには，次のボーレートが表示されます。“ システム ” を選択した場合は，デバイス・マネージャの設定に従います。
システム，1200，2400，4800，9600，19200，38400，115200

[機能ボタン]

ボタン	機能
接続／切断	COM ポートに接続（デフォルト）／切断します。 COM ポートに接続状態の場合は，ボタンの表示名は “ 切断 ” になります。 COM ポートから切断状態の場合は，ボタンの表示名は “ 接続 ” になります。
キャンセル	COM ポートの設定を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

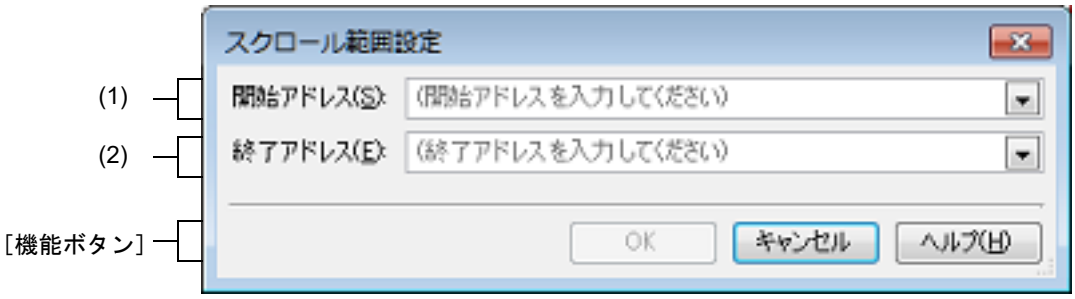
スクロール範囲設定 ダイアログ

メモリ パネル／逆アセンブル パネルの垂直スクロール・バーのスクロール範囲の設定を行います。
適正な範囲を設定することにより，パネルの垂直スクロール・バー上のスライダの大きさが変化し，マウスによるドラッグなどの操作性が向上します。

注意 このダイアログによりスクロール範囲を設定したのち，ライン・アセンブルなどの実行により指定したアドレス式が表すアドレスに変更が生じても，スクロール範囲の修正は行いません。

備考 ［Page Up］／［Page Down］／［↑］／［↓］キー，スクロール・バー端のボタン，またはジャンプ系のメニュー項目の選択による移動は，スクロール範囲外でも可能です。

図 A.65 スクロール範囲設定 ダイアログ



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］
- ［機能ボタン］

［オープン方法］

- メモリ パネルにおいて，ツールバーの [表示] → ボタンをクリック
- メモリ パネルにおいて，コンテキスト・メニューの［表示］→［スクロール範囲の設定 ...］を選択
- 逆アセンブル パネルにおいて，ツールバーの [表示] → ボタンをクリック
- 逆アセンブル パネルにおいて，コンテキスト・メニューの［表示］→［スクロール範囲の設定 ...］を選択

［各エリアの説明］

(1) ［開始アドレス］エリア
スクロールする範囲の開始アドレスを指定します。
アドレス式をテキスト・ボックスに直接入力するか（最大指定文字数：1024 文字），またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
なお，ドロップダウン・リスト内の“全範囲”を指定すると，スクロール範囲の設定は行いません（範囲は制限されません）。

備考 このテキスト・ボックスで［Ctrl］+［Space］キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます（「2.20.2 シンボル名の入力補完機能」参照）。

(2) ［終了アドレス］エリア
スクロールする範囲の終了アドレスを指定します。
アドレス式をテキスト・ボックスに直接入力するか（最大指定文字数：1024 文字），またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
ただし，［開始アドレス］において，“全範囲”を指定している場合，このエリアは無効となります。

備考 このテキスト・ボックスで［Ctrl］+［Space］キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます（「2.20.2 シンボル名の入力補完機能」参照）。

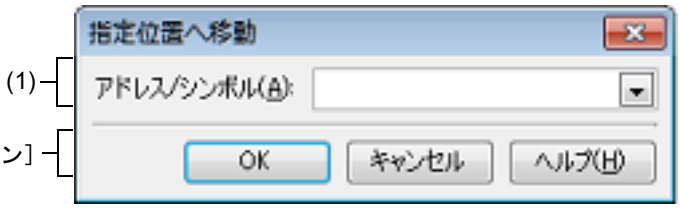
［機能ボタン］

ボタン	機能
OK	指定したスクロール範囲を対象パネルに設定し，開始アドレスを表示の先頭として対象パネルにキャレットを移動します。
キャンセル	設定を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

指定位置へ移動 ダイアログ

指定した位置にキャレットを移動します。

図 A.66 指定位置へ移動 ダイアログ



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- [編集] メニュー→ [移動 ...] を選択
- 逆アセンブル パネルにおいて，コンテキスト・メニューの [移動 ...] を選択
- IOR パネルにおいて，コンテキスト・メニューの [移動 ...] を選択

[各エリアの説明]

(1) [アドレス / シンボル] ／ [IOR] エリア
キャレットを移動したい箇所を指定します。
テキスト・ボックスに直接入力するか（最大指定文字数：1024 文字），またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
対象となるパネルにより，指定内容は次のように異なります。

対象パネル	指定内容
逆アセンブル パネル	アドレス式
IOR パネル	I/O レジスタ名

備考 逆アセンブル パネルよりこのダイアログをオープンした場合，このテキスト・ボックスで [Ctrl] + [Space] キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます（「2.20.2 シンボル名の入力補完機能」参照）。

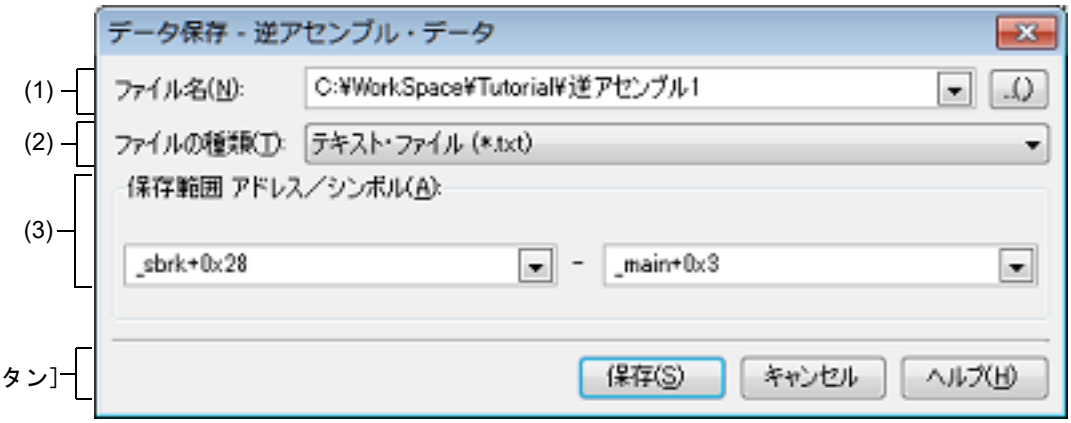
[機能ボタン]

ボタン	機能
OK	指定した位置を表示の先頭として対象パネルにキャレットを移動します。
キャンセル	設定を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

データ保存 ダイアログ

逆アセンブル パネル／メモリ パネル／トレース パネルの表示内容，およびアップロード・データの保存（「2.5.3 アップロードを実行する」参照）を行います。
なお，このダイアログは，デバッグ・ツールと接続時のみオープンすることができます。

図 A.67 データ保存 ダイアログ



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- 逆アセンブル パネルにフォーカスがある状態で，[ファイル] メニュー→ [名前を付けて逆アセンブル・データを保存 ...] を選択
- メモリ パネルにフォーカスがある状態で，[ファイル] メニュー→ [名前を付けてメモリ・データを保存 ...] を選択
- トレース パネルにフォーカスがある状態で，[ファイル] メニュー→ [名前を付けてトレース・データを保存 ...] を選択
- [デバッグ] メニュー→ [デバッグ・ツールからアップロード ...] を選択

[各エリアの説明]

(1) [ファイル名] エリア
保存するファイル名を指定します。
テキスト・ボックスに直接入力するか（最大指定文字数：259 文字），またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
また，[...] ボタンをクリックすることでオープンするデータ保存ファイルを選択 ダイアログにより，ファイルを選択することもできます。
なお，パス情報を含まずファイル名のみを指定した場合は，プロジェクト・フォルダが対象となります。

(2) [ファイルの種類] エリア
保存するファイルの形式を次のドロップダウン・リストにより選択します。
保存する対象により，選択できるファイルの形式が次のように異なります。

(a) パネルの表示内容を保存する場合

テキスト・ファイル (*.txt)	テキスト形式（デフォルト）
CSV(カンマ区切り) (*.csv)	CSV 形式注

注 各データを “,” で区切り保存します。
なお，データ内に “,” が含まれている際の不正形式を避けるため，各データを “""” （ダブルクォーテーション）で括り出力します。

(b) アップロード・データを保存する場合
選択できるファイル形式は，「表 2.3 アップロード可能なファイル形式」を参照してください。

(3) ［保存範囲 *xxx*］エリア
ファイルに保存する際の保存範囲を指定します。
それぞれのテキスト・ボックスに直接入力するか，またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
保存する対象により，指定方法が次のように異なります。

保存対象	説明
逆アセンブル パネル	保存するアドレス範囲を，開始アドレスと終了アドレスで指定します。 16 進数の数値，またはアドレス式による入力が可能です。 パネル上で範囲選択している場合は，デフォルトでその選択範囲が指定されます。 範囲選択していない場合は，現在のパネルの表示範囲が指定されます。
メモリ パネル	保存するメモリ範囲を，開始アドレスと終了アドレスで指定します。 16 進数の数値，またはアドレス式による入力が可能です。 範囲選択していない場合は，現在のパネルの表示範囲が指定されます。
トレース パネル	- 保存範囲を指定する場合 保存するトレース範囲を開始トレース番号[注]と終了トレース番号で指定します。 10 進数の数値のみ入力が可能です。 - すべてのトレース・データを保存する場合 左側のドロップダウン・リストにより，［すべてのトレース・データ］を選択します。右側のテキスト・ボックスが無効となり，現在取得しているトレース・データのすべてが保存の対象となります。 デフォルトでは，現在のパネルの表示範囲が指定されます。
アップロード・データ	保存するメモリ範囲を開始アドレスと終了アドレスで指定します。 16 進数の数値，またはアドレス式による入力が可能です。

注 トレース パネル上の［番号］エリアに表示されている番号を示します。

備考 このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより，現在のキャレット位置のシンボル名を補完することができます（「2.20.2 シンボル名の入力補完機能」参照）。

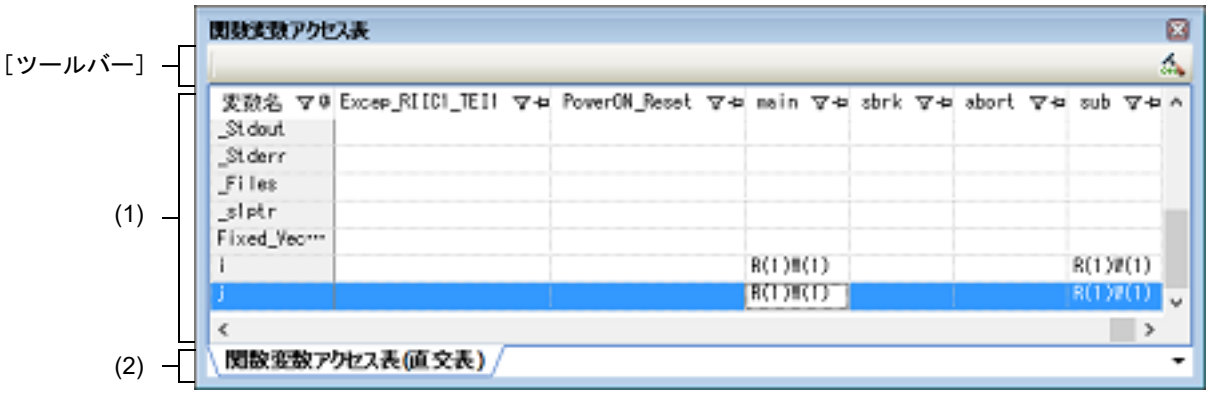
［機能ボタン］

ボタン	機能
保存	指定したファイルに，指定した形式でデータを保存します。
キャンセル	データ保存の設定を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

関数変数アクセス表 パネル

変数にアクセスしている関数を直交表で表示するパネルです。

図 A.68 関数変数アクセス表 パネル



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]

[オープン方法]

- [表示] メニュー→ [関数変数アクセス表] を選択

[各エリアの説明]

(1) 表エリア

(a) [関数変数アクセス表 (直交表)] タブ
アクティブ・プロジェクトにおいて，変数にアクセスしている関数を直交表で表示します。
この情報はクロス・リファレンス情報とマップ情報を用いて作成しています。情報が表示されない場合は，ツールバーの右端の （直交表を生成するためのビルド・オプションを有効にしてビルドするボタン）をクリックして情報を生成してください。

また，C ソースを静的解析して得られた関数の変数へのアクセス状態を表示します。

- “R” と記載されているセルは変数値を読み込んだことを表します。かっこ内の数字は関数内での変数の読み込み箇所の数を示します。
- “W” と記載されているセルは変数値を書き込んだことを表します。かっこ内の数字は関数内での変数の書き込み箇所の数を示します。

変数名，または変数へのアクセス状態を表しているセルをダブルクリックすることで，その定義箇所をエディタで表示します。

(2) タブ選択エリア
変数にアクセスしている関数を直交表で表示する [関数変数アクセス表 (直交表)] タブがあります。

[ツールバー]

ツールバーは，コンパイラがクロスリファレンスの生成に対応している場合のみ表示します。

	クロス・リファレンスの生成に関するビルド・オプションが無効の場合，有効にしてビルドします。その際，テキスト・エディタで編集中のファイルがある場合は，該当ファイルを保存します。

B. 入出力機能

ここでは，低水準インタフェース・ルーチン（アセンブリ言語部分）におけるシミュレータ用の入出力機能別の指定内容について説明します。
シミュレータでは，標準入出力，およびファイル入出力用の機能を提供しています。
シミュレータ用の入出力機能は，プロパティ パネル上の［デバッグ・ツール設定］タブの［ストリーム入出力］【シミュレータ】カテゴリ内の［ストリーム入出力モードを選択］プロパティにおいて，［シミュレータモード］を指定した場合使用できます。

B.1 標準入出力／ファイル入出力

低水準インタフェース・ルーチンから呼ばれる，入出力を行うファイルの内容を入出力機能用のプログラムに置き換えます。
標準入出力用の機能 GETC，PUTC については，1 文字の入出力を行う関数 “charput”，“charget”（_charput，_charget）をデバッグ・コンソール機能用のプログラムに置き換え，デバッグ・コンソール パネル上へデータの入出力を行います。
以下に入出力機能の一覧を示します。

表 B.1 入出力機能

分類	機能名	説明
標準入出力	GETC	標準入力から 1 バイト入力します。
	PUTC	標準出力へ 1 バイト出力します。
ファイル入出力	FOPEN	ファイルをオープンします。
	FCLOSE	ファイルをクローズします。
	FGETC	ファイルから 1 バイト入力します。
	FPUTC	ファイルへ 1 バイト出力します。
	FEOF	ファイルの終端をチェックします。
	FSEEK	ファイル・ポインタを指定された位置に移動します。
	FTELL	ファイル・ポインタの現在位置を取得します。

備考 低水準インタフェース・ルーチンについては，「CC-RX コンパイラ ユーザーズマニュアル」を参照してください。

シミュレータの入出力機能を実現するためには，まず，プロパティ パネルの［デバッグ・ツール設定］タブ上の［ストリーム入出力］【シミュレータ】カテゴリ内の［ストリーム入出力用アドレス］プロパティにおいて，入出力用の特定のアドレスを指定します。0 ～ “ アドレス空間の終了アドレス ” の範囲のアドレス式を直接入力で指定してください。

図 B.1 ［ストリーム入出力］カテゴリ

シミュレータは，プログラムの命令を実行中に，指定したアドレスへのサブルーチン分岐命令（BSR，JSR）を検出すると，R1，R2 レジスタの内容をパラメータとして入出力処理を行います。
したがって，入出力処理を行う前にプログラム内に次の設定をしておきます。

- 機能コード（R1 レジスタ）

MSB 1 バイト	1 バイト	1 バイト	1 バイト LSB
0x01	機能コード	0x00	0x00

- パラメータ・ブロックのアドレス（R2 レジスタ）
  パラメータ・ブロックの内容は，各入出力機能の説明を参照してください。

  MSB LSB

- パラメータ・ブロックおよびストリーム入出力バッファ領域の確保
  パラメータ・ブロックの各パラメータにアクセスする場合は，該当するパラメータのサイズでアクセスしてください。

GETC

標準入力から 1 バイト入力します。

[機能コード（R1 レジスタ）]

1 バイト	1 バイト	1 バイト	1 バイト
0x01	0x21	0x00	0x00

[パラメータ・ブロック（R2 レジスタ）]

	1 バイト	1 バイト
+0	*入力バッファ先頭アドレス*	
+2		

[パラメータ]

パラメータ	説明
入力バッファ先頭アドレス（入力）	入力データを書き込むバッファの先頭アドレス

[詳細説明]

- 標準入力から 1 バイト入力します。

[使用例]

```
_charget:
        MOV.L       #1210000h,R1     ; GETC の機能コードを R1 に設定
        MOV.L       #IO_BUF,R2       ; 入力バッファ先頭アドレスを R2 に設定
        MOV.L       #PARM,R3         ; パラメータ・ブロックのアドレスを R3 に設定
        MOV.L       R2,[R3]          ; 入力バッファ先頭アドレスをパラメータ・ブロックに設定
        MOV.L       R3,R2            ; R3（パラメータ・ブロックのアドレス）を R2 に設定
        MOV.L       #SIM_IO,R3       ; システム・コールのアドレスを R3 に設定
        JSR         R3               ; システム・コール
        MOV.L       #IO_BUF,R2       ; 入力バッファ先頭アドレスを R2 に設定
        MOVU.B      [R2],R1          ; 入力バッファ先頭 1 バイト(取得した 1 バイト文字)を R1 に設定
        RTS                          ; 呼び出し元のアドレスに戻る

         .SECTION  B,DATA,ALIGN=4
PARM:       .BLKL   1                ; パラメータ・ブロック領域
         .SECTION  B_1,DATA
IO_BUF:     .BLKL   1                ; 入出力バッファ
```

PUTC

標準出力へ 1 バイト出力します。

[機能コード（R1 レジスタ）]

1 バイト	1 バイト	1 バイト	1 バイト
0x01	0x22	0x00	0x00

[パラメータ・ブロック（R2 レジスタ）]

	1 バイト	1 バイト
+0 +2	*出力バッファ先頭アドレス*	

[パラメータ]

パラメータ	説明
出力バッファ先頭アドレス（入力）	出力データを格納しているバッファの先頭アドレス

[詳細説明]

- 標準出力へ 1 バイト出力します。

[使用例]

```
_charput:
        MOV.L       #IO_BUF,R2      ; 出力バッファ先頭アドレスを R2 に設定
        MOV.B       R1,[R2]         ; R1（出力文字）を出力バッファに設定
        MOV.L       #1220000h,R1    ; PUTC の機能コードを R1 に設定
        MOV.L       #PARM,R3        ; パラメータ・ブロックのアドレスを R3 に設定
        MOV.L       R2,[R3]         ; 出力バッファ先頭アドレスを出力バッファに設定
        MOV.L       R3,R2           ; R3（パラメータ・ブロックのアドレス）を R2 に設定
        MOV.L       #SIM_IO,R3      ; システム・コールのアドレスを R3 に設定
        JSR         R3              ; システム・コール
        RTS                         ; 呼び出し元のアドレスに戻る

         .SECTION  B,DATA,ALIGN=4
PARM:       .BLKL   1               ; パラメータ・ブロック領域
         .SECTION  B_1,DATA
IO_BUF:     .BLKL   1               ; 入出力バッファ
```

FOPEN

ファイルをオープンします。

[機能コード（R1 レジスタ）]

1 バイト	1 バイト	1 バイト	1 バイト
0x01	0x25	0x00	0x00

[パラメータ・ブロック（R2 レジスタ）]

	1 バイト	1 バイト
+0	*実行結果*	*ファイル番号*
+2	*オープン・モード*	未使用
+4	*ファイル名先頭アドレス*	
+6		

[パラメータ]

パラメータ	説明
実行結果（出力）	0：正常終了 -1：エラー
ファイル番号（出力）	オープン処理以降のファイルへのアクセスで使用する番号
オープン・モード（入力）	0x00 ："r" 0x01 ："w" 0x02 ："a" 0x03 ："r+" 0x04 ："w+" 0x05 ："a+" 0x10 ："rb" 0x11 ："wb" 0x12 ："ab" 0x13 ："r+b" 0x14 ："w+b" 0x15 ："a+b" 各モードの内容は以下の通りです。 "r" ：読み出し用にオープンします。 "w" ：空ファイルを書き込み用にオープンします。 "a" ：ファイルの最後から書き込み用にオープンします。 "r+" ：読み出し，書き込み用にオープンします。 "w+" ：空ファイルを読み出し，書き込み用にオープンします。 "a+" ：読み出し追加用にオープンします。 "b" ：バイナリ・モードでオープンします。
ファイル名先頭アドレス（入力）	ファイル名を格納している領域の先頭アドレス

[詳細説明]

- [FOPEN] によってファイルをオープンすると，ファイル番号を返します。
  以後のファイル入出力，ファイルのクローズ等では，このファイル番号を用います。
- 同時にオープンできる最大ファイル数は 256 です。

［使用例］

```
_fileopen:
        MOV.L       R2,R5             ; R2（オープン・モード）を R5 に設定
        MOV.L       #PARM,R2          ; パラメータ・ブロックのアドレスを R2 に設定
        MOV.L       R1,4h:5[R2]       ; R1（ファイル名先頭アドレス）を R2 ＋ 4 バイトに設定
        MOV.B       R5,2h:5[R2]       ; R5 を R2 ＋ 2 バイト（オープン・モード）に設定
        MOV.L       #01250000h,R1     ; FOPEN の関数コードを R1 に設定
        MOV.L       #SIM_IO,R5        ; システム・コールのアドレスを R5 に設定
        JSR         R5                ; システム・コール
        NOP
        MOV.L       #PARM,R2          ; パラメータ・ブロックを R2 に設定
        MOV.B       1h:5[R2],R1       ; R2 ＋ 1 バイト（ファイル番号）を R1 に設定
        MOV.B       R1,[R3]           ; R1 を R3（ファイル番号のポインタ）の指す先に設定
        MOV.B       [R2],R1           ; R2 の先頭 1 バイト（実行結果）を R1 に設定
        RTS                           ; 呼び出し元のアドレスに戻る

         .SECTION  B,DATA,ALIGN=4
PARM:       .BLKL   2                 ; パラメータ・ブロック領域
```

FCLOSE

ファイルをクローズします。

[機能コード（R1 レジスタ）]

1 バイト	1 バイト	1 バイト	1 バイト
0x01	0x06	0x00	0x00

[パラメータ・ブロック（R2 レジスタ）]

	1 バイト	1 バイト
+0	*実行結果*	ファイル番号

[パラメータ]

パラメータ	説明
実行結果（出力）	0：正常終了 -1：エラー
ファイル番号（入力）	ファイルのオープン時に返す番号

[詳細説明]

- ファイルをクローズします。

[使用例]

```
_fileclose:
        MOV.L       #PARM,R2          ; パラメータ・ブロックのアドレスを R2 に設定
        MOV.B       R1,1h:5[R2]       ; R1（ファイル番号）を R2 ＋ 1 バイトに設定
        MOV.L       #01060000h,R1     ; FCLOSE の関数コードを R1 に設定
        MOV.L       #SIM_IO,R3        ; システム・コールのアドレスを R3 に設定
        JSR         R3                ; システム・コール
        NOP
        MOV.L       #PARM,R2          ; パラメータ・ブロックを R2 に設定
        MOV.B       [R2],R1           ; R2 の先頭 1 バイト（実行結果）を R1 に設定
        RTS                           ; 呼び出し元のアドレスに戻る

         .SECTION  B,DATA,ALIGN=4
PARM:       .BLKL   1                 ; パラメータ・ブロック領域
```

FGETC

ファイルから 1 バイトのデータを読み出します。

[機能コード（R1 レジスタ）]

1 バイト	1 バイト	1 バイト	1 バイト
0x01	0x27	0x00	0x00

[パラメータ・ブロック（R2 レジスタ）]

	1 バイト	1 バイト
+0	*実行結果*	*ファイル番号*
+2	未使用	
+4	*入力バッファ先頭アドレス*	
+6		

[パラメータ]

パラメータ	説明
実行結果（出力）	0：正常終了 -1：EOF 検出
ファイル番号（入力）	ファイルのオープン時に返す番号
入力バッファ先頭アドレス（入力）	入力データを書き込むバッファの先頭アドレス

[詳細説明]

- ファイルから 1 バイトのデータを読み出します。

[使用例]

```
_fcharget:
        MOV.L       R2,R5             ; R2（ファイル番号）を R5 に設定
        MOV.L       #PARM,R2          ; パラメータ・ブロックのアドレスを R2 に設定
        MOV.L       R1,4h:5[R2]       ; R1（入力バッファ）を R2 ＋ 4 バイトに設定
        MOV.B       R5,1h:5[R2]       ; R5 を R2 ＋ 1 バイト（ファイル番号）に設定
        MOV.L       #01270000h,R1     ; FGETC の関数コードを R1 に設定
        MOV.L       #SIM_IO,R3        ; システム・コールのアドレスを R3 に設定
        JSR         R3                ; システム・コール
        NOP
        MOV.L       #PARM,R1          ; パラメータ・ブロックを R1 に設定
        MOV.B       [R1],R1           ; R1 の先頭 1 バイト（実行結果）を R1 に設定
        RTS                           ; 呼び出し元のアドレスに戻る

         .SECTION  B,DATA,ALIGN=4
PARM:       .BLKL   2                 ; パラメータ・ブロック領域
```

FPUTC

ファイルへ 1 バイトのデータを書き込みます。

[機能コード（R1 レジスタ）]

1 バイト	1 バイト	1 バイト	1 バイト
0x01	0x28	0x00	0x00

[パラメータ・ブロック（R2 レジスタ）]

	1 バイト	1 バイト
+0	*実行結果*	*ファイル番号*
+2	未使用	
+4	*出力バッファ先頭アドレス*	
+6		

[パラメータ]

パラメータ	説明
実行結果（出力）	0：正常終了 -1：エラー
ファイル番号（入力）	ファイルのオープン時に返す番号
出力バッファ先頭アドレス（入力）	出力データを格納しているバッファの先頭アドレス

[詳細説明]

- ファイルへ 1 バイトのデータを書き込みます。

[使用例]

```
_fcharput:
        MOV.L       R2,R5             ; R2（ファイル番号）を R5 に設定
        MOV.L       #PARM,R2          ; パラメータ・ブロックのアドレスを R2 に設定
        MOV.L       #IO_BUF,R4        ; 出力バッファ先頭アドレスを R4 に設定
        MOV.L       R4,4h:5[R2]       ; R4（出力バッファ）を R2+4 バイトに設定
        MOV.B       R1,[R4]           ; R1（出力文字）を R4 の指す先（出力バッファ）に設定
        MOV.B       R5,1h:5[R2]       ; R5 を R2+1 バイト（ファイル番号）に設定
        MOV.L       #01280000h,R1     ; FPUTC の関数コードを R1 に設定
        MOV.L       #SIM_IO,R3        ; システム・コールのアドレスを R3 に設定
        JSR         R3                ; システム・コール
        NOP
        MOV.L       #PARM,R1          ; パラメータ・ブロックのアドレスを R1 に設定
        MOV.B       [R1],R1           ; R1 の先頭 1 バイト（実行結果）を R1 に設定
        RTS                           ; 呼び出し元のアドレスに戻る

         .SECTION  B,DATA,ALIGN=4
PARM:       .BLKL   2                 ; パラメータ・ブロック領域
         .SECTION  B_1,DATA
IO_BUF:     .BLKL   1                 ; 入出力バッファ
```

FEOF

ファイルの終端をチェックします。

[機能コード（R1 レジスタ）]

1 バイト	1 バイト	1 バイト	1 バイト
0x01	0x0B	0x00	0x00

[パラメータ・ブロック（R2 レジスタ）]

	1 バイト	1 バイト
+0	*実行結果*	*ファイル番号*

[パラメータ]

パラメータ	説明
実行結果（出力）	0：EOF でない -1：EOF 検出
ファイル番号（入力）	ファイルのオープン時に返す番号

[詳細説明]

- ファイルの終端をチェックします。

[使用例]

```
_fpeof:
        MOV.L       #PARM,R2          ; パラメータ・ブロックのアドレスを R2 に設定
        MOV.B       R1,1h:5[R2]       ; R1（ファイル番号）を R2 ＋ 1 バイトに設定
        MOV.L       #010B0000h,R1     ; FEOF の関数コードを R1 に設定
        MOV.L       #SIM_IO,R3        ; システム・コールのアドレスを R3 に設定
        JSR         R3                ; システム・コール
        NOP
        MOV.L       #PARM,R2          ; パラメータ・ブロックを R2 に設定
        MOV.B       [R2],R1           ; R2 の先頭 1 バイト（実行結果）を R1 に設定
        RTS                           ; 呼び出し元のアドレスに戻る

         .SECTION  B,DATA,ALIGN=4
PARM:       .BLKL   1                 ; パラメータ・ブロック領域
```

FSEEK

ファイル・ポインタを指定された位置に移動します。

[機能コード（R1 レジスタ）]

1 バイト	1 バイト	1 バイト	1 バイト
0x01	0x0C	0x00	0x00

[パラメータ・ブロック（R2 レジスタ）]

	1 バイト	1 バイト
+0	*実行結果*	*ファイル番号*
+2	*ディレクション*	未使用
+4	*オフセット*	
+6		

[パラメータ]

パラメータ	説明
実行結果（出力）	0：正常終了 -1：エラー
ファイル番号（入力）	ファイルのオープン時に返す番号
ディレクション（入力）	0：オフセットはファイルの先頭からのバイト数 1：オフセットは現在のファイル・ポインタからのバイト数 2：オフセットはファイルの最後尾からのバイト数
オフセット（入力）	ディレクションで指定した位置からのバイト数

[詳細説明]

- ファイル・ポインタを指定された位置に移動します。

［使用例］

```
_fpseek:
        MOV.L       R2,R5             ; R2（オフセット）を R5 に設定
        MOV.L       #PARM,R2          ; パラメータ・ブロックのアドレスを R2 に設定
        MOV.L       R5,4h:5[R2]       ; R5 を R1 ＋ 4 バイト（オフセット）に設定
        MOV.B       R1,1h:5[R2]       ; R1（ファイル番号）を R1 ＋ 1 バイトに設定
        MOV.B       R3,2h:5[R2]       ; R3（ディレクション）を R2 ＋ 2 バイト（ディレクション）
                                      ; に設定
        MOV.L       #010C0000h,R1     ; FSEEK の関数コードを R1 に設定
        MOV.L       #SIM_IO,R5        ; システム・コールのアドレスを R5 に設定
        JSR         R5                ; システム・コール
        NOP
        MOV.L       #PARM,R1          ; パラメータ・ブロックのアドレスを R1 に設定
        MOV.B       [R1],R1           ; R1 の先頭 1 バイト（実行結果）を R1 に設定
        RTS                           ; 呼び出し元のアドレスに戻る

         .SECTION  B,DATA,ALIGN=4
PARM:       .BLKL   2                 ; パラメータ・ブロック領域
```

FTELL

ファイル・ポインタの現在位置を取得します。

[機能コード（R1 レジスタ）]

1 バイト	1 バイト	1 バイト	1 バイト
0x01	0x0D	0x00	0x00

[パラメータ・ブロック（R2 レジスタ）]

	1 バイト	1 バイト
+0	*実行結果*	*ファイル番号*
+2	未使用	
+4		
+6	*オフセット*	

[パラメータ]

パラメータ	説明
実行結果（出力）	0：正常終了 -1：エラー
ファイル番号（入力）	ファイルのオープン時に返す番号
オフセット（出力）	現在のファイル・ポインタの位置（ファイル先頭からのバイト数）

[詳細説明] ファイル・ポインタ

- の現在位置を取得します。

[使用例]

```
_fptell:
        MOV.L       R2,R5           ; R2（オフセット）を R5 に設定
        MOV.L       #PARM,R2        ; パラメータ・ブロックのアドレスを R2 に設定
        MOV.B       R1,1h:5[R2]     ; R1（ファイル番号）を R2 ＋ 1 バイトに設定
        MOV.L       #010D0000h,R1   ; FTELL の関数コードを R1 に設定
        MOV.L       #SIM_IO,R3      ; システム・コールのアドレスを R3 に設定
        JSR         R3              ; システム・コール
        NOP
        MOV.L       #PARM,R2        ; パラメータ・ブロックのアドレスを R2 に設定
        MOV.L       4h:5[R2],R1     ; R2 ＋ 4 バイト（新しいオフセット値）を R1 に設定
        MOV.L       R1,[R5]         ; R1 を R5（オフセットのポインタ）の先に設定
        MOV.B       [R2],R1         ; R2 の先頭 1 バイト（実行結果）を R1 に設定
        RTS                         ; 呼び出し元のアドレスに戻る

         .SECTION  B,DATA,ALIGN=4
PARM:       .BLKL   2               ; パラメータ・ブロック領域
```

改訂記録

Rev.	発行日	改定内容	
		ページ	ポイント
1.00	2017.06.01	-	初版発行

CS+ V6.00.00 ユーザーズマニュアル
RX デバッグ・ツール編

発行年月日　2017年 6月 1日　Rev.1.00

発行　　ルネサス エレクトロニクス株式会社
　　〒135-0061　東京都江東区豊洲3-2-24（豊洲フォレシア）

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 エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■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：https://www.renesas.com/contact/

© 2017 Renesas Electronics Corporation. All rights reserved.
Colophon 3.1

CS+ V6.00.00

RENESAS

ルネサス エレクトロニクス株式会社

R20UT3995JJ0100