

CubeSuite+ V2.01.00

統合開発環境

User's Manual

ユーザーズマニュアル RH850 デバッグ編

対象デバイス RH850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。

Rev.1.00 2013.09

ご注意書き

- 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、 応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアお よびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これ らの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負い ません。
- 2. 本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないこと を保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害が お客様に生じた場合においても、当社は、一切その責任を負いません。
- 3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の 使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当 社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特 許権、著作権その他の知的財産権を何ら許諾するものではありません。
- 4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
- 5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、 各品質水準は、以下に示す用途に製品が使用されることを意図しております。 標準水準: コンピュータ、OA 機器、通信機器、計測機器、AV 機器、

家電、工作機械、パーソナル機器、産業用ロボット等 高品質水準: 輸送機器(自動車、電車、船舶等)、交通用信号機器、 防災・防犯装置、各種安全装置等

当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム(生命維持装置、人体に埋め込み使用するもの等)、もしくは多大な物的損害を発生させるおそれのある機器・システム(原子力制御システム、軍事機器等)に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。

- 6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件 その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の 故障および事故につきましては、当社は、一切その責任を負いません。
- 7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障 が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放 射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事 故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対 策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとし ての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、 お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
- 8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せく ださい。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境 関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令 を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
- 9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
- 10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
- 11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
- 注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネ サス エレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する 会社をいいます。
- 注 2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造 製品をいいます。

このマニュアルの使い方

このマニュアルは、RH850 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 CubeSuite+ について説明します。

CubeSuite+は、RH850 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、CubeSuite+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、CubeSuite+の持つソフトウェ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ェア、ソフトウェ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。

第1章 概 説
第2章 機 能
付録A ウインドウ・リファレンス
付録B 索 引

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	:	本文中につけた注の説明
		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

関連資料 関連資料は暫定版の場合がありますが、この資料では「暫定」の表示をしておりません。あらかじめご 了承ください。

資	:料 名	資料番号		
		和文	英文	
CubeSuite+ 統合開発環境	起動編	R20UT2682J	R20UT2682E	
ユーザーズ・マニュアル	RX 設計編	R20UT2683J	R20UT2683E	
	V850 設計編	R20UT2134J	R20UT2134E	
	RL78 設計編	R20UT2684J	R20UT2684E	
	78K0R 設計編	R20UT2137J	R20UT2137E	
	78K0 設計編	R20UT2138J	R20UT2138E	
	RH850 コーディング編	R20UT2584J	R20UT2584E	
	RX コーディング編	R20UT2470J	R20UT2470E	
	V850 コーディング編	R20UT0553J	R20UT0553E	
	コーディング編(CX コンパイラ)	R20UT2659J	R20UT2659E	
	RL78,78K0R コーディング編	R20UT2140J	R20UT2140E	
	78K0 コーディング編	R20UT2141J	R20UT2141E	
	RH850 ビルド編	R20UT2585J	R20UT2585E	
	RX ビルド編	R20UT2472J	R20UT2472E	
	V850 ビルド編	R20UT0557J	R20UT0557E	
	ビルド編(CX コンパイラ)	R20UT2142J	R20UT2142E	
	RL78,78K0R ビルド編	R20UT2143J	R20UT2143E	
	78K0 ビルド編	R20UT0783J	R20UT0783E	
	RH850 デバッグ編	このマニュアル	R20UT2685E	
	RX デバッグ編	R20UT2702J	R20UT2702E	
	V850 デバッグ編	R20UT2446J	R20UT2446E	
	RL78 デバッグ編	R20UT2445J	R20UT2445E	
	78K0R デバッグ編	R20UT0732J	R20UT0732E	
	78K0 デバッグ編	R20UT0731J	R20UT0731E	
	解析編	R20UT2686J	R20UT2686E	
	メッセージ編	R20UT2687J	R20UT2687E	

注意 上記関連資料は、予告なしに内容を変更することがあります。設計などには、必ず最新の資料を 使用してください。

この資料に記載されている会社名、製品名などは、各社の商標または登録商標です

目 次

- 第1章 概 説 … 7
 - 1.1 概 要 … 7
 - 1.2 特 長 … 7

第2章機能…9

2.1 概 要 … 9 2.2 デバッグを始める前の準備 … 12 2.2.1 ホスト・マシンとの接続を確認する … 12 2.3 デバッグ・ツールの動作環境設定 … 14 2.3.1 使用するデバッグ・ツールを選択する … 14 2.3.2 【Full-spec emulator】の場合 … 15 2.3.3 【E1】の場合 … 22 2.3.4 【E20】の場合 … 30 2.3.5 【シミュレータ】の場合 … 38 2.4 デバッグ・ツールとの接続/切断 … 43 2.4.1 デバッグ・ツールを接続する … 43 2.4.2 デバッグ・ツールを切断する … 43 2.5 ダウンロード/アップロード … 45 2.5.1 ダウンロードを実行する … 45 2.5.2 応用的なダウンロード方法 … 47 2.5.3 アップロードを実行する … 55 2.6 プログラムの表示と変更 … 57 2.6.1 ソース・ファイルを表示する … 57 2.6.2 逆アセンブル結果を表示する … 67 2.6.3 他の処理と平行してビルドを実行する … 71 2.6.4 ライン·アセンブルを行う ··· 72 2.7 コア(PE)の選択 … 76 2.7.1 コア(PE)を切り替える … 78 2.8 プログラムの実行 … 79 2.8.1 マイクロコントローラ (CPU) をリセットする … 79 2.8.2 プログラムを実行する … 80 2.8.3 プログラムをステップ実行する … 81 2.9 プログラムの停止(ブレーク) … 84 2.9.1 ブレーク動作の設定をする【Full-spec emulator】【E1】【E20】 … 85 2.9.2 プログラムの実行を手動で停止する … 86 2.9.3 任意の場所で停止する(ブレークポイント) … 86 2.9.4 任意の場所で停止する(ブレーク・イベント) … 88 2.9.5 変数 /I/O レジスタへのアクセスで停止する … 90 2.9.6 その他のブレーク要因 … 93 2.10 メモリ、レジスタ、変数の表示/変更 … 94 2.10.1 メモリを表示/変更する … 94

2.10.2 CPU レジスタを表示/変更する … 104 2.10.3 I/O レジスタを表示/変更する … 106 2.10.4 グローバル変数/スタティック変数を表示/変更する … 110 2.10.5 ローカル変数を表示/変更する … 110 2.10.6 ウォッチ式を表示/変更する … 112 2.11 スタックからの関数呼び出し情報の表示 … 120 2.11.1 コール・スタック情報を表示する … 120 2.12 実行履歴の収集 … 122 2.12.1 トレース動作の設定をする … 122 2.12.2 実行停止までの実行履歴を収集する … 128 2.12.3 任意区間の実行履歴を収集する … 129 2.12.4 条件を満たしたときのみの実行履歴を収集する【シミュレータ】 … 132 2.12.5 実行履歴の収集を停止/再開する … 134 2.12.6 実行履歴を表示する … 134 2.12.7 トレース・メモリをクリアする … 137 2.12.8 トレース・データを検索する … 137 2.12.9 実行履歴の表示内容を保存する … 143 2.13 実行時間の計測 … 145 2.13.1 実行停止までの実行時間を計測する … 145 2.13.2 任意区間の実行時間を計測する【シミュレータ】 … 146 2.13.3 測定可能時間の範囲 … 149 2.14 カバレッジの測定【シミュレータ】 … 150 2.14.1 カバレッジ測定の設定をする … 150 2.14.2 カバレッジ測定結果を表示する … 151 2.15 プログラム内へのアクションの設定 … 154 2.15.1 printf を挿入する … 154 2.16 イベントの管理 … 157 2.16.1 設定状態(有効/無効)を変更する … 158 2.16.2 特定のイベント種別のみ表示する … 158 2.16.3 イベントのアドレスにジャンプする … 159 2.16.4 イベントを削除する … 159 2.16.5 イベントにコメントを入力する … 159 2.16.6 イベント設定に関する留意事項 … 159 2.17 フック処理を設定する … 163 2.18 入力値について … 166 2.18.1 入力規約 … 166 2.18.2 シンボル名の入力補完機能 … 170 2.18.3 入力不備箇所に対するアイコン表示 … 171 付録 A ウインドウ・リファレンス … 172

A.1 説 明 … 172

付録 B 索 引 … 391

第1章概 説

CubeSuite+ 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。

CubeSuite+ 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。

この章では、CubeSuite+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

CubeSuite+ が提供するデバッグ機能を使用することにより、RH850 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、CubeSuite+が提供するデバッグ機能の特長を示します。

- マルチコア対応版における同期実行/同期ブレーク 使用するマイクロコントローラがマルチコア対応版の場合、同期実行/同期ブレークを行います。 コアの選択を切り替えることにより、コアごとの情報がパネルに表示されます。

- 各種デバッグ・ツールとの接続 フルスペック・エミュレータ(Full-spec emulator),オンチップ・デバッギング・エミュレータ(E1/E20),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 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 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 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。
- フラッシュ・セルフ・プログラミング・エミュレーション機能の対応 (コード・フラッシュ)
 フラッシュ・セルフ・プログラミング・エミュレーション機能のフラッシュ・セルフ・ライブラリを使用し、
 コード・フラッシュの書き換えを行うことができます。

- 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。



第2章機能

この章では、CubeSuite+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

CubeSuite+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) CubeSuite+ を起動する

Windows の [スタート] メニューから CubeSuite+ を起動します。

備考 "CubeSuite+ を起動する"についての詳細は、「CubeSuite+ 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、「CubeSuite+ 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-RH 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、「CubeSuite+ 統合開発 環境 ユーザーズマニュアル RH850 ビルド編」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(Full-spec emulator/E1/E20/ 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。

- (6) デバッグ・ツールの動作環境設定を行う
 - (5) で選択したデバッグ・ツールの動作環境を設定します。
 - 【Full-spec emulator】の場合
 - 【E1】の場合
 - 【E20】の場合
 - 【シミュレータ】の場合

- (7) デバッグ・ツールを接続する CubeSuite+ 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) で作成したロード・モジュールを, デバッグ・ツールヘ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(「2.9.3 任意の場所で停止する(ブレークポイント)」/「2.9.4 任意の場所で 停止する(ブレーク・イベント)」/「2.9.5 変数/I/O レジスタへのアクセスで停止する」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、「2.16.6 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 選択しているマイクロコントローラがマルチコア対応版の場合、プログラムを実行する前に、デバッグ対象とするコア(PE: プロセッサ・エレメント)を選択してください(「2.7 コア(PE)の選択」
 参照)。
- (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) でブレークポイント/ブレーク・イベントを設定している場合、設定したブレーク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】
- 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、「2.16.6 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
 - 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(インテル拡張へキサ・フォーマット/モト ローラ・Sタイプ・フォーマット/バイナリ・フォーマットなど)で保存します。

(14) デバッグ・ツールを切断する

CubeSuite+ 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、「CubeSuite+ 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。

- (1) 【Full-spec emulator】の場合
- (2)【E1】の場合
- (3)【E20】の場合
- (4) 【シミュレータ】の場合

(1)【Full-spec emulator】の場合

ホスト・マシン, Full-spec 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, Full-spec emulator 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【Full-spec emulator】



(2)【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。 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 2-2 ホスト・マシンとデバッグ・ツールとの接続例【E1】



注意 1. ターゲット・ボードとの通信方式として, Low Pin Debug Interface (以降, LPD 通信方式と略します)のみをサポートしています。

2. デバッグ MCU ボードを使用する場合の接続例については、デバッグ MCU ボードのユーザーズ・ マニュアルを参照してください。

(3)【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。 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---3 ホスト・マシンとデバッグ・ツールとの接続例【E20】



- 注意 1. ターゲット・ボードとの通信方式として, Low Pin Debug Interface (以降, LPD 通信方式と略し ます)のみをサポートしています。
 - デバッグ MCU ボードを使用する場合の接続例については、デバッグ MCU ボードのユーザーズ・ マニュアルを参照してください。

(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 2-4 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・ツールはメイン・プロジェクト/サブ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・メニューから行ってください。



図 2-5 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。

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 【Full-spec emulator】の場合

Full-spec 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—6 動作環境設定【Full-sp	er emulator】(ブロバティ バネル)
----------------------	-------------------------

プロパティ	8			
🚑 RH850 Full-spec emulator のプロパティ				
□ <u>20</u> 22				
メイン・クロック周波数[MHz]	10.00			
■ CPUクロック周波数[MHz]	[2]			
□ ターゲット・ボードとの接続				
ターゲット・ボードを接続している	いいえ			
□ 75991				
Code Flashのセルフ・プログラミングを行う	いいえ			
□ CPU仮想化支援機能				
仮想マシン・スレッドを使用する	いいえ			
クロック				
│ 接続用設定 / デバッグ・ツール… / ダウ	ンロード・フ 🖌 フラッシュ・オブ 🖌 フック処理設定 / 🔻			

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。

- (1) [接続用設定] タブ
- (2) [デバッグ・ツール設定] タブ
- (3) [ダウンロード・ファイル設定] タブ
- (4) [フラッシュ・オプション設定] タブ
- (5) [フック処理設定] タブ

(1) [接続用設定] タブ

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。

- (a) [クロック]
- (b) [ターゲット・ボードとの接続]
- (c) [フラッシュ]
- (d) [CPU 仮想化支援機能]
- (a) [クロック]

クロックに関する設定を行います。



 \Box

Ξ

⊡ [1]

コア名称

<u> ////////////////////////////////////</u>		
メイン・クロック周波数[MHz]	10.00	
CPUクロック周波数[MHz]	[2]	
⊟ [0]	CPU1 - 10.00	
コア名称	CPU1	
CPUクロック周波数	10.00	

PCU

10.00

PCU - 10.00

図 2—7 [クロック]カテゴリ【Full-spec emulator】

- [メイン・クロック周波数 [MHz]]

CPUクロック周波数

メイン・クロック周波数(逓倍前)を指定します。

ドロップダウン・リストによる選択, または直接入力により, 0.001 ~ 999.999(単位: MHz)の範囲の周波数を指定します(デフォルト: [10.00])。

 - [CPU クロック周波数 [MHz]]
 CPU クロック周波数をコアごとに指定します。
 サブプロパティとして、選択しているマイクロコントローラが持つコア名称が表示されます。
 各コアの CPU クロック周波数を、直接入力により、0.001 ~ 999.999(単位:MHz)の範囲の数値で 指定してください。
 なお、表示されるコア名称の数は、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。

備考 CPU クロック周波数は、トレースのタイム・スタンプ情報を実時間に換算する際に使用されます。

(b) [ターゲット・ボードとの接続]

Full-spec 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。

注意 Full-spec emulator が CubeSuite+ に接続している場合, 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—8 [ターゲット・ボードとの接続] カテゴリ【Full-spec emulator】

E	ターゲット・ボードとの接続		
	ターゲット・ボードを接続している	いいえ	

備考 このプロパティを空欄の状態でデバッグ・ツールと接続した場合,自動設定した値で接続処理 を行い,接続後に決定値を表示します。

- [ターゲット・ボードを接続している] Full-spec emulator にターゲット・ボードを接続しているか否かを選択します。 ターゲット・ボードと接続している場合は [はい] を選択してください(デフォルト:[いいえ])。

(c) [フラッシュ]

フラッシュ・セルフ・プログラミング機能に関する設定を行います。

ただし、このカテゴリは、選択しているマイクロコントローラがフラッシュ・セルフ・プログラミング 機能をサポートしている場合のみ表示されます。

注意 Full-spec emulator が CubeSuite+ に接続している場合, 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—9 [フラッシュ]カテゴリ

日 フラッシュ Code Flashのセルフ・プログラミングを行う いいえ

- [Code Flash のセルフ・プログラミングを行う]
 フラッシュ・セルフ・プログラミング機能のフラッシュ・セルフ・ライブラリを使用して、Code
 Flash の書き換えを行うか否かを選択します。
 Code Flash の書き換えを行う場合は、[はい]を選択してください(デフォルト:[いいえ])。
 なお、[はい]を選択した場合、Code Flash のキャッシュは行われません。
- (d) [CPU 仮想化支援機能]

このカテゴリ内のプロパティは、常に無効です。



(2) [デバッグ・ツール設定] タブ

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。

- (a) [メモリ]
- (b) [実行中のメモリ・アクセス]
- (c) [実行中のイベント設定]
- (d) [ブレーク]
- (e) [トレース]
- (f) [入力信号のマスク]
- (a) [メモリ]

メモリに関する設定を行います。

🗆 አモリ	
Ⅲ メモリ・マッピング	[26]
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]

展開表示することにより、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。

义	2—11	メモリ	•	マッ	ピン	ノグ	の詳細表示
---	------	-----	---	----	----	----	-------

Ξ	メモリ	
	メモリ・マッピング	[26]
	⊟ [0]	Code Flash
	メモリ種別	Code Flash
	開始アドレス	HEX ()
	終了アドレス	HEX 3FFFFF
	⊞ [1]	Access prohibited

注意 メモリ・マッピングの追加/削除はできません。

備考 選択しているマイクロコントローラがマルチコア対応版の場合、コア(PEn)の選択を切り替えることにより、PEnごとのメモリ・マッピングの状況を表示します(「2.7 コア(PE)の選択」参照)。

- [メモリ書き込み時にベリファイを行う]
 メモリ書き込み時に、ベリファイを行うか否かを選択します。
 [はい]を選択した場合、ダウンロードの際、またはウォッチ パネル/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(デフォルト)。

(b) [実行中のメモリ・アクセス]

プログラム実行中におけるメモリ・アクセス(リアルタイム表示更新機能)に関する設定を行います。 リアルタイム表示更新機能についての詳細は、「(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—12 [実行中のメモリ・アクセス] カテゴリ【Full-spec emulator】

□ 実行中のメモリ・アクセス

_		
	実行中にアクセスする	()()Ż
	実行中に表示更新を行う	(สถา
	表示更新間隔[ms]	500

- [実行中にアクセスする]

プログラム実行中に内蔵 RAM 領域にアクセスするか否かを選択します。 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:[いいえ])。

- [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選択します。 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)。

- [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]を選択した場合のみ表示 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: [500])。

(c) [実行中のイベント設定]

このカテゴリでは、プログラム実行中におけるイベントの設定に関する設定を行います。

図 2-13 [実行中のイベント設定] カテゴリ

□ 実行中のイベント設定

実行を一瞬停止してイベントを設定する いいえ

- [実行を一瞬停止してイベントを設定する]

プログラム実行中,またはトレーサ/タイマ動作中には設定することができないイベントを,プログ ラムの実行,またはトレーサ/タイマの動作を強制的に一瞬停止させることで設定を行うか否かを選 択します。

このプロパティの対象となるイベント種別については、「(2)実行中に設定/削除可能なイベント種別」を参照してください。

プログラム実行中,またはトレーサ/タイマ動作中に,イベントの設定を行う場合は[はい]を選択 してください(デフォルト:[いいえ])。

(d) [ブレーク]

ブレーク機能に関する設定を行います。

ブレーク機能,およびこのカテゴリ内の設定についての詳細は,「2.9 プログラムの停止(ブレーク)」 を参照してください。

(e) [トレース]

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,「2.12 実行履歴の収集」を参照して ください。

(f) [入力信号のマスク]

入力信号のマスクに関する設定を行います。

図 2—14 [入力信号のマスク] カテゴリ【Full-spec emulator】

🛛 入力信号のマスク

コ ハノルョックション		
	WAIT 信号をマスクする	はい
	RESET 信号をマスクする	はい
	マスクするRESET信号の選択	TARGET RESET信号

- [WAIT 信号をマスクする]

WAIT 信号をマスクするか否かを選択します。

WAIT 信号を Full-spec emulator に入力する場合は, [いいえ] を選択してください (デフォルト : [いいえ])。

注意 [接続設定用] タブ上の [ターゲット・ボードを接続している] プロパティにおいて [いいえ] を選択している場合, このプロパティは [はい] に固定となります (変更不可)。

- [RESET 信号をマスクする]

RESET 信号をマスクするか否かを選択します。

RESET 信号を Full-spec emulator に入力する場合は, [いいえ] を選択してください (デフォルト: [いいえ])。

注意 [接続設定用] タブ上の [ターゲット・ボードを接続している] プロパティにおいて [いいえ] を選択している場合, このプロパティは [はい] に固定となります (変更不可)。

- [マスクする RESET 信号の選択]

このプロパティは、[RESET 信号をマスクする]プロパティにおいて[はい]を選択した場合にのみ 表示されます。

マスクする RESET 信号を次のドロップダウン・リストより選択します。

- TARGET RESET 信号(デフォルト)
- TARGET RESET 信号と INTERNAL RESET 信号

(3) [ダウンロード・ファイル設定] タブ

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。

(4) [フラッシュ・オプション設定] タブ

マイクロコントローラに搭載されているフラッシュ・メモリのためのオプション設定を行います。

ただし、このタブは、選択しているマイクロコントローラがフラッシュ・オプションに対応している場合の み表示されます。

なお,各オプションの設定はこのタブ上では行わず,[フラッシュ・オプション]カテゴリ内[フラッシュ・ オプション]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によりオープン する,フラッシュ・オプションの設定ダイアログ【Full-spec emulator】【E1】【E20】で行います([...]ボタ ンは,デバッグ・ツールと接続している場合のみ表示されます)。

このダイアログにおいて、各項目の設定を完了したのち、[書き込み] ボタンをクリックしてください。 設定内容についての詳細は、フラッシュ・オプションの設定 ダイアログ【Full-spec emulator】【E1】【E20】 を参照してください。

図 2-15 フラッシュ・オプションの設定 ダイアログのオープン



図 2—16 フラッシュ・オプションの設定(フラッシュ・オプションの設定 ダイアログ)

フラッシュ・オブションの設定		
フラッシュ・オブション・プロパティ(P):		
□ オブション・バイト設定		
OPBTO	HEX FFFFFFF	
OPBT1	HEX FFFFFFF	
OPBT2	HEX FFFFFFF	
OPBT3	HEX FFFFFFF	
OPBT4	HEX FFFFFFF	
OPBT5	HEX FFFFFFF	
OPBT6	HEX FFFFFFF	
OPBT7	HEX FFFFFFF	
オプション・バイト設定 読み込み(R) 書き込み(W) 閉じる(C) キャンセル ヘルプ(H)		

(5) [フック処理設定] タブ

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,「2.17 フック処理を設定する」を参照してく ださい。

2.3.3 【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。

注意 ターゲット・ボードとの通信方式として、LPD 通信方式のみをサポートしています。

プロパティ	X	
🚑 RH850 E1(LPD) のプロパティ	₽ −+	
日 クロッ ク		
メイン・クロック周波数[MHz]	10.00	
■ GPUクロック周波数[MHz]	[2]	
□ ターゲット・ボードとの接続		
LPDモード	4ピン	
LPDクロック周波数[kHz]	Default	
接続時にOPJTAGをLPD接続に設定する	(t()	
切断時にOPJTAGをJTAG接続に設定する	いいえ	
□ 75991		
セキュリティID	IN FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	
Code Flashのセルフ・プログラミングを行う	いいえ	
□ CPU仮想化支援機能		
仮想マシン・スレッドを使用する	เหเน้	
ሳዐቃク		
接続用設定 デバッグ・ツール… / ダウン	ンロード・フ 🖌 フラッシュ・オブ 🖌 フック処理設定 / 🔻	

図 2-17 動作環境設定【E1】(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。

- (1) [接続用設定] タブ
- (2) [デバッグ・ツール設定] タブ
- (3) [ダウンロード・ファイル設定] タブ
- (4) [フラッシュ・オプション設定] タブ
- (5) [フック処理設定] タブ

(1) [接続用設定] タブ

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。

- (a) **[クロック]**
- (b) [ターゲット・ボードとの接続]
- (c) [フラッシュ]
- (d) [CPU 仮想化支援機能]
- (a)[クロック]

クロックに関する設定を行います。

図 2—18 [クロック]カテゴリ【E1】

Ξ	3 /0%/	
	メイン・クロック周波数[MHz]	10.00
Ξ	CPUクロック周波数[MHz]	[2]
	□ [0]	CPU1 - 10.00
	コア名称	CPU1
	CPUクロック周波数	10.00
	⊡ [1]	PCU - 10.00
	コア名称	PGU
	CPUクロック周波数	10.00

- [メイン・クロック周波数 [MHz]]

メイン・クロック周波数(逓倍前)を指定します。

ドロップダウン・リストによる選択, または直接入力により, 0.001 ~ 999.999(単位: MHz)の範囲の数値を指定します(デフォルト: [10.00])。

- [CPU クロック周波数 [MHz]]

CPU クロック周波数をコアごとに指定します。 サブプロパティとして、選択しているマイクロコントローラが持つコア名称が表示されます。 各コアの CPU クロック周波数を、直接入力により、0.001 ~ 999.999(単位:MHz)の範囲の数値で 指定してください。

なお、表示されるコア名称の数は、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。

備考 CPU クロック周波数は、トレースのタイム・スタンプ情報を実時間に換算する際に使用されます。

(b) [ターゲット・ボードとの接続]

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。

注意 E1 が CubeSuite+ に接続している場合、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 2-19 [ターゲット・ボードとの接続] カテゴリ【E1】

□ ターケット・ホードとの接続

シーン クト・ホート くい かた	
LPDモード	4ピン
LPDクロック周波数[kHz]	Default
接続時にOPJTAGをLPD接続に設定する	はい
切断時(EOPJTAGをJTAG接続(ご設定する	いいえ

- [LPD モード]

使用する LPD 通信方式のモードを選択します。

ドロップダウン・リスト内に表示されるピン数は、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。

ただし、選択できるモードが1つしか存在しない場合は、このプロパティを変更することはできません。

「ボーレート [kbps]]
 このプロパティは、[LPD モード] プロパティにおいて、[1 ピン] を選択した場合のみ表示されます。
 LPD 通信方式の通信速度を選択します(デフォルト: [500])。

- [LPD クロック周波数 [kHz]]

このプロパティは、[LPD モード] プロパティにおいて、[4 ピン] を選択した場合のみ表示されます。 LPD 通信方式のクロック周波数を選択します(デフォルト:[Default])。

なお、[Default]を選択した場合は、マイクロコントローラ固有のデフォルト値で接続処理を行います。

「接続時に OPJTAG を LPD 接続に設定する」
 デバッグ・ツールとの接続時にシリアル・プログラミング・モードで起動し、オプション・バイトの
 設定を LPD 接続に変更するか否かを選択します。
 [はい]を選択した場合、接続時にシリアル・プログラミング・モードで起動し、OPJTAG をチェック
 します。この際に LPD 設定でない場合は LPD 設定に変更し、その後デバッグ・モードに移行します

(デフォルト)。

[いいえ]を選択した場合,接続時にデバッグ・モードで起動し,OPJTAGをチェックします。この際にLPD設定でない場合はメッセージ・ダイアログを表示します。

- [切断時に OPJTAG を JTAG 接続に設定する] このプロパティは、[接続時に OPJTAG を LPD 接続に設定する] プロパティにおいて、[はい] を選択 した場合のみ変更することができます。

デバッグ・ツールとの接続を切断する際に、オプション・バイトの設定を JTAG 接続に変更するか否 かを選択します。

切断時にオプション・バイトの設定を JTAG 接続に変更する場合は [はい]を選択してください。 なお、[いいえ](デフォルト)を選択している場合、切断時にオプション・バイトを変更しないため、 ピン・モードは LPD モード設定となります。

備考 E1 との接続時に CubeSuite+ は、オプション・バイトが LPD モード設定になっていない場合、
 オプション・バイトの値を書き換えます。このため、マイクロコントローラのオプション・バ
 イトが E1 との接続前と接続後で異なる可能性があります。

(c) [フラッシュ]

フラッシュ書き換えに関する設定を行います。

注意 E1 が CubeSuite+ に接続している場合, 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—20 [フラッシュ] カテゴリ【E1】

フラッシュ
 セキュリティID
 Code Flashのセルフ・プログラミングを行う いいえ

- [Code Flash のセルフ・プログラミングを行う]
 フラッシュ・セルフ・プログラミング機能のフラッシュ・セルフ・ライブラリを使用して、Code
 Flash の書き換えを行うか否かを選択します。
 Code Flash の書き換えを行う場合は、[はい]を選択してください(デフォルト:[いいえ])。
 なお、[はい]を選択した場合、Code Flash のキャッシュは行われません。

(d) [CPU 仮想化支援機能]

このカテゴリ内のプロパティは、常に無効です。

(2) [デバッグ・ツール設定] タブ

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。

- (a) [メモリ]
- (b) [実行中のメモリ・アクセス]
- (c) [実行中のイベント設定]
- (d) [ブレーク]
- (e) [トレース]
- (f) [入力信号のマスク]
- (a) [メモリ]

メモリに関する設定を行います。

図 2—21 [メモリ]カテゴリ【E1】

🗆 አモリ		
Ⅲ メモリ・マッピング	[26]	
メモリ書き込み時にベリファイを行う	(まい)	

- [メモリ・マッピング]

展開表示することにより、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。

図 2-22 メモリ・マッピングの詳細表示

Ξ	メモリ	
Ξ	メモリ・マッピング	[26]
	⊟ [0]	Code Flash
	メモリ種別	Code Flash
	開始アドレス	HEX ()
	終了アドレス	HEX 3FFFFF
	⊕ [1]	Access prohibited

注意 メモリ・マッピングの追加/削除はできません。

- 備考 選択しているマイクロコントローラがマルチコア対応版の場合、コア(PEn)の選択を切り替えることにより、PEnごとのメモリ・マッピングの状況を表示します(「2.7 コア(PE)の選択」参照)。
- [メモリ書き込み時にベリファイを行う]
 メモリ書き込み時に、ベリファイを行うか否かを選択します。
 [はい]を選択した場合、ダウンロードの際、またはウォッチ パネル/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(デフォルト)。
- (b) [実行中のメモリ・アクセス]

プログラム実行中におけるメモリ・アクセス(リアルタイム表示更新機能)に関する設定を行います。 リアルタイム表示更新機能についての詳細は、「(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 2-23 [実行中のメモリ・アクセス] カテゴリ【E1】

□ 実行中のメモリ・アクセス

ロ 美口中のメモリ・アクビス		
実行中にアクセスする	いいえ	
実行中に表示更新を行う	(まい)	
表示更新間隔[ms]	500	

- [実行中にアクセスする] プログラム実行中に内蔵 RAM 領域にアクセスするか否かを選択します。 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:[いいえ])。

- [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選択します。 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)。

- [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]を選択した場合にのみ有 効となります。 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: [500])。

(c) [実行中のイベント設定]

このカテゴリでは、実行中のイベント設定に関する設定を行います。

図 2-24 [実行中のイベント設定] カテゴリ

□ 実行中のイベント設定

実行を一瞬停止してイベントを設定する いいえ

- [実行を一瞬停止してイベントを設定する]

プログラム実行中,またはトレーサ/タイマ動作中には設定することができないイベントを,プログラムの実行,またはトレーサ/タイマの動作を強制的に一瞬停止させることで設定を行うか否かを選択します。

このプロパティの対象となるイベント種別については、「(2)実行中に設定/削除可能なイベント種別」を参照してください。

プログラム実行中,またはトレーサ/タイマ動作中に,イベントの設定を行う場合は[はい]を選択 してください(デフォルト:[いいえ])。

(d) [ブレーク]

ブレーク機能に関する設定を行います。

ブレーク機能,およびこのカテゴリ内の設定についての詳細は,「2.9 プログラムの停止(ブレーク)」 を参照してください。

(e) [トレース]

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,「2.12 実行履歴の収集」を参照して ください。

(f) [入力信号のマスク]

入力信号のマスクに関する設定を行います。

Ξ		入力信号のマスク	
		WAIT 信号をマスクする	(า(าวั
		RESET 信号をマスクする	はい
		マスクするRESET信号の選択	TARGET RESET信号とINTERNAL RESET信号



- [WAIT 信号をマスクする] WAIT 信号をマスクするか否かを選択します。 WAIT 信号を E1 に入力しない場合は, [はい] を選択してください(デフォルト: [いいえ])。

- [RESET 信号をマスクする] RESET 信号をマスクするか否かを選択します。 RESET 信号を E1 に入力しない場合は, [はい]を選択してください(デフォルト:[いいえ])。

- [マスクする RESET 信号の選択] このプロパティは, [RESET 信号をマスクする]プロパティにおいて[はい]を選択した場合にのみ 表示されます。 マスクされる RESET 信号を表示します。

このプロパティを変更することはできません。

(3) [ダウンロード・ファイル設定] タブ

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。

(4) [フラッシュ・オプション設定] タブ

マイクロコントローラに搭載されているフラッシュ・メモリのためのオプション設定を行います。

ただし、このタブは、選択しているマイクロコントローラがフラッシュ・オプションに対応している場合の み表示されます。

なお,各オプションの設定はこのタブ上では行わず,[フラッシュ・オプション]カテゴリ内[フラッシュ・ オプション]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によりオープン する,フラッシュ・オプションの設定ダイアログ【Full-spec emulator】【E1】【E20】で行います([...]ボタ ンは,デバッグ・ツールと接続している場合のみ表示されます)。

このダイアログにおいて、各項目の設定を完了したのち、[書き込み]ボタンをクリックしてください。 設定内容についての詳細は、フラッシュ・オプションの設定 ダイアログ【Full-spec emulator】【E1】【E20】 を参照してください。



□ **フラッシュ・オフション** フラッシュ・オフション **フラッシュ・オフション**□



図 2-27 フラッシュ・オプションの設定(フラッシュ・オプションの設定 ダイアログ)

。 フラッシュ・オブションの設定		
フラッシュ・オプション・プロパティ(P):		
□ オブション・バイト設定		
OPBT0	HEX FFFFFF	
OPBT1	HEX FFFFFFF	
OPBT2	HEX FFFFFFF	
OPBT3	HEX FFFFFF	
OPBT4	HEX FFFFFFF	
OPBT5	HEX FFFFFFF	
OPBT6	HEX FFFFFFF	
OPBT7	HEX FFFFFFF	
オブション・バイト設定 読み込み(R)(書き込み(W)) 閉じる(C) キャンセル ヘルブ(H)		

(5)[フック処理設定] タブ

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,「2.17 フック処理を設定する」を参照してく ださい。



2.3.4 【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。

注意 ターゲット・ボードとの通信方式として、LPD 通信方式のみをサポートしています。

プロパティ	8
🔊 RH850 E20(LPD)のプロパティ	P -+
日 クロッ ク	
メイン・クロック周波数[MHz]	10.00
■ GPUクロック周波数[MHz]	[2]
□ ターゲット・ボードとの接続	
LPDモード	4ピン
LPDクロック周波数[kHz]	Default
接続時にOPJTAGをLPD接続に設定する	(はい)
切断時にOPJTAGをJTAG接続に設定する	いいえ
□ フラッシュ	
セキュリティID	FFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Code Flashのセルフ・プログラミングを行う	いいえ
□ CPU仮想化支援機能	
仮想マシン・スレッドを使用する	いいえ
00 9 0	
接続用設定 デバッグ・ツール… / ダウン	ンロード・フ 🖌 フラッシュ・オブ 🖌 フック処理設定 / 🔻

図 2—28 動作環境設定【E20】(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。

- (1) [接続用設定] タブ
- (2) [デバッグ・ツール設定] タブ
- (3) [ダウンロード・ファイル設定] タブ
- (4) [フラッシュ・オプション設定] タブ
- (5) [フック処理設定] タブ

(1) [接続用設定] タブ

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。

- (a) [クロック]
- (b) [ターゲット・ボードとの接続]
- (c) [フラッシュ]
- (d) [CPU 仮想化支援機能]
- (a)[クロック]

クロックに関する設定を行います。

図 2—29 [クロック]カテゴリ【E20】

Ξ	<i>ካ</i> ロ ッ ク	
	メイン・クロック周波数[MHz]	10.00
Ξ	CPUクロック周波数[MHz]	[2]
	□ [0]	CPU1 - 10.00
	コア名称	CPU1
	CPUクロック周波数	10.00
	⊡ [1]	PCU - 10.00
	コア名称	PCU
	CPUクロック周波数	10.00

- [メイン・クロック周波数 [MHz]]

メイン・クロック周波数(逓倍前)を指定します。

ドロップダウン・リストによる選択, または直接入力により, 0.001 ~ 999.999(単位: MHz)の範囲の数値を指定します(デフォルト: [10.00])。

- [CPU クロック周波数 [MHz]]

CPU クロック周波数をコアごとに指定します。 サブプロパティとして,選択しているマイクロコントローラが持つコア名称が表示されます。 各コアの CPU クロック周波数を,直接入力により,0.001 ~ 999.999(単位:MHz)の範囲の数値で 指定してください。

なお、表示されるコア名称の数は、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。

備考 CPU クロック周波数は、トレースのタイム・スタンプ情報を実時間に換算する際に使用されます。

(b) [ターゲット・ボードとの接続]

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。

注意 E20 が CubeSuite+ に接続している場合、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—30 [ターゲット・ボードとの接続] カテゴリ【E20】

□ ターケット・ホードとの接続

ダニクタト・ホートという致める	
LPDモード	4ピン
LPDクロック周波数[kHz]	Default
接続時(EOPJTAGをLPD接続に設定する	はい
切断時にOPJTAGをJTAG接続に設定する	いいえ

- [LPD モード]

使用する LPD 通信方式のモードを選択します。

ドロップダウン・リスト内に表示されるピン数は、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。

ただし、選択できるモードが1つしか存在しない場合は、このプロパティを変更することはできません。

「ボーレート [kbps]]
 このプロパティは、[LPD モード] プロパティにおいて、[1 ピン] を選択した場合のみ表示されます。
 LPD 通信方式の通信速度を選択します(デフォルト: [500])。

- [LPD クロック周波数 [kHz]]

このプロパティは、[LPD モード] プロパティにおいて、[4 ピン] を選択した場合のみ表示されます。 LPD 通信方式のクロック周波数を選択します(デフォルト:[Default])。

なお、[Default]を選択した場合は、マイクロコントローラ固有のデフォルト値で接続処理を行います。

- [接続時に OPJTAG を LPD 接続に設定する]
 デバッグ・ツールとの接続時にシリアル・プログラミング・モードで起動し、オプション・バイトの
 設定を LPD 接続に変更するか否かを選択します。
 [はい]を選択した場合、接続時にシリアル・プログラミング・モードで起動し、OPJTAG をチェック
 します。この際に LPD 設定でない場合は LPD 設定に変更し、その後デバッグ・モードに移行します
 (デフォルト)。

[いいえ]を選択した場合,接続時にデバッグ・モードで起動し,OPJTAGをチェックします。この際にLPD設定でない場合はメッセージ・ダイアログを表示します。

- [切断時に OPJTAG を JTAG 接続に設定する] このプロパティは、[接続時に OPJTAG を LPD 接続に設定する] プロパティにおいて、[はい] を選択 した場合のみ変更することができます。

デバッグ・ツールとの接続を切断する際に、オプション・バイトの設定を JTAG 接続に変更するか否かを選択します。

切断時にオプション・バイトの設定を JTAG 接続に変更する場合は [はい]を選択してください。 なお、[いいえ](デフォルト)を選択している場合、切断時にオプション・バイトを変更しないため、 ピン・モードは LPD モード設定となります。

備考 E20 との接続時に CubeSuite+ は、オプション・バイトが LPD モード設定になっていない場合、 オプション・バイトの値を書き換えます。このため、マイクロコントローラのオプション・バ イトが E1 との接続前と接続後で異なる可能性があります。

(c) [フラッシュ]

フラッシュ書き換えに関する設定を行います。

注意 E20 が CubeSuite+に接続している場合、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—31 [フラッシュ] カテゴリ【E20】

D フラッシュ
 セキュリテ・
 セキュリテ・



- [Code Flash のセルフ・プログラミングを行う]
 フラッシュ・セルフ・プログラミング機能のフラッシュ・セルフ・ライブラリを使用して、Code
 Flash の書き換えを行うか否かを選択します。
 Code Flash の書き換えを行う場合は、[はい]を選択してください(デフォルト:[いいえ])。
 なお、[はい]を選択した場合、Code Flash のキャッシュは行われません。

(d) [CPU 仮想化支援機能]

このカテゴリ内のプロパティは、常に無効です。

(2) [デバッグ・ツール設定] タブ

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。

- (a) [メモリ]
- (b) [実行中のメモリ・アクセス]
- (c) [実行中のイベント設定]
- (d) [ブレーク]
- (e) [トレース]
- (f) [入力信号のマスク]
- (a) [メモリ]

メモリに関する設定を行います。

図 2—32 [メモリ] カテゴリ【E20】

🗆 አモリ	
Ⅲ メモリ・マッピング	[26]
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]

展開表示することにより、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。

図 2-33 メモリ・マッピングの詳細表示

Ε	メモリ	
Ε	メモリ・マッピング	[26]
	□ [0]	Code Flash
	メモリ種別	Code Flash
	開始アドレス	HEX ()
	終了アドレス	HEX 3FFFFF
	⊡ [1]	Access prohibited

注意 メモリ・マッピングの追加/削除はできません。

- 備考 選択しているマイクロコントローラがマルチコア対応版の場合、コア(PEn)の選択を切り替えることにより、PEnごとのメモリ・マッピングの状況を表示します(「2.7 コア(PE)の選択」参照)。
- [メモリ書き込み時にベリファイを行う]
 メモリ書き込み時に、ベリファイを行うか否かを選択します。
 [はい]を選択した場合、ダウンロードの際、またはウォッチ パネル/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(デフォルト)。
- (b) [実行中のメモリ・アクセス]

プログラム実行中におけるメモリ・アクセス(リアルタイム表示更新機能)に関する設定を行います。 リアルタイム表示更新機能についての詳細は、「(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 2-34 [実行中のメモリ・アクセス] カテゴリ【E20】

□ 実行中のメモリ・アクセス

ロ 美口中のメモリ・アクビス	美山子のメモリーアクピス		
実行中にアクセスする	いいえ		
実行中に表示更新を行う	(まい)		
表示更新間隔[ms]	500		

- [実行中にアクセスする] プログラム実行中に内蔵 RAM 領域にアクセスするか否かを選択します。 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: [いいえ])。

- [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選択します。 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)。

- [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]を選択した場合にのみ有 効となります。 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: [500])。

(c) [実行中のイベント設定]

このカテゴリでは、実行中のイベント設定に関する設定を行います。

図 2---35 [実行中のイベント設定] カテゴリ

□ 実行中のイベント設定

実行を一瞬停止してイベントを設定する いいえ

- [実行を一瞬停止してイベントを設定する]

プログラム実行中,またはトレーサ/タイマ動作中には設定することができないイベントを,プログラムの実行,またはトレーサ/タイマの動作を強制的に一瞬停止させることで設定を行うか否かを選択します。

このプロパティの対象となるイベント種別については、「(2)実行中に設定/削除可能なイベント種別」を参照してください。

プログラム実行中,またはトレーサ/タイマ動作中に,イベントの設定を行う場合は[はい]を選択 してください(デフォルト:[いいえ])。

(d) [ブレーク]

ブレーク機能に関する設定を行います。

ブレーク機能,およびこのカテゴリ内の設定についての詳細は,「2.9 プログラムの停止(ブレーク)」 を参照してください。

(e) [トレース]

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,「2.12 実行履歴の収集」を参照して ください。

(f) [入力信号のマスク]

入力信号のマスクに関する設定を行います。

図 2---36 [入力信号のマスク] カテゴリ【E20】

	ΘŻ	し力信号のマスク	
	W	VAIT 信号をマスクする	いいえ
	R	RESET 信号をマスクする	はい
	3	7スクするRESET信号の選択	TARGET RESET信号とINTERNAL RESET信号



- [WAIT 信号をマスクする] WAIT 信号をマスクするか否かを選択します。 WAIT 信号を E20 に入力しない場合は, [はい] を選択してください(デフォルト:[いいえ])。

- [RESET 信号をマスクする] RESET 信号をマスクするか否かを選択します。 RESET 信号を E20 に入力しない場合は, [はい]を選択してください(デフォルト:[いいえ])。

- [マスクする RESET 信号の選択] このプロパティは, [RESET 信号をマスクする]プロパティにおいて[はい]を選択した場合にのみ 表示されます。 マスクされる RESET 信号を表示します。

このプロパティを変更することはできません。

(3) [ダウンロード・ファイル設定] タブ

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。

(4) [フラッシュ・オプション設定] タブ

マイクロコントローラに搭載されているフラッシュ・メモリのためのオプション設定を行います。

ただし、このタブは、選択しているマイクロコントローラがフラッシュ・オプションに対応している場合の み表示されます。

なお,各オプションの設定はこのタブ上では行わず,[フラッシュ・オプション]カテゴリ内[フラッシュ・ オプション]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によりオープン する,フラッシュ・オプションの設定ダイアログ【Full-spec emulator】【E1】【E20】で行います([...]ボタ ンは,デバッグ・ツールと接続している場合のみ表示されます)。

このダイアログにおいて、各項目の設定を完了したのち、[書き込み]ボタンをクリックしてください。 設定内容についての詳細は、フラッシュ・オプションの設定 ダイアログ【Full-spec emulator】【E1】【E20】 を参照してください。



□ **フラッシュ・オフション** フラッシュ・オフション **フラッシュ・オフション**□


図 2—38 フラッシュ・オプションの設定(フラッシュ・オプションの設定 ダイアログ)

フラッシュ・オプションの設定 🛛 🛛 🛛			
フラッシュ・オブション・プロパティ(円):			
日 オブション・バイト設定			
OPBT0	HEX FFFFFFF		
OPBT1	HEX FFFFFFF		
OPBT2	HEX FFFFFFF		
OPBT3	HEX FFFFFFF		
OPBT4	HEX FFFFFFF		
OPBT5	HEX FFFFFFF		
OPBT6	HEX FFFFFFF		
OPBT7	HEX FFFFFFF		
オブション・バイト設定 読み込み(R)(書き込み)(開じる(C)) キャンセル ヘルブ(H)			

(5)[フック処理設定]タブ

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,「2.17 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—39	動作環境設定	【シミュレータ】	(プロパティ	パネル)
	ヨリート・ネーブレルストレ			

プロパティ	×
🚑 RH850 シミュレータ のプロパティ	P -+
□ クロック	
メイン・クロック周波数[MHz]	320.00
タイマ/トレース用クロック周波数の選択	CPUクロック周波数
タイマ/トレース用クロック周波数の単位	MHz
タイマ/トレース用クロック周波数	2021201
B コンフィギュレーション	
シミュレータ・コンフィギュレーション・ファイルを使用する	いいえ
□ CPU仮想化支援機能	No. 1992
仮想マシン・スレッドを使用する	いいえ
クロック	
│ 接続用設定 / デバッグ・ツール設定 / ダウンロード・	ファイル設定/フック処理設定/ 🗸 🗸

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。

- (1) [接続用設定] タブ
- (2) [デバッグ・ツール設定] タブ
- (3) [ダウンロード・ファイル設定] タブ
- (4) [フック処理設定] タブ

(1)[接続用設定]タブ

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。

- (a) [クロック]
- (b) [コンフィギュレーション]
- (c) [CPU 仮想化支援機能]
- (a) [クロック]

クロックに関する設定を行います。

図 2—40 [クロック]カテゴリ【シミュレータ】

Ξ	クロック	
	メイン・クロック周波数[MHz]	320.00
	タイマ/トレース用クロック周波数の選択	CPUクロック周波数
	タイマ/トレース用クロック周波数の単位	MHz
	タイマ/トレース用クロック周波数	



- [メイン・クロック周波数 [MHz]] メイン・クロック周波数を指定します。 ドロップダウン・リストによる選択、または直接入力により、0.001 ~ 999.999(単位:MHz)の範囲 の周波数を指定します(デフォルト: [320.00])。

- 注意 命令シミュレータ版では、CPU クロック周波数は、常にここで指定するメイン・クロック周波 数の値となります。
- [タイマ/トレース用クロック周波数の選択] タイマ/トレース機能に使用するクロック周波数を表示します。 このプロパティ値を変更することはできません。
- [タイマ/トレース用クロック周波数の単位] タイマ/トレース機能に使用するクロック周波数の単位を表示します。 このプロパティ値を変更することはできません。
- [タイマ/トレース用クロック周波数] タイマ/トレース機能に使用するクロック周波数の値を表示します。 ただし、デバッグ・ツールと切断時は "---_---"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。
- (b) [コンフィギュレーション] このカテゴリ内のプロパティは、常に無効です。
- (c)[CPU 仮想化支援機能]

このカテゴリ内のプロパティは、常に無効です。

(2) [デバッグ・ツール設定] タブ

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。

- (a) [メモリ]
- (b) [実行中のメモリ・アクセス]
- (c) [トレース]
- (d) [タイマ]
- (e) [カバレッジ]
- (f) [シミュレータ GUI]



(a) [メモリ]

メモリに関する設定を行います。

D 777		
回 リエロ・フッピンガ	[96]	
四 メモリ・ イッヒノク	[20]	

- [メモリ・マッピング]

展開表示することにより、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。

ヨ メモリ		
ヨ メモリ・マッピング	[26]	
□ [0]	Code Flash	
メモリ種別	Code Flash	
開始アドレス	HEX ()	
終了アドレス	HEX 3FFFFF	
⊞ [1]	Access prohibited	
	 ⇒ メモリ ⇒ メモリ・マッピング □ [0] ⇒ メモリ種別 開始アドレス 終了アドレス ① [1] 	 ⇒ メモリ ■ メモリ・マッピング [26] □ [0] Code Flash メモリ種別 Gode Flash 開始アドレス 純EX WEX 3FFFFF ① [1] Access prohibited

図 2-42 メモリ・マッピングの詳細表示

注意 メモリ・マッピングの追加/削除はできません。

- 備考 選択しているマイクロコントローラがマルチコア対応版の場合、コア(PEn)の選択を切り替えることにより、PEnごとのメモリ・マッピングの状況を表示します(「2.7 コア(PE)の選択」参照)。
- (b) [実行中のメモリ・アクセス]

プログラム実行中におけるメモリ・アクセス(リアルタイム表示更新機能)に関する設定を行います。 リアルタイム表示更新機能についての詳細は、「(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 2-43 [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

-	夫打中のメモリ・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選択します。 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)。

図 2—41 [メモリ]カテゴリ【シミュレータ】

- [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]を選択した場合にのみ有 効となります。

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: [500])。

(c) [トレース]

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,「2.12 実行履歴の収集」を参照して ください。

(d) [タイマ]

タイマ機能に関する設定を行います。

タイマ機能についての詳細は、「2.13 実行時間の計測」を参照してください。

図 2—44 [タイマ]カテゴリ

タイマ
 タイマ機能を使用する
 いいえ

- [タイマ機能を使用する]

タイマ機能を使用するか否かを選択します。

タイマ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:[いいえ])。

(e) [カバレッジ]

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,「2.14 カバレッジの測定【シミュレータ】」を参照してください。

(f) [シミュレータ GUI]

シミュレータ GUI 機能に関する設定を行います。

注意 選択しているマイクロコントローラのシミュレータが周辺機能シミュレーションをサポートしてい ない場合(命令シミュレータ版の場合)、このカテゴリ内のすべてのプロパティは無効となります。

図 2—45 [シミュレータ GUI]カテゴリ

□ シミュレータGUI

シミュレータGUIを表示する	はい
実行開始時に最前面表示する	(สถา

- [シミュレータ GUI を表示する]

シミュレータ GUI ウインドウを表示するか否かを選択します。

シミュレータ GUI の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)。

シミュレータ GUI の機能を使用しない場合は [いいえ] を選択することにより, シミュレータ GUI ウ インドウがクローズします。

- [実行開始時に最前面表示する]

このプロパティは, [シミュレータ GUI を表示する] 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。

プログラムの実行開始時に、シミュレータ GUI ウインドウを最前面に表示するか否かを選択します。 最前面に表示する場合は [はい] を選択してください (デフォルト)。

(3) [ダウンロード・ファイル設定] タブ

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。

(4) [フック処理設定] タブ

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,「2.17 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、CubeSuite+とデバッグ・ツールの接続方法、および切断方法について説明します。

2.4.1 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、CubeSuite+ はアクティブ・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。





- 注意 CubeSuite+ のサポート範囲外のコンパイラを使用している場合、[デバッグ・ツールへ接続] は無効となります。
- 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。

2.4.2 デバッグ・ツールを切断する

デバッグ・ツールバーの i ボタンをクリックすることにより, CubeSuite+ は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・ツールとの通信が切断すると、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 2-47 デバッグ・ツールを切断したステータスバー

注意 プログラム実行中にデバッグ・ツールを切断することはできません。

備考 デバッグ・ツールを切断すると、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。



2.5 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 CubeSuite+ ヘダウンロー ドする方法と、デバッグ中のメモリ内容を CubeSuite+ 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

Ξ	ダウンロード	
Ξ	ダウンロードするファイル	[1]
	⊟ [0]	Debug Build¥RH850test.abs
	ファイル	Debug Build¥RH850test.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
	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	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—48 [ダウンロード] カテゴリ

(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します)。

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(CubeSuite+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, 「CubeSuite+ 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。

(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]【Full-spec emulator】【E1】【E20】

ダウンロード実行前にフラッシュ ROM を消去するか否かを指定します。

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(デフォルトでは [いいえ] が指定されます)。

(d) [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 2-49 [デバッグ情報] カテゴリ

Ξ	デバッグ情報	
	CPUリセット後に指定シンボル位置まで実行する	はい
	指定シンボル	_main

(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]を選択している場合のみ)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]を選択している場合では、この プロパティで [はい]を選択すると、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]を選択した場合では、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] を選択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。

直接入力により、0~ "*アドレス空間の終了アドレス*"の範囲のアドレス式を指定してください(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。
 アセンブリ・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

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] を選択してください。

(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。

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(「2.17 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、およびダウンロード条件は変更することができます。 CubeSuite+ では、次のファイルをダウンロードすることができます。

ダウンロード可能なファイル	拡張子	ファイル形式
ロード・モジュール・ファイル	.abs	ロード・モジュール・フォーマット
	.out【GHS】 ^注	
インテル拡張へキサ・ファイル	.hex	インテル拡張へキサ・フォーマット

表 2-1 ダウンロード可能なファイル



ダウンロード可能なファイル	拡張子	ファイル形式
モトローラ・Sタイプ・ファイル	.mot	モトローラ・Sタイプ・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
バイナリ・ファイル	.bin	バイナリ・フォーマット

注 GHS コンパイラ (米国 Green Hills Software, Inc. 製) 使用上の注意点

- 対応バージョン

- MULTI (Ver.6.1.4), GHS コンパイラ (Ver.2012.5.5)

- 対応オプション

デバッグ・オプション	:	-G, -dual_debug, -cpu=rh850 / -cpu=v850e3
最適化オプション	:	-Odebug, -O, -Ospeed, -Onone
その他	:	-prepare_dispose, -callt

- 非対応オプション

リンカ最適化オプション : -shorten_loads, -code_factor, -delete

(リンカによって実コードが変更され、デバッグ情報にその変更が反映されないため)

- デバッグ上の注意点
 - 作成したロード・モジュール・ファイル(*.out)は、デバッグ専用プロジェクトに追加してください。
 - 次のプログラムのデバッグはできません。
 - C++ 言語
 - C99 言語固有の型/修飾子を使用したプログラム
 - GNUC拡張仕様を使用したプログラム
 - ステップ、および実行関連機能について

```
次の関数内からのリターン・アウト実行は失敗することがあります。この場合は、コール・スタッ
クパネルの呼び出し履歴も正しく表示されません。
```

- callt で呼び出された関数
- 割り込み関数
- 式を使用した変数の参照機能について
- long long 型,または double 型の変数がレジスタに割り付いて場合,ウォッチパネルの[アドレス]エリアには下位4バイト側のレジスタ名のみ表示されます。また,この時,CubeSuite+は,上位4バイトの値を表示したレジスタの次のレジスタから値を取得します。
 - 例) [アドレス] エリアに"R4"が表示されている場合, CubeSuite+ は, 上位 4 バイトの値を R5 レジスタから取得します。
- 構造体型の変数がレジスタに割り付いた場合,ウォッチパネルでは構造体メンバの値を正しく表示できません。値は[アドレス]エリアに表示したレジスタを CPU レジスタパネルで参照してください。
- スコープ指定付きの式を使用しても関数内に定義したスタティック変数は参照できません (PC カウンタ値が変数を定義した関数内にある場合のみ参照可能です)。

```
func() {
    static sta = 100;
}
```

上記例の場合

func() 関数をデバッグ中の場合, "sta" および "func()#sta"の両方で参照できます。 func() 関数以外をデバッグ中の場合, "sta", または "func()#sta"のどちらでも参照できません。

- 関数の先頭位置ではスタック・フレームを形成していないため、スタック渡しの引数のアドレス が正しくありません。引数の値は、関数内へステップしてから参照してください。
- その他の機能について
 - シンボル名の入力補完機能を無効化することはできません(ダウンロード・ファイル ダイアログの[入力補完機能用の情報を生成する]項目の指定は無視されます)。

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 をクリックするとオープンします。

図 2—50 ダウンロード・ファイル ダイアログのオープン

Ξ	ダウンロード	
Ŧ	ダウンロードするファイル	[1] (🛄)
	ダウンロード後にCPUをリセットする	(th)
	ダウンロード前にフラッシュ 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

ダウンロード・ファイル		X
ダウンロード・ファイル一覧(E):	ダウンロード・ファイルのプロパティ(<u>P</u>):	
RH850.abs	□ ダウンロード・ファイル情報	
-	ファイル	DefaultBuild¥RH850.abs
[10)	ファイルの種類	ロード・モジュール・ファイル
+ (<u>D</u>)	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
	入力補完機能用の情報を生成する	はい
<u>追加(A)</u> 「ダウンロード・ファイルー覧」エ	ダウンロード・ファイル情報 [ダウンロード・ファイルの	プロパティ] エリア

図 2-51 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)



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。

- (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) ダウンロード・ファイル (*.hex/*.mot/*.bin) を追加する
- (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
- (4) ロード・モジュール・フォーマット以外のファイルでソース・レベル・デバッグを行う
- (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

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。

(a) 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。

(b) ダウンロード条件の変更

[ダウンロード・ファイルのプロパティ] エリアでは、選択しているロード・モジュール・ファイルのダ 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	指定したファ	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選択します。			
ウンロードする	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			
	指定可能值	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	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選択します ^{注1} 。		
ロードする	デフォルト	はい	はい		
	変更方法	ドロップダ	ウン・リストによる選択		
	指定可能值	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。		
入力補完機能用の 情報を生成する	ダウンロード す ^{注 2} 。	時に、シンボ	ル名の入力補完機能のための情報を生成するか否かを選択しま		
	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		
	指定可能值	はい	シンボル名の入力補完機能用の情報を生成します(入力補完		
			機能を使用します)。		
		いいえ	シンボル名の入力補完機能用の情報を生成しません(入力補		
			完機能を使用しません)。		

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。

- **2.** [はい]を選択した場合、ダウンロード時間、およびホスト・マシンのメモリ消費量が増加しま す。シンボル名の入力補完機能を使用しない場合は、[いいえ]を選択することを推奨します。
- (c) [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) ダウンロード・ファイル(*.hex/*.mot/*.bin)を追加する

ロード・モジュール・フォーマット以外のファイル(インテル拡張ヘキサ・ファイル(*.hex)/モトロー ラ・S タイプ・ファイル(*.mot)/バイナリ・ファイル(*.bin))をダウンロード対象に追加する場合は、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。

(a) [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・ファイルー覧] エリアの最終行に空欄の項目("-") が表示されます。

(b) 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(インテル拡張ヘキサ・ファイル(*.hex)/モトロー					
	ラ・Sタイプ	ラ・S タイプ・ファイル(*.mot)/バイナリ・ファイル(*.bin))を指定します(最大指定					
	文字数:259 文字)。						
	デフォルト	空欄	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					
	指定可能值	「表 2—1 ダウンロード可能なファイル」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選択します。					
	ここでは, [ロ	ード・モジュール・ファイル]以外を選択します。					
	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					
	指定可能值	ロード・モジュール・ファイル	ロード・モジュール・フォーマット				
			(*.abs)を指定します。				
		ヘキサ・ファイル	インテル拡張ヘキサ・フォーマット				
			(*.hex)を指定します。				
		S レコード・ファイル モトローラ・S タイプ・フォーマット					
		(*.mot)を指定します。					
		バイナリ・データ・ファイル バイナリ・フォーマット(*.bin)を指定し					
			ます。				



オフセット	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。					
	なお、この項目は、[ファイルの種類] に [ヘキサ・ファイル]、または [S レコード・フ イル] を選択している場合のみ表示されます。					
	デフォルト 0					
	変更方法 キーボードからの直接入力					
	指定可能值	0x0~0xFFFFFFFFの16進数				
開始アドレス	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。				
	なお、この項目	目は、[ファイルの種類]に[バイナリ・データ・ファイル]を選択している				
	場合のみ表示。	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				
	指定可能值	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。
- (c) ダウンロードの際の実行順序の確認

[ダウンロード・ファイルー覧] 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。

(d) [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)。

(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。
- (a) [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・ファイルー覧] エリアの最終行に空欄の項目("-") が表示されます。

(b) 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・ファイルー覧] 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		
	指定可能值	「表 2—1 🔮	ダウンロード可能なファイル」 参照	
ファイルの種類	追加するダウ	ンロード・フ	アイルのファイル形式を指定します。	
	ここでは, [ロ	ュード・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	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選択します。	
ウンロードする	デフォルト	ット はい k ドロップダウン・リストによる選択		
	変更方法			
	指定可能值	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	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選択します ^{注1} 。	
ロードする	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	
	指定可能值	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。	
入力補完機能用の 情報を生成する	ダウンロード す ^{注 2} 。	時に、シンボ	ル名の入力補完機能のための情報を生成するか否かを選択しま	
	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		
	指定可能值	はい	シンボル名の入力補完機能用の情報を生成します(入力補完	
			機能を使用します)。	
		いいえ	シンボル名の入力補完機能用の情報を生成しません(入力補	
			完機能を使用しません)。	

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。

- **2.** [はい]を選択した場合、ダウンロード時間、およびホスト・マシンのメモリ消費量が増加しま す。シンボル名の入力補完機能を使用しない場合は、[いいえ]を選択することを推奨します。
- 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)。
- (c) 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。 (d) [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)。

(4) ロード・モジュール・フォーマット以外のファイルでソース・レベル・デバッグを行う

インテル拡張へキサ・ファイル (*.hex), モトローラ・Sタイプ・ファイル (*.mot), またはバイナリ・ファ イル (*.bin) をダウンロード対象のファイルと指定している場合でも, これらのファイルの作成元となった 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。

(a) [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・ファイルー覧] エリアの最終行に空欄の項目("-") が表示されます。

(b) 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	ダウンロードするインテル拡張ヘキサ・ファイル(*.hex),モトローラ・S タイプ・ファ					
	イル (*.mot), またはバイナリ・ファイル (*.bin) を作成する基となったロード・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	ジュール・ファ	ァイル]を選択します(デフォルト)。			
オブジェクトをダ	[いいえ] を選	選択します。				
ウンロードする						
シンボルをダウン	[はい] を選択します(デフォルト)。					
ロードする						
入力補完機能用の	ダウンロード時に、シンボル名の入力補完機能のための情報を生成するか否かを選択しま					
情報を生成する	す ^注 。					
	デフォルト	はい				
	変更方法	方法 ドロップダウン・リストによる選択				
	指定可能值	はい シンボル名の入力補完機能用の情報を生成します(入力補完				
		機能を使用します)。				
		いいえ シンボル名の入力補完機能用の情報を生成しません(入				
		完機能を使用しません)。				

注 [はい]を選択した場合、ダウンロード時間、およびホスト・マシンのメモリ消費量が増加します。 シンボル名の入力補完機能を使用しない場合は、[いいえ]を選択することを推奨します。

(c) [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)。

2.5.3 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—52	メモリ内容のアップロード	(データ保存 ダイアログ)
--------	--------------	---------------

データ保存 - デバ	ッグ・ツールからアップロード	
ファイル名(N):		() 💌
ファイルの種類(工):	インテル・ヘキサ・フォーマット(*)	.hex)
保存範囲 アドレス	ノシンボル(<u>A</u>):	
(保存範囲の開始	☆ モンガレてく ●● ● -	(保存範囲の終了位置を入力してく 🕨 💙
1		
		保存(S) キャンセル ヘルプ(H)
		1*14 27 270 101

(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—	27	ップロー	ド可能なフ	アイ	ル形式
------	----	------	-------	----	-----

リスト表示	ファイル形式		
インテル・ヘキサ・フォーマット (*.hex)	インテル拡張へキサ・フォーマット (常に拡張リニア・アドレス・レコードを使用)		
モトローラ S フォーマット (*.mot)	モトローラ・Sタイプ・フォーマット		



リスト表示	ファイル形式
バイナリ・データ (*.bin)	バイナリ・フォーマット

(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(「2.18.2 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/命令レベル・デバッグ(「2.8.3 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】(「2.14 カバレッジの測定【シミュレータ】」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(「2.8.3 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】(「2.14 カバレッジの測定【シミュレータ】」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、インテル拡張ヘキサ・ファイル(*.hex)、モトローラ・Sタイプ・ファ イル(*.mot)、またはバイナリ・ファイル(*.bin)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(「(4) ロード・モジュール・フォーマット以外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、エディタパネルの項を参照してください。

備考 [ファイル] メニュー→ [エンコードを指定して開く ...] の選択によりオープンするファイル・エンコー ドの選択 ダイアログにより、エンコードを指定してファイルをオープン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。

- (1) 表示モードを変更する
- (2) 表示カラムを設定する
- (3) 複数のソース・ファイルを1枚のパネルで表示する
- (4) 変数値を表示する
- (5) 文字列を検索する
- (6) 指定行へ移動する
- (7) 関数ヘジャンプする
- (8) タグ・ジャンプする
- (9) ブックマークを登録する

(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, エディタ パネルの表示モードを切り替えることができます。

- 通常表示モード

行番号/アドレス/ソース・テキストなどを表示する、デフォルトの表示モードです。



図 2-54 通常表示モード(エディタ パネル)

12 01000220 🍂	🔷 (void main(int args)
13	E {
14	gc_pe1 = 0x12;
1 <u>た</u>	gs_pe3 = 0x1/34;
行番号	gi = 0x123 ソース・テキスト
・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	g = 0x120+0070720+0070;

- 混合表示モード

通常表示モードの表示内容に加え,各ソース・テキストに対応した,命令コード/ラベル/逆アセンブ ル・テキストも併せて表示します。



- 注意 1. 混合表示モードは、デバッグ・ツールと接続中で、ダウンロードしたソース・ファイルをオープン している場合のみ有効となる機能です。
 - 混合表示モードでは、ダウンロードしているロード・モジュール・ファイルからソース・テキスト とそれに対応したアセンブラ・コードの情報を取得して表示します。そのため、編集したソース・ テキストの結果を混合表示モードで反映する場合は、リビルド→ダウンロードを実行する必要があ ります。
 - 混合表示モードでは、ソース・ファイルの編集はできません。
 また、[編集] メニュー→ [やり直し] / [切り取り] / [貼り付け] / [削除] / [すべて選択]
 / [置換 ...] / [ブックマーク] / [コードのアウトライン] / [高度な設定] 項目は無効となります。
- 備考 [ファイル] メニュー→ [名前をつけて 混合表示 を保存 ...] を選択することにより、現在表示してい る混合表示の内容をテキスト形式/ CSV 形式で保存することができます(非表示に設定しているカ ラムの内容は保存されません)。
- (2) 表示カラムを設定する

ツールバーの次の項目の選択により, エディタ パネルで表示するカラム, またはマークを設定することができます。

なお、この表示カラムの設定は、すべてのエディタパネルに適用されます。

1	コラム	すべてのエディタ パネルで表示するカラム、またはマークの表示/非表示を切り替
		える次の項目を表示します。
		チェックを外すことにより非表示となります(デフォルトではすべての項目が
		チェックされています)。
	行	行番号エリアにおいて、行番号を表示します。
	選択	行番号エリアにおいて、行の編集状態を示すマークを表示します。
	ソース・ファイルがロー	行番号エリアにおいて、ダウンロードしたロード・モジュール・ファイルの更新状
	ドモジュールより新しい	態を示すマークを表示します。
	ことを示す表示	ただし、デバッグ・ツールと接続中の場合のみ有効となります。
	カバレッジ	カバレッジ・エリアを表示します。
		ただし、デバッグ・ツールと接続中の場合のみ有効となります。
	アドレス	アドレス・エリアを表示します。
		ただし、デバッグ・ツールと接続中の場合のみ有効となります。
	命令コード	命令コード・エリアを表示します。
		ただし、デバッグ・ツールと接続中で、かつ混合表示モードの場合のみ有効となり
		ます。
	ラベル	ラベル・エリアを表示します。
		ただし、デバッグ・ツールと接続中で、かつ混合表示モードの場合のみ有効となり
		ます。
	イベント	イベント・エリアを表示します。
		ただし、デバッグ・ツールと接続中の場合のみ有効となります。
	メイン	メイン・エリアを表示します。
	カラム・ヘッダ	カラム・ヘッダを表示します。

(3) 複数のソース・ファイルを1 枚のパネルで表示する

デバッグ作業時,ステップ実行などで,カレント PC が複数のソース・ファイルをまたいで移動する場合, 複数のソース・ファイルのエディタ パネルをオープンせず,1枚のエディタ パネル内で順に複数のソース・ ファイルを表示することができます(リサイクル・モード)。

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-テキスト・エディタ]カテゴリ内の[リ サイクル・モード]をチェックします。



図 2---56 通常の動作



図 2---57 リサイクル・モードの動作

🧹 ma	ain.c [リサイクル]	
1		
20	0100025a 🕪 sub(gi);	
22 23	01000262 if(args == 0x1)	
24	「sub.c [リサイクル] 実行	
26		
47	行	^
	8 0100029a 🕪 void sub(int g)	
	9 10 🖂 1	
	11 12 int a.b.c;	
	13 14	~
		2

- 注意 1. リサイクル・モードは、デバッグ・ツールと接続中で、ダウンロードしたソース・ファイルをオー プンしている場合のみ有効となる機能です。
 - 編集を行ったリサイクル・モードのエディタパネル上にカレント PC が存在する状態でプログラム を実行した場合、編集を行ったエディタパネルは、リサイクル・モードを解除し、新たにオープン するエディタパネルがリサイクル・モードとなります。

備考 すでに該当するソース・ファイルを表示しているエディタ パネルがオープンしている場合, リサイク ル・モードのパネルには表示せず, すでにオープンしているエディタ パネルを表示します。

(4) 変数値を表示する

ソース・テキスト上の変数にマウス・カーソルを重ねることにより、"<*変数名*> = < *変数値*>"をポップ アップ表示します。

この際の変数値の表示形式は、変数値の型に依存します(「表 A—7 ウォッチ式の表示形式(デフォルト)」と同等)。

図 2-58 変数値のポップアップ表示(エディタ パネル)

<pre>m_minute_= 0: m_second = 0;</pre>	— マウス・カ ー ソル
dot_fla /* init	=0 (0x0)

- 注意 変数値のポップアップ表示は、デバッグ・ツールと接続中で、ダウンロードしたソース・ファイルを オープンしている場合のみ有効となる機能です。
- (5) 文字列を検索する

ソース・テキスト内の文字列の検索は、ツールバーの 満 ボタンを選択することによりオープンする検索・ 置換 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 2-59 ソース・テキスト内の文字列検索(検索・置換 ダイアログ)

検索·置換				
クイック検索	一括検索	クイック置換	一括置換	
検索する文字	⊆歹川(<u>T</u>): m	ain		
置換後の文字	字列(业):			×.
検索場所(1)	Ę	見在のパネル(r	nain.c)	
<u>オプション(0</u>)]			前を検索(P) 次を検索(N) キャンセル ヘルブ(H)

(a) [検索する文字列]の指定

検索する文字列を入力します。

デフォルトでは、エディタパネル上のキャレット位置の単語(変数/関数)が表示されます。 変更する場合は、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)。

(b) [検索場所]の指定

ドロップダウン・リストより、 [現在のパネル (ファイル名)]を選択します。

(c) [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、指定した検索場所でアドレスの小さい方向に検索を行い、検索結 果箇所をエディタ 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、指定した検索場所でアドレスの大きい方向に検索を行い、検索結 果箇所をエディタ パネル上で選択状態にします。

- 備考1. [オプション] ボタンをクリックすることにより、ワイルド・カードの使用/大文字と小文字の区 別/単語単位の検索などを指定することができます。
 - **2.** 検索・置換 ダイアログでは、[一括検索] / [クイック置換] / [一括置換] タブを選択すること により、様々な検索/置換操作を行うことができます。
- (6) 指定行へ移動する

ソース・テキスト上の指定行への移動は、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 プンする指定行へのジャンプ 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 2-60 ソース・テキストの指定行へ移動(指定行へのジャンプダイアログ)

指定行へのジャンプ	¢.	X
行番号 (1 - 44) また	±はシンボル(L):	
12		
ОК	キャンセル	ヘルプ(円)

(a) [行番号(*有効な行の範囲*)またはシンボル]エリア

キャレットを移動したい行番号(10進数)/シンボル名^{注1}/アドレス値^{注2}を直接入力により指定します。

なお、"(*有効な行の範囲*)"には、現在のファイルの有効な行番号の範囲が表示されます。 デフォルトでは、エディタパネル上の現在のキャレット位置の行番号が表示されます。

- 注1. 次の注意が必要となります。
 - シンボル名として、関数名/変数名のみ指定可能です。
 - 使用するビルド・ツールのプロパティ パネルにおいて、クロス・リファレンス情報を出力する 設定([共通オプション] タブ→ [出力ファイルの種類と場所] カテゴリ→ [クロス・リファレ ンス情報を出力する] プロパティ→ [はい (-Xcref)]) にしたのち、ビルドの実行を完了させて ください。
 - ビルドの際にエラーが発生した場合, エラー発生前のクロス・リファレンス情報を使用します。
 - 2. 次の注意が必要となります。
 - アドレス値は、"["と"]"で囲んで指定してください。
 - ビルドの実行を完了させてください。

- ビルドの際にエラーが発生した場合、エラー発生前の情報を使用します。

- (b) [OK] ボタンのクリック 指定した行番号へキャレットを移動します。
- (7) 関数ヘジャンプする

現在選択している文字列, またはキャレット位置の単語を関数名と判断し, 該当する関数へジャンプすることができます(対象関数内で最初の実行可能行へジャンプ)。

ソース・テキスト上で、対象関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[関数へジャンプ]を選択してください。

注意 1 行に複数のステートメントを記述している場合、不正な箇所へジャンプすることがあります。



図 2-61 関数ヘジャンプ

なお、この機能は、使用するビルド・ツールに依存して次の条件を満たしている場合のみ有効となります。

(a) CC-RH の場合

- デバッグ・ツールと切断している場合

- アクティブ・プロジェクトの種類が"アプリケーション"である。
- 対象の関数がグローバル関数である。
- [ダウンロードするファイル]の1番目に指定されたファイルに対象の関数が定義されており,
- また、このファイルにはシンボル情報が含まれている。

- デバッグ・ツールに接続し、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
 - ロード・モジュール・ファイルにシンボル情報が含まれている。

- プログラム・カウンタ(PC)の指すアドレスから呼び出し可能な関数である。 例えば、PCの指すアドレスのファイル以外で定義したスタティック関数へはジャンプできません。

- 備考 同名の関数が複数存在する場合は、関数へジャンプダイアログがオープンし、ジャンプ先の関数 を選択することができます。 ただし、この機能は、使用するビルド・ツールのプロパティ パネルにおいて、クロス・リファレ ンス情報を出力する設定([共通オプション] タブ→[出力ファイルの種類と場所] カテゴリ→ [クロス・リファレンス情報を出力する] プロパティ→[はい (-Xcref)]) となっている場合のみ 有効です。
- (b) 外部ビルド・ツールの場合
 - 対象がアクティブ・プロジェクト内の関数である。
 - ・シンボル情報を持つファイルが [ダウンロードするファイル] に指定されている (ロード・モジュール・ファイル以外の場合,シンボル情報をダウンロードする設定が必要です (「(4) ロード・モジュール・フォーマット以外のファイルでソース・レベル・デバッグを行う」参照))。
 - ただし, デバッグ・ツールと切断している場合は, [ダウンロードするファイル] の1番目に指定されている。

注意 デバッグ・ツールと切断している場合は、スタティック関数へのジャンプはできません。

備考 単語の判断は、現在のビルド・ツールに依存します。

(8) タグ・ジャンプする

現在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,該当ファイルを新たなエディタパネルに オープンし,該当行/桁へジャンプすることができます(すでにオープンしている場合は,該当エディタパネ ルにジャンプ)。

ソース・テキスト上で、対象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 [タグ・ジャンプ] を選択してください。

タグ・ジャンプの動作は次のとおりです。

文字列の例	動作
C: ¥ work ¥ src.c	ファイル"C:¥work¥src.c"の先頭行にジャンプ
Tmp ¥ src.c	ファイル"Tmp¥src.c"の先頭行にジャンプ
C: ¥ work ¥ src.c(10)	ファイル "C:¥work¥src.c"の10行目にジャンプ
C:¥ [″] work sub¥ src.c [″] (10)	ファイル "C:¥work sub¥src.c"の10行目にジャンプ
C: ¥ work ¥ src.c(10,5)	ファイル "C:¥work¥src.c"の10行5桁目にジャンプ

表 2---3 タグ・ジャンプの動作

図 2—62 タグ・ジャンプ



備考1. 大文字/小文字の区別は行いません。

- **2.** 相対パスによる指定の場合は、ファイルが登録されているプロジェクト・フォルダを基点としま す。ただし、プロジェクトに属さない場合は、アクティブ・プロジェクトを基点としてます。
- 3. パス指定(パス/ファイル名)にスペースを含む場合は、必要があります。
- (9) ブックマークを登録する

ブックマーク・ツールバーの 「ボタンをクリックすることにより,現在キャレットのある行にブックマー クを登録することができます。

ブックマークを登録すると, [メイン] エリアにブックマーク() が表示されます。 なお、すでにブックマークを登録している行でこの操作を行った場合は、そのブックマークを削除します。 ブックマークは、エディタ パネルごとに最大 50 個まで設定することができます。

- 注意 1. 混合表示モードを選択している場合、ブックマークの登録/表示を行うことはできません。
 - ブックマークを含む行を削除したのち、[編集]メニュー→ [元に戻す]を選択しても、ブック マーク情報は復元しません。



図 2-63 ブックマークの登録



- 備考1. ブックマーク情報は現在開いているプロジェクト・ファイルに保存され,再びプロジェクトをオー プンした際に復元します。そのため,プロジェクトに属していないファイルに対してブックマーク の設定を行った場合は、ブックマークの復元を行いません。
 - ブックマーク・ツールバーの / 「ボタンをクリックすることにより、前/次のブックマーク 位置へキャレットを移動することができます。

なお、この場合の移動順序は、ブックマークを登録した順となります(行番号順ではありません)。

 ブックマーク・ツールバーの ・ ブックマーク・ツールバーの ・ ボタンのクリックでオープンするブックマーク ダイアログにより、現在登録しているブックマークをリスト表示します。

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。

34:	void main() {		
	_main:		100 March 100 Ma
▶ 00000394	a515	br	_main+0x24
36:	func();	iarl	func In
37.	sfunc():	Jari	_runc, rp
0000039a	80ff2200	iarl	sfunc, lp
38:	nosource();		
0000039e	80ff0e13	jarl	_nosource, lp
39:	parent_num_2();		
000003a2	80ff2a00	jarl	_parent_num_2, I
40:	parent_num_3();	1	0 I
41.	func2():	Jari	_parent_num_s, i
S 000003aa	bfffc6ff	iarl	func2. lp
42:	sub02 main();	2	
000003ae	- 80ffba00	jarl	_sub02_main, lp
<			

図 2-64 逆アセンブル結果の表示(逆アセンブルパネル)

ここでは、次の操作方法について説明します。

- (1) 表示モードを変更する
- (2) 表示形式を変更する
- (3) 指定アドレスへ移動する
- (4) シンボル定義箇所へ移動する
- (5) 逆アセンブル結果の表示内容を保存する

(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。

12:	void main(int args)		
	_main:	テキスト	
□ ↓ 01000220	80072	epare	r2U, Ip, UxU
01000224	06a0	MOV	r6, r20
14:	(gc_pe1 = 0x12;		
01000226	4016e0fe	movhi	Oxfee0, r0, r2
0100022a	202e1200	movea	0x12, r0, tp
0100022e	422f0c80	st.b 📐	tp, -0x7ff4[r2]
15:	gs_pe3 = 0 <u>×123</u>	14:	
01000232	4016e0fe 逆ア・	センブル・テキスト	0xfeeO, rO, r2
01000236	202e3412	movea	0x1234, r0, tp
0100023a	622f0e80	st.h	tp, -0x7ff2[r2]
0100023e	220678563412	mov	0×12345678, r2

図 2---65 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 2-66 逆アセンブル表示モード(逆アセンブル パネル)

	_main:				
🔲 🔶 01000220		80072108		prepare	r20, lp, 0x0
01000224		06a0		MOV	r6, r20
01000226		4016e0fe	!	movhi	0xfee0, r0, r2
0100022a		202e1200		movea	0x12, r0, tp
0100022e		422f0c80		st.b	tp, -0x7ff4[r2]
01000232		4016e0fe	!	movhi 📐 👘	0xfeeO, rO, r2
01000236		202e3412	溢アムンゴ		0×1234, r0, tp
0100023a		622f0e8(近りビノノ		tp, -0x7ff2[r2]
0100023e		22067856	3412	mov	0x12345678, r2

(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。	
	1	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。	
	ALBO C	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。	
		レジスタ名を機能名称で表示します(デフォルト)。	
		レジスタ名を絶対名称で表示します。	

(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 2-67 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)

指定位置へ移動		8
アドレス/シンボル(<u>A</u>):		~
ОК	キャンセル	~117(H)

(a) [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(「2.18.2 シンボル名の入力補完機能」参照)。
- (b) [OK] ボタンのクリック

指定したアドレスヘキャレットを移動します。

(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。

(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 2-68 逆アセンブル・データの保存(データ保存 ダイアログ)

データ保存		
ファイル名(N):	C:¥Test¥sample¥RH850¥逆アセンブル1	_ Q
ファイルの種類(工):	テキスト・ファイル (*.txt)	~
保存範囲 アドレス _sub03_sub01+0>	/シンボル(<u>A</u>): :00000026	~
	(保存(S) キャンセル	ヘルプ田

(a) [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。

(b) [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。

(c) [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(「2.18.2 シンボル名の入力補完機能」参照)。

(d) [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。

 ラベル(シンボル名) :			◆ ラベル(シンボル)行
<i>ファイル名</i> :	<i>行番号</i> :	<i>c ソース</i> :	← ソース・テキスト行
アドレス オフセット : :	ゴード	<i>逆アセンブル結果</i> :	◀── 逆アセンブル結果行

図 2-69 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。
 - **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の画像 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

CubeSuite+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)。

- 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/アセンブリ・ソース・ファイル/ヘッダ・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/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、およびライブラリ・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

- 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/アセンブリ・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/アセンブリ・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[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。
 - 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。
 - **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。

- (1) 命令を編集する
- (2) 命令コードを編集する

(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。

(a) 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。

(b) 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。

(c) 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。

編集前	0432	mov	0x4, r6
	1d38	mov	r29, r7
	8f071b0effff0000	prepare	r20, r21, r22, 0x1c, 0x0000ffff
	0132	mov	0x1, r6
編集後	0432	mov	0x4, r6
	1d38	mov	r29, r7
	bfffe265	jarl	0x100, lp
	0000	nop	
	0000	nop	
	0132	mov	0x1, r6

例 1. 3 行目の prepare 命令(8 バイト命令)を jarl 命令(4 バイト命令)で上書きした場合

2. 2 行目の mov 命令(2 バイト命令)を jarl 命令(4 バイト命令)で上書きした場合

編集前	0432	mov	0x4, r6
	1d38	mov	r29, r7
	8f071b0effff0000	prepare	r20, r21, r22, 0x1c, 0x0000ffff
	0132	mov	0x1, r6
編集後	0432	mov	0x4, r6
	bfffe265	jarl	0x100, lp
	0000	nop	
	0000	nop	
	0000	nop	
	0132	mov	0x1, r6

注意 prepare 命令 /dispose 命令の扱いについて

prepare 命令 /dispose 命令の命令形式は,次のとおりで,オペランドの "list12" には, 12 ビットの値 が入り,各ビットごとに対応するレジスタが割り当てられます。

prepare 命令の命令形式	prepare	list12, imm5
	prepare	list12, imm5, sp/imm
dispose 命令の命令形式	dispose	imm5, list12
	dispose	imm5, list12, [reg1]

逆アセンブル パネルでは, prepare 命令 /dispose 命令の逆アセンブル結果を表示する場合, オペランドの "list12" は値ではなく, 次の例のように対応するレジスタ名を表示します。

例1. 命令コードが "0x91, 0x07, 0xe1, 0xff" (prepare の 4 バイト命令) の場合



 表示
 prepare
 r20, r21, r22, r23, r24, r25, r26, r27, r28, r29, r30, r31, 0x20

 正表記
 prepare
 0xfff, 0x20

2. 命令コードが "0x90, 0x07, 0xbb, 0xaa 0xff, 0xff, 0xff, 0xff "(prepare の 8 バイト命令)の場合

表示	prepare	r20, r22, r24, r26, r28, r31, 0x20, 0x7fffffff
正表記	prepare	0x555, 0x20, 0x7fffffff

3. 命令コードが "0x51, 0x06, 0xe0, 0xff" (dispose の4バイト命令) の場合

 表示
 dispose
 0x20, r20, r21, r22, r23, r24, r25, r26, r27, r28, r29, r30, r31

 正表記
 dispose
 0x20, 0xfff

4. 命令コードが "0x50, 0x06, 0xaa, 0xaa" (dispose の4バイト命令) の場合

表示	dispose	0x20, r20, r22, r24, r26, r28, r31, [r10]
正表記	dispose	0x20, 0x555, [r10]

ただし, prepare 命令 /dispose 命令をライン・アセンブルする場合は, オペランドの "list12" には, 値/レジスタ名の両方の指定が可能です。

例 1. (1) と(2) の指定は、ライン・アセンブルの結果、同じ値の "0x91, 0x07, 0xe1, 0xff" となり ます。

(1)	prepare	r20, r21, r22, r23, r24, r25, r26, r27, r28, r29, r30, r31, 0x20
(2)	prepare	0xfff, 0x20

2. (1) と (2) の指定は、ライン・アセンブルの結果、同じ値の "0xbe, 0x07, 0xbb, 0xaa 0xff, 0xff, 0x7f" となります。

(1)	prepare	r20, r22, r24, r26, r28, r31, 0x20, 0x7fffffff
(2)	prepare	0x555, 0x20, 0x7fffffff

3. (1) と(2) の指定は、ライン・アセンブルの結果、同じ値の"0x51, 0x06, 0xe0, 0xff"となり ます。

 (1)
 dispose
 0x20, r20, r21, r22, r23, r24, r25, r26, r27, r28, r29, r30, r31

 (2)
 dispose
 0x20, 0xfff

4. (1) と(2) の指定は、ライン・アセンブルの結果、同じ値の "0x50, 0x06, 0xaa, 0xaa"となります。

(1) dispose 0x20, r20, r22, r24, r26, r28, [r10]

(2) dispose 0x20, 0x555, [r10]

(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。

(a) 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。

(b)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。

(c) 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 コア(PE)の選択

この節では、選択しているマイクロコントローラがマルチコア対応版の場合における、デバッグ対象となるコア (PE:プロセッサ・エレメント)の選択方法について説明します。

CubeSuite+では、デバッグ対象とするコア(PEn)の選択を切り替えることにより(「2.7.1 コア(PE)を切り 替える」参照)、PEnごとの情報を表示します(PEnごとに複数のパネルの表示は行いません)。

なお、マルチコア対応版を対象とした CubeSuite+の各機能の振る舞いは次のとおりです。

(1) プログラムの実行制御

全 PE において, 原則として同期実行/同期ブレークを行います。

ただし、ステップ実行については、次の動作となります。

- [Full-spec emulator] [E1] [E20]

命令レベル単位で1命令ずつ実行します。

-【シミュレータ】

動作周波数に従い同期してステップ実行します。

- 注意 ステップ実行は、現在選択している PEn でのみ行います。 ただし、ソース・レベル単位のステップ実行の場合は、選択外の PEn が実行される場合があります。
- (2) イベントの発生

イベントは全 PE で有効となるように自動設定されます。

注意 1. 【Full-spec emulator】【E1】【E20】

Local RAM self 領域については、現在選択している PEn にブレークポイントを設定します。

2. 【シミュレータ】

Local RAM self 領域へのアクセスであれば, それがどの PEn からのアクセスであってもイベント が発生します。

ただし、Local RAM self 領域の実態に直接アクセスした場合、イベントは発生しません。

- (3) メモリ/レジスタ/変数などの情報
 - (a) メモリ・マップ

現在選択している PEn により、メモリ・マップが異なる場合があります。 この場合、プロパティ パネルの [デバッグ・ツール設定] タブ上 [メモリ] カテゴリ、およびメモリ・ マッピング ダイアログでは、PEn を切り替えることによって対応したメモリ・マップを表示します。

(b)メモリ範囲と値

現在選択している PEn にかかわらず,同じ値を表示/設定します。

ただし、Local RAM self 領域については、現在選択している PEnの値を取得し表示/設定します。



- (c) レジスタ (IOR/PC レジスタを含む)の値 現在選択している PEnの値を取得し表示/設定します。
- (d) シンボル (ウォッチ式/変数名を含む) 現在選択している PEn の PC 値を基にアドレスと値を決定します (たとえば、シンボルが特定の PE の みで有効であった場合でも、現在選択している PEn を基にアドレスと値を決定します)。
- (e)コール・スタック情報

現在選択している PEn の値を取得し表示/設定します。

- (4) その他の機能
 - (a) 実行履歴の収集
 - 【Full-spec emulator】【E1】【E20】 動作は、プロパティ パネルの[デバッグ・ツール設定]タブの[トレース]カテゴリ内[トレースの取 得対象設定]プロパティの指定に依存します。
 - [デバッグ対象のコアのみ]を選択している場合(デフォルト)
 現在選択している PEn を対象にトレース・データを収集します。
 したがって、目的のトレース・データを収集するためには、プログラムを実行する前に、PEnの選択を行う必要があります(トレース・データ収集後に PEn を切り替えても、トレースパネルの表示内容は変化しません)。
 - [全てのコア]を選択している場合
 全 PE を対象にトレース・データの収集を行います。
 トレース・データ収集後、トレースパネルでは、PEnを切り替えることによって対応したトレース・データを表示します。
 - -【シミュレータ】

現在選択している PEn を対象にトレース・データを収集します。

したがって、目的のトレース・データを収集するためには、プログラムを実行する前に、PEnの選択を 行う必要があります(トレース・データ収集後に PEn を切り替えても、トレースパネルの表示内容は 変化しません)。

(b) 実行時間の計測

全 PE を対象に実行時間の計測を行います。

計測完了後、PEnを切り替えることによって対応した測定結果を表示します。

(c) カバレッジ測定

全 PE のアクセスを対象にカバレッジ測定を行います。

ただし, Local RAM self 領域については,現在選択している PEn のアクセスのみを対象に測定結果を表示します。

2.7.1 コア(PE)を切り替える

デバッグ対象とするコア(PE)の切り替えは、次のいずれかの方法により行うことができます。

(1) ステータスバーで切り替える

メイン・ウインドウのステータスバー上の次のドロップダウン・リストにより、任意の PEn を選択します。

図 2—70 メイン・ウインドウのステータスバー

CPU リセットを行います。 46 行 13 桁 編集不可	PE1 💌	BREAK	Halt	ᅌ 0x000009ae	タ	🕜 6.210 ms	

(2) デバッグ・マネージャー パネルで切り替える

[表示] メニュー→ [デバッグ・マネージャ] を選択することによりオープンするデバッグ・マネージャパ ネルにおいて,任意の PEn を選択します。

図 2—71	デバッグ・	マネージャ	パネル
--------	-------	-------	-----

デバッグ・マネージ	¥.					
🖓 🗘 🐂 I 📵) 🕑 🙌 🕫 Ci çi 🐺					
デバッグ対象コア	の指定:					
⊙ PE1	O PE3					
デバッグ対象コア	D.状態:					
実行状態:	BREAK					
コア・ステータス:	コア・ステータス:					
カレントPC:	0x0000086e					
C						



2.8 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。
- 備考 マルチコア対応版を対象とした"プログラムの実行制御"については、「2.7 コア(PE)の選択」も参照し てください。

図 2-72 デバッグ・ツールバー(フローティング状態)



図 2—73 [デバッグ]メニュー

デバ	ッグ(<u>D</u>)	
D.	デバッグ・ツールへダウンロード(<u>D</u>)	
6	ビルド&デバッグ・ツールへダウンロ	ード(<u>B</u>) F6
00	デバッグ・ツールへ接続(<u>C</u>)	
D)	デバッグ・ツールからアップロード(<u>し</u>	<u>J</u>)
d to	デバッグ・ツールから切断(<u>N</u>)	Shift+F6
	停止(S)	Shift+F5
	実行(G)	F5
D	ブレークせずに実行(<u>E</u>)	F8
93	ステップ・イン(型	F11
Ģ≡	ステップ・オーバー(<u>O</u>)	F10
¢_	リターン・アウト(<u>B</u>)	Shift+F11
K٦	CPUリセット(<u>T</u>)	Ctrl+F5
147	リスタート(<u>A</u>)	

2.8.1 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 ブレーク中の 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(「2.17 フック処理を設定する」参照)。

2.8.2 プログラムを実行する

プログラムの実行方法には次の種類があります。

- デバッグの目的に応じて実行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、「2.9 プログラムの停止(ブレーク)」を参照してください。
 - (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) 現在のアドレスから実行する
 - (3) PC 値を変更してから実行する
- 備考 プログラムの実行開始直前に、 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(「2.17 フック処理を設定する」参照)。
- (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 ディ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。

- 间 ボタンのクリック(「2.9.2 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(「2.9.3 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(「2.9.4 任意の場所で停止する(ブレーク・イベント)」/「2.9.5 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、 Marka かくしょう この したのち、 A ないない いっかい した場合と同等です。

(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。

(a) 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。

- PC がブレークポイントに到達(「2.9.3 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(「2.9.4 任意の場所で停止する(ブレーク・イベント)」/「2.9.
- 5 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
- (b) 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(「2.9.2 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、アクション・イベントの発生も無視されます。

(c) 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
- 间 ボタンのクリック(「2.9.2 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、アクション・イベントの発生も無視されます。

(3) PC 値を変更してから実行する

カレント PC 値を任意のアドレス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)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。

2.8.3 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、次の設定に依存します。

- エディタ パネルのツールバーの 🛐 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。 - エディタ パネルのツールバーの 数 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。

注意 🚯 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。

- (1) 関数内にステップ・インする(ステップ・イン実行)
- (2) 関数をステップ・オーバする (ステップ・オーバ実行)
- (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/アクション・イベントを 発生しません。
 - 2. 関数のプロローグ/エピローグ処理中、および戻りアドレスが取得できない場合は、エラー・メッセー ジを表示します。
 - 3. 【Full-spec emulator】【E1】【E20】
 ステップ実行中は、割り込みが禁止されます。
 ステップ実行中は、スタンバイ・モードに移行しません。
 - 【シミュレータ】
 ステップ実行中に割り込みハンドラに飛ぶことがあります。
- (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。
 - 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。
 - 3. 関数の入口の処理(プロローグ処理)はスキップされません。プロローグ処理をスキップさせたい 場合は、再度ステップ・イン実行してください。
- (2) 関数をステップ・オーバする (ステップ・オーバ実行)

jarl 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(jarl 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、jarl 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 longjmp 関数のステップ・オーバ実行は、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。

(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 **ご** 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。
 - 2. 関数にステップ・インした直後にリターン・アウト実行はできません。
 - 3. 関数のプロローグ/エピローグ処理中からリターン・アウト実行はできません。
 - 4. 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。
 - 5. 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。



2.9 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。

注意 1. スタンバイ・モード(HALT/STOP/IDLE)中に強制ブレークを行った場合、カレント PC 値はスタンバ イ・モード命令以降の次命令のアドレスとなります。

また、使用するデバッグ・ツールによって、次のように動作が異なります。

- [Full-spec emulator] [E1] [E20]

強制ブレークによりスタンバイ・モードを解除します。

- 【シミュレータ】
 強制ブレークによりスタンバイ・モードを解除しません。
 スタンバイ・モードが解除されているように見えますが、スタンバイ・モードが解除されているか否か
 は、メイン・ウインドウのステータス・バー上の CPU 状態で確認してください。
- 【Full-spec emulator】【E1】【E20】 ブレーク時にターゲット・システムの電圧を下げないようにしてください。ブレーク中に低電圧検出回路 (LVI)、またはパワーオン・クリア(POC)によるリセットが発生した場合、CubeSuite+の不正動作や 通信エラーの原因となる場合があります。 なお、ターゲット電源 OFF のエミュレーション中でのブレークもこれに該当します。
- 備考1. マルチコア対応版を対象とした "プログラムの実行制御", または "イベントの発生" については, 「2.7 コア(PE)の選択」も参照してください。
 - **2.** 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。
- なお、CubeSuite+では、次のブレーク機能を使用して任意の箇所でプログラムを停止させることができます。

(1) 強制ブレーク機能

強制的にプログラムの実行を停止する機能です。

(2) ハードウェア・ブレーク機能

デバッグ・ツールが、ハードウエアの資源を使用して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る機能です。

ハードウェア・ブレーク・イベントを設定すると、指定したアドレスの命令実行前にプログラムがブレーク します(実行前ブレーク)。

- 備考 ハードウェア・ブレーク・イベント(アクセス系)を使用する場合では(「(1) ブレーク・イベント (アクセス系)を設定する」参照),次の場合のみ"実行後ブレーク"となります。 - コンテキスト・メニューの[ブレークの設定]→[読み込みブレークを設定]/[読み書きブレー
 - クを設定]の選択によるブレーク・イベントの設定において、データ条件を設定した場合

- コンテキスト・メニューの[ブレークの設定]→[書き込みブレークを設定]/[読み書きブレー クを設定]の選択によるブレーク・イベントの設定において、リードモディファイライト系の命令 のライト・アクセスを検出した場合
- (3) ソフトウエア・ブレーク機能【Full-spec emulator】【E1】【E20】

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 ムを停止させる機能です。

ソフトウエア・ブレーク・イベントを設定すると、指定したアドレスの命令実行前にプログラムがブレークし ます(実行前ブレーク)。

注意 命令コードをブレーク用の命令に書き換えるため ソフトウエア・ブレーク・イベントの設定/削除 を行うたびに、次のタイミングでフラッシュ・メモリの書き換えが行われます。 - プログラムの実行開始時([デバッグ] メニュー→ [ブレークせずに実行]の選択を含む) - デバッグ・ツールと切断時

2.9.1 ブレーク動作の設定をする【Full-spec emulator】【E1】【E20】

ブレーク機能を使用するためには、あらかじめブレーク動作に関する設定を行う必要があります。 ブレーク動作の設定は、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]カテゴリ内で行い ます。

備考 【シミュレータ】

ブレーク動作の設定は必要ありません。

図 2---74 [ブレーク] カテゴリ

- - **- - - -**

フレーク	
ソフトウエア・ブレークを使用する	はい
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
停止時に周辺エミュレーションを停止する	いいえ

(1) [ソフトウエア・ブレークを使用する]

ソフトウエア・ブレーク機能【Full-spec emulator】【E1】【E20】を使用するか否かを選択します。 ソフトウエア・ブレーク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:「いいえ])。

- 注意1.1度ソフトウエア・ブレーク機能を使用したのち[いいえ]を選択した場合、それまで設定してい たすべてのソフトウエア・ブレーク・イベント および Printf イベントは無効状態となります。こ の場合、このプロパティを[はい]に再設定しても自動的に有効状態には戻りません(手動で設定 を行う必要があります)。
 - 2. プログラム実行中は、このプロパティを変更することはできません。

(2) [優先的に使用するブレークポイントの種類]

このプロパティは, [ソフトウエア・ブレークを使用する] 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。

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設定するブレークポイントの 種別を選択します。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。

ハードウェア・ブレーク	ハードウェア・ブレーク機能を使用した。ハードウェア・ブレークポイントを優先的
	に設定します(デフォルト)。
	設定すると、ハードウエア・ブレーク・イベント(実行系)として扱われます。
ソフトウエア・ブレ ー ク	ソフトウエア・ブレーク機能【Full-spec emulator】【E1】【E20】を使用した。 ソフト
	ウエア・ブレークポイントを優先的に設定します。
	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。

- 注意 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(「(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。
- (3) [停止時に周辺エミュレーションを停止する]

ブレーク時に、エミュレータの周辺エミュレーション機能を停止(Peripheral Break)するか否かを選択します。 停止する場合は[はい]を選択してください(デフォルト:[いいえ])。

2.9.2 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 (強制ブレーク機能)。

2.9.3 任意の場所で停止する(ブレークポイント)

ブレークポイントは、マウスのワン・クリック操作で設定することができるブレーク・イベントの1つです。 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。 ここでは、次の操作方法について説明します。

- (1) ブレークポイントを設定する
- (2) ブレークポイントを削除する
- (1) 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。

アドレス表示のあるメイン・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。[優先的に使用するブレークポイントの種 類] プロパティで選択している種別のブレークポイントが、クリックした行に対応する先頭アドレスの命令に 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(ハードウェア・ブレーク・イベント/ソフトウェア・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(「2.16 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

ブレークポイント種別	イベント種別	イベント・マーク
ハードウエア・ブレークポイント	ハードウェア・ブレーク・イベント ^注	10
ソフトウエア・ブレークポイント	ソフトウエア・ブレーク・イベント ^注	10
[Full-spec emulator] [E1] [E20]		

注 イベント パネルにおける [名前] エリアでは、イベント種別名が"ブレーク"として表示されます。



図 2—75 ブレークポイントの設定例(逆アセンブル パネルの場合)

図 2—76 イベント パネルのブレークポイントの設定例

<u> </u>		X
× 🖲 🗑 🖬 🖏 🐼		
名前	△ 訂羊和 青幸履	איגר
🗹 警 Run-Break प्रति	未計測	
🕑 🐝 ブレーク0001	実行後 CG_main.c#72 0x295	
🗹 警 無条件トレース		

- 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)、「2.16.6 イベント設定に関する留意事項」も参照してください。
 - 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。
 - 3. 【Full-spec emulator】【E1】【E20】 ソフトウエア・ブレークポイントは、コード・フラッシュ領域にのみ設定することができます。
- 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(「2.16.1 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(<u></u>)が表示されます。

2. [Full-spec emulator] [E1] [E20]

次に示す操作により, [優先的に使用するブレークポイントの種類] プロパティの選択に依存する ことなく, ハードウェア・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。

ただし、"操作方法 1"は、逆アセンブルパネルでのみ有効です。

種別	操作方法 1	操作方法 2
ハードウエア・ブレーク	[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]
ポイント		→ [ハード・ブレークを設定]を選択
ソフトウエア・ブレーク	[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]
ポイント		→ [ソフト・ブレ ー クを設定]を選択

3. 【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。

(2) 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で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)。

2.9.4 任意の場所で停止する(ブレーク・イベント)

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。 ここでは,次の操作方法について説明します。

- (1) ブレーク・イベント(実行系)を設定する
- (2) ブレーク・イベント (実行系) を削除する
- (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。

各パネルのアドレス表示のある行にキャレットを移動したのち,目的のイベント種別に従って,コンテキスト・メニューより次の操作を行います。

イベント種別	操作方法	説明
ハードウエア・ブレーク	[ブレークの設定]→[ハード・ブレーク	ハードウェア・ブレーク機能を使用してブ
	の設定]を選択	レーク・イベントを設定します。
ソフトウエア・ブレーク	[ブレークの設定]→[ソフト・ブレーク	ソフトウエア・ブレーク機能【Full-spec
[Full-spec emulator]	の設定]を選択	emulator】【E1】【E20】を使用してブレー
[E1] [E20]		ク・イベントを設定します。

ブレーク・イベント(実行系)は、キャレット位置の行に対応する先頭アドレスの命令に設定されます。

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, イベント パネルにおいて, ハードウェア・ブレーク・イベント (実行系) /ソフトウェア・ブレー ク・イベント (実行系) として管理されます (「2.16 イベントの管理」参照)。

表 2--5 ブレーク・イベント(実行系)のイベント・マーク

イベント種別	イベント・マーク
ハードウエア・ブレーク	-
ソフトウエア・ブレーク	1
[Full-spec emulator] [E1] [E20]	

図 2-77 ブレーク・イベント(実行系)の設定例(逆アセンブルパネルの場合)



図 2-78 イベント パネルのハードウェア・ブレーク・イベント(実行系)の設定例

イベント		8
	🖓 되 🗏 🖏 🐺	
名前	✓ ■詳希田情報報	
🗹 警 Run-Breakタイ፣	マ 未計測	
🔽 % ハードウエア・ブレ	ノーク0001 実行後 CG_main.c#72 0x295	
🗹 🚏 無条件トレース		

- 注意 1. 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),「2.16.6 イベン ト設定に関する留意事項」も参照してください。
 - [Full-spec emulator] [E1] [E20]
 ソフトウエア・ブレーク・イベントは、コード・フラッシュ領域にのみ設定することができます。
- 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(「2.16.1 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

(2) 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、エディタ パネル/逆アセンブル パネルにおいて、 表示されているイベント・マークをクリックします。

また、イベント パネルにおいて、対象となるソフトウエア・ブレーク・イベント/ハードウエア・ブレー ク・イベントを選択したのち、ツールバーのX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. 16.4 イベントを削除する」参照)。

2.9.5 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。

アクセス種別	説明
リード	指定した変数 /l/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 /l/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 /l/O レジスタに、リード・アクセス/ライト・アクセスした(読み書きを 行った)際に実行中のプログラムを停止します。

表 2---6 変数へのアクセス種別

注意 DMA (Direct Memory Access) によるアクセスは対象となりません。

ここでは、次の操作方法について説明します。

- (1) ブレーク・イベント (アクセス系) を設定する
- (2) ブレーク・イベント(アクセス系)を削除する
- (1) 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・イベントの設定は, 次のいずれかの操作により行います。

- 注意 ブレーク・イベントの設定に関しては(有効イベント数の制限など),「2.16.6 イベント設定に関す る留意事項」も参照してください。
- (a) エディタパネル/逆アセンブルパネル上の変数 //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,目 的のアクセス種別に従って,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。

アクセス種別	操作方法
リード	[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
ライト	[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
リード/ライト	[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

なお、この際に、コンテキスト・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/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/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。

- 2. ブレーク・イベントは、アドレス表示がない行上の変数 *IVO* レジスタを選択しても設定することはできません。
- 図 2---79 エディタ パネル上の変数に対するハードウエア・ブレーク・イベント(アクセス系)の





(b) 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),目的のアクセス種別に従って,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 /l/O レジスタのみとなります。

アクセス種別	操作方法		
リード	[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		
ライト	[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		
リード/ライト	[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		

なお、この際に、コンテキスト・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/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 ているウォッチ式に読み込み/書き込みを行った場合にブレークします。

注意 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 2-80 ウォッチ式に対するブレーク・イベントの設定例

<u> </u>		
🗟 🌒 🛃 📆 🗙	表記(№) ▼ 100	
ウォッチ式	値 型情報(バイト数) アドレス	
r3:REG	10・・・ 10・・・ 10・・・ アクセス・ブレークの設定 ● 読み込みブレークを設定(R) この出力(T) ● 書き込みブレークを設定(W) 0xb リアルタイム> 更新設定(R) ● ● 読み書きブレークを設定(A)	>
2 2 2	最新の情報に更新(A) ウォッチ式 "global_a" 上のコンテキスト・メニュー 値を強制読み込み(F) より、[アクセス・ブレークの設定] → [書き込みブ 新規ウォッチ式を追加(M) レークを設定]内に値を入力したのち、[Enter] キー	-
× ×	カテゴリを作成(C) を押下します。 削除(D) ここでは、ウォッチ式 "global_a" に "0xb" が書き 切り取り(T) 込まれた場合にブレークする設定をしています。	
	JE-(<u>C)</u>	

以上の操作を行うことにより、ブレーク・イベント(アクセス系)が設定されると、イベントパネルにおい て、ハードウェア・ブレーク・イベント(アクセス系)として管理されます(「2.16 イベントの管理」参照)。

図 2-81 イベント パネルのハードウエア・ブレーク・イベント(アクセス系)の設定例

1421		2
名前 🛛 🕹	言羊糸田「青幸履	コメント
🗹 警 Run-Breakタイマ	未計測	
🗹 懸 ハードウエア・ブレーク0001	ライト global_a Oxfefba - Oxfefbb == Oxb	
🗹 🚏 無条件トレース		

(2) 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、イベントパネルにおいて、対象となるハードウェア・ブレーク・イベントを選択したのち、ツールバーのXボタンをクリックします(「2.16 イベントの管理」参照)。

2.9.6 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のステータス・メッセージ で確認することができます。

安区	Full-spec emulator	E1/E20	シミュレータ
トレース・メモリを使い切った ^{注1}	0	0	0
ノン・マップ領域へのアクセス	_	_	0
書き込み禁止領域への書き込み	—	_	0
テンポラリ・ブレークの発生 ^{注 2}	0	0	0
ステップ実行回数オーバ	0	0	0

表 2---7 その他のブレーク要因

- **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
 - 2. CubeSuite+内部でのみ使用するブレーク(ユーザは使用不可)



2.10 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。

2.10.1 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。

メモリ1											E
2 🤫	表記()	Ð-	サイズ表	記(乙) -	-בעד	+≍(<u>C</u>) +	表示	Ø• -	ツールバー		
🔽 停正時	に移動				∠表	示位置指	旨定エリ	7		移動	Ī
00000000 00000020 00000020 00000040 00000040 00000050 00000050 00000050 00000050 000000	+0 +1 80 07 E5 57 0A 00 00 00 01 00 03 00 A6 FF 61 62 FF FF FF FF FF FF	+2 +3 06 00 40 00 85 05 02 2F 02 2F 40 16 63 00 FF FF FF FF FF FF	+4 +5 0 7F 00 0 0B 51 5 80 07 6 00 00 6 02 00 6 C0 FE 6 62 00 0 FF FF FF FF FF FF FF FF	+6 +7 +1 23 06 0 EA 2F 2 00 00 8 43 2F 0 43 2F 0 43 2F 0 43 17 0 43 17 0 FF FF F FF FF F	+9 +a 4 82 BF 0 00 80 2 07 21 0 00 02 0 00 02 0 00 02 1 80 22 1 80 22 1 80 22 1 80 22 1 80 42 F FF FF F FF FF	+b +c FE 40 FF 54 00 22 2F 01 17 03 36 2C 06 3F FF FF FF FF	+d +e 5E 01 00 80 00 70 00 43 00 43 01 BF 00 7F FF FF FF FF	+f 00 FF 00 2F 17 FF 00 FF FF FF FF	ASCII ?	???@^ .??T.?? C/ C/ C? ?~6,.?? ?? ?? ????????????????????	10.00
ー ドレス・エ	リア			メモリ値	エリア				文字列エ	」 リア	

図 2-82 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(開始アドレス/終了アドレス)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。

- (1) 表示位置を指定する
- (2) 値の表示形式を変更する
- (3) メモリの内容を変更する
- (4) プログラム実行中にメモリの内容を表示/変更する
- (5) メモリの内容を検索する
- (6) メモリの内容を一括して変更(初期化)する
- (7) メモリの表示内容を保存する

備考 マルチコア対応版を対象とした "メモリ/レジスタ/変数などの情報"については、「2.7 コア(PE)の選 択」も参照してください。

(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---83 表示位置指定エリア(メモリ パネル)



(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(「2.18.2 シンボル名の入力補完機能」参照)。
 - **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) 値の表示形式を変更する

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	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。
Hex	メモリ値を 16 進数で表示します(デフォルト)。
SDec	メモリ値を符号付き 10 進数で表示します。
U Dec	メモリ値を符号なし 10 進数で表示します。
Oct	メモリ値を8進数で表示します。
Bin	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	メモリ値を4ビット幅で表示します。
8	メモリ値を8ビット幅で表示します(デフォルト)。
16	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。
32	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。
64	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。
	文字列を ASCII コードで表示します(デフォルト)。
	文字列を Shift_JIS コードで表示します。
EUC	文字列を EUC-JP コードで表示します。
	文字列を UTF-8 コードで表示します。
	文字列をUTF-16コードで表示します。
Flo	文字列を単精度浮動小数点数値注で表示します。
Dbl	文字列を倍精度浮動小数点数値 ^注 で表示します。
Fig	文字列を単精度浮動小数点数の複素数 ^注 で表示します。
Del	文字列を倍精度浮動小数点数の複素数 ^注 で表示します。
Fig	文字列を単精度浮動小数点数の虚数 ^注 で表示します。
Dbl	文字列を倍精度浮動小数点数の虚数 ^注 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 ダイアログをオープンしま
	す。
表示アドレス・オフ	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
セット値を設定 …	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。

(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、 「(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。

- 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。
- 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。
- 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。
- (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。

なお、リアルタイム表示更新機能は、CPU/デバッグ・ツールが持つ RRM 機能(読み込み)【シミュレー タ】、RAM モニタ機能(読み込み)【Full-spec emulator】【E1】【E20】、および DMM 機能(書き込み)により 実現され、それぞれの機能による読み込み/書き込みが可能な対象領域は異なります。

リアルタイム表示更新機能を有効にするために、プロパティパネルの[デバッグ・ツール設定] タブ上にお いて、次の基本設定を行ってください。

カテゴリ	プロパティ	設定値
[実行中のメモリ・アクセス]	[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]

表 2---8 リアルタイム表示更新機能の基本設定

注意 1. 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。

- RRM 機能や RAM モニタ機能で値を読み出す変数のサイズが複数バイト(2バイト/4バイト/8バイト)の場合、変数へ値を代入する処理が2回に分けて行われる場合があります。
 この2回の代入処理の間で変数の読み出しが行われると、変数へ値が代入される途中の値が読み出され、実際には代入していない値が表示されることがあるため注意が必要です。
- 3. 選択しているマイクロコントローラがマルチコア対応版の場合では、全 PE のアクセスを対象に読み込み可能です。

ただし、Local RAM self 領域は、現在選択している PEn のみが対象となります。

- 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、「(3) メモリの内容を 変更する」/「(6) ウォッチ式の内容を変更する」を参照してください。
- (a)RRM 機能(読み込み)【シミュレータ】

プログラム実行中に、リアルタイムにメモリ/ウォッチ式の内容を読み込む機能です。 この領域に割り当てられているメモリ/ウォッチ式は、常にリアルタイムな表示が可能です。 RRM 機能による読み込みが可能な領域は次のとおりです。

対象領域	シミュレータ
内蔵 ROM	0
内蔵 RAM	0
周辺 I/O 領域	_
データフラッシュ	_
エミュレーション・メモリ	_
ターゲット・メモリ	_
CPU レジスタ	O ^注
I/O レジスタ	0
(読み込み保護対象 IOR を除く)	

表 2—9 RRM 機能の対象領域

注 トレーサ/タイマ動作中は不可

(b) RAM モニタ機能(読み込み) 【Full-spec emulator】 【E1】 【E20】

CPUのRAMモニタ機能を使用してメモリ/ウォッチ式の内容を読み込む機能です。 RAMモニタ機能による読み込みが可能な領域は次のとおりです。

注意 CPU ステータスがスタンバイ・モード(HALT/STOP/IDLE)に移行すると、タイムアウト・エ ラーを発生します。

対象領域	Full-spec emulator	E1/E20
内部 ROM	_	_
内部 RAM	0	0
周辺 I/O 領域	_	_
データフラッシュ	_	—
ターゲット・メモリ	_	_
CPU レジスタ	—	_
I/O レジスタ	—	_

表 2—10 RAM モニタ機能の対象領域

ただし, RAM モニタ機能を有効にするためには, リアルタイム表示更新機能の基本設定に加え, 次の設 定が必要となります。

デバッグ・ツール	カテゴリ	プロパティ	設定値
Full-spec emulator	[実行中のメモリ・アクセス]	[実行中にアクセスする]	[はい]
E1/E20			

(c) DMM 機能(書き込み)

プログラム実行中に、リアルタイムにメモリ/ウォッチ式に値を書き込む機能です。 DMM 機能による書き込みが可能な領域は次のとおりです。

- 注意 1. DMM 機能を使用して書き込みを行った場合のアトミック性の保証はありません。
 - 2. CPU ステータスがスタンバイ・モード(HALT/STOP/IDLE)に移行すると、タイムアウト・エ ラーを発生します。

対象領域	Full-spec emulator	E1/E20	シミュレータ
内部 ROM	—	_	0
内部 RAM	0	0	0
周辺 I/O 領域	—	_	_
エミュレーション・メモリ	—	_	_
ターゲット・メモリ	_		
CPU レジスタ	_	_	O ^注
I/O レジスタ	—	_	0
(読み込み保護対象 IOR を除く)			

表 2—11 DMM 機能の対象領域

注 トレーサ/タイマ動作中は不可

ただし, DMM 機能を有効にするためには、リアルタイム表示更新機能の基本設定に加え、次の設定が 必要となります。

デバッグ・ツール	カテゴリ	プロパティ	設定値	
Full-spec emulator E1/E20	[実行中のメモリ・アクセス]	[実行中にアクセスする]	[はい]	
シミュレータ	設定不要	<u> </u>		

なお、リアルタイム表示更新機能を行っているメモリ値/ウォッチ式は、メモリ パネル/ウォッチ パネルに おいてピンク色に強調表示されます。

図 2-84 リアルタイム表示更新を行っているメモリ表示の例(メモリパネル)

メモリ1		×
	表記(№ + サイズ表記(2) + Iンコード(<u>C</u>) + 表示(<u>0</u> +	
[] 停正時	[[78章]]	移動
	+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +a +b +c +d +e +f	ASCII
00000090	30 01 38 01 F0 00 F8 00 00 01 08 01 10 01 C8 00	0.8.?.??.
000000a0	D0 00 D8 00 E0 00 E8 00 A0 00 A8 00 B0 00 B8 00	7.7.7.7.7.7.7.7.
00000060	CO 00 78 00 80 00 88 00 90 00 98 00 CC 01 D0 01	?.x.?.?.?.?.?.?.
000000c0	D4 01 D8 01 DC 01 B8 01 BC 01 C0 01 C4 01 C8 01	?.?.?.?.?.?.?.?.
000000000	A4 01 A8 01 AC 01 B0 01 B4 01 90 01 94 01 98 01	?.?.?.?.?.?.?.?.
000000e0	9C 01 A0 01 7C 01 80 01 84 01 88 01 8C 01 68 01	?.?. .?.?.?.?.h.
000000f0	6C 01 70 01 74 01 78 01 80 07 E1 70 E0 07 44 01	l.p.t.x.?.?p?.D.
00000100	82 07 E1 70 E0 07 44 01 84 07 E1 70 E0 07 44 01	?.?p?.D.?.?p?.D.
00000110	86 07 E1 70 E0 07 44 01 88 07 E1 70 E0 07 44 01	?.?p?.D.?.?p?.D.
00000120	80 07 E1 30 E0 07 44 01 82 07 E1 30 E0 07 44 01	?.?0?.D.?.?0?.D. 💌

(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—85 メモリ内谷の検索(メモリ検索 タイアロク)	义 2—8	5 メモ	リ内容の検索	(メモリ	検索	ダイ	アログ)
------------------------------	-------	------	--------	------	----	----	------

検索するデータ(D):		*
検索する範囲(<u>R</u>):	アドレス範囲を指定する	
アドレス(<u>A</u>):	0x0 🗸 -	0xffffffff

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。

2. 浮動小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。

(a) [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。

(b) [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。
	表示形式:< <i>メモリ種別</i> > < <i>アドレス範囲</i> > < <i>サイズ</i> >

(c) [アドレス] の指定

この項目は、「(b) [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(0x03FFFFF)です(ミラー領域は検索対象となりません)。

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(「2.18.2 シンボル名の入力補完機能」参照)。
 - 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。
 - **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。
- (d) [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。 (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 2-86 メモリ内容の一括変更(メモリ初期化 ダイアログ)

· · · ·	(タフアドレフ / パンボルを入力 回い
	MG1710077270787071
初期化データの:	
■■ ②期化データを16進数で入力して	(ださい。スペース区切りで複数指 🕞 🗸

(a) [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(「2.18.2 シンボル名の入力補完機能」参照)。
- (b) [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,初期化対象のアドレス範囲のエンディアンのバイト列に変換 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	書き込みイメージ(バイト単位)	
(初期化データ)	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



入力文字列	書き込みイメージ(バイト単位)	
(初期化データ)	リトル・エンディアン	ビッグ・エンディアン
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

(c) [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。

(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。

ノークルAFF クロ ファイル名(N):	xey1
ファイルの種類(I): - 保存範囲 アドレン	テキスト・ファイル (*.txt)
	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓
000000000	

(a) [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。

(b) [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。

(c) [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(「2.18.2 シンボル名の入力補完機能」参照)。
- (d) [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 2-88 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)】

00000010 11 11 11 11 11 11 11 11 11 11 11 11 1

【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)】

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。

2.10.2 CPU レジスタを表示/変更する

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 2-89 CPU レジスタの内容の表示(CPU レジスタ パネル)

CPUレジスタ		X
◎ 表記(№) - 198 - ツールバー		
レジスタ名	値	^
□ 🚽 ブログラム・レジスタ		
□1 汎用レジスタ	0	
■ 🗂 整数		
🗉 🗇 プログラム・カウンタ		
🖙 🖯 システム・レジスタ		
⊟ ゚゚゚゚゚ グループ0		
🗉 🖃 EIPC	0×0000000	
🖃 🚍 EIPS₩	0×0000020	
🕀 📑 FEPC	0×0000000	
🕀 📑 FEP SW	0×0000020	
🕀 📑 PSW	0×0000020	
EIIC	0×0000000	~

ここでは、次の操作方法について説明します。

- (1) 値の表示形式を変更する
- (2) CPU レジスタの内容を変更する
- (3) プログラム実行中に CPU レジスタの内容を表示/変更する
- (4) CPU レジスタの表示内容を保存する

(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。

疌	記	値の表示形式を変更する次のボタンを表示します。	
	auto	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)。	
	Hex	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。	
	SDec	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。	
	Dec	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。	
	Oct	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。	
	Bin	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。	
	Asc	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。	
選択している項目を Float で表示します。		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。	
	Dbl	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。	
1	彩	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) 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。

(3) プログラム実行中に CPU レジスタの内容を表示/変更する

対象となる CPU 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 CPU 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、「2.10.6 ウォッチ式を表示/変更する」を参照してください。

(4) 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—90 CPU レジスタ保存の際の出力イメージ

レジスタ名	値	
カテ <i>ゴリ名</i> - <i>レジス</i> タ名 :	<i>值</i> :	

2.10.3 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---91 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。

- (1) I/O レジスタを検索する
- (2) I/O レジスタを整理する
- (3) 値の表示形式を変更する
- (4) I/O レジスタの内容を変更する
- (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
- (6) I/O レジスタの表示内容を保存する

(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(最大履歴数:10個)。

次のいずれかのボタンをクリックします。

۶	テキスト・ボックスで指定している文字列を含む I/O レジス名を上方向に検索し、 態にします。	検索結果を選択状
>	テキスト・ボックスで指定している文字列を含む I/O レジス名を下方向に検索し 態にします。	検索結果を選択状

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)。

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) 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。
 - 2. I/O レジスタの追加/削除はできません。
- (a) 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。

(b) 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。

(c) 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 x 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。

(d) 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。

(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。
	Hex	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)。
	SDec	選択している項目の値を符号付き 10 進数で表示します。
	UDec	選択している項目の値を符号なし 10 進数で表示します。
	Oct	選択している項目の値を8進数で表示します。
	Bin	選択している項目の値を2進数で表示します。
	Asc	選択している項目の値を ASCII コードで表示します。
Hest		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。
(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。
 - 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。
 - 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。
 - 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 "OSTMnXX"の値に "0x41"を書き込んだ場合
 - → OSTM*nXX* に, "0x41"が書き込まれます。
 - I/O レジスタ名 "OSTMnXX" の値に ASCII 文字 "'A'"を書き込んだ場合
 - → OSTM*nXX* に, "0x41"が書き込まれます。

(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、「2.10.6 ウォッチ式を表示/変更する」を参照してください。

(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 2-92 I/O レジスタ保存の際の出力イメージ

IOR 名	值	型情報(バイト数)	アドレス
<i>カテゴリ名</i> -IOR <i>名</i> :	<i>値</i> :	<i>型情報(バイト数)</i> :	アドレス :



2.10.4 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,「2.10.6 ウォッチ式を表示/変更する」を参照してください。

2.10.5 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 2-93 ローカル変数の内容の表示(ローカル変数 パネル)

ローカル変数			8
2 表記(N) ▼ 1	בעב <u>(0</u>)	· < <u>9= nn=</u>	
カレント		^	
名前	値	型情報(バイト数)	アドレス
👽 result	290899	int (4)	R11:REG
😜 local_a	58182	int (4)	R26:REG
💊 local_b	58182	int (4)	R27:REG
💊 local_c	58182	int(4)	R28:REG
0 I	58179	unsigned long(4)	R29:REG
	П		П
」 「名前]エリア 「·	ー」 値]エリア 「 ³	 型情報(バイト数)] エ!	

ここでは、次の操作方法について説明します。

(1) 値の表示形式を変更する

- (2) ローカル変数の内容を変更する
- (3) ローカル変数の表示内容を保存する

(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。

値の表示形式を変更する次のボタンを表示します。

表記



auto	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)。
Hex	このパネル上の値を 16 進数で表示します。
Dec	このパネル上の値を 10 進数で表示します。
Oct	このパネル上の値を8進数で表示します。
Bin	このパネル上の値を2進数で表示します。
[Dec]	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)。
(Hex)	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。
Flo	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。
Dbl	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。
Heg	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。
Asc	文字列変数を ASCII コードで表示します(デフォルト)。
	文字列変数を Shift_JIS コードで表示します。
EUC	文字列変数を EUC-JP コードで表示します。
UTF	文字列変数を UTF-8 コードで表示します。
UTF	文字列変数を UTF-16 コードで表示します。

(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。

- 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。
- 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。
- 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

- 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

(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。

スコープ: <i>現在のスコ</i> [V] 変数 [P] 引数 名前	<i>コープ</i> [F]関数 値	型情報(バイト数)	アドレス	
[V] <i>変数名</i> [1]	<i>値</i>	<u>型情報(バイト数)</u>	アドレス	
- [V] <i>変数名</i> [0]	<i>値</i>	型情報(バイト数)	アドレス	
:	:	:	:	

2.10.6 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(「(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。



図 2—94 ローカル変数保存の際の出力イメージ

図 2---95 ウォッチ式の内容の表示(ウォッチ パネル)

ウォ ッ チ1	(and the second s	ツールバー	
	★ 表記(N) ★ 📑		
ウォッチ式	値	型情報(バイト数)	アドレス メモ
🖃 🧺 Reg_Set			
	0×0000000	システム・レジスタ(4)	an a
😜 у	80	int (4)	0xfebf8000
💊 global_a	58180 (0x0000e344)	int (4)	0×03ff010c
[ウォッチ式] エリア	[値] エリア	[型情報(バイト数)] エリア	[アドレス] エリア [メモ] エ

- ここでは、次の操作方法について説明します。
 - (1) ウォッチ式を登録する
 - (2) 登録したウォッチ式を整理する
 - (3) 登録したウォッチ式を編集する
 - (4) ウォッチ式を削除する
 - (5) 値の表示形式を変更する
 - (6) ウォッチ式の内容を変更する
 - (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
 - (8) ウォッチ式をエクスポート/インポートする
 - (9) ウォッチ式の表示内容を保存する
- (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 3000 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)。
 - 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。
 - 2. ウォッチ式は、同名を複数登録することができます。
- (a) 他のパネルから登録する場合

CubeSuite+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。

道アセンブル1		
2 11 🔁 1	ఓ 🐉 表示@▼	ここでは、逆アセンブル パネル上の変数
	_recursive_call01:	"global_val01++"をウォッチ式として登録します。
00000314	d515 br 63370100 st.w	逆アセンブル パネル上の変数 "global_val01++"を選
0000031a	28870100 Id.w	択したのち、ウォッチ パネル上へ直接ドラッグ・アン
0000031e 00000322	100al Valu ++; ウォッチ1	ド・ドロップします。
00000324	2 ● 2 1 本 表記(1)・	19
<u> </u>	ウォッチ式	値 型情報(バイト数) アドレス
	🥥 global_a 🔪 10 (0x000	10000a) int(4) 0x3ff710c
	ka ka	
	0.000	
	<	

図 2-96 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)。

(b) 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—97 ウォッチ式のエントリ・ボックス

ウォッチ式	Δ.	値	型情報(バイ	ト数)	アドレス
📦 arg_a	1003761	(0×000f50f1)	int(4)		0×00006
🐳 arg_c	1003761	(0×000f50f1)	int(4)		0×00008
	\geq	?	?		?
Ę	のエリアに直接	妾ウォッチ式を入	カします。		

エントリ・ボックス内に、キーボート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の詳細は、次を参照してください。

- 「表 2---27 ウォッチ式の基本入力形式」
- 「表 A—3 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」
- 「表 A—5 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—6 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(「2.18.2 シンボル名の入力補完機能」参照)。
- (c) 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) 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。
 - 1 つのウォッチ パネルにおいて、カテゴリは 1500 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)。
- (a) 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 新規カテゴリ名を直接入力します。

(b) 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。

(c) 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。

(d) 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。
- (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。

(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 X ボタンをクリックします。

(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。
auto	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—7 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)。
Hex	選択している項目の値を16進数で表示します。
SDec	選択している項目の値を符号付き 10 進数で表示します。
Dec	選択している項目の値を符号なし 10 進数で表示します。
Oct	選択している項目の値を8進数で表示します。
Bin	選択している項目の値を2進数で表示します。
Asc	選択している項目の値を ASCII コードで表示します。
FIO	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。
Dbl	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。
Hex	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。

(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,「(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。

- 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。
- **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。
- 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
- (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、「(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。

(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。

(a) 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(*.csv)を指定します。

[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—98 ウォッチ式のエクスポート

名前を付けて保存					20
保存する場所型:	🚞 sample		~	3 🕫	⊳ 🖽
していたつたつアイル	C DefaultBuild				
ごう デスクトップ					
אנעב¥א אד					
र्ग २७४७-७					
(ファイル名(<u>N</u>):	Watch_Export.csv			保存(S)
	ファイルの種類(工):	インポート可能CSV(カンマ	区切り)(*.csv)		キャンセル

(b) ウォッチ式をインポートする

(a) 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。

ウォッチ式データ・フ	ァイルを聞く	? 🛛
ファイルの場所①:	🗁 sample 🕑 🗿 🕫 📰	
していたつアイル	C DefaultBuild C src Watch_Export.csv	
デスクトップ		
کر کر انجاع کا کر		
ער דאר דא 😨		
र्ग रूग-७		
	ファイルの種類(T): インポート可能CSV(カンマ区切り) (*.csv) 開いの	9 94

図 2---99 ウォッチ式のインポート



(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。

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)が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。

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 2-100 ウォッチ・データ保存の際の出力イメージ

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	ウォッチ式	值	型情報(バイト数)	アドレス	メモ
- カナコウ石 変数式 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
	-)) コッ名 変数式	值	型情報(バイト数)	アドレス	×ŧ

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。



2.11 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。

CubeSuite+が提供するコンパイラ(CC-RH)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。

備考 マルチコア対応版を対象とした "コール・スタック情報"については、「2.7 コア(PE)の選択」も参照し てください。

2.11.1 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 2-101 コール・スタック情報の表示(コール・スタック パネル)

コール・	<u></u>	×
2	表記(N)・ Iンコード(E)・ 😽 👯 🦂 🧹 ツールバー	
深さ	コール・スタック	
0	main0[sample.abs\$main.c#71]	
<		>
深さ] エ	ニリア [コール・スタック] エリア	

ここでは、次の操作方法について説明します。

- (1) 値の表示形式を変更する
- (2) ソース行へジャンプする
- (3) ローカル変数を表示する
- (4) コール・スタック情報の表示内容を保存する



(1) 値の表示形式を変更する

このパネルの表示形式は、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。
	at	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)。
	Hex	このパネル上の値を 16 進数で表示します。
	B	このパネル上の値を 10 進数で表示します。
	ot	このパネル上の値を8進数で表示します。
	Bin	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。
	Asc	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)。
		このパネル上の文字列変数を Shift_JIS コードで表示します。
	3	このパネル上の文字列変数を EUC-JP コードで表示します。
	E.	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) 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、その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 [逆アセンブルヘジャンプ]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)。
- (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。

(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 2-102 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	<i>コール・スタック情報 コール・スタック情報</i> <i>コール・スタック情報</i>
:	:

2.12 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。

ー般的に、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。

備考 マルチコア対応版を対象とした"実行履歴の収集"については、「2.7 コア(PE)の選択」も参照してくだ さい。

2.12.1 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。

- (1) 【Full-spec emulator】の場合
- (2)【E1】/【E20】の場合
- (3) 【シミュレータ】の場合

(1)【Full-spec 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。

注意 プログラム実行中は、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—103	[トレース]	カテゴリ	[Full-spec emulator]
---------	--------	------	----------------------

(a) [トレース・データの選択]

収集するトレース・データの種類を次のドロップダウン・リストにより選択します。

分岐命令	プログラム実行中に発生した分岐処理の分岐元/分岐先の命令の PC 値をトレース・デー タとして収集します。
データ・アクセス	プログラム実行中に成立したアクセス系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。



分岐命令とデー	プログラム実行中に発生した分岐処理の分岐元/分岐先の命令の PC 値	および成立した
タ・アクセス	アクセス系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	(デフォルト)。

注意 このプロパティを変更すると、トレース・メモリがクリアされます。

(b) [トレースの優先度]

トレース機能を使用する際の優先度を次のドロップダウン・リストにより選択します。

スピード優先	リアルタイム性を優先してトレースを行います(デフォルト)。
データ優先	データの取りこぼしが発生しないように、CPU の実行パイプラインを一時的に停止して
	トレースを行います。

注意 このプロパティを変更すると、トレース・メモリがクリアされます。

(c) [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)するか否かを選択します。 クリアする場合は[はい]を選択してください(デフォルト)。

- 備考 トレース パネルのツールバーの ポ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。
- (d) [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・ リストにより選択します。

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] を選択している場 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。
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)。 ただし、[トレースの優先度] プロパティで [データ優先] を選択している場合は、 この項目は表示されません。
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。

注意 このプロパティを変更すると、トレース・メモリがクリアされます。



(e) [トレースの取得範囲設定]

トレース・データの取得範囲を次のドロップダウン・リストにより選択します。 ただし、このプロパティは、デバッグ・ツールと接続時のみ変更することができます。

区間をトレース	トレース開始イベントとトレース終了イベントで設定した区間の実行履歴をトレー ス・データとして収集します(デフォルト)。
範囲外をトレース	トレース開始イベントとトレース終了イベントで設定した範囲外の実行履歴をトレー

- 注意 このプロパティを変更した場合,現在設定しているトレース開始イベント/トレース終了イベン トはすべて無効となります。
- 備考 [範囲外をトレース]を選択した場合では、次のように、トレース開始イベント/トレース終了イ ベントで設定するアドレスを下限/上限とし、トレース・データの取得範囲が決定されます。



(f) [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)を選択します。

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(デフォルト:[8K])。

注意 このプロパティを変更すると、トレース・メモリがクリアされます。

(g) [トレースを補完する]

収集したトレース・データをトレース パネルで表示する際に、補完表示を行うか否かを選択します。 補完表示を行うことにより、ハードウェアではトレースできない分岐命令間の命令の表示が可能となり ます。

補完表示を行う場合は[はい]を選択してください(デフォルト)。

なお、この設定は、次回取得するトレース・データより反映されます。

(h) [トレースの取得対象設定]

トレースの対象となるコアを次のドロップダウン・リストにより選択します。

デバッグ対象コアのみ	現在デバッグ対象に選択している PEn のみを対象にトレース・データを収集します	
	(デフォルト)。	
	トレース・データ収集後、PEnを切り替えてもトレース パネルの表示内容は変わり	
	ません。	
全てのコア	全 PE を対象にトレース・データを収集します。	
	トレース・データ収集後、PEnを切り替えることにより、対応するトレース・データ	
	の内容をトレース パネルに表示します。	

(2)【E1】/【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。

- 注意 1. 接続したマイクロコントローラがトレース機能を搭載していない場合,デバッグ・ツールと接続後,このカテゴリ内のプロパティは変更不可状態となります(トレース機能を使用することはできません)。
 - 2. プログラム実行中は、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—104 [トレース] カテゴリ【E1】【E20】

ロトレース

1	<u> </u>	
	トレース・データの選択	分岐命令とデータ・アクセス
	トレースの優先度	スピード優先
	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
	トレースの取得範囲設定	区間をトレース
	トレースの取得対象設定	デバッグ対象コアのみ
	トレースの取得対象設定	デバッグ対象コアのみ

(a) [トレース・データの選択]

収集するトレース・データの種類を次のドロップダウン・リストにより選択します。

分岐命令	プログラム実行中に発生した分岐処理の分岐元/分岐先の命令の PC 値をト	
	レース・データとして収集します。	
データ・アクセス	プログラム実行中に成立したアクセス系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。	
分岐命令とデータ・アクセス	プログラム実行中に発生した分岐処理の分岐元/分岐先の命令の PC 値。およ	
	び成立したアクセス系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(デフォルト)。	

注意 このプロパティを変更すると、トレース・メモリがクリアされます。

(b) [トレースの優先度]

トレース機能を使用する際の優先度を次のドロップダウン・リストにより選択します。

スピード優先	リアルタイム性を優先してトレースを行います(デフォルト)。
データ優先	データの取りこぼしが発生しないように、CPU の実行パイプラインを一時的に停止して
	トレースを行います。

注意 このプロパティを変更すると、トレース・メモリがクリアされます。

(c) [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)するか否かを選択します。 クリアする場合は[はい]を選択してください(デフォルト)。

- 備考 トレース パネルのツールバーの ポ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。
- (d) [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・ リストにより選択します。

トレース・メモリを上	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		
書きし実行を続ける	(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] を選択している場		
	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。		
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。		

注意 このプロパティを変更すると、トレース・メモリがクリアされます。

(e) [トレースの取得範囲設定]

トレース・データの取得範囲を次のドロップダウン・リストにより選択します。 ただし、このプロパティは、デバッグ・ツールと接続時のみ変更することができます。

区間をトレース	トレース開始イベントとトレース終了イベントで設定した区間の実行履歴をトレー ス・データとして収集します(デフォルト)。
範囲外をトレース	トレース開始イベントとトレース終了イベントで設定した範囲外の実行履歴をトレー ス・データとして収集します。

注意 このプロパティの設定を変更した場合,現在設定しているトレース開始イベント/トレース終了 イベントはすべて無効となります。 備考 [範囲外をトレース]を選択した場合では、次のように、トレース開始イベント/トレース終了イ ベントで設定するアドレスを下限/上限とし、トレース・データの取得範囲が決定されます。



(f) [トレースの取得対象設定]

トレースの対象となるコアを次のドロップダウン・リストにより選択します。

デバッグ対象コアのみ	現在デバッグ対象に選択している PEn のみを対象にトレース・データを収集します		
	(デフォルト)。		
	トレース・データ収集後、PEnを切り替えてもトレース パネルの表示内容は変わり		
	ません。		
全てのコア	全 PE を対象にトレース・データを収集します。		
	トレース・データ収集後、PEnを切り替えることにより、対応するトレース・データ		
	の内容をトレース パネルに表示します。		

(3)【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—105 [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	はい
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]	4K
トレース・タイム・タグの分周率	1/1

(a) [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選択します。

トレース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:[いいえ])。

(b) [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)するか否かを選択します。 クリアする場合は[はい]を選択してください(デフォルト)。

- 備考 トレース パネルのツールバーの **い**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。
- (c) [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・ リストにより選択します。

トレース・メモリを上	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
書きし実行を続ける	(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] を選択している場
	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。
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。

(d) [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選択します。

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] を選択して ください (デフォルト)。

(e) [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・リストにより選択します。 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(デフォルト:[4K])。

(f) [トレース・タイム・タグの分周率]

トレースのタイム・タグ(トレース パネルの[時間]表示)で使用するカウンタの分周率を,ドロップ ダウン・リストにより選択します(デフォルト:[1/1])。

2.12.2 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、「2.12.6 実行履歴を表示する」を参照してく ださい。

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。

したがって、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を外し、無効状態にした場合、プログラムの実行開始に連動したトレース・データの収集は行いません(無条件トレース・イベントはデフォルトで有効状態に設定されています)。

なお、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(「2.12.3 任意区間の実行履歴を収 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。

2.12.3 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。

なお、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。

- (1) トレース・イベントを設定する
- (2) プログラムを実行する
- (3) トレース・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),「2.16.6 イベント設定に関する 留意事項」も参照してください。
 - 2.【シミュレータ】
 トレーサ動作中は、トレース開始イベント/トレース終了イベントの設定/削除はできません。
- (1) トレース・イベントを設定する

トレース・イベントを設定するため、トレース・データの収集を開始/終了するトレース開始イベント/ト レース終了イベントを設定します。

トレース開始イベント/トレース終了イベントの設定は、次いずれかの操作により行います。

(a) 実行系イベントの場合

実行系イベントをトレース開始イベント/トレース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。

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。

各パネルのアドレス表示のある行にキャレットを移動したのち、目的のイベント種別に従って、コンテ キスト・メニューより次の操作を行います。

イベント種別	操作方法
トレース開始	[トレース設定]→[トレース開始の設定]
トレース終了	[トレース設定]→[トレース終了の設定]

注意 【シミュレータ】

トレース終了イベントはトレース・データとして表示されません。

トレース・データとして表示する場合は、1行下にトレース終了イベントを設定してください。

トレース開始イベント/トレース終了イベントは、キャレット位置の行に対応する先頭アドレスの命令 に設定されます。

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 2-12 トレース開始イベント/トレース終了イベント・マーク

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	0

図 2—106 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)



備考 【Full-spec emulator】 【E1】 【E20】

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 [トレースの取 得範囲設定] プロパティを [範囲外をトレース] に指定することにより,設定した範囲外の実行 履歴をトレース・データとして取得することができます。

(b) アクセス系イベントの場合

今版では、この機能はサポートしていません。

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、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(「2.16 イベントの管理」参照)、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。 図 2—107 イベント パネルのトレース開始イベント/トレース終了イベント(実行系)の設定例

150	۱ ۲		8
×	🖲 🗑 🗑 🤞		
	名前	> 言羊糸田!青幸履	
	Y 🔮 Run-Breakタイマ	7 未計測	
E	א-ע-ג	開始/終了の総数:2	
	言羊翁田惜幸履	×	
	開始 実行後 CG_r	nain.c#79 0x65e	
	終了 実行後 CG_r	nain.c#90 0x674	
	名前	✓ 言羊糸田!青幸履	
	🛾 💕 無条件トレース	12 <u>-</u> 0	

- 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、プログラムの実行開始に連動した トレース・データの収集は行いません(設定したトレース開始イベントの条件が成立するまでト レーサは動作しません)。
 - 2. トレース終了イベントが不要な場合は、未設定でもかまいません。

トが設定されていることを示すイベント・マーク(配)が表示されます。

- 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(「2.16.1 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
- 【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) プログラムを実行する

プログラムを実行します(「2.8 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が設定されている命令が実行された際に、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、「2.12.6 実行履歴を表示する」を参照し てください。

(3) トレース・イベントを削除する

設定したトレース・イベ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において、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、イベントパネルにおいて、対象となるトレース・イベントを選択したのち、ツールバーの × ボタンをクリックする操作でも削除することができます(「2.16.4 イベントを削除する」参照)。

注意 トレース・イベント内のトレース開始イベント、またはトレース終了イベントのみを削除することは 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のいずれかのイベント・マークを削除し た場合、対応したすべてのイベント・マークが削除されます)。

2.12.4 条件を満たしたときのみの実行履歴を収集する【シミュレータ】

ある条件を満たした場合にのみプログラムの実行履歴を収集することができます。

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。

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。

- (1) ポイント・トレース・イベントを設定する
- (2) プログラムを実行する
- (3) ポイント・トレース・イベントを削除する
- (1) ポイント・トレース・イベントを設定する

ポイント・トレース・イベントの設定は、次のいずれかの操作により行います。

- 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),「2.16.6 イベン ト設定に関する留意事項」も参照してください。
 - 2. トレース動作中は、ポイント・トレース・イベントの設定/削除はできません。
 - 3. DMA によるアクセスはトレース対象外です。
- 備考 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。

(a) エディタ パネル/逆アセンブル パネル上の変数 ///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。

各パネルにおいて,対象となる変数,または I/O レジスタを選択したのち,目的のアクセス種別に従って,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。

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。		

注意 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、目的のアクセス種別に従って、コンテキスト・メニューより次の操作を行います(「2.10.6 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 /l/O レジスタのみとなります。

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。		

注意 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。

以上の操作を行うことにより、対象変数、I/O レジスタ、または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(「2.16 イベントの管理」参照)。

図 2-108 イベント パネルのポイント・トレース・イベントの設定例

1121		8
× 🖲 🗑 🖬 🖏 🖓 🖥		
名前	シスト 言羊糸田 情幸服	איאר
🔽 警 Run-Break १४/ 🖓	未計測	
🖻 🗹 🌮 ポイント・トレース		
言羊命四情幸履		
ライト g_count Oxfefba		
名前	✓ 言羊糸田惜幸服	אעאב 🛛
🔲 💕 無条件トレース		-

(2) プログラムを実行する

プログラムを実行します(「2.8 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,「2.12.6 実行履歴を表示する」を参照してください。

図 2-109 ポイント・トレース・イベントの結果表示例

10-	λ				
2	🕅 🛞 🛞 🕌 表記(10) - 🗊 🏦 🚮 🐴 🕤				-
番号	時間(h:min:s,ms,us,ns) 時間(クロック) 行番号/アドレス	- ソース/逆アセンブル	アドレス	データ	~
4087	00h00min00s000ms000µs015ns 5	//////	0xfedf800c	R 4	
4088	00h00min00cc000mc000uc015cc 5	(((sc_per-////	0xfedf800c	R 7	
4089	200h00min0 200 gc_pe1 へのリード・アクセス	<< <gc_pel=1u>>></gc_pel=1u>	0xfedf800c	R 10	
4090 4090	でのポイント・トレース結果の例 00h00min0	<< <gc_pe1=13>>></gc_pe1=13>	0xfedf800c	R 13	
4091	0050001-000-0000-0015	<< <gc_pe1=16>>>></gc_pe1=16>	0vfodf900o	D 10	
4031	001000000000000000000000000000000000000	<< <gc_pe1=19>>></gc_pe1=19>	UXIEd1000C	n 10	
4092 4093	ՍՍԻՍՍminUՍՏՍՍՍmՏՍՍՍµՏՍ15ns 5	<< <gc_pe1=22>>></gc_pe1=22>	Uxfedf800c	R 19	~

(3) ポイント・トレース・イベントを削除する

設定したポイント・トレース・イベントを削除するには、イベントパネルにおいて、対象となるポイント・ トレース・イベントを選択したのち、ツールバーのXボタンをクリックします(「2.16 イベントの管理」 参照)。

2.12.5 実行履歴の収集を停止/再開する

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。

- (1) 実行履歴の収集を一時的に停止する
- (2) 実行履歴の収集を再開する
- (1) 実行履歴の収集を一時的に停止する

プログラムの実行を停止することなく、トレーサの動作のみを停止するには、トレースパネルのツールバーの () ボタンをクリックします。

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) 実行履歴の収集を再開する

プログラム実行中にトレース機能を停止したのち,再度トレース・データの収集を開始するには,トレース パネルのツールバーの ポタンをクリックします。

なお、再開前に収集したトレース・データは一度クリアされます。

2.12.6 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、デフォルトでソース・テキストと逆アセンブル・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—110 トレース・データの表示(トレース パネル【Full-spec emulator】【E1】【E20】)

番号	時間 (h:min:s,ms,µs,ns)	時間(クロック)	行番号/アドレス	リソース/逆ア	⁷ センブル	アドレス	データ	1
18	00h00min00s000ms000µs500ns	5				0×01000438	R 1074339512	
9	00h00min00s000ms000µs100ns	1	0×010003f6	+118	st.w r14, 0x0[r12]			-
0	00h00min00s000ms000µs000ns	0				0xfedf8008	W 1074339512	
1			0×010003fa	+122	add 0x4, r10			-
1-1			0x010003fc	+124	add 0x4, r12			
1-2			0x010003fe	+126	cmp r10, r11			
1-3			0×01000400	+128	bh INITSCT RH+0x6e			
1-4			0×01000402	+130	addi -0x2, r11, r13			
1-5			0×01000406	+134	cmp r10, r13			
1-6			0×01000408	+136	bc INITSCT RH+0×98			
1-7	00h00min00s000ms000µs900ns	9	0×01000418	+152	cmp r10, r11			
2			0×0100041a	+154	boh INITSCT RH+0xaa			
2-1	00h00min00s000ms000us400ns	4	0×0100042a	+170	add 0xc, r6			
3			0×0100042c	+172	br INITSCT RH+0×4a			
3-1	00h00min00s000ms000us400ns	4	0x010003ca	+74	cmp r6, r7			
4			0x010003cc	+76	boh INITSCT RH+0xae			
4-1	00h00min00s000ms000us300ns	3	0x0100042e	+174	ime [le]			
				PE1:				
5			cstart.asm#84	MOV 0x1.r6				-
5	00h00min00s000ms000us400ns	4	0x0100031c	+78	mov 0x1, r6			1
1502	1	S.					11	÷

図 2—111 トレース・データの表示(トレース パネル【シミュレータ】)

トレース		20 - 21 - 21							
2	• @ @ #	🏦 🕹 表記(N)- 📋 📆 🚮	も 🦉 🚽 ッ	ールハー					
番号	ターゲット	時間 (h:min:s,ms,µs,ns)	時間(クロック)	行番号/アドレ	ス バイブ	ライン ソー	ス/逆アセンブル	アドレス	データ
0				vecttbl.asm#1	3	jr32_	_start ; RESET		
0	CPU1	00h00min00s000ms000µs000ns	0	0×01000000			jrstart		
1	CPU1	OOhOOminOOsOOOmsOOOµsOOOns	0						
2	CPU1				FI	DE			
3				boot.asm#30		stsrO	, r10, 2; get HTCFGO		
3	CPU1	00h00min00s000ms000µs084ns	27	0×01000a00		+0	stsr 0x0, r10, 0x3	2	
4	CPU1				DE	F			
5				boot.asm#31		shr16	, r10; set PEID		
5	CPU1	00h00min00s000ms000µs056ns	18	0×01000a04		+4	shr 0x10, r10		
6	CPU1				DI	EF			
7				boot.asm#33		shl2,	r10		
7	CPU1	00h00min00s000ms000µs003ns	1	0×01000a06		+6	shl 0x2, r10		
8	CPU1			1		DE			
9				boot.asm#34		mo∨#_	entry_table, r11		
9	CPU1	00h00min00s000ms000µs003ns	1	0×01000a08		+8	mov 0×1000b20, r1	1	
10	CPU1	00h00min00s000ms000µs000ns	0						
11	CPU1			and the second response of		DE			
12				boot.asm#35		addr1	1, r10		
12	CPU1	00h00min00s000ms000µs003ns	1	0x01000a0e		+14	add r11, r10		
13	CPU1					FDE			
14				boot.asm#36		Id.w0	[r10], r10		
14	CPU1	00h00min00s000ms000µs003ns	1	0×01000a10	(304)	+16	ld.w 0x0[r10], r10)	
15	CPU1				F	DE			
16	CPU1							0×01000b24	R 16779798
						11-	ドウエア・ブレークにより停」	2	
[/	ターゲット]	エリア 「時間(クロック)] エ	:リア		「ソース	ス /	7 I	1
		· · · · · · · · · · · · · · · · · · ·						I	_
【畨号]	エリア		L行番-	号 / アドレス	.」 エリア		[アド	レス」エリ	ア
		[時間 (h:m:s,ms,µs,ns)]	エリア	[[パイプライ	ン] エリア		[データ] エリア

ここでは、次の操作方法について説明します。

(1) 表示モードを変更する

- (2) 値の表示形式を変更する
- (3) 他のパネルと連動させる

(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、トレーサが動作中の場合は無効となります。

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/ブレーク要因を表示します(デフォルト)。
1	逆アセンブル	命令(逆アセンブル)/ラベル名/ポイント・トレース結果/ブレーク要因
	表示モード	を表示します。
	ソース表示	ソース・テキスト(対応するソース行)/ブレーク要因を表示します。
	モード	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。

表 2—13 トレース パネルの表示モード

図 2-112 ソース表示モードの例(トレース パネル)

Trace						
2 💱	🕐 🛞 🕌 🛔 表記(N) -	1580	1			
番号	時間(h:min:s,ms,µs,ns)	時間(クロック)	行番号/アドレス	ソース/逆アセンブル	アドレス データ	হ 🔺
51 52 54 55 56 57 59 61			mainc#208 mainc#210 mainc#210 mainc#212 mainc#213 mainc#213 mainc#175	tmp = sub02_sub01 (arg_a, arg_b, arg_c), result = tmp + global_b; result = tmp + global_b; return result; } } <no debug="" information=""> result = sub02(static_global_a, static_glob</no>		

(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、トレーサが動作中の場合は無効となります。

表記 値の表示形式を変更		値の表示形式を変更する次のボタンを表示します。
	Hex	このパネル上の値を 16 進数で表示します(デフォルト)。
	ß	このパネル上の値を 10 進数で表示します。
	Oct	このパネル上の値を8進数で表示します。
	Bin	このパネル上の値を2進数で表示します。

(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)。

2.12.7 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 **い**ボタンをクリックします。 ただし、トレーサが動作中は無効となります。

備考 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] を選択している場合は, 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。

2.12.8 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 ぶ 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。

フェッチ・アドレス(T):		v -	(範囲を指定する場合) >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
命令(<u>M</u>):		~	
アクセス・アドレス(<u>A</u>):		-	(範囲を指定する場合 🕨 🗸
アクセスの種類(S):	(指定なし)	~	
データ(<u>D</u>):	HEX	-	HE8] (範囲を指定する) 🍺 🗸
検索範囲の指定			
番号(<u>U)</u> :		~ -	~

図 2-113 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。

- (1) 命令レベルで検索する
- (2) ソース・レベルで検索する

(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。

注意 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。

図 2-114 命令レベルでのトレース・データの検索

トレース検索					×
命令レベル ソース・レイ	402				
検索条件の指定					
フェッチ・アドレス(工):		*	- (範	囲を指定する場合	
命令(<u>M</u>):		~			
アクセス・アドレス(<u>A)</u> :		v	- (軍	囲を指定する場合	
アクセスの種類(S):	(指定なし)	~			
データ(<u>D</u>):	HEX	×	HEX] (範囲を指定する	5 14 1
検索範囲の指定					
番号(<u>U</u>):		*			*
100					
	前を検索(日) 次を検索(Ð (キャンセル	ヘルプ(比)

(a) [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。

(b) [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。

(c) [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。

(d) [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。

(e) [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。

(f) [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。

(g) [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) 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。

注意 ソース・レベルで検索を行う場合は、トレースパネルを混合表示モード、またはソース表示モードで 表示している必要があります。

検索対象の指定 シソース行を指定	て実行箇所を検索(①)	
 関数名を指定し グローバル変数 	て先頭アドレスの実行箇所を検索(T) な指定してアクセス箇所を検索(G)	
検索条件の指定		
ソース行(<u>S</u>):		~
関数名(N):		~
変数名(⊻):		2
種業頁(<u>K</u>):	参照/代入	
変数値(<u>A</u>):	HEX	- HEX
検索範囲の指定		
番号(山):	~	-

図 2-115 ソース・レベルでのトレース・データの検索

(a) 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
 - 2. main.c
 - 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。

(b) 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。

(c) 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

ル上で選択状態にします。

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



2.12.9 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 2-116 実行履歴の保存(データ保存 ダイアログ)

データ保存 - トレ	-Z·データ 🔟
ファイル名(N):	C:¥Test¥sample¥トレース・データ 💽
ファイルの種類(T): 保存範囲 番号(E	テキスト・ファイル (*.txt)
4088	 4094 4094 (保存⑤) キャンセル ヘルブ(出)

(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プドロップダウン・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 2-117 トレース・データ保存の際の出力イメージ

番号 ターゲット 時間 クロック 行番号 / アドレス パイプライン ソース / 逆アセンブル アドレス データ 番号 ターゲット 時間 クロック 行番号 / アドレス パイプライン ソース / 逆アセンブル アドレス データ

備考 出力されるトレース・データの項目は、使用しているデバッグ・ツールにより異なります。


2.13 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。

備考 マルチコア対応版を対象とした"実行時間の計測"については、「2.7 コア(PE)の選択」も参照してくだ さい。

2.13.1 実行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-Break 時間)を計測する機能 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。

注意 1. ステップ実効中は正しい実行時間が表示されません。

2. 【シミュレータ】

Run-Break 時間を計測するためには、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] カテゴリ内 [タイマ機能を使用する] プロパティにおいて、[はい] が指定されている必要があります。

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。

(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)。

図 2—118 Run-Break タイマ・イベントの測定結果例(ステータスバー)

CPU リセットを行います。46 行 13 桁 編集不可 PE1 💽 ● BREAK Halt 💭 0x000009ae 🚥 RH850 ジミュレータ 🔅 6210 ms 💽 🔯 🖽

(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。



1421			
× 🖲 🗑 🖬 🖗			
名前	言羊糸田!青幸履	אלאב	<u></u>
🗹 学 無条件トレ	ノース -		
🗹 字 Run-Brea	kタイマ 総実行時間:62095	i00 ns 🔵	

2.13.2 任意区間の実行時間を計測する【シミュレータ】

タイマ計測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。

なお、タイマ計測イベントは、タイマ開始イベント/タイマ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。

- (1) タイマ計測イベントを設定する
- (2) プログラムを実行する
- (3) タイマ計測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),「2.16.6 イベント設定に関する 留意事項」も参照してください。
 - 2. タイマ機能を使用するためには、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] カ テゴリ内 [タイマ機能を使用する] プロパティにおいて、[はい] が指定されている必要があります。
- (1) タイマ計測イベントを設定する

タイマ計測イベントを設定するため、タイマ計測を開始/終了するタイマ開始イベント/タイマ終了イベントを設定します。

タイマ開始イベント/タイマ終了イベントの設定は、次のいずれかの操作により行います。

(a)実行系イベントの場合

実行系イベントをタイマ開始イベント/タイマ終了イベントに設定することにより,任意の区間の実行 時間を計測することができます。

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。

各パネルのアドレス表示のある行にキャレットを移動したのち、目的のイベント種別に従って、コンテ キスト・メニューより次の操作を行います。

イベント種別	操作方法
タイマ開始	[タイマ設定]→[実行時にタイマ開始]→[タイマ n に設定 ^注]
タイマ終了	[タイマ設定]→[実行時にタイマ終了]→[タイマ <i>n</i> に設定 ^注]



図 2—119 Run-Break タイマ・イベントの測定結果例(イベント パネル)

注 タイマ・イベントを設定するチャネル番号(n:1~8)を選択します。

注意 タイマ終了イベントは時間測定結果に含まれません。

時間測定結果に含める場合は、1行下にタイマ終了イベントを設定してください。

タイマ開始イベント/タイマ終了イベントは、キャレット位置の行に対応する先頭アドレスの命令に設 定されます。

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、設定した行のイベント・エリアに次のイベ ント・マークが表示されます。

表 2-14 タイマ開始イベント/タイマ終了のイベント・マーク

イベント種別	イベント・マーク
タイマ開始	19
タイマ終了	19

図 2—120 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)



(b) アクセス系イベントの場合

今版では、この機能はサポートしていません。

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、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(「2.16 イベントの管理」参照)、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詳細情報が表示されます。



図 2-121 イベント パネルのタイマ開始イベント/タイマ終了イベント(実行系)の設定例

2F		ole		
10 10 10	37 🚱 🖻	S & B		
名前		2 副羊部門 青菊		コメント
🔽 🙅 Run-Breal	kタイマ	未計測		
🗹 🤔 タイマ計測	1	総実行時間:	0 ns 開始/終了(の総数:0
総実行時間	パスカウント	平均実行時間	最大実行時間	最小実行時間
0 ns	0	0 ns	0 ns	0 ns
詳細情報		· · · · · · · · · · · · · · · · · · ·	•	
開始 実行 C0	G_timer_user.	c#66 0x18b		
終了 実行 C0	G_timer_user.	c#73 0x1ae		
名前		2 詳細情報		コメント
□ 🝼 無条件トレ	/ース		-	

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(「2.16.1 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(**6**)が表示されます。

(2) プログラムを実行する

プログラムを実行します(「2.8 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 2-122 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

	31 🚳 📄	RDG		
名前	1	計部時報		עאב
🧐 🗐 Run-Brea	k\$17	総実行時間:10	06660312 ns	
🕼 タイマ計測]1	総実行時間:65	6 ns 開始/終了(D総裝:2
総実行時間	パスカウント	平均実行時間	最大実行時間	最小実行時間
656 ns	1	656 ns	656 ns	656 ns
詳細情報		•	•	
開始 実行 С	G_timer_user.	c#66 0×18b		
終了 実行 C	G_timer_user.	c#73 0x1ae		
名前	Λ.	詳細情報		עאב
1.1.2.2.2.2.	and the second second			1. States



(3) タイマ計測イベントを削除する

設定したタイマ計測イベ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において、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、イベントパネルにおいて、対象となるタイマ計測イベント選択したのち、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(「2.16.4 イベントを削除する」参照)。

注意 タイマ計測イベント内のタイマ開始イベント、またはタイマ終了イベントのみを削除することはでき ません(タイマ開始イベント/タイマ終了イベントのいずれかのイベント・マークを削除した場合、 対応したすべてのイベント・マークが削除されます)。

2.13.3 測定可能時間の範囲

Run-Break タイマ・イベント(「2.13.1 実行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(「2.13.2 任意区間の実行時間を計測する【シミュレータ】」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。

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
Full-spec emulator	最小	約 60 ナノ秒	_
	最大	約4分20秒(LPD 4Pin 使用時)	
		オーバフロー検出あり	
E1/E20	最小	約 60 ナノ秒	_
	最大	約 4 分 20 秒(LPD 4Pin 使用時)	
		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		CPU クロック周波数に依存

表 2-15 測定可能時間の範囲



2.14 カバレッジの測定【シミュレータ】

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、CubeSuite+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。

CubeSuite+では、次の両領域がカバレッジ測定の対象となります。

- 内蔵 ROM 領域(アドレス 0x000000 ~ 0x0FFFFF)の 1M バイト空間(固定の測定領域)

- 上記の固定測定領域以外の任意の1Mバイト空間(「2.14.1 カバレッジ測定の設定をする」参照)

- 備考 1. 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。
 - **2.** マルチコア対応版を対象とした"カバレッジの測定"については、「2.7 コア(PE)の選択」も参照して ください。

2.14.1 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]カテゴリ内で行います。

図 2—123 [カバレッジ]カテゴリ

🗆 カバレッジ

11(17)	
カバレッジ機能を使用する	はい
カバレッジ結果を再利用する	ບບາ້
カバレッジ測定領域(IMバイト単位)	HEX 3F00000

(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選択します。

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:[いいえ])。

(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] を選択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選択します。

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:[いいえ])。

注意 内蔵 ROM 領域のみがこの機能の対象となります。

(3) [カバレッジ測定領域 (1M バイト単位)]

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] を選択した場合のみ表示されます。

コード・カバレッジ測定の対象領域を指定します。

カバレッジ測定を行う内蔵 ROM 領域(0x000000 - 0x0FFFFF)以外の任意の 1M バイト空間の開始アドレスを, 直接入力により 16 進数で指定してください(デフォルト: [100000])。

2.14.2 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

カバレッジ測定結果の表示は、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—16 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—17 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)。

 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—16 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

表 2—17 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

RENESAS

注意 マルチコア対応版を対象としている場合, Local RAM self 領域は, 現在選択している PEn (「2. 7 コア(PE)の選択」参照)のアクセスのみを対象に測定結果を表示します。

- 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。
 - 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。
 - 3. 上記の背景表示は、対象領域外の行に対しては行われません。
 - 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—124 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—125 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 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、「CubeSuite+ 統合開発環境 ユーザーズマニュアル 解析編」を参照してください。

RENESAS

(2) データ・カバレッジ率

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、「CubeSuite+ 統合開発環境 ユーザーズマニュアル 解析 編」を参照してください。



2.15 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。

2.15.1 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。

注意 1. 【Full-spec emulator】【E1】【E20】

Printf イベントはソフトウエア・ブレーク機能【Full-spec emulator】【E1】【E20】により実現されま す。そのため、Printf イベントを設定するためには、あらかじめ、プロパティ パネルの [デバッグ・ ツール設定] タブの [ブレーク] カテゴリ内 [ソフトウエア・ブレークを使用する] プロパティで [は い] を選択してくさい。

- 2. 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),「2.16.6 イベント設定に関する留意 事項」も参照してください。
- (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。 各パネルのアドレス表示のある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 [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。

このダイブログにおいて、次の採作を打ってくたさい。

図 2—126 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)

rintf イベント		
出力文字列位	◎ 入力例 サンプル:	
変数式(⊻):	入力例) aaa, bbb, ccc	
にこに変数式	を入力してください)	
アドレス(<u>A</u>):		
出力 パネル で サンプル: aaa =	の表示例) : 10, bbb = 20 ccc = 30	



(a) [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)。

(b) [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—11 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(「2.18.2 シンボル名の入力補完機能」参照)。
- (c) [アドレス] の指定

Printf イベントを設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(「2.18.2 シンボル名の入力補完機能」参照)。
- (d) [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(「2.16 イベントの管理」参照)。

(2) プログラムを実行する

プログラムを実行します(「2.8 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。

(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—47 Printf イベントの出力結果フォーマット」参照)。

RENESAS

図 2—127 Printf イベントの出力結果例



(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 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コンの特定の状態を指します。

CubeSuite+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。

1421			×
× 🖲 🗑 🗑 🔄			
名前	言羊糸町情幸履	コメント	4
🔲 💕 無条件トレース	1 		114
🔽 警 Run-Break १४/ २	総実行時間:6093875 ns		
🗉 🗹 🖉 タイマ計測1	総実行時間:1875 ns 開始/終了の総数:2		
🖻 🗹 🎎 ኑレース	開始/終了の総数:2		
言羊 糸田(青幸服			
開始 実行 main.c#1	79 0x9b0		
終了 実行 main.c#2	210 0xa26		
名前	言羊糸田惜幸履	コメント	X
🗹 🧐 ブレーク0001	実行 main.c#201 0x9fa		
└───── [【名前] エリア	L [詳細情報] エリア	」 [コメント] エ	リア

図 2-128 設定したイベントの表示(イベントパネル)

- 備考1. マルチコア対応版を対象とした"イベントの発生"については、「2.7 コア(PE)の選択」も参照して ください。
 - 2. 各種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。
 - 「2.9.3 任意の場所で停止する (ブレークポイント)」
 - 「2.9.4 任意の場所で停止する (ブレーク・イベント)」
 - 「2.9.5 変数 /I/O レジスタへのアクセスで停止する」
 - 「2.12.3 任意区間の実行履歴を収集する」
 - 「2.12.4 条件を満たしたときのみの実行履歴を収集する【シミュレータ】」
 - 「2.13.2 任意区間の実行時間を計測する【シミュレータ】」
 - 「2.15.1 printf を挿入する」

2.16.1 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 2-129 イベント名のチェック・ボックス



表 2-18 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. 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の両方の設 定が必要となります。
 - 2. Run-Break タイマ・イベントを無効状態/保留状態にすることはできません。
 - **3.** 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。
 - 4. 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。

2.16.2 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。

1	ハードウェア・ブレーク関連のイベントを表示します。
	ソフトウエア・ブレーク関連のイベントを表示します。
[Full-spec emulator] [E1] [E20]	
1 1 1	トレース関連のイベントを表示します。
3	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)を表示します。
S	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。

2.16.3 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。

2.16.4 イベントを削除する

設定したイベント,およびイベント条件を削除するには,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。

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. 実行系のブレーク・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。
 - 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、メボタンをクリックします(ビルトイン・イベントを除く)。

2.16.5 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。

2.16.6 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。

(1) 有効イベント数の制限

- (2) 実行中に設定/削除可能なイベント種別
- (3) その他の注意事項

(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。

소하는 비 태리에	デバッグ・ツール		
イベント裡別	Full-spec emulator	E1/E20	シミュレータ
ハードウェア・ブレーク	-	-	—
(実行系:実行後)			
ハードウェア・ブレーク	1	₂ 注1	64
(実行系:実行前)			
ハードウェア・ブレーク	2	注1	
(アクセス系)			
ソフトウエア・ブレ ー ク	2000(コード・	フラッシュのみ)	—
トレース	8 +	- 7	64
(トレース開始/トレース終了)			
ポイント・トレース	-	-	64
タイマ計測	-	-	₁ 注2
(タイマ開始/タイマ終了)			
アクション	10	00	₆₄ 注3
(Printf イベント)			

表 2-19 有効イベント数の制限

"x+y": "実行系イベント:x個"+ "アクセス系イベント:y個"

- 注1. ハードウェア・ブレーク(実行系:実行前)の4個はハードウェア・ブレーク(アクセス系)と兼用 (ハードウェア・ブレーク(アクセス系)を4個設定すると、ハードウェア・ブレーク(実行系:実 行前)は残り8個までしか設定できません)
 - 2. タイマ開始イベント/タイマ終了イベントもそれぞれ1つのみ設定可
 - 3. ハードウェア・ブレークと兼用
- (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。

ᄼᅅᆞᆞᆝᆧᇊᄜ		デバッグ・ツール	
イベント裡別	Full-spec emulator	E1/E20	シミュレータ
ハードウエア・ブレーク	-	-	—
(実行系:実行後)			
ハードウエア・ブレーク	Z	7	A
(実行系:実行前)			
ハードウエア・ブレーク	Z	7	A
(アクセス系)			
ソフトウエア・ブレ ー ク	×	<	—
トレース			A
(トレース開始/トレース終了)			
ポイント・トレース	-	-	A
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)			

表 2-20 実行中に設定/削除可能なイベント種別

△ : プログラムの実行を一瞬停止することで可能注

- ▲ :トレーサ/タイマ動作中は不可
- ×:不可
- -: 非サポート
- 注 プロパティ パネルの [デバッグ・ツール設定] タブ上の [実行中のイベント設定] カテゴリ内 [実行を 一瞬停止してイベントを設定する] プロパティにおいて, [はい] を指定することにより可能となります。

(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。

RENESAS

- 内蔵 ROM/ 内蔵 RAM のサイズを変更することにより、イベント設定箇所がノン・マップ領域になった場合、 設定しているイベントは発生しません(イベントパネル上でも無効状態/保留状態に変更されません)。
- 関数名や変数名を先頭のアンダー・バーの有無などで区別している場合,シンボル変換やブレーク・イベントの設定が不正になる場合があります。
 - 例: "_reset"と"__reset"などの2つの関数が存在する場合



2.17 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]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。

Ξ	フック処理設定	
Ð	ダウンロード前	ダウンロード前[0]
Ð	ダウンロード後	ダウンロード後[0]
Ŧ	ブレーク中のCPUリセット後	ブレーク中のCPUリセット後[0]
Ð	実行開始前	実行開始前[0]
Ŧ	ブレーク後	ブレーク後[0]

図 2—130 [フック処理]カテゴリ

表 2-21 [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	ブレーク中の 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] 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—131 テキスト編集 ダイアログのオープン

🛛 フック処理設定 🖌		
田 ダウンロード前	ダウンロード前[0]	(🛄)
田 ダウンロード後	ダウンロード後[0]	



図 2—132 フック処理を設定する(テキスト編集 ダイアログ)

テキスト編集	
テキスト(<u>T</u>):	
I	~
	<u>×</u>
<u>(</u>	
	OK キャンセル ヘルプ(H)

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。 各処理の指定形式は次のとおりです。

【処理 1】

I/O レジスタの内容を、数値に自動的に書き換えます。 指定形式:

1/0 レジスタ名 数値

【処理 2】

CPU レジスタの内容を, 数値に自動的に書き換えます。

指定形式:

CPU レジスタ名 数値

【処理 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。

指定形式:

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。

2. 半角スペースは、タブ文字でも代用可能です。

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—133 フック処理設定の例

テキスト編集	
テキスト(<u>T</u>):	
WTM 0×00	
<u>s</u>	
	OK キャンセル ヘルプ(H)



2.18 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。

2.18.1 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—22 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	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
(Unicode)	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 2-23 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
¥ν	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 值

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—24 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
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	Ox 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、Ox の接頭辞は必要 ありません。

(4) 式と演算子

式とは、定数、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。

式	説明
C 言語変数名 ^{注 1}	C言語の変数のアドレス
_{式[式]} 注2	配列のアドレス
<i>式</i> .メンバ名	構造体/共用体のメンバのアドレス
<i>式 -></i> メンバ名	ポインタの指し示す構造体/共用体のメンバの
	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 3} EQU シンボル名 ^{注 3} [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

表 2—25 アドレス式の基本入力形式

注1. 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。

2. 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。

 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—25 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。

式	説明
(演算順序の指定
- <i>±</i> ť	符号反転
! <i>式</i>	論理否定
~ 式	ビット反転
式* 式 ^注	棄算
<i>式 / 式</i> 注	除算
式%式 ^注	剰余算
<i>式</i> +式 ^注	加算
式-式 ^注	減算
<i>式 & 式</i> 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
<i>式 式</i> 注	ビットごとの論理和

表 2-26 演算子を用いたアドレス式の構成

- 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。

式	説明
C言語変数名	C 言語の変数の値
式[式]	配列の要素値
<i>式</i> .メンバ名	構造体/共用体のメンバ値
<i>式 -></i> メンバ名	ポインタの指し示す構造体/共用体のメンバ値
*	ポインタの変数の値

表 2-27 ウォッチ式の基本入力形式



式	説明
& <i>z</i> ť	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 EQU シンボル名 ^注 [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	文字定数值

 注 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)。なお"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:1:REG)。

また,「表 2-27 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。

式	説明
(演算順序の指定
! <i> </i>	論理否定
~ 式	ビット反転
<i>式* 式</i> 注	乗算
<i>式 / 式</i> 注	除算
<i>式 % 式</i> 注	剰余算
<i>式</i> + 式 ^注	加算
<i>式 - 式</i> 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

表 2-28 演算子を用いたウォッチ式の構成

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)。

2.18.2 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。
 - シンボル名の入力補完機能に必要な情報は、ロード・モジュール・ファイルのダウンロード時に生成されるため、この情報を生成するとダウンロード時間、およびホスト・マシンのメモリ消費量が増加します。シンボル名の入力補完機能を使用しない場合は、ダウンロード・ファイル ダイアログの [入力補完機能用の情報を生成する]項目で[いいえ]を選択し、この機能を無効化することを推奨します(デフォルトでは[はい]が選択されます)。

ただし、GHS コンパイラを使用している場合、この機能を無効化することはできません([入力補完機 能用の情報を生成する]項目の選択は無視されます)。

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。

メモリ1	
② ● 表記(N)▼	サイズ表記(2) - Iンコード [Ctrl] + [Space] キーを押下します。
+0 +1 +2 +3	aultBuild¥lcd_sample.abs C¥Project¥DefaultBuild¥lcd_sample.abs\$lcd_sample.c#init_cgx
000000 D8 00 FF OU OU GO 000000000000000000000000000000000000	aultBuild¥lcd_sample.abs\$lcd_sample.c# <u>init_</u> display aultBuild¥lcd_sample.abs\$lcd_sample.c# <u>init_</u> intc 表示されるリストの中から目的の シンボル名を選択します。
<	

図 2—134 シンボル名の入力補完機能



2.18.3 入力不備箇所に対するアイコン表示

CubeSuite+ 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。



付録 A ウインドウ・リファレンス

この付録では、CubeSuite+ 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。

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
デバッグ・マネージャ パネル	デバッグ対象とするコア(PEn)の選択、およびその状態の表示
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
エディタ パネル	ファイルの表示/編集、およびソース・レベル・デバッグ
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
メモリ・マッピング ダイアログ	メモリ・マッピングの表示
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
フラッシュ・オプションの設定 ダイアログ【Full-	フラッシュ・メモリのオ プション設定
spec emulator] [E1] [E20]	

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
優先するブート・ローダ・プロジェクトの選択 ダ	優先してデバッグ対象とするブート・ローダ・プロジェクトの選
イアログ	択
テキスト編集 ダイアログ	複数行のテキストの入力、編集
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
ファイル・エンコードの選択 ダイアログ	ファイル・エンコードの選択
ファイルの保存設定 ダイアログ	ファイルのエンコード、および改行コードの設定
ブックマーク ダイアログ	ブックマークの表示/削除
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
Print Preview ウインドウ	印刷する前のソース・ファイルのプレビュー
トレース検索 ダイアログ	トレース・データの検索
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 パネルのスクロール範囲の設定
指定行へのジャンプ ダイアログ	指定した行にキャレットを移動
関数ヘジャンプ ダイアログ	キャレットを移動する関数の選択
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
処理中表示 ダイアログ	処理の進捗状況の表示
オプション ダイアログ	各種環境の設定
ダウンロードするファイルを選択 ダイアログ	ダウンロード・ファイルの選択
ウォッチ式データ・ファイルを開く ダイアログ	ウオッチ式をインポートするためのファイルの選択
ファイルを開く ダイアログ	オープンするファイルの選択
名前を付けて保存 ダイアログ	ファイル、またはパネルの内容の新規保存
データ保存ファイルを選択 ダイアログ	データを保存する際のファイルの選択
オプション設定ファイルを開く ダイアログ	オプション ダイアログにインポートするオプション設定ファイル
	の選択
オプション設定ファイルを保存 ダイアログ	オプション ダイアログの設定内容をオプション設定ファイルに保存



メイン・ウインドウ

CubeSuite+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] → [プログラム] → [Renesas Electronics CubeSuite+] → [CubeSuite+] を選択

[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)。

_		
4	デバッグ・マネージャ	デバッグ・マネージャ パネルをオープンします。
		ただし、選択しているマイクロコントローラがシングルコア版の場合、またはデ
		バッグ・ツールと切断時は無効となります。
ŗ	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。
1	コーカル変数	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。
ŀ	OR	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。
C	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。
i	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。
Ŧ	見在の 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	定義箇所ヘジャンプ(「(7)関数ヘジャンプする」/「(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。

RENESAS

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
	(「(8) タグ・ジャンプする」参照)。

(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・ツールヘダウン	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
ロード	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&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。
ビルド&デバッグ・ツール	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
ヘダウンロード	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、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、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。
デバッグ・ツールからアッ	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
プロード …	す。
	ただし、プログラム実行中、[ビルド&デバッグ・ツールへダウンロード] 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、[ビルド&デバッグ・ツールへダウンロード] 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。
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&デバッグ・ツールへダウンロード]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&デバッグ・ツールへダウンロード] 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&デバッグ・ツールへダウンロード]実行

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。
	jarl 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(jarl 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、jarl 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&デバッグ・ツールへダウンロード]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&デバッグ・ツールへダウンロード] 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&デバッグ・ツールへダウンロード]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&デバッグ・ツールへダウンロード]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。

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、「2.8.3 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。
 - 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。

7	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・ツールヘダウンロード] 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&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。
	[デバッグ] メニュー→ [デバッグ・ツールヘダウンロード] の選択と同等です。
κJ.	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&デバッグ・ツールへダウンロード]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。

RENESAS

	 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&デバッグ・ツールへダウンロード]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&デバッグ・ツールへダウンロード]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&デバッグ・ツールへダウンロード]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。
9e	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&デバッグ・ツールへダウンロード]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。
Ç=	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・オーバ実行)。
	jarl 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(jarl 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、jarl 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&デバッグ・ツールへダウンロード]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&デバッグ・ツールへダウンロード] 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。
<u>*</u>	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&デバッグ・ツールへダウンロード]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、「2.8.3 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

CPU リセットを行います。4	6 行 13 桁 編集不可	T PE1	BREAK	Halt	💫 0x000009ae	🗖 🚥 RH850 シミュレータ	6.210 ms	
Ť	Î	Î	Î	Î	Î	Î	Î	Î
(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)

(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(「2.9 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) デバッグ対象コアの指定

デバッグ対象とするコア(PEn)を指定します(「2.7 コア(PE)の選択」参照)。

ただし,選択しているマイクロコントローラがシングルコア版の場合,またはデバッグ・ツールと切断 時の場合は非表示となります。

(d)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

(e) CPU 状態

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
Full-spec emulator	Halt	HALT モード中
E1/E20	Stop	STOP モード中
	Reset	リセット状態
	Pow Off	ターゲットに電源が供給されていない状態
シミュレータ	Halt	HALT モード中
	StopIdle	STOP/IDLE モード中
	Reset	リセット状態

(f) 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。

また, このエリアにマウスを重ねることにより, "カレント PC: 0x *カレント PC 値(ソース名 # 行数* 注)"をポップアップ表示します。

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。

注 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。

備考 プログラム実行中は、"実行中"と表示します。

ただし、リアルタイム表示更新を行っている場合、設定している表示更新間隔で PC 位置を更新 して表示します。

(g) 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

(h) Run-Break タイマ結果

Run-Break タイマの計測結果(「2.13.1 実行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


(i) 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	使用する		使用しない
	動作中	停止中	
トレース	34	34	Υ _γ
タイマ	3	3	0
カバレッジ			

備考 【シミュレータ】

プログラム実行が停止している場合、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(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] / [タイマ] / [カバレッジ] カテゴリ内 [トレース機能を使用する] / [タイマ機能を使用する] / [カバレッジ機能を使用する] プロパティの指定に反映されます)。



デバッグ・マネージャ パネル

選択しているマイクロコントローラがマルチコア対応版の場合において、デバッグ対象とするコア(PE:プロセッサ・エレメント)の選択、およびその状態の表示を行います(「2.7 コア(PE)の選択」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 選択しているマイクロコントローラがシングルコア版の場合、このパネルをオープンすることはできません。

	デバッグ・マネージャ	£	
[ツールバー] -	n 🖓 🖓 🔘	🕑 🕑 🙌 🖙 Çi Či 🕌	
Ē	デバッグ対象コアの)指定:	
(1) -	⊙ PE1]	O PE3	
Г	デバッグ対象コアの	0状態:	
	実行状態:	BREAK	
(2) —	コア・ステータス:		
	カレントPC:	0x0000086e	
L	L		

図 A---3 デバッグ・マネージャ 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→ [デバッグ・マネージャ] を選択

[各エリアの説明]

(1) [デバッグ対象コアの指定] エリア

デバッグ対象とする PE n をオプション・ボタンにより指定します。 ただし、プログラム実行中はこのエリアは無効となります。

備考 デバッグ対象コアの指定は、メイン・ウインドウのステータスバー上においても行うことができます。

(2) [デバッグ対象コアの状態] エリア

現在選択しているコアの状態を表示します。

備考 このエリアに表示される内容は、メイン・ウインドウのステータスバー上においても確認することが できます。

(a) [実行状態]

現在のプログラムの実行状態を次のアイコンと文字列で示します。

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

(b) [コア・ステータス]

現在のデバッグ・ツールのコアの状態を示します。

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。

デバッグ・ツール	表示内容	状態
Full-spec emulator	Halt	HALT モード中
E1/E20	StopIdle	ハードウェア STOP/ ソフトウエア STOP/IDLE モード中
	Hold	バス・ホールド中
	Wait	ウエイト状態
	Reset	リセット状態
	Pow Off	ターゲットに電源が供給されていない状態
シミュレータ	Halt	HALT モード中
	StopIdle	STOP/IDLE モード中
	Reset	リセット状態

(c) [カレント PC]

現在のカレント PC 位置の値を 16 進数で示します。

このボタンをクリックすると、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。

[ツールバー]

ツールバーの機能は、メイン・ウインドウ上のデバッグ・ツールバーの機能と同等です。 各ボタンの機能についての詳細は、「(2)デバッグ・ツールバー」を参照してください。



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—4 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	- 品種名
(デバッグ・ツール)	プロジェクトで選択しているマイクロコントローラの品種名(RH850)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(Full-spec emulator/E1(LPD)/
	E20(LPD)/ シミュレータ)を表示します ^注 。
	なお、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。

RENESAS

注 選択しているマイクロコントローラの種類により、使用可能なデバッグ・ツールは異なります。

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。
	品種名 Full-spec emulator	Full-spec emulator を使用します。
	<i>品種名</i> E1(LPD)	E1 を LPD 通信方式で使用します。
	<i>品種名</i> E20(LPD)	E20 を LPD 通信方式で使用します。
	品種名 シミュレータ	シミュレータを使用します。
	[゚] ゚゚゚ロパティ	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。





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) 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,田マークは、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。

(2) 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
- [フラッシュ・オプション設定] タブ【Full-spec emulator】【E1】【E20】
- [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文字列をすべて選択	
	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。	

[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) [クロック]
- (2) [ターゲット・ボードとの接続] 【Full-spec emulator】 【E1】 【E20】
- (3) [フラッシュ] 【Full-spec emulator】 【E1】 【E20】
- (4) [コンフィギュレーション] 【シミュレータ】
- (5) [CPU 仮想化支援機能]





図 A---7 プロパティ パネル: [接続用設定] タブ【E1】【E20】

E h	በቃን	the second s
	イン・クロック周波数[MHz]	10.00
ΞŌ	PUクロック周波数[MHz]	[2]
日夕	ーケット・ボードとの接続	
L	PDモード	4ピン
L	PDクロック周波数[kHz]	Default
持	統時にOPJTAGをLPD接続に設定する	(t (1)
切	断時にOPJTAGをJTAG接続に設定する	いいえ
日7	59921	
セ	パキュリティID	🗰 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
C	ode Flashのセルフ・プログラミングを行う	いいえ
ΞC	PU仮想化支援機能	
仍	刺マシン・スレッドを使用する	いいえ



図 A---8 プロパティ パネル: [接続用設定] タブ【シミュレータ】

🚉 RH850 シミュレータ のプロパティ	م
ם לפסל	
メイン・クロック周波数[MHz]	320.00
タイマ/トレース用クロック周波数の選択	GPUクロック周波数
タイマ/トレース用クロック周波数の単位	MHz
タイマ/トレース用クロック周波数	202 <u>202</u>
3 3774436-932	
シミュレータ・コンフィギュレーション・ファイルを使用する	いいえ
□ CPU仮想化支援機能	
仮想マシン・スレッドを使用する	いいえ
<u>////////////////////////////////////</u>	

[各カテゴリの説明]

(1)[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・クロック周波数	メイン・クロック周波数(逓倍前)を MHz 単位で指定します。		
[MHz]	デフォルト	[Full-spec emulator] [E1] [E20]	
		10.00	
		【シミュレータ】	
		320.00	
	変更方法	ドロップダウン・リストによる選択 またはキーボードからの直接入力	
	指定可能值	- ドロップダウン・リストによる次のいずれか	
		[Full-spec emulator] [E1] [E20]	
		10.00, 20.00(単位:MHz)	
		【シミュレータ】	
		1.00, 2.00, 3.00, 3.57, 4.00, 4.19, 4.91, 5.00, 6.00, 7.20,	
		8.00, 8.38, 9.60, 10.00, 12.00, 16.00, 20.00, 25.00, 30.00,	
		32.00, 33.33, 34.00, 40.00, 48.00, 50.00, 64.00, 80.00, 160.00,	
		240.00, 320.00(単位:MHz)	
		- テキスト入力による次の範囲	
		0.001 ~ 999.999(単位:MHz)	
CPU クロック周波数 [MHz]	CPU クロック周波数をコアごとに指定します。		
[Full-spec emulator]	各コアの CPU クロック周波数は、下段のサブプロパティで指定します。CPU クロック		
[E1] [E20]	周波数は、トレースのタイム・スタンプ情報を実時間に換算する際に使用されます。		
	なお、表示す	るサブプロパティの数は選択しているマイクロコントローラの種類に依存	
	します。		



コア名称	選択しているマイクロコントローラが持つコアの名称を表示します。					
(サブプロパティ)	デフォルト	選択しているマイクロコントローラに依存				
[Full-spec emulator]	変更方法	変更不可				
[E1] [E20]	ļ'					
CPU クロック 周波数	<i>コア名称</i> の CPU クロック周波数を指定します。					
(サブプロパティ)	デフォルト	選択しているマイクロコントローラに依存				
[Full-spec emulator]	変更方法	キーボードからの直接入力				
[E1] [E20]	指定可能值	0.001 ~ 999.999(単位:MHz)				
タイマ/トレース用ク ロック周波数の選択 【シミュレータ】	タイマ/トレース機能に使用するクロック周波数を表示します。					
	デフォルト	CPU クロック周波数				
	変更方法	変更不可				
タイマ/トレース用ク	タイマ/トレ	タイマ/トレース機能に使用するクロック周波数の単位を表示します。				
ロック周波数の単位	デフォルト	MHz				
【シミュレータ】	変更方法	変更不可				
タイマ/トレース用ク	タイマ/トレ	, ース機能に使用するクロック周波数の値を表示します。				
ロック周波数	ただし、デバ	、ッグ・ツールと切断時は""を表示します。				
【シミュレータ】	デフォルト	320.00				
	変更方法	変更不可				

(2) [ターゲット・ボードとの接続] 【Full-spec emulator】 [E1] [E20]

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。

注意 デバッグ・ツールが CubeSuite+ に接続している場合, 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。

ターゲット・ボードを接	Full-spec emulator にターゲット・ボードを接続しているか否かを選択します。				
続している	デフォルト	いいえ			
[Full-spec emulator]	変更方法	ドロップタ	マウン・リストによる選択		
	指定可能值	はい	ターゲット・ボードを接続しています。		
		いいえ	ターゲット・ボードを接続していません。		
LPD モード	LPD 通信方式				
【E1】【E20】	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	ドロップタ	ドロップダウン・リストによる選択		
	指定可能值	選択してい	るマイクロコントローラに依存		
ボーレート [kbps]	LPD 通信方式の通信速度を選択します。				
【E1】【E20】	なお, このプロパティは, [LPD モード] プロパティにおいて [1 ピン] を選択した場				
	合のみ表示さ	れます。			
	デフォルト	500			
	変更方法	ドロップダウン・リストによる選択			
	指定可能值	500, 1000	0, 2000(単位:kbps)		



LPD クロック周波数 [kHz]	LPD 通信方式	、のクロック	周波数を選択します。		
[E1] [E20]	Default を選択した場合、マイクロコントローラ固有のデフォルト値で接続処理を行い				
	ます。	す。			
	なお, このプ	プロパティは, [LPD モード] プロパティにおいて [4 ピン] を選択した場			
	合のみ表示されます。				
	デフォルト	フォルト Default			
	変更方法	ドロップタ	「ウン・リストによる選択		
	指定可能值	Default, 5	500, 11000(単位:kHz)		
接続時に OPJTAG を	デバッグ・ツー	ールとの接続時にシリアル・プログラミング・モードで起動し、オプション・バ			
LPD 接続に設定する	イトの設定を	LPD 接続に変	更するか否かを選択します。		
[E1] [E20]	デフォルト	はい			
	変更方法	ドロップタ	「ウン・リストによる選択		
	指定可能值				
			クします。この際に LPD 設定でない場合は LPD 設定に変更し、そ		
			の後デバッグ・モードに移行します。		
		いいえ	デバッグ・モードで起動し,OPJTAG をチェックします。この際		
			に LPD 設定でない場合はメッセージ・ダイアログを表示します。		
切断時に OPJTAG を	デバッグ・ツー	ールとの接続	を切断する際に、オプション・バイトの設定を JTAG 接続に変更す		
JTAG 接続に設定する	るか否かを選択	尺します。			
[E1] [E20]	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, 【接続時に OPJTAG を LPD 接続に設定する】プロパティで[い			
		いえ]を選	『択している場合は変更不可		
	指定可能值	はい	切断時にオプション・バイトの設定をJTAGに変更します。		
		いいえ			

(3) [フラッシュ] 【Full-spec emulator】 【E1】 【E20】

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。

注意 デバッグ・ツールが CubeSuite+ に接続している場合, 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。

セキュリティ ID 【E1】【E20】	内蔵 ROM a を指定します	Eたは内蔵 フラッシュ・メモリ上のコードを読み出す際の ID 認証用のキーコード 主 。			
	デフォルト FFFFFFFFFFFFFFFFFFFFFFFFFFFFF				
	変更方法	キーボードからの直接入力			
	指定可能值	32 桁の 16 進数(16 バイト)			



Code Flash のセルフ・プ	フラッシュ・	セルフ・プログラミング機能のフラッシュ・セルフ・ライブラリを使用し		
ログラミングを行う	て、Code Fla	de Flash を書き換えるか否かを選択します。		
	デフォルト			
	変更方法	ドロップダウン・リストによる選択		
	指定可能値	はい	Code Flash の書き換えを行います。この場合、Code Flash のキャッシュは行われません。	
		いいえ	Code Flash の書き換えを行いません。	

- **注** ID 認証用のキーコードについての詳細は、使用するデバッグ・ツールのユーザーズ・マニュアルを参照 してください。
- (4) [コンフィギュレーション]【シミュレータ】このカテゴリ内のプロパティは、常に無効です。

(5) [CPU 仮想化支援機能]

このカテゴリ内のプロパティは、常に無効です。



[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) [実行中のイベント設定] 【Full-spec emulator】 【E1】 【E20】
- (4) [ブレーク] 【Full-spec emulator】 【E1】 【E20】
- (5) [トレース]
- (6) [タイマ] 【シミュレータ】
- (7) [入力信号のマスク] 【Full-spec emulator】 【E1】 【E20】
- (8) [カバレッジ]【シミュレータ】
- (9) [シミュレータ GUI] 【シミュレータ】

図 A—9 プロパティ パネル : [デバッグ・ツール設定] タブ【Full-spec emulator】

🔍 RH850 Full-spec emulator のプロパティー		2
コ メモリ・マッピング	[26]	
メモリ書き込み時にベリファイを行う	(\$(.)	
ヨ 実行中のメモリ・アクヤス	1	
実行中にアクセスする	いいえ	
実行中に表示更新を行う	(はい)	
表示更新間隔[ms]	500	
日実行中のイベント設定		
実行を一瞬停止してイベントを設定する	いいえ	
田 ブレーク		
ヨ トレース		
🛛 入力信号のマスク		
WAIT 信号をマスクする	はい	
RESET 信号をマスクする	はい	
マスクするRESET信号の選択	TARGET RESET信号	
1 T II		
メモリ		



プロパティ		
🚉 RH850 XXXXX のプロパティ		P
🗆 አモリ		
田 メモリ・マッピング	[26]	
メモリ書き込み時にベリファイを行う	はい	
□ 実行中のメモリ・アクセス		
実行中にアクセスする	いいえ	
実行中に表示更新を行う	はい	
表示更新間隔[ms]	500	
日実行中のイベント設定		
実行を一瞬停止してイベントを設定する	いいえ	
団 ブレーク		
田 トレース		
□ 入力信号のマスク		
WAIT 信号をマスクする	いいえ	
RESET 信号をマスクする	いいえ	
hu-7		

図 A—10 プロパティ パネル:[デバッグ・ツール設定]タブ【E1】【E20】

図 A—11 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】

プロパティ	
🚉 RH850 シミュレータ のプロパティ	P -(
ロ メモリ	
田 メモリ・マッピング	[26]
🗆 実行中のメモリ・アクセス	
実行中に表示更新を行う	(tí)
表示更新間隔 [ms]	500
団 トレース	
団 タイマ	
⊞ カバレッジ	
🗆 シミュレータGUI	
シミュレータGUIを表示する	いいえ
አ ቲ ሀ	
接続用設定、デバッグ・ツール設	雇 ↓ ダウンロード・ファイル ↓ フック処理設定 /



[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。

メモリ・マッピング	メモリ・マッ	・ ・マッピングの状況をメモリ領域の種別 ^注 ごとに表示します。		
	デフォルト	[マイクロニ	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	編集不可		
	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		
メモリ書き込み時にベリ	メモリ値の初			
ファイを行う	デフォルト	はい		
[Full-spec emulator] [E1] [E20]	変更方法	ドロップダ	ウン・リストによる選択	
	指定可能值	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 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。

(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(「(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。

実行中にアクセスする	プログラム実行中に内蔵 RAM 領域にアクセスするか否かを選択します。			
[Full-spec emulator] [E1] [E20]	デフォルト	いいえ		
	変更方法	ドロップタ	ドロップダウン・リストによる選択	
	指定可能值	はい	実行中に内蔵 RAM 領域にアクセスします。	
		いいえ	実行中に内蔵 RAM 領域にアクセスしません。	
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			
	選択します。			
	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		
	指定可能值	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 ⁻	100 ms 単位で指定します。				
	なお、このブ	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				
	選択した場合	選択した場合のみ表示されます。				
	デフォルト	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)					

(3) [実行中のイベント設定]【Full-spec emulator】【E1】【E20】

実行中のイベント設定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。

実行を一瞬停止してイベ ントを設定する	プログラム実 ト(「(2)実行 たはトレーサ します。	ログラム実行中、またはトレーサ/タイマ動作中には設定することができないイベン (「(2)実行中に設定/削除可能なイベント種別」参照)を、プログラムの実行、ま はトレーサ/タイマの動作を強制的に一瞬停止させることで設定を行うか否かを選択 ます。		
	デフォルト	オルト いいえ		
	変更方法	ドロップダウン・リストによる選択 はい プログラムの実行、またはトレーサ/タイマの動作を一瞬		
	指定可能值			
		停止してイベントを設定します。		
		いいえ プログラム実行中、またはトレーサ/タイマ動作中に対象		
			イベントを設定することはできません。	

(4) [ブレーク]【Full-spec emulator】【E1】【E20】

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。

ソフトウエア・ブレ ー ク	ソフトウエア・ブレーク機能【Full-spec emulator】【E1】【E20】 ^注 を使用するか否かを						
を使用する	選択します。	星択します。					
	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	ただし、プログラム実行中は変更不可				
	指定可能值	はい	ソフトウエア・	・ブレーク機能を使用します。			
		いいえ	ソフトウエア	・ブレーク機能を使用しません。			
優先的に使用するブレー	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						
クポイントの種類	してマウスの	ワンクリック	ク操作でブレーク	フ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選択します。						
	デフォルト	ソフトウェ	ア・ブレーク				
	変更方法	ドロップダ	ウン・リストに	よる選択			
	指定可能值	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。			
		ハードウェ	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。			



停止時に周辺エミュレー	実行停止時に、エミュレータの周辺エミュレーション機能を停止(Peripheral Break)				
ションを停止する	するか否かを選択します。				
	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			
	指定可能值	はい 周辺エミュレーション機能を停止します。			
		いいえ	周辺エミュレーション機能を停止しません。		

- 注 1度ソフトウエア・ブレーク機能を使用したのち[いいえ]を選択した場合、それまで設定していたす べてのソフトウエア・ブレークポイント、および Printf イベントは無効状態となります。この場合、こ のプロパティを[はい]に再設定しても自動的に有効状態には戻りません(手動で設定を行う必要があ ります)。
- (5) [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(「2.12.1 トレース動作の設定をする」参照)。

注意 1. 【Full-spec emulator】【E1】【E20】

プログラム実行中は、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. 【E1】【E20】 接続したマイクロコントローラがトレース機能を搭載していない場合, デバッグ・ツールと接続

後, このカテゴリ内のプロパティは変更不可状態となります(トレース機能を使用することはでき ません)。

トレース・データの選択	収集するトレース・データの種類を選択します ^{注1} 。				
[Full-spec emulator]	デフォルト	分岐命令とデータ・アクセス			
[E1] [E20]	変更方法	ドロップダウン・リストによる選択			
	指定可能值	分岐命令	プログラム実行中に発生した分岐処理の分岐元/分岐先の		
			命令の PC 値をトレース・データとして収集します。		
		データ・ア	プログラム実行中に成立したアクセス系イベントのデータ		
		クセス	情報をトレース・データとして収集します。		
		分岐命令と	プログラム実行中に発生した分岐処理の分岐元/分岐先の		
		データ・ア	命令の PC 値、および成立したアクセス系イベントのデー		
		クセス	タ情報をトレース・データとして収集します。		
トレースの優先度	トレース・テ	ー ータを収集す	る際の優先度を選択します ^{注1} 。		
[Full-spec emulator]	デフォルト	スピード優先			
[E1] [E20]	変更方法	ドロップダウン・リストによる選択			
	指定可能值	スピード優	リアルタイム性を優先してトレースを行います。		
		先			
		データ優先	データの取りこぼしが発生しないように、CPU の実行パイ		
			プラインを一時的に停止してトレースを行います。		



し、フ機能た体田ナス	レレーフナ総合					
トレース懱能を使用する	トレース協能	1を10月9 つい				
【ンミュレーダ】	デフォルト	いいえ	いいえ			
	変更方法	ドロップダウ	ドロップダウン・リストによる選択			
		ただし、プロ	リグラム実行	中は変更不可		
	指定可能値	はい	トレース機	能を使用します。		
	ļ	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	実行前にトレ	ノース・メモリ	をクリアす	るか否かを選択します。		
リをクリアする	デフォルト	はい				
	変更方法	ドロップダウ	マ・リスト	による選択		
	指定可能值	はい	トレース・	メモリをクリアします。		
		いいえ	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	トレース・メ	モリが、収集し	したトレース	・データで満たされた際の動作を選択します ^{注1} 。		
切った後の動作	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	マ・リスト	による選択		
	指定可能值	トレース・メモリを上書 きし実行を続ける		トレース・メモリを使い切ると、古いトレー		
	ļ			ス・データに上書きを続けます。		
	ļ ļ	トレースを停	_{止する} 注3	トレース・メモリを使い切ると、トレース		
				データの書き込みを停止します(実行は停止し		
	ļ			ません)。		
	ļ	停止する		トレース・メモリを使い切ると、トレース		
	ļ			データの書き込みを中止すると同時に実行を停		
				止します。		
トレース・タイム・タグ	トレースパス	ネルに表示する	トレース時	間の表示方法を選択します。		
を積算する	デフォルト	いいえ				
【シミュレータ】	変更方法	ドロップダウ	·ン・リスト	による選択		
	指定可能值	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・タイム・タグ	トレースのタ	· イム・タグ表	示 (トレー;	スパネルの [時間] 表示)で使用するカウンタ		
の分周率	の分周率を運	፪択します 。				
【シミュレータ】	なお、分周率	፤の設定を変更	すると、タ・	イム・タグで表示されるカウンタのカウント・		
	アップに必要	アップに必要なクロック数も変更されます。				
	デフォルト					
	変更方法	ドロップダウ	ッン・リスト	による選択		
	指定可能值	1/1, 1/2, 1/-	4, 1/8, 1/1	6, 1/32, 1/64, 1/128, 1/256, 1/512, 1/1K,		
		1/4K, 1/8K, 1/16K, 1/64K, 1/256K, 1/1M				



トレースの取得範囲設定	トレース・ラ	データの取得範囲を選択します。					
[Full-spec emulator]	デフォルト	区間をトレー	-3				
[E1] [E20]	変更方法	ドロップダウ	 ドロップダウン・リストによる選択				
	指定可能值	区間をト	トレース開始イベントとトレース終了イベントで指定した				
		レース	区間の実行履歴をトレース・データとして収集します。				
		範囲外をト	トレース開始イベントとトレース終了イベントで指定した				
		レース	範囲外の実行履歴をトレース・データとして収集します。				
トレース・メモリ・サイ	トレース・デ	ータを格納する	メモリ・サイズをトレース・フレーム ^{注4} 数で選択します ^{注1} 。				
ズ[フレーム]	デフォルト	[Full-spec er	nulator]				
[Full-spec emulator]		8K					
【シミュレータ】		【シミュレー	۶]				
		4K					
	変更方法	ドロップダウン・リストによる選択					
	指定可能值	[Full-spec er	nulator]				
		8K, 32K, 64K, 128K, 256K, 512K 【シミュレータ】 4K, 8K, 12K, 16K, 20K, 24K, 28K, 32K, 36K, 40K, 44K, 48I					
		52K, 56K,	60K, 64K, 128K 192K, 256K, 320K, 384K, 448K,				
		512K, 576	бК, 640К, 704К, 768К, 832К, 896К, 960К, 1М, 2М,				
		3M					
トレースを補完する	収集したトレ	ノース・データを表示する際に、補完表示を行うか否かを選択します。					
[Full-spec emulator]	補完表示を行	うことにより、	ハードウエアではトレースできない分岐命令間の命令を表				
	ホしまり。 たむ 三の認	空け 次回取	得するトレーフ ・データトリ 反映 されます				
			すりるトレース・リーダより反映されまり。				
	テフォルト	はい					
	変更方法	ドロップダウ	シン・リストによる選択				
	指定可能值	はい	トレース・データの補完表示を行います。				
		いいえ	トレース・データの補完表示を行いません。				
トレースの取得対象設定	トレースのヌ	1象となるコア	を選択します。				
[Full-spec emulator]	デフォルト	ト デバッグ対象コアのみ					
[E1] [E20]	変更方法	ドロップダウ	ン・リストによる選択				
	指定可能值	デバッグ対	現在デバッグ対象に選択している PEnのみを対象にトレー				
		象コアのみ	ス・データを収集します(デフォルト)。				
		全てのコア	全 PE を対象にトレース・データを収集します。				

注 1. 【Full-spec emulator】【E1】【E20】

このプロパティを変更すると、トレース・メモリがクリアされます。

- 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。
- 3. [トレースの優先度] プロパティで [データ優先]を選択している場合, この項目は表示されません。
- **4.** 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。

(6) [タイマ]【シミュレータ】

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。

タイマ機能を使用する	タイマ機能を	- タイマ機能を使用するか否かを選択します。			
	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			
	指定可能值	はいタイマ機能を使用します。			
		いいえ	タイマ機能を使用しません。		

(7) [入力信号のマスク]【Full-spec emulator】【E1】【E20】

入力信号のマス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。

WAIT 信号をマスクする	WAIT 信号を	エミュレータに入力しないようにマスクするか否かを選択します。				
	デフォルト	[Full-spec	[Full-spec emulator]			
		はい	はい			
		【E1】 【E2	[E1] [E20]			
		いいえ				
	変更方法	ドロップ	ダウン・リストに	よる選択 ^注		
	指定可能值	はい	WAIT 信号をマス	くクします。		
		いいえ	WAIT 信号をマス	スクしません。		
RESET 信号をマスクす	RESET 信号:	をエミュレ	ータに入力しない	ようにマスクするか否かを選択します。		
る	デフォルト	[Full-spec	emulator]			
		はい				
		【E1】 【E2	0]			
		いいえ				
	変更方法	ドロップ:	ダウン・リストに	よる選択 ^注		
	指定可能值	はい	RESET信号をマ	マスクします。		
		いいえ	RESET 信号をマ	マスクしません。		
マスクする RESET 信号	マスクする R	 ESET 信号を選択します。				
の選択	なお, このプ	ロパティは, [RESET 信号をマスクする] 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	[Full-spec	emulator			
		TARGE	T RESET 信号			
		[E1] [E2	0]			
		TARGE	T RESET 信号と	INTERNAL RESET 信号		
	変更方法	[Full-spec	emulator]			
		ドロッフ	プダウン・リスト	による選択		
		【E1】 【E2	0]			
		変更不可	য			
	指定可能值	TARGET	RESET 信号	TARGET RESET 信号をマスクします。		
		TARGET	RESET 信号と	TARGET RESET 信号と INTERNAL RESET		
		INTERNA	L RESET 信号	信号をマスクします。		



注 【Full-spec emulator】

[接続用設定] タブ上の [ターゲット・ボードとの接続] [Full-spec emulator] [E1] [E20] カテゴリ内 [ターゲット・ボードを接続している] プロパティを [いいえ] に指定している場合, このプロパティ は, デバッグ・ツールとの接続時に自動的に [はい] に固定されます (変更不可)。

(8) [カバレッジ]【シミュレータ】

カバレッジ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。

カバレッジ機能を使用する	カバレッジ機能	能を使用するか否かを選択します。			
	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		
	指定可能值	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	デバッグ・ツ	ールと接続時	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		
する	否かを選択し	ます。			
	なお, このブ	゚ロパティは	[カバレッジ機能を使用する] プロパティにおいて [はい]		
	を選択した場	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			
	指定可能值	はいカバレッジ測定結果のロード/セーブを行います。			
		いいえ	カバレッジ測定結果のロード/セーブを行いません。		
カバレッジ測定領域 (1M	内蔵 ROM 領	域以外のカハ	「レッジ測定の対象領域を指定します。		
バイト)	カバレッジ測	定を行う任意	気の 1M バイト空間の開始アドレスを指定します。		
	なお, このブ	『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]			
	を選択した場	場合のみ表示されます。 100000 キーボードからの直接入力			
	デフォルト				
	変更方法				
	指定可能值	内蔵 ROM 領域以外のアドレス(シンボル指定不可)			

(9) [シミュレータ GUI] 【シミュレータ】

シミュレータ GUI 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 選択しているマイクロコントローラのシミュレータが周辺機能シミュレーションをサポートしていな い(命令シミュレーション版)場合,このカテゴリ内のプロパティはすべて無効となります。



シミュレータ GUI を表示	シミュレータ GUI の機能を使用するため、シミュレータ GUI ウインドウを表示するか				
する	否かを選択し	かを選択します。			
	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			
		ただし、プ	ログラム実行中は変更不可		
	指定可能值	はい	シミュレータ GUI ウインドウを表示します。		
		いいえ	ーーーーーーーーーーーー シミュレータ GUI ウインドウを表示しません。		
実行開始時に最前面表示	プログラムの実行開始時に、シミュレータ GUI ウインドウを最前面に表示するか否か				
する	を選択します。				
	な <u>お</u> , このプ	゚ロパティは	[シミュレータ GUI を表示する] プロパティにおいて [は		
	い]を選択し	た場合のみ表	夏示されます。		
	デフォルト	'ォルト はい			
	変更方法				
	指定可能值	はい	最前面に表示します。		
		いいえ	最前面に表示しません。		



[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-12 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 		
		(このパネル	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	ダウンロード	·ド後に CPU をリセットするか否かを選択します。		
をリセットする	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		
	指定可能值	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	ダウンロード	ダウンロード前にフラッシュ ROM を消去するか否かを選択します。					
シュ ROM を消去する 【Full-spec emulator】	デフォルト	いいえ					
	変更方法	ドロップダイ					
[E1] [E20]	指定可能值	はい	はい ダウンロー ド前にフラッシュ ROM を消去します。				
	ļ	いいえ	ダウンロー	ド前にフラッシュ ROM を消去しません。			
イベント設定位置の自動	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)						
変更方法	が命令の途中	になる場合の	再設定方法を	F選択します ^{注 2} 。			
	デフォルト	イベントを作	イベントを保留にする				
	変更方法	ドロップダイ					
指定可能们		命令の先頭に	こ移動する	命令の先頭アドレスに対象イベントを再設定し			
				ます。			
		イベントを作	呆留にする	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—1 ダウンロード可能なファイル」を参 照してください。
 - **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	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選択します。				
ンボル位置まで実行する	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		
	指定可能值	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			
	指定可能值	0~" <i>アドレス空間の終了アドレス</i> "のアドレス式			



[フラッシュ・オプション設定] タブ【Full-spec emulator】【E1】【E20】

[フラッシュ・オプション設定] タブでは、マイクロコントローラに搭載されているフラッシュ・メモリのためのオプ ション設定を行います。

ただし、このタブは、選択しているマイクロコントローラがフラッシュ・オプションに対応している場合のみ表示されます。

- 注意 1. デバッグ・ツールと接続している場合のみ、このタブの設定を行うことができます。
 - 2. このタブの設定を変更すると、選択しているマイクロコントローラの種類により CPU リセットが発生する 場合があります。
 - (1) [フラッシュ・オプション]

S RH950 YYYY M→□パティ		
RH000 XXXX 070/171		-
🛙 フラゥシュ・オプション		
フラッシュ・オプション	フラッシュ・オブション	
フラッシュ・オプション		
フラッシュ・オプション		
フラッシュ・オフション		
フラッシュ・オフション		

図 A-13 プロパティ パネル:[フラッシュ・オプション設定] タブ

[各カテゴリの説明]

(1) [フラッシュ・オプション]

フラッシュ・オプ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。

フラッシュ・オプション	フラッシュ・メモリのオプションを指定します。		
	デフォルト	フラッシュ・オプション	
	変更方法	フラッシュ・オプションの設定 ダイアログ【Full-spec emulator】【E1】	
		【E20】による指定	
		フラッシュ・オプションの設定 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(このパネル上でフラッシュ・メモリのオプションを指定するこ	
		とはできません)。	
		なお、設定したフラッシュ・メモリのオ プションの内容は、このパネル	
		上では表示されません。	



[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、「2.17 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-14 プロパティ パネル:[フック処理設定] タブ

🚑 RH850 XXXXX のプロパティ) ه
日 フック処理設定	- 16 - 16 -
田 ダウンロード前	ダウンロード前[0]
田 ダウンロード後	ダウンロード後[0]
田 ブレーク中のCPUリセット後	ブレーク中のCPUリセット後[0]
田 実行開始前	実行開始前[0]
田ゴレーカ2条	ブレーク(後[0]

[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。

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 ^注 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	
		テキスト編集 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(このパネル上で処理	
		を指定することはできません)。	
	指定可能值	128 個までの処理(テキスト編集 ダイアログ上の1行が1処理に相当)	
		ただし、1 処理につき 64 文字まで入力可	
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 ^注 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	
		テキスト編集 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(このパネル上で処理	
		を指定することはできません)。	
	指定可能值	128 個までの処理(テキスト編集 ダイアログ上の1行が1処理に相当)	
		ただし、1 処理につき 64 文字まで入力可	



ブレーク中の CPU	ブレーク中の CPU リセット直後に行う処理 ^注 を指定します。		
リセット後	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	
		テキスト編集 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(このパネル上で処理	
		を指定することはできません)。	
	指定可能值	128 個までの処理(テキスト編集 ダイアログ上の1行が1処理に相当)	
		ただし、1 処理につき 64 文字まで入力可	
実行開始前	プログラムの	。 D実行開始直前に行う処理 ^注 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	
		テキスト編集 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(このパネル上で処理	
		を指定することはできません)。	
	指定可能值	128 個までの処理(テキスト編集 ダイアログ上の1行が1処理に相当)	
		ただし、1 処理につき 64 文字まで入力可	
ブレーク後	プログラムの) D実行がブレークした直後に行う処理 ^注 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	
		テキスト編集 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(このパネル上で処理	
		を指定することはできません)。	
	指定可能值	128 個までの処理(テキスト編集 ダイアログ上の1行が1処理に相当)	
		ただし。1 処理につき 64 文字まで入力可	

注 次の3つの処理の中から目的に応じた処理の指定形式をテキスト編集ダイアログに入力します。

【処理 1】

I/O レジスタの内容を、数値に自動的に書き換えます。

指定形式:

I/O レジスタ名 数値

【処理 2】

CPU レジスタの内容を、数値に自動的に書き換えます。

指定形式:

CPU レジスタ名 数値

【処理 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。 指定形式:

Source Pythonスクリプト・パス

RENESAS

エディタ パネル

ファイルの表示,編集を行います。

デバッグ・ツールと接続中で、ダウンロードしたソース・ファイルをオープンしている場合では、ソース・レベル・ デバッグ/命令レベル・デバッグ(「2.8.3 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】(「2.14 カバレッジの測定【シミュレータ】)参照)を行うことができます。

また, 混合表示モードを選択することにより, ソース・テキストに対応した命令コード/ラベル/逆アセンブル・テキストを合わせて表示することができます(「(1)表示モードを変更する」参照)。

自動的にファイルのエンコードと改行コードを判別してオープンし,保存の際は元のフォーマットで保存します。ただし,ファイル・エンコードの選択ダイアログによりエンコードを指定してオープンすることができます。また,ファイルの保存設定ダイアログでエンコードと改行コードを指定した場合は,それに従って保存します。

このパネルは複数オープンすることができます(最大表示個数:100 個)。

- 注意 1. プロジェクトをクローズすると、該当プロジェクト内で登録されているファイルをオープンしているすべてのエディタ パネルがクローズします。
 - 2. プロジェクトからファイルの登録を外すと、該当ファイルをオープンしているエディタパネルがクローズします。
- 備考 ツールバーの 100% ▼, または [Ctrl] キーを押下しながらマウス・ホイールを前後方に動かすこと により、本パネルの表示を拡大/縮小することができます。



図 A—15 エディタ パネル





図 A—16 エディタ 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】



図 A—17 エディタ パネル(混合表示モードの場合)

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 (エディタ パネル専用部分)]
- [[編集] メニュー (エディタ パネル専用部分)]
- [[ウインドウ] メニュー (エディタ パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- デバッグ情報を持つロード・モジュール・ファイルのダウンロード直後、自動的にオープン
- プロジェクト・ツリー パネルにおいて,ファイルをダブルクリック
- プロジェクト・ツリー パネルにおいて、ファイルを選択したのち、コンテキスト・メニューの [開く] を選択
- プロジェクト・ツリー パネルにおいて、ファイルを選択したのち、コンテキスト・メニューの [内部エディタで 開く] を選択
- プロジェクト・ツリー パネルにおいて、コンテキスト・メニューの [追加] → [新しいファイルを追加] を選択 したのち、テキスト・ファイル/ソース・ファイルを作成
- ・ 逆アセンブル パネル/コール・スタック パネル/トレース パネル/イベント パネルにおいて、コンテキスト・メニューの [ソースへジャンプ]を選択
- カレント PC 値を強制的に変更した場合, またはプログラムの実行が停止した際に, カレント PC 値に対応する ソース・テキスト行が存在する場合に自動的にオープン

[各エリアの説明]

(1) タイトルバー

オープンしているファイルのファイル名を表示します。

なお、ファイル名の末尾に表示されるマークの意味は次のとおりです。

マーク	意味
*	ファイルをオープンしたのち、編集している場合に表示します。
!	ダウンロードしているロード・モジュール・ファイルの更新日時より、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に表示します。
	ただし、デバッグ・ツールと接続中で、ダウンロードしたソース・ファイルをオープンしている場
	合のみ有効となります。
[リサイクル]	リサイクル・モード(「(3)複数のソース・ファイルを1枚のパネルで表示する」参照)を有効と
	している場合に表示します。
	ただし、デバッグ・ツールと接続中で、ダウンロードしたソース・ファイルをオープンしている場
	合のみ有効となります。
(編集不可)	書き込み禁止状態のファイルをオープンしている場合に表示します。

(2) カラム・ヘッダ

エディタ パネルの各列のタイトルを表示します。

マウス・カーソルを重ねることによりタイトル名をポップアップ表示します。

RENESAS

表示	タイトル名	説明
行	行	行番号を表示します(「(4) 行番号エリア」参照)。
(表示なし)	選択	編集状況に応じた色表示を行います(「(5)選択エリア」参照)。
		ただし、混合表示モード時は表示されません。
(表示なし)	ソース・ファイル	ソース・ファイルの更新日時がロード・モジュール・ファイルの更新日時より
	がロード・モ	新しい場合に色表示を行います(「(6) ソース・ファイルがロード・モジュール
	ジュールより新し	より新しいことを示す表示エリア」参照)。
	いことを示す表示	ただし、デバッグ・ツールと切断時、または混合表示モード時は表示されません。
	カバレッジ	カバレッジ情報を表示します(「(7)カバレッジ・エリア」参照)。
		ただし、デバッグ・ツールと切断時は表示されません。
アドレス	アドレス	アドレスを表示します(「(8) アドレス・エリア」参照)。
		ただし、デバッグ・ツールと切断時は表示されません。
命令コード	命令コード	命令コードを表示します(「(9)命令コード・エリア」参照)。
		ただし、混合表示モード時のみ表示されます。
ラベル	ラベル	ラベルを表示します(「(10) ラベル・エリア」参照)。
		ただし、混合表示モード時のみ表示されます。
	イベント	イベントを設定します(「(11) イベント・エリア」参照)。
		ただし、デバッグ・ツールと切断時は表示されません。
(†	メイン	ブックマーク/アドレス・マーク/カレント PC マークを表示します。
		また、ブレークポイントを設定します(「(12) メイン・エリア」参照)。

備考 カラム・ヘッダは、ツールバーの設定により、表示/非表示を切り替えることができます。

(3) 分割バー

縦と横の分割バーを使うことにより、エディタパネルを分割して表示することができます。分割の上限は、 縦2分割、横2分割までです。

- 分割表示するには、分割バーを下方向/右方向の目的の位置までドラッグします。または、分割バーを ダブルクリックします。

- 分割表示を解除するには、分割バーをダブルクリックします。

注意 分割表示は、通常表示モードでのみ有効となります(混合表示モードに設定すると、分割表示は解除 されます)。



図 A-18 エディタ パネル(縦横2分割表示した場合)

前 1 世	אַלָּל 🖍 יי און 🖒					
行債	•	^	行	G		^
4	char gc_pel;		12		void main(int args)	
5	short gs pe3;		13		Ξ{	-
6	int si;		14		gc_pe1 = 0x12;	
7	long long gll;		15		gs_pe3 = 0x1234;	
8	int g_i = 0x123;	_	16		gi = 0×12345678;	
9	double g_d = 3.14;	~	17		gll = 0×123456781234	5 💙
<			<		10. J	
行債	a	^	行	G		^
21	sub(gi);		34		while(1)	
22			35		{	
23	if(args == 0x1)	-	36		gs_pe3++;	
24	{	-	37		gs_pe3++;	
25	while(1)		38		gs_pe3++;	
26	{	(219)	39		//halt()	;
27	80. I	DI 🎽	40		1	~
<	100		<		111	

(4) 行番号エリア

オープンしているファイルの行番号を表示します。

(5) 選択エリア

行の編集状況に応じて次の色表示を行います(混合表示モード時を除く)。

新規、または変更したが保存されていない行を示します。
新規、または変更後に保存済みの行を示します。
パネルをクローズしたのち、再度該当ソース・ファイルを表示するとこのマークは消失します。



(6) ソース・ファイルがロード・モジュールより新しいことを示す表示エリア

デバッグ・ツールと接続中で、ダウンロードしたソース・ファイルをオープンしている場合のみ有効となる エリアです(混合表示モード時を除く)。

ダウンロードしたロード・モジュール・ファイルの更新日時より、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、次の色表示を行います(マークの色はオプションダイアログにおける [全般 - フォント と色] カテゴリの警告色に依存)。

なお、ビルドを実行したのち、再度ロード・モジュール・ファイルをダウンロードするとこのマークは消失 します。





(7) カバレッジ・エリア

デバッグ・ツールと接続中で、ダウンロードしたソース・ファイルをオープンしている場合のみ有効となる エリアです。

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(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] カテゴリのカバレッジ色に依存)。

なお,カバレッジ測定についての詳細は,「2.14 カバレッジの測定【シミュレータ】」を参照してください。

(8) アドレス・エリア

デバッグ・ツールと接続中で、ダウンロードしたソース・ファイルをオープンしている場合のみ有効となる エリアです。

選択しているマイクロコントローラのメモリ空間での命令に対応するアドレスを表示します。

このエリアは 16 進表示固定です。

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。

(9) 命令コード・エリア

デバッグ・ツールと接続中で、ダウンロードしたソース・ファイルを混合表示モードでオープンしている場 合のみ有効となるエリアです。

ソース・テキストに対応した命令コードを表示します。

(10) ラベル・エリア

デバッグ・ツールと接続中で、ダウンロードしたソース・ファイルを混合表示モードでオープンしている場 合のみ有効となるエリアです。

ラベルが定義されている場合、ラベル名を表示します。

(11) イベント・エリア

デバッグ・ツールと接続中で、ダウンロードしたソース・ファイルをオープンしている場合のみ有効となる エリアです。

このエリアは、次の機能を備えています。

(a) 各種イベントの設定/削除

アドレス・マーク() がある行のコンテキスト・メニューにより、タイマ・イベント、トレース・イベント、またはアクション・イベント(Printf イベント)を設定/削除することができます。

イベントを設定すると,設定した行にイベント種別を示すイベント・マークが表示されます。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。

(b)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。

RENESAS

(12) メイン・エリア

このエリアは、次の機能を備えています。

(a) ブックマークの表示

登録したブックマーク()を表示します。

なお、ブックマークについての詳細は、「(9) ブックマークを登録する」を参照してください。

注意 混合表示モードを選択している場合、この機能は無効となります。

また、デバッグ・ツールと接続中で、ダウンロードしたソース・ファイルをオープンしている場合では、上 記機能に加え、次の機能を備えています。

(b) アドレス·マークの表示

アドレス表示のある行にアドレス・マーク() を表示します。 アドレス・マークのある行は、ブレークポイントや各種イベントを設定することができます。

(c) カレント PC マークの表示

カレント PC 位置(PC レジスタ値)を示すカレント PC マーク(눧)を表示します。

なお、カレント PC マークは、プログラムが停止状態の際にカレント PC 値がソース・テキスト行と対応する場合のみ表示します。

- 備考 混合表示モードを選択している場合で、ツールバーの (数) ボタンにより命令レベル単位のステッ プ実行を選択した場合は、カレント PC マークは逆アセンブル・テキスト行へ移動します。
- (d) ブレークポイントの設定/削除

アドレス・マーク () がある行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。

ブレークポイントを設定すると,設定行にイベント・マークを表示します。また,設定したブレークポ イントの詳細情報をイベント パネルに反映します。

すでにブレークポイントを設定している箇所でこの操作を行った場合は、そのブレークポイントを削除 します (ブレークポイントの設定は行いません)。

なお,ブレークポイントの設定方法についての詳細は,「2.9.3 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。

備考 ブレークポイントの設定,および設定したブレークポイントの状態変更は,このエリアのコンテ キスト・メニューからも行うことができます。

(e) 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。 (13) 文字列エリア

ファイルの文字列の表示/編集を行います。

このエリアは、次の機能を備えています。

(a) 文字列の編集

キーボードより、IME などの日本語入力システムを使用した文字列を入力することができます。 また、編集機能を充実させるための様々なショートカットキーを使用することができます。

注意 混合表示モードを選択している場合、この機能は無効となります。

- 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
 - タブ幅
 - 空白記号の表示/非表示
 - 予約語/コメントの色分け
- (b) コードのアウトライン表示

ソース・コード・ブロックの展開/折りたたみ表示を行い、現在編集中、またはデバッグ中のコード領域に集中して作業することができます。使用できるファイルの種類は、Cソース・ファイルです。 展開/折りたたみを行うには、それぞれ文字列エリアの左にあるプラス/マイナス記号をクリックします。 展開/折りたたみ可能なソース・コード・ブロックの種類を次に示します。

左中かっこと右中かっこ("{"と"}")	¥[]
複数行のコメント("/*"と"*/")	€ [/**/]
プリプロセッサ文("if", "elif", "else", "endif")	<pre>##if[Preprocessor block] ##elif[Preprocessor block] ##else[Preprocessor block] #endif</pre>

注意 混合表示モードを選択している場合、この機能は無効となります。

(c) カレント行の強調表示

オプションダイアログの[全般 - テキスト・エディタ]カテゴリ内の[カレント行のハイライト表示] チェック・ボックスをチェックすることにより,現在キャレットのある行を四角い枠で強調表示します (枠の色は同ダイアログの[全般 - フォントと色]カテゴリの強調色に依存)。

図 A-19 カレント行の強調表示

11	
12	void main(int args)
13	
14	sc_pe1 = 0×12;


(d) 括弧の強調表示

キャレット位置の括弧と、それに対応する括弧を強調表示します。 強調表示の対象となる括弧は次のとおりです。

ファイルの種類	対応する括弧
C言語ファイル	(と), {と}, および[と]
Python 言語ファイル	
HTML言語ファイル	< と >
XML 言語ファイル	

備考 対応する括弧を表示する際に、コメント内の括弧、および文字定数/文字列/文字列定数内の括 弧を考慮しません。そのため、これらが存在する場合では、実際に対応する括弧とは異なる括弧 が強調表示されることがあります。

(e) 複数行選択とブロック選択

次の操作により、複数行選択、またはブロック選択を行うことができます。

- 複数行選択を行う場合:
- 左マウス・ボタンでドラッグ
- [Shift] キーを押下しながら, [←] / [→] / [↓] ボタンを押下
- ブロック選択を行う場合:
- [Alt] キーを押下しながら, 左マウス・ボタンでドラッグ
- [Alt] + [Shift] キーを押下しながら, [←] / [→] / [↓] ボタンを押下

図 A-20 複数行選択とブロック選択



注意 ブックマーク情報は、選択内容に含まれません。

備考 選択した内容は、[編集]メニューの [切り取り] / [コピー] / [貼り付け] / [削除] の対象 となります。

(f) 関数へのジャンプ

現在選択している文字列, またはキャレット位置の単語を関数名と判断し, 該当する関数へジャンプす ることができます。

関数へのジャンプの動作の詳細については、「(7) 関数へジャンプする」を参照してください。

(g) タグ・ジャンプ

現在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、コンテキスト・メニューの[タグ・ジャンプ]を選択することにより、該当ファイルを新たなエディタパネルにオープンし、該当行/該当桁 ヘジャンプすることができます(該当ファイルがすでにオープンしている場合は、そのエディタパネルに ジャンプ)。

タグ・ジャンプの動作の詳細については、「(8) タグ・ジャンプする」を参照してください。

(h) ブックマークの登録

ブックマーク・ツールバーの □ ボタンをクリック,またはこのエリアのコンテキスト・メニューの [ブックマーク] → [ブックマークの挿入/削除]を選択することにより,現在キャレットのある行にブッ クマークを登録することができます。

なお、ブックマークについての詳細は、「(9) ブックマークを登録する」を参照してください。

注意 混合表示モードを選択している場合、この機能は無効となります。

(i) ファイルの監視機能

CubeSuite+ 以外によって,現在表示しているファイルの内容が変更(リネーム/削除を含む)された場合,ファイルを更新するか否かのメッセージを表示し,どちらかを選択することができます。

(j) スマート・エディット機能

スマート・エディット機能とは、コーディング中に関数情報や変数情報、および関数の引数情報などを リスト表示することにより、ユーザの入力を補完する機能です。

スマート・エディット機能では、次の情報の入力補完を行います。

- C 言語におけるグローバル関数

- C 言語におけるグローバル変数

図 A—21 スマート・エディット機能の表示例(関数/変数の候補表示)



- 注意 1. この機能は、ビルド・ツールとして CC-RH を使用している場合のみサポートされる機能です (GHS コンパイラを使用している場合、この機能は無効となります)。
 - 2. 混合表示モードを選択している場合、この機能は無効となります。

なお、スマート・エディット機能を使用するためには、次の設定が必要となります。

- オプション ダイアログの [全般 テキスト・エディタ] カテゴリ内の [スマート・エディット] チェック・ボックスを選択してください (デフォルト)。
- スマート・エディット機能は、ビルド・ツールが出力するクロス・リファレンス情報を使用して入力候補表示を行います。したがって、使用するビルド・ツールのプロパティパネルにおいて、クロス・リファレンス情報を出力する設定注にしたのち、ビルドの実行を完了させてください。
 なお、ビルドの際にエラーが発生した場合、エラー発生前のクロス・リファレンス情報が存在する場合はそれを使用します。
 - 注 [共通オプション] タブ→ [出力ファイルの種類と場所] カテゴリ→ [クロス・リファレンス情報を出力する] プロパティ→ [はい (-Xcref)] この設定が無効な場合, クロスリファレンス情報がクリアされるため, スマート・エディット 機能は使用不可となります。

- 関数/変数の候補表示

- 表示方法

関数/変数の候補は、次のいずれかの場合により表示されます。

- C 言語において、"."、または"->"を入力した時点で左辺に対して該当するメンバがある場合 - キーボードより、[Ctrl] + [Space] キーを押下した場合(すべての候補を表示)

ただし、候補が1つのみの場合は候補表示を行わず、該当文字列がこの時点で挿入されます。

- 文字列の挿入方法

候補表示のリストより文字列を [↑] / [↓] キー, またはマウスにより選択したのち, [Enter], または [TAB] キーを押下します。

- 各エリアの説明



図 A-22 関数/変数の候補表示

- 候補リスト

関数/変数の候補をアルファベット順に表示します。

キャレット位置の文字列と一致する文字列がある場合は、その文字列が強調表示されます(大文 字/小文字不問)。

なお、各候補の先頭には、次のアイコンを表示します。

アイコン	説明
	候補が typedef であることを示します。
=0	候補が関数であることを示します。
۵	候補が変数であることを示します。
}	候補が構造体であることを示します。
\diamond	候補が共用体であることを示します。

- ツールバー

関数/変数の候補の表示/非表示を切り替えます。

ボタン	説明
=0	選択することにより、関数の候補を表示します。
٢	選択することにより、変数の候補を表示します。

- タブ

表示するメンバを切り替えます。

タブ名	説明
All members	すべての候補を表示します。
Public members	Public 属性の候補のみ表示します。

- 詳細表示

現在選択されている関数/変数の詳細情報を表示します。

項目	説明
(1)種別	次の種別を表示します。
	(function):関数であることを示します。
	(variable)∶変数であることを示します。
(2)型	関数/変数の型を表示します。
(3) 名称	関数/変数の名称を表示します。
(4) 名称と引数	関数/変数の名称を表示します。関数の場合は引数も表示します。

- 引数の候補表示

- 表示方法

引数の候補は、次のいずれかの場合に表示されます。

- 関数名において、"("を入力した時点で左辺に対して該当する関数の引数がある場合
- 関数の引数位置にキャレットがある状態で、キーボードより、[Ctrl] + [Shift] + [Space] キー を入力した場合

- 各エリアの説明





項目	説明
(1)型	関数の型を表示します。
(2) 名称と引数	関数の名称と引数表示します。現在のキャレット位置の引数が強調表示されます。
(3) 名称と引数	関数の名称と引数表示します。

- 候補表示の消去

候補表示は、次のいずれかの操作により消去されます。

- [ESC] キーの押下
- 英数字以外のキーの入力 候補リストで何も選択していない場合: 何もしません。 候補リストで選択している場合: 選択している候補の文字列を挿入します。

- 候補表示の際の注意

- 次の項目は候補表示に対応していません。
 - マクロ定義
 - ローカル変数
 - Typedef
- 関数中に構造体宣言/共用体宣言を行った場合、宣言以降関数内で候補表示は行いません。
- 変数のサイズに影響するコンパイル・オプションを設定した場合、表示される変数の型が実際の宣言と異なる場合があります。
- 備考 ソース・テキスト上の関数名/変数名にマウス・カーソルを重ねると、その関数/変数の情報をポップアップ表示します。
 ただし、次の注意が必要です。
 デバッグ・ツールと接続中にこの機能は使用できません。
 - const, static, および volatile 属性は表示しません。

- クラス型/構造体型/共用体型/列挙型の変数の場合、メンバの一覧を次のように表示します。
 - クラス型/構造体型/共用体型の場合、メンバの型とメンバ名を表示
 - なお、クラス型の場合でメンバが関数の場合は、返却値の型と関数名を表示し、関数名の最後 に"("")"を付加
- 列挙型の場合、メンバ名のみ表示
- メンバ表示については、ソース・ファイルに定義されている順番で1行に1つを表示(最大 20 個まで)
- なお、ポップアップ表示するアイコンの意味は次のとおりです。

アイコン	説明
	対象が typedef であることを示します。
=	対象が関数であることを示します。
٢	対象が変数であることを示します。
>	対象が構造体であることを示します。
\diamond	対象が共用体であることを示します。
_	対象が列挙型であることを示します。

図 A-24 スマート・エディット機能によるポップアップ表示

72	hera = tashizan(hera,4);
73	g_send. int tashizan()
74	g_cha = [function]
75	

また、デバッグ・ツールと接続中で、ダウンロードしたソース・ファイルをオープンしている場合では、上 記機能に加え、次の機能を備えています。

(k) カレント PC 行の強調表示

カレント PC 位置(PC レジスタ値)がソース・テキスト行と対応する場合,該当行を強調表示します (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 PC 位置色に依存)。

(I) ブレークポイント設定行の強調表示

ブレークポイントが設定されている行を強調表示します(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] カテゴリのブレークポイント色に依存)。

(m) コード・カバレッジ測定結果の表示【シミュレータ】

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(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] カテゴリのカバレッジ色に依存)。

カバレッジ測定についての詳細は、「2.14 カバレッジの測定【シミュレータ】」を参照してください。

(n) 変数値のポップアップ表示

変数上にマウス・カーソルを重ねることにより、変数名と変数値をポップアップ表示します(「(4) 変数値を表示する」参照)。

(o) 各種イベントの設定

アドレス・マーク () がある行のコンテキスト・メニューにより, 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と、対応するイベント・マークをイベント・エリア、またはメイン・エリアに表示 します。また、設定したイベントの詳細情報をイベントパネルに反映します。

なお、各種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。

- 「2.9.4 任意の場所で停止する (ブレーク・イベント)」
- 「2.9.5 変数 /I/O レジスタへのアクセスで停止する」
- 「2.12.3 任意区間の実行履歴を収集する」
- 「2.12.4 条件を満たしたときのみの実行履歴を収集する【シミュレータ】」
- 「2.13.2 任意区間の実行時間を計測する【シミュレータ】」
- 「2.15.1 printf を挿入する」
- (p) ウォッチ式の登録

表示している 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、「(1) ウォッチ式を登録する」を参照してください。

1 2 - 12/13 - 1	(—1	レバ	ール	「ツ	ſ
-----------------	-------------	----	----	----	---

8 .	このパネルの表示モードとして、通常表示モード(デフォルト)と混合表示モードを切り替え
	ます(「(1)表示モードを変更する」参照)。
	ただし、デバッグ・ツールと接続中で、かつダウンロードしたソース・ファイルをオープンし
	ている場合のみ有効となります。
	ステップ実行を行う際の単位として、ソース・レベル(デフォルト)と命令レベルを切り替え
	ます(「2.8.3 プログラムをステップ実行する」参照)。
	ステップ実行の単位を命令レベルに設定すると、カレント PC マークが逆アセンブル・テキス
	ト行へ移動します
	ただし、デバッグ・ツールと接続中で、かつ混合表示モードの場合のみ有効となります。
	現在の PC 位置を表示します。
	ただし、デバッグ・ツールと接続中の場合のみ有効となります。
Ç.	[コンテキスト・メニュー]→[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。
	ただし,混合表示モードの場合は無効となります。
5	[コンテキスト・メニュー]→[関数ヘジャンプ]を実行する前の位置へ戻ります。
	ただし、混合表示モードの場合は無効となります。

カラム	エディタ パネルで表示するカラム、またはマークの表示/非表示を切り替える次の項目を表示
	します。チェックを外すことにより非表示となります。
	なお、この設定は、すべてのエディタ パネルに反映します。
行	行番号エリアにおいて、行番号を表示します。
選択	行番号エリアにおいて、行の編集状態を示すマークを表示します。
ソース・ファイルが	行番号エリアにおいて、ダウンロードしたロード・モジュール・ファイルの更新状態を示す
ロードモ ジュールより	マークを表示します。
新しいことを示す表示	ただし、デバッグ・ツールと接続中の場合のみ有効となります。
カバレッジ	カバレッジ・エリアを表示します。
	ただし、デバッグ・ツールと接続中の場合のみ有効となります。
アドレス	アドレス・エリアを表示します。
	ただし、デバッグ・ツールと接続中の場合のみ有効となります。
命令コード	命令コード・エリアを表示します。
	ただし、デバッグ・ツールと接続中で、かつ混合表示モードの場合のみ有効となります。
ラベル	ラベル・エリアを表示します。
	ただし、デバッグ・ツールと接続中で、かつ混合表示モードの場合のみ有効となります。
イベント	イベント・エリアを表示します。
	ただし、デバッグ・ツールと接続中の場合のみ有効となります。
メイン	メイン・エリアを表示します。
カラム・ヘッダ	カラム・ヘッダを表示します。

[[ファイル] メニュー (エディタ パネル専用部分)]

エディタ 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)。

<i>ファイル名</i> を閉じる	現在編集しているエディタ パネルをクローズします。
	なお、パネルの内容が保存されていない場合は、確認メッセージを表示します。
<i>ファイル名</i> を保存	現在編集しているエディタ パネルの内容を上書き保存します。
	なお、ファイルが一度も保存されていない、またはファイルが書き込み禁止の場合は、
	[名前を付けて <i>ファイル名</i> を保存] の選択と同等の動作となります。
	ただし、混合表示モードの場合は無効となります。
名前を付けて <i>ファイル名</i> を保存	現在編集しているエディタ パネルの内容を新規保存するために、名前を付けて保存 ダ
	イアログをオープンします。
	なお、混合表示モードの場合は、" <i>ファイル名</i> "を"混合表示"と表示します。
<i>ファイル名</i> の保存設定	現在編集しているエディタ パネルでオープンしているファイルのエンコードと改行
	コードを変更するために、ファイルの保存設定 ダイアログをオープンします。
ページ設定	この項目は常に無効です。
印刷	現在編集しているエディタ パネルの内容を印刷するために、Windows の印刷用 ダイア
	ログをオープンします。
印刷プレビュー	印刷するファイル内容のプレビューを行うために、Print Preview ウインドウをオープン
	します。



[[編集] メニュー (エディタ パネル専用部分)]

エディタパネル専用の[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)。

元に戻す	前回行った操作をキャンセルし、文字とキャレット位置を元に戻します(最大 100 回
	まで)。
	ただし、混合表示モードの場合は無効となります。
やり直し	前回行った[元に戻す]の操作をキャンセルし、文字とキャレット位置を元に戻します。
	ただし、混合表示モードの場合は無効となります。
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。
	何も選択されていない場合は、その行を切り取ります。
	ただし、混合表示モードの場合は無効となります。
⊐ピ—	選択範囲の文字列をクリップ・ボードにコピーします。
	何も選択されていない場合は、その行をコピーします。
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、挿入モードの場合は
	挿入し、上書きモードの場合は上書きします。
	ただし、クリップ・ボードの内容を文字列として認識できない場合、または混合表示
	モードの場合は無効となります。
削除	キャレット位置の文字を1文字削除します。
	範囲選択している場合は、選択されている文字列を削除します。
	ただし、混合表示モードの場合は無効となります。
すべて選択	現在編集中のテキストの先頭から最終までを選択状態にします。
	ただし、混合表示モードの場合は無効となります。
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。
置换	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。
	ただし、混合表示モードの場合は無効となります。
移動	指定した行へキャレットを移動するため、指定行へのジャンプ ダイアログをオープンし
	ます。
ブックマーク	ブックマークに関するカスケード・メニューを表示します(「(9) ブックマークを登録
	する」参照)。
	ただし、混合表示モードの場合は無効となります。
ブックマークの挿入/削除	キャレット位置の行にブックマークを登録します。
	すでにブックマークを登録している場合は、そのブックマークを削除します。
次のブックマーク	アクティブなエディタ パネルにおいて、次に登録したブックマーク位置にキャレット
	を移動します。
前のブックマーク	アクティブなエディタ パネルにおいて,前に登録したブックマーク位置にキャレット
	を移動します。
すべてのブックマークの削除	アクティブなエディタ パネルにおいて、登録しているブックマークをすべて削除しま
	す。
ブックマークをリスト表示する	登録しているブックマークをリスト表示するブックマーク ダイアログをオープンしま
	す。

コードのアウトライン	ソース・ファイルのコードの展開/折りたたみ表示を行うためのカスケード・メニューを
	表示します(「(b) コードのアウトライン表示」参照)。
	ただし、混合表示モードの場合は無効となります。
定義を折りたたむ	関数定義など、実装ブロックとして登録されているすべてのノードを折りたたみます。
アウトラインを切り替える	折りたたまれた部分で、カーソルが置かれている最も内側のアウトライン部分の現在の 状態を切り替えます。
すべてのアウトラインを切り	すべてのノードの状態を切り替え、すべて同じ状態(展開または折りたたみ)に設定し
替える	ます。折りたたまれているノードと展開されたノードが混在している場合。すべてを展 開します。
アウトラインを中止する	コードのアウトラインを中止します。 現在の ソース・ファイルからすべてのアウトライ ン情報を削除します。
自動アウトラインを開始する	コードの自動アウトラインを開始します。サポートしているソース・ファイルのアウト ライン情報を自動的に表示します。
高度な設定	エディタ パネルに関する高度な操作を行うためのカスケード・メニューを表示します。 ただし、混合表示モードの場合は無効となります。
行インデントを増やす	現在カーソルのある行のインデントをタブ1個分増やします。
	現在カーソルのある行のインデントをタブ1個分減らします。
行コメントを削除する	現在カーソルのある行の先頭から、言語に応じた行コメントの区切り記号の最初のセット
	を削除します。現在のソース・ファイルが行コメントの区切り記号が指定されている言語 を使用している場合のみ使用できます。
行コメントを付ける	現在カーソルのある行の先頭に、言語に応じた行コメントの区切り記号を設定します。現 在のソース・ファイルが行コメントの区切り記号が指定されている言語を使用している場
	合のみ使用できます。
タブをスペースに変換する	現在カーソルのある行のすべてのタブをスペースに変換します。
スペースをタブに変換する	現在カーソルのある行の連続したスペースの一組をタブに変換します。ただし、そのス ペースの各組がタブ1個以上の幅に等しい場合に限ります。
選択行をタブ化する	現在の行をタブ化します。行の先頭にある(テキストの前の)すべてのスペースを可能な 限りタブに変換します。
選択行を非タブ化する	現在の行を非タブ化します。行の先頭にある(テキストの前の)すべてのタブをスペース に変換します。
大文字にする	選択しているすべての文字を大文字に変換します。
小文字にする	選択しているすべての文字を小文字に変換します。
大文字/小文字を切り替える	選択しているすべての文字を、大文字または小文字に切り替えます。
先頭を大文字にする	選択しているすべての単語の先頭文字を大文字に変換します。
前後の空白を削除する	カーソル位置の前後にある余分な空白を削除し、空白文字を1個だけ残します。カーソル が単語内にある場合、または前後に空白文字がない場合、何も行いません。
末尾の空白を削除する	カーソルのある行で、最後の非空白文字の後にある空白を削除します。
行を削除する	現在カーソルのある行を完全に削除します。
行をコピーする	現在カーソルのある行をコピーして、その直後に挿入します。
空白行を削除する	カーソルのある行が空である場合。または空白文字しかない場合。その行を削除しま す。



[[ウインドウ] メニュー (エディタ パネル専用部分)]

エディタ パネル専用の [ウインドウ] メニューは次のとおりです (その他の項目は共通)。

分割	アクティブのエディタ パネルを水平方向に分割します。
	分割の対象は、アクティブのエディタ パネルのみで、他のパネルは分割されません。
	分割の上限は、4分割までです。
分割の解除	エディタ パネルの分割表示を解除します。

[コンテキスト・メニュー]

- (1) タイトルバー・エリア
- (2) カバレッジ・エリア【シミュレータ】
- (3) イベント・エリア
- (4) メイン・エリア (デバッグ・ツールと接続中の場合)
- (5) 文字列エリア(デバッグ・ツールと切断時の場合)
- (6) 文字列エリア(デバッグ・ツールと接続中の場合)

(1) タイトルバー・エリア

閉じる	選択しているパネルを閉じます。
このタブ以外すべて閉じる	選択しているパネルと同じパネル表示エリアに表示されているパネルを、選択して いるパネルのみ残し、すべて閉じます。
<i>ファイル名</i> の保存	ファイルの内容を保存します。
完全パスのコピー	ファイルの絶対パスをクリップ・ボードにコピーします。
含んでいるフォルダを開く	テキスト・ファイルが保存されているフォルダをエクスプローラで開きます。
新しい水平タ ブグループ	 アクティブなパネルの表示領域を水平方向に均等に2分割して、新たなタブ・グループを表示します。新たなタブ・グループには、アクティブなパネルが1つだけ入ります。分割の上限は、4分割までです。 次の場合は、この項目は表示されません。 タブ・グループにパネルが1つしか開いていない 垂直方向にタブ・グループが分割されている タブ・グループが4分割されている
新しい垂直タ ブグループ	 アクティブなパネルの表示領域を垂直方向に均等に2分割して、新たなタブ・グループを表示します。新たなタブ・グループには、アクティブなパネルが1つだけ入ります。分割の上限は、4分割までです。 次の場合は、この項目は表示されません。 タブ・グループにパネルが1つしか開いていない 水平方向にタブ・グループが分割されている タブ・グループが4分割されている



次のタブグループへ移動	表示領域を水平方向に分割している場合、選択しているパネルを表示しているタ
	ブ・グループの下側のタブ・グループに移動します。
	表示領域を垂直方向に分割している場合、選択しているパネルを表示しているタ
	ブ・グループの右側のタブ・グループに移動します。
	移動する側にタブ・グループがない場合は、この項目は表示されません。
前のタブグループへ移動	表示領域を水平方向に分割している場合、選択しているパネルを表示しているタ
前のタブグループへ移動	表示領域を水平方向に分割している場合、選択しているパネルを表示しているタ ブ・グループの上側のタブ・グループに移動します。
前のタ ブグルー プへ移動	表示領域を水平方向に分割している場合、選択しているパネルを表示しているタ ブ・グループの上側のタブ・グループに移動します。 表示領域を垂直方向に分割している場合、選択しているパネルを表示しているタ
前のタ ブグルー プへ移動	表示領域を水平方向に分割している場合、選択しているパネルを表示しているタ ブ・グループの上側のタブ・グループに移動します。 表示領域を垂直方向に分割している場合、選択しているパネルを表示しているタ ブ・グループの左側のタブ・グループに移動します。

(2) カバレッジ・エリア【シミュレータ】

カバレッジ情報をクリア	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。

(3) イベント・エリア

実行時にタイマ開始	選択している行が実行された際に、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(「2.13.2 任意区間の実行時間を計測する【シミュ レータ】」参照)。
実行時にタイマ終了	選択している行が実行された際に、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(「2.13.2 任意区間の実行時間を計測する【シミュ レータ】」参照)。
トレース開始の設定	選択している行が実行された際に、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(「2.12.3 任意区間の実行履歴 を収集する」参照)。
トレース終了の設定	選択している行が実行された際に、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(「2.12.3 任意区間の実行履歴 を収集する」参照)。
アクション・イベントの登録…	選択している行に対応する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(「2.15.1 printf を挿入する」参 照)。
有効化	選択しているイベントを有効状態にします。 複数のイベントを設定していることを示すイベント・マーク(PPG))を選択してい る場合は、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。 複数のイベントを設定していることを示すイベント・マーク(PPG))を選択してい る場合は、すべてのイベントを有効状態にします。
イベント削除	選択しているイベントを削除します。 複数のイベントを設定していることを示すイベント・マーク(PPG))を選択してい る場合は、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。

(4) メイン・エリア(デバッグ・ツールと接続中の場合)

ブレークの設定/削除	キャレット位置の行にブレークポイントを設定します(「2.9.3 任意の場所で停止
	する (ブレークポイント)」参照) 注。
	すでにブレークポイントを設定している場合は、そのブレークポイントを削除します。
ハードウェア・ブレークポイ	キャレット位置の行にブレークポイント(ハードウェア・ブレーク・イベント)を
ントの設定	設定します。
[Full-spec emulator]	
【E1】【E20】	
ソフトウエア・ブレークポイ	キャレット位置の行にブレークポイント(ソフトウエア・ブレーク・イベント)を
ントの設定	設定します。
[Full-spec emulator]	
【E1】【E20】	
ハードウエア・ブレーク優先	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ェア・ブレーク
[Full-spec emulator]	ポイントとします(プロパティ パネル上の [デバッグ・ツール設定] タブの [ブ
【E1】【E20】	レーク]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 ク優先	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
[Full-spec emulator]	イントとします(プロパティ パネル上の[デバッグ・ツール設定]タブの[ブレー
【E1】【E20】	ク]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)。
有効化	選択しているブレークポイントを有効状態にします。
	複数のイベントを設定し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、すべてのイベントを有効状態にします。
無効化	選択しているブレークポイントを無効状態にします。
	複数のイベントを設定し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、すべてのイベントを有効状態にします。
イベント削除	選択しているブレークポイントを削除します。
	複数のイベントを設定し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、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。

注 【Full-spec emulator】【E1】【E20】

デフォルトは、リソースが使用可能であれば、デバッグ・ツールはハードウエア・ブレークポイントを 設定します。[ハードウエア・ブレーク優先]、または[ソフトウエア・ブレーク優先]を選択すること で、この動作をカスタマイズすることができます。

(5) 文字列エリア(デバッグ・ツールと切断時の場合)

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。
	何も選択していない場合は、その行を切り取ります。
コピー	選択範囲の文字列をクリップ・ボードにコピーします。
	何も選択していない場合は、その行をコピーします。



貝	らり付け	クリップ・ボードにコピーしている文字列をキャレット位置に 挿入モードの場合
		は挿入し、上書きモードの場合は上書きします。
		ただし、クリップ・ボードの内容を文字列として認識できない場合は無効となります。
枹	ē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。
利	多動	指定した行へキャレットを移動するため、指定行へのジャンプ ダイアログをオープン
		します。
:-	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。
~``	ジャンプ前の位置へ戻る	[関数ヘジャンプ]を実行する前の位置へ戻ります。
月	関数ヘジャンプ	選択している文字列、またはキャレット位置の単語を関数と判断し、該当する関数
		ヘジャンプします(「(7)関数ヘジャンプする」参照)。
5	マグ・ジャンプ	キャレット位置の行にファイル名/行/桁の情報がある場合、該当するファイルの
		該当行/該当桁ヘジャンプします(「(g) タグ・ジャンプ」参照)。
11	ブックマーク	ブックマークに関するカスケード・メニューを表示します(「(9) ブックマークを登
		録する」参照)。
	ブックマークの挿入/削除	キャレット位置の行にブックマークを登録します。
		すでにブックマークを登録している場合は、そのブックマークを削除します。
	次のブックマーク	アクティブなエディタ パネルにおいて、前に登録したブックマーク位置にキャレッ
		トを移動します。
	前のブックマーク	アクティブなエディタ パネルにおいて、次に登録したブックマーク位置にキャレッ
		トを移動します。
	すべてのブックマークの削除	アクティブなエディタ パネルにおいて、登録しているブックマークをすべて削除し
		ます。
	ブックマークをリスト表示する	登録しているブックマークをリスト表示するブックマーク ダイアログをオープンし
		ます。
JuF	高度な設定	エディタ パネルに関する高度な操作を行うためのカスケード・メニューを表示しま
		す。
	行インデントを増やす	現在カーソルのある行のインデントをタブ1個分増やします。
	行インデントを減らす	現在カーソルのある行のインデントをタブ1個分減らします。
	行コメントを削除する	現在カーソルのある行の先頭から、言語に応じた行コメントの区切り記号の最初の
		セットを削除します。現在のソース・ファイルが行コメントの区切り記号が指定され
		ている言語を使用している場合のみ使用できます。
	行コメントを付ける	現在カーソルのある行の先頭に、言語に応じた行コメントの区切り記号を設定します。
		現在のソース・ファイルが行コメントの区切り記号が指定されている言語を使用して
		いる場合のみ使用できます。
	タブをスペースに変換する	現在カーソルのある行のすべてのタブをスペースに変換します。
	スペースをタブに変換する	現在カーソルのある行の連続したスペースの一組をタブに変換します。
		ただし、そのスペースの各組がタブ1個以上の幅に等しい場合に限ります。
	選択行をタブ化する	現在の行をタブ化します。行の先頭にある(テキストの前の)すべてのスペースを可
		能な限りタブに変換します。
	選択行を非タブ化する	現在の行を非タブ化します。行の先頭にある(テキストの前の)すべてのタブをス
		ペースに変換します。



大文字にする	選択しているすべての文字を大文字に変換します。
小文字にする	選択しているすべての文字を小文字に変換します。
大文字/小文字を切り替	選択しているすべての文字を、大文字または小文字に切り替えます。
える	
先頭を大文字にする	選択しているすべての単語の先頭文字を大文字に変換します。
前後の空白を削除する	カーソル位置の前後にある余分な空白を削除し、空白文字を1個だけ残します。カー
	ソルが単語内にある場合、または前後に空白文字がない場合、何も行いません。
末尾の空白を削除する	カーソルのある行で、最後の非空白文字の後にある空白を削除します。
行を削除する	現在カーソルのある行を完全に削除します。
行をコピーする	現在カーソルのある行をコピーして、その直後に挿入します。
空白行を削除する	カーソルのある行が空である場合、または空白文字しかない場合、その行を削除し
	ます。

(6) 文字列エリア(デバッグ・ツールと接続中の場合)

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)。
	ただし、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無効となります。
解析グラフに登録	選択している文字列、またはキャレット位置の単語を変数として、解析ツールの解
	析グラフ パネルに登録します。
	なお、既にすべてのチャネルに変数が登録されている場合は、メッセージを表示し
	て登録は行いません。
	ただし、アクティブ・プロジェクトが解析ツールのプラグインに対応していない場
	合は無効となります。
アクション・イベントの登	キャレット位置の行に対応するアドレスにアクション・イベントを設定するため
録…	アクション・イベント ダイアログをオープンします ^{注 1} 。
	ただし、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無効となります。
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。
	何も選択されていない場合は、その行を切り取ります。
	ただし、混合表示モードの場合は無効となります。
コピー	選択範囲の文字列をクリップ・ボードにコピーします。
	何も選択されていない場合は、その行をコピーします。
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、挿入モードの場
	合は挿入し、上書きモードの場合は上書きします。
	ただし、クリップ・ボードの内容を文字列として認識できない場合は無効となります。
	ただし、混合表示モードの場合は無効となります。
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。
移動	指定した行へキャレットを移動するため、指定行へのジャンプ ダイアログをオープン
	します。
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。
	ただし、混合表示モードの場合は無効となります。

ジャンプ前の位置へ戻る		[関数ヘジャンプ]を実行する前の位置へ戻ります。
		ただし、混合表示モードの場合は無効となります。
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 ^{注 1} 。
		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。
F	C 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 ^{注1} 。
		ただし、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、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。
	関数ヘジャンプ	選択している文字列、またはキャレット位置の単語を関数と判断し、該当する関数
		ヘジャンプします(「(7)関数ヘジャンプする」参照)。
!	ヌ グ・ジャンプ	キャレット位置の行にファイル名/行/桁の情報がある場合、該当するファイルの
		該当行/該当桁ヘジャンプします(「(g) タグ・ジャンプ」参照)。
ì	逆アセンブルヘジャンプ	キャレット位置の行に対応するアドレスを逆アセンブル パネルでオープンします ^{注1} 。
		ただし、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無効となります。
-	ブックマーク	ブックマークに関するカスケード・メニューを表示します(「(9) ブックマークを登
		録する」参照)。
		ただし、混合表示モードの場合は無効となります。
	ブックマークの挿入/削除	キャレット位置の行にブックマークを登録します。
		すでにブックマークを登録している場合は、そのブックマークを削除します。
	前のブックマーク	アクティブなエディタ パネルにおいて、次に登録したブックマーク位置にキャレッ
		トを移動します。
	次のブックマーク	アクティブなエディタ パネルにおいて、前に登録したブックマーク位置にキャレッ
		トを移動します。
	すべてのブックマークの削除	アクティブなエディタ パネルにおいて、登録しているブックマークをすべて削除し
		ます。
	ブックマークをリスト表示する	登録しているブックマークをリスト表示するブックマーク ダイアログをオープンし
		ます。
	「「」」である。	エディタ パネルに関する高度な操作を行うためのカスケード・メニューを表示しま
		₫.
		ただし、混合表示モードの場合は無効となります。
	行インデントを増やす	現在カーソルのある行のインデントをタブ1個分増やします。
	行インデントを減らす	現在カーソルのある行のインデントをタブ1個分減らします。
	行コメントを削除する	現在カーソルのある行の先頭から、言語に応じた行っメントの区切り記号の最初の
		セットを削除します。現在のソース・ファイルが行コメントの区切り記号が指定さ
		れている言語を使用している場合のみ使用できます。
	ーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーー	現在カーソルのある行の先頭に「言語に応じた行コメントの区切り記号を設定」す
		す。現在のソース・ファイルが行コメントの区切り記号が指定されている言語を使
		用している場合のみ使用できます。
	ー タブをスペースに変換する	現在カーソルのある行のすべてのタブをスペースに変換します。
	フペーフたクゴに赤海ナフ	
	へい―へとダ ノに変換する	- 現在カーフルののつけの運転しに入へ一人の一組をダブに変換します。ただし、そのスペースの久組がタゴ 1 個以上の幅に笑」 い場合に阻止ます
		のハ ・ への百幅ガ ノ 」 画以上の幅に守しい物口に限ります。



選択行をタブ化する	現在の行をタブ化します。行の先頭にある(テキストの前の)すべてのスペースを 可能な限りタブに変換します。
選択行を非々ブ化する	明在の行を非々ブ化します。行の失頭にある(テキストの前の)すべてのタブをス
	ペースに変換します。
大文字にする	選択しているすべての文字を大文字に変換します。
小文字にする	選択しているすべての文字を小文字に変換します。
大文字/小文字を切り替える	選択しているすべての文字を、大文字または小文字に切り替えます。
先頭を大文字にする	選択しているすべての単語の先頭文字を大文字に変換します。
前後の空白を削除する	カーソル位置の前後にある余分な空白を削除し、空白文字を1個だけ残します。
	カーソルが単語内にある場合、または前後に空白文字がない場合、何も行いません。
末尾の空白を削除する	カーソルのある行で、最後の非空白文字の後にある空白を削除します。
行を削除する	現在カーソルのある行を完全に削除します。
行をコピーする	現在カーソルのある行をコピーして、その直後に挿入します。
空白行を削除する	カーソルのある行が空である場合、または空白文字しかない場合、その行を削除し
	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。
ハード・ブレークの設定	キャレット位置の行にブレークポイント(ハードウェア・ブレーク・イベント)を
	設定します(「2.9.3 任意の場所で停止する(ブレークポイント)」参照) ^{注1} 。
ソフト・ブレークの設定	キャレット位置の行にブレークポイント(ソフトウエア・ブレーク・イベント)を
[Full-spec emulator]	設定します(「2.9.3 任意の場所で停止する(ブレークポイント)」参照) ^{注1} 。
[E1] [E20]	
読み込みブレークを設定	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)///0 レンスタに、リート・アクセスのフレーク・イベントを設定します(「2.9.4 任音の場所で停止する(ブレーク・イベン
書き込みブレークを設定	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、ライト・アクセスのブレー
	ク・イベントを設定します(「2.9.4 任意の場所で停止する(ブレーク・イベン
	ト)」参照)。
読み書きブレークを設定	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)///O レジスタに、リード/ライト・アクセスの
	ノレーク・1 ヘノトを設定します (12.9.4 仕息の場所で停止する (ノレーク・1 ベント) 1 参昭)
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。
 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。
トレース開始の設定	キャレット位置の行が実行された際に 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. 12. 3 任意区間の実行
	履歴を収集する」参照)注1,2。
トレース終了の設定	キャレット位置の行が実行された際に、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(「2.12.3 任意区間の実行
	履歴を収集する」参照) 注 1, 2。

-		
ſ	値をトレースに記録(読	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
	み込み時)	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(「2. │
		12.4 条件を満たしたときのみの実行履歴を収集する【シミュレータ】」参照)。
	値をトレースに記録(書	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
	き込み時)	│ 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(「2. │
l		12.4 条件を満たしたときのみの実行履歴を収集する【シミュレータ】」参照)。
	 値をトレースに記録 (読み	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
	書き時)	┃ 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(「2.12.4 条件を満たしたときのみの実行履歴を収集する【シミュレータ】」
		参照)。
l	 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・データの最終行を表示します。
I	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	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
		(「2.13.2 任意区間の実行時間を計測する【シミュレータ】」参照)。
٦	 実行時にタイマ開始	キャレット位置の行が実行された際に、プログラムの実行時間の計測を開始するタ
l		イマ開始イベントを設定します ^{注 1, 3} 。
	タイマnに設定	タイマ開始イベントを設定するチャネル(n:1~8)を指定します。
ſ	実行時にタイマ終了	キャレット位置の行が実行された際に、プログラムの実行時間の計測を終了するタ
l		イマ終了イベントを設定します ^{注1,3} 。
-	タイマ n に設定	タイマ開始イベントを設定するチャネル(n:1~8)を指定します。
ſ	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		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。
[]	シミュレータ】	
名	↓前を付けて 混合表示 を保	現在編集しているエディタ パネルの内容を新規保存するために、名前を付けて保存
存	<u></u>	ダイアログをオープンします。
		ただし、混合表示モードの場合のみ有効となります。

- **注1.** ダウンロードされているロード・モジュールの更新日時よりオープンしているソース・ファイルの更新日が新しい場合、この項目を選択するとメッセージを表示します。
 - 【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。
 - 【シミュレータ】 プロパティ パネル上の [タイマ]【シミュレータ】カテゴリ内 [タイマ機能を使用する] プロパティ の設定を自動的に [はい] にします。

メモリ パネル

メモリの内容の表示、変更を行います(「2.10.1 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。

- 注意 データフラッシュ領域のメモリ値を変更すると、選択しているマイクロコントローラの種別により、CPU リ セットを発生する場合があります。
- 備考1. ツールバーの 100% ▼, または [Ctrl] キーを押下しながらマウス・ホイールを前後方に動かすこ とにより、本パネルの表示を拡大/縮小することができます。
 - ツールバーの [表示] → ゴ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―25 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(「2.18.2 シンボル名の入力補完機能」参照)。
 - 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		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		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、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

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) 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・		
	背景色	標準色	メモリに書き込まれ	.ます)	
<u>00</u>	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	
(下線)	背景色	標準色	できます)		
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	
	背景色	クリーム	ッールバーの 🛞 ボ	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。	
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			
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	
	背景色	薄緑		リ値のアクセス状態	
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	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・マッピングされていない領域		
	背景色	標準色			
	文字色	グレー	書き換え不可能領域(I/O レジスタ領域 /I/O 保護領域など)。またはメモリ値の		
	背景色	標準色	取得に失敗した場合		
**	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		
	背景色	標準色	合、またはメモリ値	合、またはメモリ値の取得に失敗した場合	

- 注 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。



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、「(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、「(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(「(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(「(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、「(7)メモリの表示内容を保存する」を参照してください。

(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/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	
Float	単精度浮	鳨小数 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	倍精度浮颤	劻小数 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><単精度浮動小数点数>*I		浮動小数点数><単精度浮動小数点数>*I	

項目	表記	サイズ
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> <倍精度浮動小数点数> * I	
Float Imaginary	単精度浮動小数点数の虚数	32 ビット
	<単精度浮動小数点数> * I	
Double Imaginary	倍精度浮動小数点数の虚数	64 ビット
	<倍精度浮動小数点数>*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

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 浮動小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、文字列を検索するためのメモリ検索 ダイ アログをオープンします(「(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]

2	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。

	Hex	メモリ値を 16 進数で表示します(デフォルト)。
	SDec	メモリ値を符号付き 10 進数で表示します。
	Dec	メモリ値を符号なし 10 進数で表示します。
	Oct	メモリ値を8進数で表示します。
	Bin	メモリ値を2進数で表示します。
+	^ト 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	メモリ値を4ビット幅で表示します。
	8	メモリ値を8ビット幅で表示します(デフォルト)。
	16	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。
	32	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。
	64	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。
Г	ニンコード	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。
	[paA]	文字列を ASCII コードで表示します(デフォルト)。
	[surs]	文字列を Shift_JIS コードで表示します。
		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。
	16	文字列を UTF-16 コードで表示します。
	Flo	文字列を単精度浮動小数点数値で表示します。
	Dbl	文字列を倍精度浮動小数点数値で表示します。
	FIP	文字列を単精度浮動小数点数の複素数で表示します。
	Dbl	文字列を倍精度浮動小数点数の複素数で表示します。
	Fig	文字列を単精度浮動小数点数の虚数で表示します。
	Dbl	文字列を倍精度浮動小数点数の虚数で表示します。
TA.	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 ダイアログをオープンします。
	 表示アドレス・オフ	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
	セット値を設定	設定 ダイアログをオープンします。

[[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	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
保存	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]

		選択しているメモリ値に定義されているシンボルをウォッチ パネル(ウォッチ 1)に登録します	
		」	
		シオッテスとして豆球される际は変数石として豆球されるため。 スコーンにより衣小さ	
		れるシン小ル石は変化しまり。	
		たたし、キャレット位置のメモリ値に対応するアトレスにシンホルが定義されていない	
		場合は無効となります(I(f)ウォッチ式の登録」参照)。	
杚	〕 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。	
衫	〕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。	
貝	ちり付け	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	
THA	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。	
	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。	



÷	ナイズ表記	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。
	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。
_	ニンコード	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。
	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。
	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。
7	हैरू	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、スクロール範囲設定 ダイアログをオープンします。
	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 ダイアログをオープンしま
		<i>च</i> .
	表示アドレス・オフセット値	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
	を設定	セット設定 ダイアログをオープンします。
5	淮調表 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)。
L		ただし、プログラム実行中は無効となります。
1	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
L		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(「2.8.3 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】(「2.14 カバレッジの測定【シミュレータ】」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。

混合表示モードを選択することにより、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(「2.8.3 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。
 - [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の画像イメージ を印刷することができます。
 - **3.** ツールバーの 100% ▼, または [Ctrl] キーを押下しながらマウス・ホイールを前後方に動かすことにより、本パネルの表示を拡大/縮小することができます。

34:	void main() {		
	_main:		
00000394	a515	br	main+0x24
36:	func(); bfff48ff	iarl	func. Ip
37:	sfunc();	2	
0000039a	80ff2200	jarl	_sfunc, lp
38:	nosource();		-
0000039e	80ff0e13	jarl	_nosource, lp
39:	parent_num_2();		
000003a2	80112a00	jarl	_parent_num_2, I
40:	parent_num_3();	lorl	porent pum 9 1
41.	func2():	Jari	_parent_num_o, n
S 000003aa	bfffc6ff	jarl	func2, lp
42:	<pre>sub02_main();</pre>	20 10	
000003ae	80ffba00	jarl	_sub02_main, lp
<			

図 A—26 逆アセンブル パネル(混合表示モードの場合)





図 A—27 逆アセンブル パネル(逆アセンブル表示モードの場合)

図 A-28 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】



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・マークを表示します。また、設定したブレー クポイントの詳細情報をイベント パネルに反映します。

なお、すでにいずれかのイベント・マークを表示している箇所で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。

なお、ブレークポイントの設定方法についての詳細は、「2.9.3 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。
詳細設定情報表示	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。

- 備考 実行停止時のデバッグ対象コアと現在のデバッグ対象コアが異なる場合など,カレント PC 位置が無 効状態となった場合は,カレント PC マークは ⇒ から ⇒ へ変化します。
- (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: < <i>ロード・モジュール名</i> > ^注 \$< <i>ラベル 名</i> >+< <i>オフセット値</i> >
	例 1: test1.abs\$main + 0x10
	例 2: subfunction + 0x20
	<u></u>
ソース行番号	形式 : 〈 <i>ロード・モジュール名</i> 〉 ^注 \$ 〈 <i>ファイル名</i> 〉 # 〈 <i>行番号</i> 〉
ソース行番号	形式: <i><ロード・モジュール名>^注 \$ <ファイル名> # <行番号</i> > 例 1: test1.abs\$main.c#40

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。

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。



⁽³⁾ 逆アセンブル・エリア

	void fun	c2()		
	3	,		
- ~ u 4	= 1			
フヘル1	T→→_tunc2:			
カレント PC 彳	ד→	8515	br	_func2+0x20
	static	global this	variable name is ve	ry_long++;
ブレークポイント設定彳	उं—▶+2	24571980	ld.w 🗌 🗌	-0x7fe8[gp], r10
	+6	4152	add	0x1, r10
	+8	64571980	st.w	r10, -0x7fe8[gp]
ソース・テキスト行	nosour	ce_variable+	+;	
	- (+c	245f1580	(Id.w	-0x7fec[gp], r11
逆アセンブル結果行・	+10	415a	add	0x1, r11
	+12	645f1580	st.w	r11, -0x7fec[gp]
				Ť
	オフセット値	」 命令コード		

図 A-29 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	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
結果行		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。
 - ツールバーの ボネシ (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)。
 - 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 説 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

表示している命令/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、「2.8.3 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] などを選択することにより、現在キャレットのあるアドレス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と、対応するイベント・マークをイベント・エリアに表示します。また、設定した イベントの詳細情報をイベント パネルに反映します。

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。

なお、各種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。

- 「2.9.4 任意の場所で停止する (ブレーク・イベント)」
- 「2.9.5 変数 /I/O レジスタへのアクセスで停止する」
- 「2.12.3 任意区間の実行履歴を収集する」
- 「2.12.4 条件を満たしたときのみの実行履歴を収集する【シミュレータ】」
- 「2.13.2 任意区間の実行時間を計測する【シミュレータ】」
- 「2.15.1 printf を挿入する」
- **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

表示している 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、「(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 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 [メモリヘジャンプ] 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】

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、「2.14 カバレッジの測定【シミュレータ】」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、「(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。

[ツールバー]

	3	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。
	50	このパネルの表示モードとして、混合表示モード(デフォルト)と逆アセンブル表示モードを切り替えま
		す(「(1)表示モードを変更する」参照)。
	2	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。
	100C	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。
	120	ご ボタンで移動する直前の位置(アドレス)へ移動します。
킢	ē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。
	1	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。
	1980 2000	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。
		レジスタ名を機能名称で表示します(デフォルト)。
	R	レジスタ名を絶対名称で表示します。
	*	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	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。
置换	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	指定したアドレスヘ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]

- (1) 逆アセンブル・エリア/アドレス・エリア
- (2) イベント・エリア [Full-spec emulator] [E1] [E20]

(1) 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。
<i>アドレス</i> 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。

	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(ハードウェア・ブレーク・イベン
		ト)を設定します(「2.9.3 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	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
	[Full-spec emulator]	ト)を設定します(「2.9.3 任意の場所で停止する(ブレークポイント)」参照)。
	【E1】【E20】	
	読み込みブレークを設定	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・アクセスのブレー
		ク・イベントを設定します(「(1) 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	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(「(1) 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	
		 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(「(1) 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。
Γ	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。
	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. 12. 3 任意
		区間の実行履歴を収集する」参照) ^{注1} 。
	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(「2.12.3 任意
		区間の実行履歴を収集する」参照) ^{注1} 。
	値をトレースに記録(読み	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
	込み時)	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(「2.
		12.4 条件を満たしたときのみの実行履歴を収集する【シミュレータ】」参照)。
	値をトレースに記録(書き	
	込み時)	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(「2.
		12.4 条件を満たしたときのみの実行履歴を収集する【シミュレータ】」参照)。
	値をトレースに記録(読み	
	書き時)	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
		定します(「2.12.4 条件を満たしたときのみの実行履歴を収集する【シミュレー
		タ】」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。
Γ	 タイマ設定	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
ו		(「2.13.2 任意区間の実行時間を計測する【シミュレータ】」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た。
	タイマnに設定	タイマ開始イベントを設定するチャネル(n:1 ~ 8)を指定します。
	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 ^{注2} 。
	タイマnに設定	タイマ開始イベントを設定するチャネル(n:1 ~ 8)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。
7	りバレッジ情報をクリア	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。
C	シミュレータ】	
Ļ	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。
WHI I	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	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)。
	示	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。
	レジスタを機能名称で表示	レジスタ名を機能名称で表示します(デフォルト)。
	レジスタを絶対名称で表示	レジスタ名を絶対名称で表示します。
	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		このパネルの表示モードとして、混合表示モード(デフォルト)と逆アセンブル表示
		モードを切り替えます(「(1)表示モードを変更する」参照)。
	ノースヘジャンプ	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
L		ディタ パネルがオープンします。
メモリヘジャンプ		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. 【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。

2. 【シミュレータ】

プロパティ パネル上の [タイマ] 【シミュレータ】カテゴリ内 [タイマ機能を使用する] プロパティ の設定を自動的に [はい] にします。



ハードウェア・ブレークを優	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
先する	イントとします(プロパティ パネル上の[ブレーク]【Full-spec emulator】【E1】
	【E20】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	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
先する	イントとします(プロパティ パネル上の[ブレーク]【Full-spec emulator】【E1】
	【E20】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)。

(2) イベント・エリア [Full-spec emulator] [E1] [E20]



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,および値の変更を行います(「2.10.2 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。

- 注意 選択しているマイクロコントローラがマルチコア対応版の場合, コア(PEn)の選択を切り替えることにより, 選択した PEn に対応した内容の表示/値の変更を行います(「2.7 コア(PE)の選択」参照)。
- 備考1. ツールバーの 100%
 ↓ または [Ctrl] キーを押下しながらマウス・ホイールを前後方に動かすことにより、本パネルの表示を拡大/縮小することができます。
 - 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。



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。

H	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。
0	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。
111	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。
11	レジスタ名(レジスタの部分を表す名称)を示します。

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、「(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。
 - 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。
- (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(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
	背景色	標準色	ゲット・メモリに書き込まれます)
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
	背景色	クリーム	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。

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、「(4) CPU レジスタの表示内容を保存する」を参照して ください。

[ツールバー]

a	デバッグ・ツールから最新の情報を取得し、表示を再新します
	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。
auto	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)。
Hex	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。
SDec	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。
UDec	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。
Oct	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。
Bin	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。
Asc	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。
FIO	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。
Dbl	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。
He	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・	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
データを保存 …	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。
⊐ ピ –		
		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。
	符号無し 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。



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(「2.10.3 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 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 VO 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 VO 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]を選択することで、1 度だけ値の読み込みが可能です。 なお、いったん [値を強制読み込み]を選択すると "?"表示ではなくなるため、読み込み保護対象の VO レ ジスタであることがわからなくなるため注意が必要です。
 - 2. 選択しているマイクロコントローラがマルチコア対応版の場合、コア(PEn)の選択を切り替えることにより、選択した PEn に対応した内容の表示/値の変更を行います(「2.7 コア(PE)の選択」参照)。
- 備考1. ツールバーの 100% ▼, または [Ctrl] キーを押下しながらマウス・ホイールを前後方に動かすことにより、本パネルの表示を拡大/縮小することができます。
 - 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 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)。
۶	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)。
- (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。

\cong	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 1 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。
 - 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、「(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。
 - 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。
- (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(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)		オルト)	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
	背景色	標準色	ゲット・メモリに書き込まれます)
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
	背景色	クリーム	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。
?	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、「(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、「(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> < 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
アクセス可能サイズ	アクセス可能なすべてのサイズを、ビット単位で小さい順に","で区切り列挙します(1	
	~ 32 ビット)	٥
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

「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"OxFF40"に割り当てられている

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]

2	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。
1	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。
	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。
*	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
*	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。



×		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)。
7	長記	値の表示形式を変更する次のボタンを表示します。
	Hex	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)。
	SDec	選択している項目の値を符号付き 10 進数で表示します。
	Dec	選択している項目の値を符号なし 10 進数で表示します。
	Oct	選択している項目の値を8進数で表示します。
	Bin	選択している項目の値を2進数で表示します。
	Asc	選択している項目の値を ASCII コードで表示します。
1	Heg	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 レジスタヘ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。	
値を強制読る	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。	
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。	
表示色をリー	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(「2.10.5 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。
 - 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。
 - 3. 選択しているマイクロコントローラがマルチコア対応版の場合、コア(PEn)の選択を切り替えることにより、選択した PEn に対応した内容の表示/値の変更を行います(「2.7 コア(PE)の選択」参照)。
- 備考1. ツールバーの 100% ▼, または [Ctrl] キーを押下しながらマウス・ホイールを前後方に動かすことにより、本パネルの表示を拡大/縮小することができます。
 - 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。



- 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,および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。

۲	変数を示します。 Auto 変数、内部ス 配列、ポインタ型 先頭に"+"マーク マークに変化)。	.タティック変数、Register 変数の表示も行います ^注 。 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。 7 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"-"
	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。	
ų.	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、「(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 [メモリヘジャンプ]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)。

(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(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	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
	背景色	標準色	ゲット・メモリに書き込まれます)
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
	背景色	クリーム	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。
?	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) 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、「(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]

	22	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。
Wei	表記	値の表示形式を変更する次のボタンを表示します。
	auto	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。
	Hex	このパネル上の値を 16 進数で表示します。
	Dec	このパネル上の値を 10 進数で表示します。
	Oct	このパネル上の値を8進数で表示します。
	Bin	このパネル上の値を2進数で表示します。
	[Dec]	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)。
	[Hex]	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。
	Flo	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。
	Dbl	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。
	Het	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。
. 1	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。
	Asc	文字列変数を ASCII コードで表示します(デフォルト)。
		文字列変数を Shift_JIS コードで表示します。
	EUC	文字列変数を EUC-JP コードで表示します。
	UTF	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	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
タを保存	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) 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]

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。	



1	6 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。	
,	・ モリヘジャンプ	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(「2.10.6 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 選択しているマイクロコントローラがマルチコア対応版の場合, コア(PEn)の選択を切り替えることにより, 選択した PEn に対応した内容の表示/値の変更を行います(「2.7 コア(PE)の選択」参照)。
- 備考1. ツールバーの 100% ▼, または [Ctrl] キーを押下しながらマウス・ホイールを前後方に動かすこ とにより、本パネルの表示を拡大/縮小することができます。
 - 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 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。

H	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。
4 9	ウォッチ式が関数であることを示します。
123	ウォッチ式が即値であることを示します。
f(∞)	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。
111	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 マンのクリック、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。

RENESAS

- 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。
 - 1 つのウォッチ パネルにおいて、カテゴリは 1500 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,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	
構造体/共用体	構造体/共用体の全メンバ	
レジスタ	レジスタを構成するビット/ビット列の名称	
	例)ECR レジスタの場合	
	FECC レジスタ	
	EICC 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。

- 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。

RENESAS

- 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
- 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-2 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 言語変数 /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 言語変数 /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
ローカル変数 パネル	C 言語変数(ローカル変数)
IOR パネル	I/O レジスタ ^注

注 自動的にスコープ指定がウォッチ式に付与されます。

- 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 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,キーボート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。

ウォッチ式の入力形式については、「(b) ウォッチ式と演算子」を参照してください。

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。

注意 1. ロード・モジュール名, またはファイル名に空白や次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

2. 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))。



スコープ指定	ロード・モ ジュール名	ソース・ファ イル名	関数名	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

表 A-3 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

注 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。

表 A—4 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	ソース・	関数名	変数名	検索対象
	ジュール名	ファイル名			
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 1}
prog\$file#var	prog	file	グローバル	var	ファイル内スタ ティック変数
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1}
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2

- **注1.** 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。
 - 2. 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—5 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—6 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
P0:IOR	P0



スコープ指定	I/O レジスタ名
P0	P0

- 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(「2.18.2 シンボル名の入力補完機能」参照)。
 - 2. 即値はアドレスとして扱われます。また、即値に演算子を使用することはできません。
 - 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はできません。
 - G名の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言語変数> CPU レジスタ> I/O レジスタ
 - 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 値のスコープを基にシンボルを解決し、値を表示します。
 - 6. 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。
 - **7.** 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。

- 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 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 3000 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)。
 - 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。
 - 2. ウォッチ式は、同名を複数登録することができます。
 - 登録した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(「(8) ウォッチ式をエクスポート/インポート する」参照)。

(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。

なお、各種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。

- 「2.9.5 変数 /I/O レジスタへのアクセスで停止する」
- 「2.12.4 条件を満たしたときのみの実行履歴を収集する【シミュレータ】」
- (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 [メモリヘジャンプ]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。

(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。



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(サイズが4バイトの場合)値に続き"()"内に 16 進数値を併記
double, long double	Double(サイズが8バイトの場合)値に続き"()"内 に 16 進数値を併記
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

表 A---7 ウォッチ式の表示形式(デフォルト)

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(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)		ルト)	説明
0×0	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
	背景色	標準色	ゲット・メモリに書き込まれます)
0×0	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
	背景色	クリーム	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。
?	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
	背景色	標準色	できなかった場合(読み込み保護対象の I/O レジスタ ^注 や、変数がスコープを外 れた場合など)

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]を選択することで、1 度だけ値の読み込みが可能です。

- 備考1. 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。
 - 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、「(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、「(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。

ウォッチ式の値の保存方法については、「(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} >)



ウォッチ式	表示形式	
単独の I/O レジスタ	< I/O レジスタの種類>(<アクセス属性> <アクセス可能サイズ> <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
	アクセス可能サイズ	アクセス可能なすべてのサイズを、ビット単位で小さい順に","
		で区切り列挙します(1 ~ 32 ビット)。
判別不能	?	
上記以外	< C <i>コンパイラの判定</i>	<i>『に従ったウォッチ式の型^{注 2} >(<サイズ^{注 1} ></i>)

- **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。
 - 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。
- (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]

22	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。
20	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 3000 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 1500 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。
auto	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—7 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)。
Hex	選択している項目の値を 16 進数で表示します。
SDec	選択している項目の値を符号付き 10 進数で表示します。
Dec	選択している項目の値を符号なし 10 進数で表示します。
Oct	選択している項目の値を8進数で表示します。
Bin	選択している項目の値を2進数で表示します。
Asc	選択している項目の値を ASCII コードで表示します。
FIO	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。
Dbl	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。
Hes	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	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
を保存	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]

7	⁷ 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(「(1) 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	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	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	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。		
ł	- 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		
		(「(1) ポイント・トレース・イベントを設定する」参照)。		

値をトレースに記録(読み込	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
み時) 	録するポイント・トレース・イベントを設定します。
値をトレースに記録(書き込	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
み時)	録するポイント・トレース・イベントを設定します。
値をトレースに記録(読み書	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
き時)	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。
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。
 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 3000 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 1500 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(「(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—7 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。		
	符号無し 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—7 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		
	配列のインデックスを 10 進 ま 記	すべての配列のインデックスを 10 進数で表示します。 		
	記列の1 フテックスを 16 進 表記	り へ この配列のイ ンテックスを 16 進数 ご衣示しまり。		
Г	ニンコード	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。		
,	・ モリヘジャンプ	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)。		
쿢	長示色 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		
Í		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。		

RENESAS

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(「2.11.1 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 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。

- 2. 選択しているマイクロコントローラがマルチコア対応版の場合、コア(PEn)の選択を切り替えることにより、選択した PEn に対応した内容の表示を行います(「2.7 コア(PE)の選択」参照)。
- 備考 ツールバーの 100% ▼, または [Ctrl] キーを押下しながらマウス・ホイールを前後方に動かすこと により、本パネルの表示を拡大/縮小することができます。

_	コール	· スタック	8
[ツールバー]-	2	表記(N) + エンコード(E) + 😵 💫	
L	深さ	コール・スタック	
	0	main0[sample.abs\$main.c#71]	
	<		>
	(1)		

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

R20UT2685JJ0100 Rev.1.00 2013.09.01



[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 シノビ 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	<i><関数</i> > (<i><引数</i> > = < <i>引数値</i> ^注 > ,…)[<i><モジュール・ファイル名</i> > \$ <
- モジュール・ファイル名を表示する	<i>ファイル名</i> ># <i><行番号</i> >](デフォルト)
- 引数を表示する	<i><関数</i> > (<i><引数</i> > = <i><引数値^注> ,…)[<i><ファイル名</i>> # <i><行番号</i>>]</i>
- モジュール・ファイル名を表示しない	
- 引数を表示しない	< <i>関数</i> > ()[< <i>モジュール・ファイル名</i> >\$ <i><ファイル名></i> # <i><行番号</i> >]
- モジュール・ファイル名を表示する	
- 引数を表示しない	< <i>関数</i> > ()[<i><ファイル名</i> > # <i><行番号</i> >]
- モジュール・ファイル名を表示しない	

- 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(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[逆アセンブルヘジャンプ]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)。

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、「(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。

[ツールバー]

2	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。
auto	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)。
Hex	このパネル上の値を 16 進数で表示します。
Dec	このパネル上の値を 10 進数で表示します。
Oct	このパネル上の値を8進数で表示します。
Bin	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。
Asc	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)。
	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	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	このパネル上の文字列変数を UTF-8 コードで表示します。
	このパネル上の文字列変数を UTF-16 コードで表示します。
V	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	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)
存	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・	- │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
データを保存 …	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					
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。				
 逆アセンブルヘジャンプ	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。				
ソースヘジャンプ	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(「2.12 実行履歴の収集」参照)。

トレース・データは、デフォルトでソース・テキストと逆アセンブル・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 【Full-spec emulator】【E1】【E20】

プロパティ パネルの [デバッグ・ツール設定] タブの [トレース] カテゴリ内 [トレースの取得対象設定] プロパティにおいて [全てのコア] を選択してトレース・データを収集した場合では、コア (PEn)の選択を切り替えることにより、選択した PEn に対応したトレース・データの表示を行います (「2.7 コア (PE)の選択」参照)。

- 備考1. ツールバーの 100% ▼, または [Ctrl] キーを押下しながらマウス・ホイールを前後方に動かすことにより、本パネルの表示を拡大/縮小することができます。
 - **2.** 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。

番写	时間 (h:min:s,ms,µs,ns)	時間(クロック)	行番号/アドレス	リリース/速ア	センブル	アドレス データ
48	00h00min00s000ms000µs500ns	5		A second		0x01000438 R 107433951
49	00h00min00s000ms000µs100ns	1	0×010003f6	+118	st.w r14, 0x0[r12]	
50	OOhOOminOOsOOOmsOOOµsOOOns	0				0xfedf8008 W 107433951
51			0x010003fa	+122	add 0×4, r10	191
51-1			0x010003fc	+124	add 0x4, r12	
51-2			0x010003fe	+126	cmp r10, r11	
51-3			0×01000400	+128	bhINITSCT_RH+0x6e	
51-4			0×01000402	+130	addi -0x2, r11, r13	
51-5			0×01000406	+134	cmp r10, r13	
51-6			0×01000408	+136	bcINITSCT_RH+0×98	
51-7	00h00min00s000ms000µs900ns	9	0×01000418	+152	cmp r10, r11	
52			0x0100041a	+154	bnhINITSCT_RH+0xaa	
52-1	00h00min00s000ms000µs400ns	4	0x0100042a	+170	add Oxc, r6	
53			0x0100042c	+172	brINITSCT_RH+0×4a	
53-1	00h00min00s000ms000µs400ns	4	0x010003ca	+74	cmp r6, r7	
54			0x010003cc	+76	bnhINITSCT_RH+0xae	
54-1	00h00min00s000ms000µs300ns	3	0x0100042e	+174	jmp [lp]	
				PE1:		
55			cstart.asm#84	MOV 0×1,r6		
55	00h00min00s000ms000µs400ns	4	0x0100031c	+78	mov 0x1, r6	
CON.			1	ſ		11 11

図 A—35	トレース パネル	[Full-spec emulator]	【E1】	[E20]



番号	ターゲット	時間 (h:min:s,ms,µs,ns)	時間(クロック)	行番号/アドレス	バイ	(プライン	ソース/逆7	Pセンブル	アドレス	デ
0				vecttbl.asm#13			jr32_start	; RESET		
0	CPU1	00h00min00s000ms000µs000ns	0	0×01000000				jrstart		
1	CPU1	00h00min00s000ms000µs000ns	0							
2	CPU1				F	DE				
3				boot.asm#30			stsr0, r10,	2; get HTCFG0		
3	CPU1	00h00min00s000ms000µs084ns	27	0×01000a00			+0	stsr 0x0, r10, 0x2		
4	CPU1					DE F				
5				boot.asm#31			shr16, r10;	get PEID		
5	CPU1	00h00min00s000ms000µs056ns	18	0x01000a04			+4	shr 0x10, r10		
6	CPU1					DEF				
7				boot.asm#33			sh12, r10			
7	CPU1	00h00min00s000ms000µs003ns	1	0×01000a06			+6	shl 0x2, r10		
8	CPU1					DE				
9				boot.asm#34			mov#_entry_	table, r11		
9	CPU1	00h00min00s000ms000µs003ns	1	0×01000a08			+8	mov 0x1000b20, r11		
10	CPU1	00h00min00s000ms000µs000ns	0							
11	CPU1					DE				
12				boot.asm#35			addr11, r10			
12	CPU1	00h00min00s000ms000µs003ns	1	0x01000a0e			+14	add r11, r10		
13	CPU1					FDE				
14				boot.asm#36			ld.w0[r10],	r10		
14	CPU1	00h00min00s000ms000µs003ns	1	0×01000a10			+16	ld.w 0x0[r10], r10		
15	CPU1					F DE				
16	CPU1								0x01000b24	4 R 1
							ハードウエ]	ア・ブレークにより停止		
	11		<u>г I</u>	I			11			
· · · ·			· · · · · · · · · · · · · · · · · · ·	/		1			<u> </u>	

図 A-36 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) [ターゲット] エリア【シミュレータ】トレースの対象となったコア名を表示します。

(3) [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 1. 【Full-spec emulator】【E1】【E20】

時間表示は相対時間となります。

- 【シミュレータ】
 時間表示を積算時間とするか差分時間とするかは、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] プロパティの設定に依存します。
- (4) [時間(クロック)] 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 CPU クロック数で表示します。

- 備考1. 【Full-spec emulator】【E1】【E20】 時間表示は差分 CPU クロック数となります。
 - 【シミュレータ】
 時間表示を積算 CPU クロック数とするか差分 CPU クロック数とするかは、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] プロパティの設定に依存します。
- (5) [行番号/アドレス] エリア

ソース・ファイルの行番号、またはアセンブル命令のアドレスを表示します。

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
ソース・テキスト	<ファイル名>#<行番号>
命令(逆アセンブル)	<アドレス>
上記以外	_

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
- オペランド・アクセス

- 無効フェッチ

(6) [パイプライン] エリア【シミュレータ】

パイプラインの実行状況を表示します。

文字数は 20 文字で、1 文字が1 クロック分のステージに相当します。各ステージは、そのステージが実行されたクロックに対する 20 の剰余の値を文字列のインデックスとして表示します。

各ステージを表す文字の意味は次のとおりです。

ステージ	文字
フェッチ	F
デコード	D
実行	E

- **例 1.** F: 10 クロック目, D: 11 クロック目, E: 13 クロック目の場合 表示: FD_E_____
 - F:18 クロック目、D:19 クロック目、E:20 クロック目の場合
 表示:E_____FD

備考 "_"は半角スペースを示します。

(7) [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-37 [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。
	なお、ソース行の実行時にアクセスされた変数 ^{注 1} /I/O レジスタの値が解析可能な場
	合は、その値をソース行に続き次の形式で表示します。
	- <<< 変数名 = 変数値 >>>
	- << <i o="" レジスタ値="" レジスタ名="I/O">>></i>
	例:a=b; << <a=5>>></a=5>
	また、ポイント・トレースの結果を表示する場合も同様の形式で表示します。
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 2} 。ニモニックは強調表示されます。
ブレーク要因	プログラムがブレークした要因を表示します。
【シミュレータ】	

- 注1. メモリへのアクセスが発生した場合、対象アドレスにシンボルが割り当たっている場合にかぎり、該 当シンボルを変数とみなして表示します。 ただし、4バイトまでの変数が対象となります。 なお、乗算などの記述が、標準ライブラリで処理されている場合、標準ライブラリで使用している SADDR 領域のラベルが表示される場合があります。
 - 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)/ポ イント・トレース結果/ブレーク要因を表示します(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/ブレーク要因を表 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)/ブレーク要因を表示します。 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 [逆アセンブルヘジャンプ] 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 パインドウ連動 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、「2.12.9 実行履歴の表示内容を保存する」を参照してください。

(8) [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。

(9) [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(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	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	1值	文字色	標準色	ライト・アクセス
		背景色	オレンジ	



表示例(デフォルト)			メモリ・アクセス種別
RW <i>データ値</i>	文字色	標準色	リードとライト・アクセス
	背景色	薄青	
VECT データ値	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]

2	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、トレーサ動作中は無効となります。			
1	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、トレーサ動作中は無効となります。			
۲	トレーサの動作を開始します。			
	現在、このパネルで表示している内容をクリアします。			
	ただし、トレーサ動作中は無効となります。			
۲	トレーサの動作を停止します。			
	新たに取得したトレース・データの内容に表示を更新します。			
	ただし、トレーサ停止中は無効となります。			
盟	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、トレーサ動作中は無効となります。			
Hex	このパネル上の値を 16 進数で表示します(デフォルト)。			
Dec	このパネル上の値を 10 進数で表示します。			
Oct	このパネル上の値を8進数で表示します。			
Bin	このパネル上の値を2進数で表示します。			
Ĵ	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。			
5	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。			
菌属	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、トレーサ動作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、トレーサ動作中無効となります。			
	表示モードをソース表示モードにします。			
	ただし、トレーサ動作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。
	ただし、トレーサ動作中は無効となります。
名前を付けてトレース・データ	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)
を保存	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)。
	- ただし、トレーサ動作中は無効となります。

[[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)。

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
	न्) _°
	ただし、トレーサ動作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。

[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、トレーサ動作中は無効となります。		
トレース開始	トレーサの動作を開始します(「(2)実行履歴の収集を再開する」参照)。		
	現在、このパネルで表示している内容をクリアします。		
	ただし、トレーサ動作中は無効となります。		
トレース停止	トレーサの動作を停止します(「(1)実行履歴の収集を一時的に停止する」参照)。		
	新たに取得したトレース・データの内容に表示を更新します。		
	ただし、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。		
	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		
	न)。		
	ただし、トレーサ動作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、トレーサ動作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、トレーサ動作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、トレーサ動作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、トレーサ動作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。
L	フインドウ連動	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	キャレット位置の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	キャレット位置の行に連動して逆アセンブル パネルをスクロールします。
ì	- 逆アセンブルヘジャンプ	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。
	ノースヘジャンプ	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。
;	イモリヘジャンプ	キャレット位置の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(「2.16 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、「2.16.6 イベント設定に関する留意事項」を参照してください。
 - 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します。
 - 3. ツールバーの 100% ▼, または [Ctrl] キーを押下しながらマウス・ホイールを前後方に動かすこ とにより、本パネルの表示を拡大/縮小することができます。
 - **4.** 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。



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

-【シミュレータ】

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

[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。

ハードウェア・ブレーク0001 \checkmark $(\dot{\mathbf{b}})$ (a)(c)

- **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. 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の 両方の設定が必要となります。
 - 2. Run-Break タイマ・イベントを無効状態/保留状態にすることはできません。
 - 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・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。

(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。

イベント種別	有効状態	無効状態	保留状態	備考
ハードウェア・ブレーク				ハードウェア・ブレーク・
				ポイントを含む
 ソフトウエア・ブレ ー ク	1°	ø	٩	ソフトウエア・ブレ ー ク・
				ポイントを含む
関数の先頭へのブレーク		-	۲	解析ツールにより設定可能
変数のアクセス・ブレーク			۲	なブレーク・イベント
無条件トレース	\$	*	_	-
Run-Break タイマ	\$	_	_	-
トレース	N	\$	39	イベント パネルでのみ表示
トレース開始	%	\$	39	エディタ パネル/逆アセン
トレース終了	1 2	*	29	ブル パネルでのみ表示
タイマ計測	19	100	Ø	イベント パネルでのみ表示
タイマ開始	19	100	Ø	エディタ パネル/逆アセン
タイマ終了	19	5	ø	ブル パネルでのみ表示
ポイント・トレース	°	**	Ť	_
Printf イベント	s v	š	S	_
上記イベントの複数設定	₽21	ば注2	┣注3	エディタ パネル/逆アセン
				ブル パネルでのみ表示

表A---8 イベント・マーク

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

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--9 イベント種別

イベント種別	説明
ハードウェア・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
(ブレーク ^{注1})	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。
	→「2.9.3 任意の場所で停止する(ブレークポイント)」参照
	→「2.9.4 任意の場所で停止する(ブレーク・イベント)」参照
	→「2.9.5 変数 /I/O レジスタへのアクセスで停止する」参照

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

イベント種別	説明
ソフトウエア・ブレーク	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
(ブレーク ^{注1})	え、その命令を実行した際にプログラムをブレークさせるイベント
	です。
	→「2.9.3 任意の場所で停止する(ブレークポイント)」参照
関数の先頭へのブレーク	解析ツールの関数一覧パネルより設定されるハードウエア・ブレー
	ク(実行系)です。
変数のアクセス・ブレ ー ク	解析ツールの変数一覧パネルより設定されるハードウエア・ブレー
	ク(アクセス系)です。
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)。
	→「2.12.2 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)。
	→「2.13.1 実行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)。
	→「2.12.3 任意区間の実行履歴を収集する」 参照
タイマ計測 n	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)。
	なお, "n"は, 設定したタイマのチャネル番号を示します。
	→「2.13.2 任意区間の実行時間を計測する【シミュレータ】」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。
	→「2.12.4 条件を満たしたときのみの実行履歴を収集する【シ
	ミュレータ】」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)。
	→「2.15.1 printfを挿入する」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(「(1) 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。
 - 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。

(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。

イベント種別		表示内容 ^{注1}
ハードウエア・ブレーク	表示形式1	< <i>発生条件</i> > <ファイル名# 行番号> <アドレス>
(発生条件:実行系)	表示例	実行前 main.c#39 0x100
		実行後 sub.c#100 0x200
		実行前 — 0x300
		実行 main.c#39 0x300 【シミュレータ】
	表示形式2	< <i>発生条件</i> > < <i>シンボル</i> +オフセット> <アドレス>
	表示例	実行前 funcA + 0x10 0x100
		実行後 funcB + 0x20 0x200
		実行前 — 0x300
ハードウェア・ブレーク	表示形式 1	<発生条件> <ファイル名# 変数名> <アドレス(範囲)>
(発生条件:アクセス系)		<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
		(<i>範囲</i>) > <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	< <i>発生条件> <変数名> <アドレス</i> (<i>範囲</i>)> <i><比較条件></i> < <i>比較値</i> >
	表示例	ライト variable1 0x100 - 0x101 == 0x10
ソフトウエア・ブレ ー ク	表示形式1	< <i>発生条件</i> > <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	< <i>発生条件</i> > <シンボル+オフセット> <アドレス>
	表示例	実行前 funcA + 0x12 0x102
無条件トレース	表示形式	
	表示例	
Run-Break タイマ	表示形式	総実行時間 : < <i>総実行時間</i> >
	表示例	総実行時間 : 1000ms
		総実行時間 : OVERFLOW

表 A-10 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}
トレース (発生条件:実行系)	表示形式	開始/終了の総数:< <i>トレース開始/トレース終了イベント</i> の総数> 注2 < 開始/終7> < トレース開始/トレース終了の詳細情報>
	表示例	開始/終了の総数:4
		- 開始 実行後 main.c#100 0x300
		-開始 実行後 funcA + 0x100 0x300
		- 終了 実行後 funcA + 0x10 0x100
タイマ計測 n (発生条件:実行系)	表示形式	総実行時間: < <i>総実行時間</i> > 開始/終了の総数: < <i>タイマ開</i> 始/タイマ終了イベントの総数> ^{注 2}
		- < 総美行時间> <ハスカワント> < + ⋻)実行時间> < 取
		- <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:4
		- 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 実行後 funcA + 0x30 0x100
		- 終了 実行後 main.c#100 0x300
		- 終了 実行後 funcA + 0x50 0x100
ポイント・トレース	表示形式1	<発生条件> <変数名> <変数のアドレス>
(発生条件:アクセス系)	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	< <i>発生条件</i> > <ファイル名 # 関数名 # <i>変数名</i> > < <i>変数のアド</i>
	表示例	リード/ライト sub.c#func1#variabl3 0x300
Printf イベント (アクション・イベント)	表示形式	< <i>発生条件</i> > < 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

注1. 表示形式の詳細は次のとおりです。

< <i>発生条件</i> >	次の条件のいずれか1つを表示します。		
	[Full-spec emulator] [E1] [E20]		
	実行系 注 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 注 実行		
	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
	<i>ド・モジュール名\$ ファイル名# 行番号</i> >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、 <i>行番号</i> をシン
	<i>ボル+ オフセット</i> 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、 <i>行番号</i> を"一"で表示します。
	- 行情報がなく、 シンボル情報がない場合
< <i>変数名</i> >	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
	κ_{\circ}
<i><比較値</i> >	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)。
<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。
<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(「2.13.3 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。
	また、未計測の場合は、"未計測"を表示します。
< <i>総実行時間</i> >	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/clock 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(「2.13.3 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。
	また、未計測の場合は、"未計測"を表示します。
< <i>平均実行時間</i> >	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/clock 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(「2.13.3 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。
	また、未計測の場合は、"未計測"を表示します。
< <i>最大実行時間</i> >	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/clock 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(「2.13.3 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。
	また、未計測の場合は、"未計測"を表示します。

<i><最小実行時間</i> >	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/clock 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(「2.13.3 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。		
	また、未計測の場合は、"未計測"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。 <i>出力文字列:変数式</i> を表示し		
	ます。		

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。

×	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。
	たたし、ビルトイン・イベント(無条件トレース・イベント /Run-Break ダイマ・イベン
	ト)を削除することはできません。
W	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)。
	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)。
[Full-spec emulator] [E1] [E20]	
	トレース関連のイベントを表示します(デフォルト)。
3	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)を表示します(デフォルト)。
5	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。
1	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。

[ツールバー]

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、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	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。
頖	转效化	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。
肖	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	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。
쿺	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	ハードウェア・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。
5	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。
	クロック表示	選択しているイベントのタイマ結果をクロック(clock)単位で表示します。
>	モリヘジャンプ	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。



逆アセンブルヘジャンプ	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。
ソースヘジャンプ	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。
条件の編集	選択しているアクション・イベント(Printf イベント)を編集するためにアクション・
	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 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。



出力 パネル

CubeSuite+ 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 (「2.15.1 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。

備考 ツールバーの 100% ▼, または [Ctrl] キーを押下しながらマウス・ホイールを前後方に動かすこと により、本パネルの表示を拡大/縮小することができます。





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。 ビルド結果/検索結果(一括検索)の表示では、ビルド/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(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	レト)	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
		背景色	標準色	ます。
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
		背景色	薄グレー	な場合に表示されます。

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)。

(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	CubeSuite+ 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	CubeSuite+ 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。
キャッシュ・レート 【シミュレータ】	ブレーク時に、キャッシュ・ヒット率(キャッシュへのアクセス回数に対する ヒット回数の割合)を表示します。
ビルド・ツール	ビルド・ツールから出力されたメッセージを表示します。



タ ブ名	説明
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 - <i>タブ名</i> 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
	ない。起動後に初めてこの項目を送水した場合は、「石船を持りてラン石を休行…」の 「 選択と同笑の動作となります
	医抗と同等の動作となりより。
	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 <i>ブ名</i> 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。	

メモリ・マッピング ダイアログ

メモリ・マッピングの状況を表示します。

注意 選択したマイクロコントローラがマルチコア対応版の場合、コア(PEn)の選択を切り替えることにより、 PEn ごとのメモリ・マッピングの状況を表示します(「2.7 コア(PE)の選択」参照)。



図 A---40 メモリ・マッピング 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 [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 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]

- 追加メモリ・マッピング指定エリア このエリアは常に無効です。
- (2) [マッピング済みメモリー覧] エリア

(a)一覧の表示

マイクロコントローラ内のメモリ・マッピング情報を表示します。 このエリアを編集することはできません。

メモリ種別	メモリ種別を表示します。
アドレス範囲	アドレス範囲を< <i>開始アドレス</i> > - < <i>終了アドレス</i> >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 ^注)。
アクセス幅	アクセス幅を表示します(単位:ビット)。

注 1024の倍数の場合のみ、Kバイト単位で表示します。

(b)ボタン

ボタン	機能
削除	このボタンは常に無効です。

[機能ボタン]

ボタン	機能
ОК	このダイアログをクローズします。
キャンセル	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)。

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。

	ダウンロード・ファイル				
	ダウンロード・ファイル一覧(E): RH850.abs -	(† (<u>U</u>) (↓ (<u>D</u>)	ダウンロード・ファイルのプロパティ(P):	DefaultBuild¥RH850.abs ロード・モジュール・ファイル はい はい	
(1) —			入力補完機能用の情報を生成する	はい	- (2)
	追加(<u>A)</u> 削除(<u>R</u>)		ダウンロード・ファイル情報		
[機能ボタン]-			ОК	キャンセル ヘルブ(円)	

図 A-41 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



[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します。

(b)ボタン

ボタン	機能
1	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。
Ļ	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。
 - 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		
		キーボードからの直接入	カーまたはこの項目を選択すると欄内右端に表示	
	夏 更月14	マーホール 50 置後八 される […] ボタン 21	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	コグによる指定	
	指定可能值	 「表 2—1 ダウンロード	可能なファイル」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冊	ジ式を選択します。	
	デフォルト	ロード・モジュール・フ	アイル	
	変更方法	ドロップダウン・リスト	による選択	
	指定可能值	ロード・モジュール・	ロード・モジュール・フォーマット(*.abs)を	
		ファイル	指定します。	
		ヘキサ・ファイル	インテル拡張ヘキサ・フォーマット(*.hex)を	
			指定します。	
		Sレコード・ファイル	モトローラ・Sタイプ・フォーマット(*.mot)	
			を指定します。	
		バイナリ・データ・	バイナリ・フォーマット(*.bin)を指定します。	
		ファイル		
オフセット	指定したファ	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。		
	なお、この項目は、[ファイルの種類] に [ヘキサ・ファイル]、または [S レコード・			
	ファイル]を:	選択している場合のみ表示	<pre></pre>	
	デフォルト	0		
	変更方法	キーボードからの直接入	. л	
	指定可能值	0x0~0xFFFFFFFFの1	16 進数	
開始アドレス	指定したファ	イルをダウンロードする閉	射始アドレスを指定します。	
	なお、この項	なお、この項目は、[ファイルの種類]に[バイナリ・データ・ファイル]を選択してい		
	る場合のみ表	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	<i>"</i> л	
	指定可能值	0x0~0xFFFFFFFFの1	16 進数	
オブジェクトをダ	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選択します。	
ウンロードする	なお、この項目は、[ファイルの種類]に[ロード・モジュール・ファイル]を選択して			
	いる場合のみ	いる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	
	指定可能值	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	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選択します ^{注2} 。			
ロードする	なお、この項	なお、この項目は、[ファイルの種類]に[ロード・モジュール・ファイル]を選択して		
	いる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	
	指定可能值	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。	
入力補完機能用の	ダウンロード時に、シンボル名の入力補完機能のための情報を生成するか否かを選択しま			
情報を生成する	_{す)} 注3 _。			
	なお、この項目は、[ファイルの種類]に[ロード・モジュール・ファイル]を選択して			
	いる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	シンボル名の入力補完機能用の情報を生成しま	
			す(入力補完機能を使用します)。	
		いいえ	シンボル名の入力補完機能用の情報を生成しま	
			せん(入力補完機能を使用しません)。	

- **注1.** [ダウンロード・ファイルー覧] 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。
 - 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。
 - **3.** [はい]を選択した場合、ダウンロード時間、およびホスト・マシンのメモリ消費量が増加しま す。シンボル名の入力補完機能を使用しない場合は、[いいえ]を選択することを推奨します。

[機能ボタン]

ボタン	機能
ОК	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。



フラッシュ・オプションの設定 ダイアログ

[Full-spec emulator] [E1] [E20]

マイクロコントローラに搭載されているフラッシュ・メモリのためのオプション設定を行います。 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。

注意 このダイアログで設定変更を行ったのち、[書き込み] ボタンをクリックすると、CPU リセットが発生します。

	フラッシュ・オブションの設定		
Γ	フラッシュ・オプション・プロパティ(<u>P</u>):		
(1) -	日 オブション・バイト設定 OPBT0 OPBT1 OPBT2 OPBT3 OPBT4 OPBT5 OPBT6	HEX FFFFFFF HEX FFFFFFF HEX FFFFFFF HEX FFFFFFF HEX FFFFFFF HEX FFFFFFF	
【機能ボタン] -	OPB17 オプション・バイト設定 読み込み(E) 書き込	HEX FFFFFFF み(W) 閉じる(Q) キャンセル (~JUJ(H)

図 A—42 フラッシュ・オプションの設定 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]

- プロパティ パネルの [フラッシュ・オプション設定] タブ【Full-spec emulator】【E1】【E20】において, [フ ラッシュ・オプション] カテゴリ内 [フラッシュ・オプション] 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



[各エリアの説明]

(1) [フラッシュ・オプション・プロパティ] エリア

(a) [オプション・バイト設定]

フラッシュ・メモリのオプション・バイトの設定を行います。

OPBT <i>0</i> ~ 7	オプション・	バイトを指定します。
	デフォルト	マイクロコントローラ出荷時の値
	変更方法	キーボードからの直接入力
	指定可能值	0x0 ~ 0xFFFFFFFF の範囲の 16 進数

注意 選択しているマイクロコントローラの種類により、表示されるオプション・バイト (OPB0~7) の個数は異なります (欠番を含む場合もあります)。

[機能ボタン]

ボタン	機能
読み込み	デバッグ・ツールに現在設定されている値を読み込み, このダイアログの各項目に反映 します。
書き込み	現在の設定値をデバッグ・ツールに書き込み、プロジェクトに反映したのちこのダイア ログをクローズします。
閉じる	現在の設定値をプロジェクトに反映し、このダイアログをクローズします。
キャンセル	設定の変更を反映せずに、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。



優先するブート・ローダ・プロジェクトの選択 ダイアログ

コア単体のデバッグを行う際において、そのアプリケーション・プロジェクトが複数のブート・ローダ・プロジェクトの構成要素となっている場合、優先してデバッグ対象とするブート・ローダ・プロジェクトを選択します。

	優先するブート・ローダ・フロジェクトの選択 🛛
	デバッグに使用するブート・ローダ・プロジェクトを選択してください。 選択されたブート・ローダ・プロジェクトを優先してデバッグに使用しま す。
	sample_multi sample_boot2
(1) -	
L [機能ボタン]-	ОК \\J\J(<u>H</u>)

図 A-43 優先するブート・ローダ・プロジェクトの選択 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]

各ブート・ローダ・プロジェクト(マルチコア設定ツール)のプロパティ パネルにおいて, [デバッグ] カテゴリ 内 [デバッグ時に優先する] プロパティがすべて [はい], またはすべて [いいえ] に選択されている状態で次の操作 を行った場合

- デバッグ・ツールとの通信開始(「2.4.1 デバッグ・ツールを接続する」参照)

[各エリアの説明]

(1) 優先するブート・ローダ・プロジェクト選択エリア 表示されるブート・ローダ・プロジェクトの一覧から、他のブート・ローダ・プロジェクトより優先してデ バッグ対象とするブート・ローダ・プロジェクトを選択します。



[機能ボタン]

ボタン	機能
ОК	選択したブート・ローダ・プロジェクトを優先してデバッグ対象とします。
	また、選択したブート・ローダ・プロジェクトのみがデバッグ対象の優先プロジェクト
	となるように、すべてのブート・ローダ・プロジェクトのプロパティ設定([デバッグ]
	カテゴリ→[デバッグ時に優先する]プロパティ)を変更します。
ヘルプ	このダイアログのヘルプを表示します。





複数行のテキストの入力、編集を行います。

図 A-44 テキスト編集 ダイアログ

1	テキスト編集	3
Г	テキスト(工):	
		1
(1) —		
L	<u>S</u>	
[機能ボタン]-	OK キャンセル ヘルプ(H)	

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 [フック処理設定] タブにおいて, [フック処理設定] カテゴリ内の各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

[各エリアの説明]

(1) [テキスト] エリア

複数行のテキストの編集を行います。

[機能ボタン]

ボタン	機能
ОК	入力したテキストをこのダイアログの呼び出し元に反映し、このダイアログをクローズ します。
キャンセル	入力したテキストをこのダイアログの呼び出し元に反映せずに、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(「2.15 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),(「2.16.6 イベント設定に関する留 意事項」も参照してください。

	アクション・イベン	21F	
(1)-	Printf イベント		
	出力文字列(0)	: 入力例)サンブル:	
(2)-	変数式(⊻):	入力例)aaa, bbb, ccc	
	にこに変数式を	入力してください)	
	アドレス(<u>A</u>):		
	"C:¥sample¥D	efaultBuild¥sample.abs″\$vecttbl.asm#14 📦 •	
	出力 パネル で(サンプル: aaa =	D表示例) 10, bbb = 20 ccc = 30	
[機能ボタン] -		OK キャンセル ヘルプ(H	\sum

図 A—45 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、設定するアクション・イベントの種類が切り替わります。 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

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
ОК	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(「2.15 プログラム内へのアクションの設定」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—46 アクション・イベント ダイアログ:[Printf イベント] タブ

アクション・イベ	ント		
Printf イベント			
出力文字列(2): 入力例)サンプル:		
変数式(V):	入力例) aaa, bbb, ccc		
にご変数式	を入力してください)		
アドレス(<u>A</u>):			
"C:¥sample¥[) efaultBuild¥sample.abs"\$v	/ecttbl.asm#14	
山力パネルで	の素子例)		
	10,111,111,000,000		
サンブル: aaa =	10, bbb = 20 ccc = 30		
		(1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	

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]

- 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。

式	出力される値
C 言語変数名	C 言語の変数の値
変数式[変数式]	配列の要素値
<u>変数式</u> .メンバ名	構造体/共用体のメンバ値
<i>変数式</i> -> メンバ名	ポインタの指し示す構造体/共用体のメンバ値
* 変数式	ポインタの変数の値
& 変数式	 配置アドレス
CPU レジスタ名	 CPU レジスタの値
	I/O レジスタの値
ラベル名 ^注 EQU シンボル名 ^注 [即値]	ラベルの値、EQU シンボルの値、即値アドレスの値

表 A-11 変数式と出力される値の関係(Printf イベント)

- 注 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)。なお、"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:1: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(「2.18.2 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(「2.18.2 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—47	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	" <i>変数式1~10</i> "に対する変数値		
	値は変数の型に応じた表示形式(「表 A—7 ウォッチ式の表示形式(デフォルト)」参照)		
	で表示します(指定された変数式が取得不能の場合は"?"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"-"を表示)。		



ファイル・エンコードの選択 ダイアログ

ファイル・エンコードの選択を行います。

図 A-48 ファイル・エンコードの選択 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]

- [ファイル] メニュー→ [エンコードを指定して開く ...]を選択してファイルを開く ダイアログをオープン→ダ イアログ上で [開く] ボタンをクリック

[各エリアの説明]

(1)[利用可能なエンコード]

設定するエンコードを選択します。

デフォルトでは、選択したファイルのエンコードが選択されています。

ボタン	機能
ОК	指定したファイル・エンコードを使用し、ファイルを開く ダイアログで選択したファ
	イルをオープンします。
キャンセル	ファイルを開く ダイアログで選択したファイルをオープンせずに、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。



ファイルの保存設定 ダイアログ

エディタ パネルで編集中のファイルのエンコードと改行コードの設定を行います。

備考 タイトルバーには、設定対象ファイルの名前が表示されます。

図 A-49 ファイルの保存設定 ダイアログ

	main.c の保存設定	×
	エンコード(<u>E)</u> :	
(1)-	日本語 (シフト JIS)	~
Г	さな行コード(止):	
(2)-	Windows (CR LF)	~
(3) —	▶ これらの設定でファイルを再読み込みする(B)	
[機能ボタン]-	OK キャンセル ヘルプ(H)	כ

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 パネルにフォーカスがある状態で, [ファイル] メニュー→ [ファイル名の保存設定 ...]を選択

[各エリアの説明]

(1) [エンコード] エリア

設定するエンコードをドロップダウン・リストにより選択します。

ドロップダウン・リストの項目は、次の順番で表示されます。

ただし、同じエンコード名、および現在の OS が対応していないエンコード名は表示されません。

- *現在のファイルのエンコード名*(デフォルト)
- 現在の OS の既定のエンコード名
- *最近使用した エンコード名*(最大4件)
- 現在のロケールでよく使用されているエンコード名
 - (例:ロケールが日本の場合)
 - 日本語 (シフト JIS)
 - 日本語 (JIS 1 バイト カタカナ可 SO/SI)
 - 日本語 (EUC)

- Unicode (UTF-8)

- 現在の OS が対応する上記以外のエンコード名(アルファベット順)

(2) [改行コード] エリア

設定する改行コードをドロップダウン・リストにより選択します。

次の項目を選択することができます。

- Windows (CR LF)
- Macintosh (CR)
- Unix (LF)

デフォルトでは、現在の改行コードが選択されます。

(3) [これらの設定でファイルを再読み込みする]



ボタン	機能
ОК	指定したエンコード、および改行コードを対象ファイルに設定し、このダイアログを クローズします。 [これらの設定でファイルを再読み込みする]をチェックした場合、指定したエンコー ド、および改行コードを対象ファイルに設定し、ファイルを読み込み直したのち、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。



~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

ブックマーク設定位置の表示や、ブックマークの削除を行います。

図 A—50 ブックマーク ダイアログ

	ブックマーク			×
Г	ファイル名	行番号	パス	表示──
	boot.s	14	C:¥Sample¥appli¥source	
	task.c	23 28	C:#Sample#appli#source	<u>則除(P)</u>
(1)—				すべて削除(<u>A</u>)
「機能ギカト」	→ (P)			
「成肥小ダン」」	E INVE		MAN HIGO	

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]

- ツールバーの 🔄 ボタンをクリック
- [編集] メニュー→ [ブックマーク] → [ブックマークをリスト表示する ...] を選択
- エディタ パネルにおいて, コンテキスト・メニューの [ブックマーク] → [ブックマークをリスト表示する ...] を選択

[各エリアの説明]

(1) ブックマーク・リスト表示エリア

登録しているブックマークをリスト表示します。

表示は、[ブックマーク]のファイル名のアルファベット順で表示します。同じファイル中のブックマークについては、行番号順に表示します。

エディタ パネルにブックマークが追加された場合、ブックマーク情報が追加されます。

ブックマーク・リスト表示エリアにおいて、行をダブルクリックすることにより、該当ブックマーク位置に キャレットを移動します。

(a) [ファイル名]

ブックマーク登録しているファイル名(パスなし)を表示します。

(b) [行番号]

ブックマーク登録している行番号を表示します。

(c) [パス]

ブックマーク登録しているファイルのパスを表示します。

(d)ボタン

表示	選択しているブックマーク位置に、キャレットを移動します。
	ただし、ブックマークが選択されていない場合、複数個のブックマークを選択している
	場、またはブックマークが登録されていない場合は、無効となります。
削除	選択しているブックマークを削除します。複数個のブックマークを選択している場合
	は、選択しているすべてのブックマークを削除します。
	ただし、ブックマークが選択されていない場合、またはブックマークが登録されていな
	い場合は、無効となります。
すべて削除	登録しているすべてのブックマークを削除します。
	ブックマークが登録されていない場合は、無効となります。

注意 エディタ パネルをクローズしても,登録したブックマークは削除されません。

ただし、ファイルを新規作成後に一度も保存していないエディタ パネルをクローズした場合、登録し たブックマークは削除されます。

ボタン	機能
前へ	選択しているブックマークの前のブックマーク位置に、キャレットを移動します。
	以下の場合は、無効となります。
	- 先頭行に表示しているブックマークを選択している
	- ブックマークが選択 されていない
	- 複数個のブックマークを選択している
	- ブックマークが登録されていない
	- ブックマークがひとつしか登録されていない
次へ	選択しているブックマークの次のブックマーク位置に、キャレットを移動します。
	以下の場合は、無効となります。
	- 最終行に表示しているブックマークを選択している
	- ブックマークが選択されていない
	- 複数個のブックマークを選択している
	- ブックマークが登録されていない
	- ブックマークがひとつしか登録されていない
閉じる	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---51 表示桁数設定 ダイアログ

	表示桁数設定	
(1)-	表示桁数(C):	16
[機能ボタン]-		OK キャンセル ヘルプ(H)

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)。



ボタン	機能
ОК	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---52 アドレス・オフセット設定 ダイアログ

	アドレス・オフセット設定	
(1)-	表示アドレス・オフセット値(_): HEN 0	
[機能ボタン]-	OK キャンセル	ヘルプ(H)

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



ボタン	機能
ОК	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(「(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---53 メモリ初期化 ダイアログ

	メモリ初期化	
Π	開始アドレス/シンボル(<u>S</u>):	終了アドレス/シンボル(E):
(1)-	✓ -	総了アドレス/シンボルを入力 🕨 🗸
Π	初期化データ(1):	·
(2)-	■■■ ②期化データを16進数で入力してくだ	さい。スペース区切りで複数指 🕨 🗸
		(++++++) (+++++++++++++++++++++++++++++
[機能ホダン]]	OK.	44701 10170

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(「2.18.2 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,初期化対象のアドレス範囲のエンディアンの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	書き込みイメージ(バイト単位)	
(初期化データ)	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

ボタン	機能
ОК	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(「(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。

注意 メモリの検索は、プログラム空間のみが対象となります(ミラー領域は検索対象となりません)。

	メモリ検索				×
(1) -	検索するデータ(<u>D</u>):				~
(2) -	検索する範囲(<u>R</u>):	アドレス範囲を指定する			~
(3) -	アドレス(<u>A</u>):	0x0 💌	-	0×fffffff	~
[機能ボタン] -	1	前を検索(B) 次を検索(E)		キャンセル ヘルフ(出	

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。	
	表示形式:<メ <i>モリ種別</i> ><ア <i>ドレス範囲</i> > <i><サイズ</i> >	

(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(0x03FFFFFF)です(ミラー領域は検索対象となりません)。

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(「2.18.2 シンボル名の入力補完機能」参照)。
 - 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。
 - 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。

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---55 印刷アドレス範囲設定 ダイアログ

	印刷アドレス範囲設定
г	印刷するアドレスの指定方法を選択してください。
(1) —	 ●現在の表示範囲(D) ●現在の選択範囲(C) ●指定した範囲(R) ■ 開始アドレス(S): ● 「開始アドレスを入力してくだ ● ● 「徐子アドレスを入力してくだ ●
[機能ボタン]-	OK キャンセル ヘルプ(H)

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(「2.18.2 シンボル名の入力補完機能」参照)。

ボタン	機能
ОК	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 タイアロクをオーフンします。
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。
ヘルプ	このダイアログのヘルプを表示します。



Print Preview ウインドウ

印刷をする前に、現在エディタパネルで表示されているファイルのプレビューを行います。

備考 [Ctrl] キーを押下しながらマウス・ホイールを前後方に動かすことにより、本ウインドウの表示を拡大/縮小 することができます。



図 A---56 Print Preview 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]
 - [ツールバー]
 - [コンテキスト・メニュー]

[オープン方法]

- エディタ パネルにフォーカスがある状態で, [ファイル] メニュー→ [印刷プレビュー] を選択

[各エリアの説明]

(1) プレビュー・エリア

印刷イメージをプレビュー表示します。 ヘッダ/フッダ部に、ファイル名(絶対パス)/ページ番号を表示します。 デバッグ・ツールと切断時、デバッグ・ツールと接続時(通常モード)、またはデバッグ・ツールと接続時 (混合表示モード)により表示するカラムが異なります。

ただし、エディタパネルにおいて、非表示に設定しているカラムは表示されません(印刷されません)。

[ツールバー]

2	印刷プレビュー表示しているアクティブなエディタ パネルの内容を印刷するために、Windows で用意されてい
	る。印刷 ダイアログをオープンします。
	選択範囲をクリップボードにコピーします。
Ð	表示サイズを拡大します。
O	表示サイズを縮小します。
幸	100% の倍率で表示します(デフォルト)。
ų	ページ幅で表示します。
	1ページ全体を表示します。
	見開き 2 ページを表示します。

[コンテキスト・メニュー]

ズームの拡大	表示サイズを拡大します。
ズームの縮小	表示サイズを縮小します。



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(「2.12.8 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---57 トレース検索 ダイアログ

	トレース検索			×
(1) —	命令レベル ソース・レイ	SV		
П	検索条件の指定			
	フェッチ・アドレス(工):		~ -	(範囲を指定する場合 🕨 🗸
	命令(<u>M</u>):		*	
	アクセス・アドレス(<u>A</u>):		-	(範囲を指定する場合 🔊 🗸
(2) —	アクセスの種類(S):	(指定なし)	~	
	データ(<u>D</u>):	HEX	-	HER (範囲を指定する) 🍺 🗸
	検索範囲の指定			
	番号(<u>U</u>):		-	~
	53			
[機能ボタン]-		前を検索(日) ///xを検索(E)	キャンセル ヘルブ(H)

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 🔛 ボタンを選択
- 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。

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。

検索条件の指定			
フェッチ・アドレス(<u>1</u>):		× -	(・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
命令(<u>M</u>):		*	
アクセス・アドレス(<u>A</u>):		~ -	(範囲を指定する場合 🕨
アクセスの種類(S):	(指定なし)	~	
データ(<u>D</u>):	HEX	-	● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●
検索範囲の指定			
番号(<u>U</u>):		-	

図 A—58 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]

[オープン方法]

- トレース パネルにおいて, ツールバーの 満 ボタンを選択
- 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。

(指定なし)	アクセスの種類を限定しません。
リード/ライト	リード、またはライトした箇所を検索します。
リード	リードした箇所を検索します。
ライト	ライトした箇所を検索します。
ベクタ・リード	ベクタ・リードした箇所を検索します。
DMA	今版では無効です。

(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。



[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。

● 快茶州家の指定 ● ソース行を指定 ● 即地々も指定	」て実行箇所を検索(Q) て生殖スドレスの実行策所を検索(T)
○ グローバル変数	C元頃アトレスの美行国所を検索(1) Sを指定してアクセス箇所を検索(G)
検索条件の指定	
ソース行(<u>S</u>):	
関数名(<u>N</u>):	
変数名(⊻):	
種類(K):	参照/代入
変数値(A):	HEX - HEX
検索範囲の指定	
番号(∐):	-

図 A—59 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]

[オープン方法]

- トレース パネルにおいて, ツールバーの 🕋 ボタンを選択
- 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	指定した関数の実行箇所を検索します。
所を検索	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	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。
所を検索	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
 - 2. main.c
 - 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) [変数值]

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。



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・バー上のスライダーの大きさが変化し、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。

	スクロール 範囲	愛定 🛛 🚺	3
(1)	開始アドレス(<u>S</u>):	1	
(2)	終了アドレス(匠):	(終了アドレスを入力して(ださい)]
 [機能ボタン]-		OK キャンセル ヘルプ(H)]

図 A-60 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(「2.18.2 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。

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)。

ボタン	機能
ОК	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。



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(「2.18.2 シンボル名の入力補完機能」参照)。

指定行へのジャンプ ダイアログ

指定した行番号/シンボル/アドレスにキャレットを移動します。

図 A---61 指定行へのジャンプダイアログ

	指定行へのジャンプ 🛛
(1)-	行番号(1 - 44)またはシンボル(<u>L</u>): 12
[機能ボタン]-	OK キャンセル ヘルプ(H)

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 パネルにフォーカスがある状態で, [編集] メニュー→ [移動 ...] を選択

- エディタ 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) [行番号(有効な行の範囲)またはシンボル] エリア

キャレットを移動したい行番号(10進数)/シンボル名^{注1}/アドレス値^{注2}を直接入力により指定します。 なお,"(*有効な行の範囲*)"には,現在のファイルの有効な行番号の範囲を表示します。 デフォルトでは,エディタパネル上の現在のキャレット位置の行番号を表示します。

- 注1. 次の注意が必要となります。
 - シンボル名として、関数名/変数名のみ指定可能です。
 - 使用するビルド・ツールのプロパティ パネルにおいて、クロス・リファレンス情報を出力する設定 ([共通オプション] タブ→ [出力ファイルの種類と場所] カテゴリ→ [クロス・リファレンス情報 を出力する] プロパティ→ [はい (-Xcref)]) にしたのち、ビルドの実行を完了させてください。
 - 2. 次の注意が必要となります。
 - アドレス値は、"0x", または"0X"を付与した 16 進数のみ指定可能です(10 進数は行番号として 扱います)。
 - ビルドの実行を完了させてください。

ボタン	機能
ОК	指定したソース行の先頭にキャレットを移動します。
キャンセル	移動を無効とし、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。



関数ヘジャンプ ダイアログ

エディタ パネル上で指定した関数へジャンプする際に同名の関数が複数存在する場合,ジャンプする関数を選択します。

- 注意 1. このダイアログは、同名の関数が複数存在し、ビルド・ツールにおいてクロス・リファレンス情報を出力す る設定([共通オプション] タブ→ [出力ファイルの種類と場所] カテゴリ→ [クロス・リファレンス情報 を出力する] プロパティ→ [はい (-Xcref)]) にしている場合のみ表示します。
 - 2. 関数ヘジャンプする機能は、プロジェクトに登録されているファイルのみが対象です。

_	関数ヘジャンプ		
	ファイル名	行番号	172
	overload.cpp	25	D:¥work¥OverlcadProject¥overload.cpp
(1)	overload.cpp	29	D:¥work¥OverloadProject¥overload.cpp
	overload.cpp	33	D:¥work¥OverloadProject¥overload.cpp
L			
能ぜない」			

図 A—62 関数ヘジャンプダイアログ

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、コンテキスト・メニュー→ [関数へジャンプ] を選択

[各エリアの説明]

(1) ジャンプ先候補表示エリア

ジャンプ先の候補をリスト表示します。

表示は、[ファイル名]のアルファベット順で表示します。同じファイル中の候補については、行番号順に表示します。

(a) [ファイル名]

関数が定義されているファイル名(パスなし)を表示します。

(b) [行番号]

関数が定義されている行番号を表示します。

(c) [パス]

関数が定義されているファイルのパスを表示します。

ボタン	機能
ОК	ジャンプ先候補表示エリアにおいて行を選択し、このボタンをクリックすると、該当
	関数の定義行へジャンプします。
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--63 指定位置へ移動 ダイアログ

	指定位置へ移動	
(1)-	アドレス/シンボル(<u>A</u>):	v
[機能ボタン]-	ОК	キャンセル ヘルプ(山)

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, [編集] メニュー→ [移動…] を選択
- IOR パネルにフォーカスがある状態で, [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 (「2.18.2 シンボル名の入力補完機能」参照)。

ボタン	機能
ОК	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—64 データ保存 ダイアログ

	データ保存		
(1) -	ファイル名(<u>N</u>):	C:¥Test¥sample¥逆アセンブル1	<u></u>
(2) -	ファイルの種類(工):	テキスト・ファイル (*.txt)	*
(3)-	保存範囲 アドレス _hdwinit+0x0000	ス/シンボル(<u>A</u>): 1000a マーmain+0x00000010	~
機能ボタン]-	-	保存(S) キャンセル 、	くいづ(巴)

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)。

RENESAS

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

インテル・ヘキサ・フォーマット (*.hex)	インテル拡張ヘキサ・フォーマット	
モトローラSフォーマット (*.mot)	モトローラ・Sタイプ・フォーマット	
バイナリ・データ (*.bin)	バイナリ・フォーマット	

備考 アップロードについての詳細は、「2.5.3 アップロードを実行する」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。	

保存対象	説明
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(「2.18.2 シンボル名の入力補完機能」参照)。

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。



処理中表示 ダイアログ

時間を要する処理を行っている際に、その進捗状況の表示を行います。 このダイアログは、実行中の処理が完了した場合、自動的にクローズします。

図 A---65 処理中表示 ダイアログ

	進捗状況	×
(1)-	プロジェクトを読み込んでいます。	
(2)-	(***************	
[機能ボタン]—	キャンセル	

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]
- [機能ボタン]

[オープン方法]

- 時間を要する処理において、メッセージが発生した際に自動的に表示

[各エリアの説明]

(1) メッセージ表示エリア

処理中に発生したメッセージを表示します(編集不可)。

(2) プログレスバー

現在実行中の処理の進捗状況をバーの長さで表示します。

なお,進捗率が100%に達した場合(右端までバーの長さが達した場合),このダイアログは自動的にクローズします。

ボタン	機能	
キャンセル	現在実行中の処理を中断し、このダイアログをクローズします。	
	ただし、実行中の処理の中断が不可能な場合、このボタンは無効となります。	



オプション ダイアログ

CubeSuite+の各種環境設定を行います。

このダイアログでの設定は、使用中のユーザの設定として保存されます。





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]

- [ツール] メニュー→ [オプション…] を選択

[各エリアの説明]

(1)カテゴリ選択エリア

設定したい項目を次のカテゴリから選択します。

カテゴリ	設定内容
[全般 - 起動と終了]カテゴリ	起動、または終了時に関連した設定を行います。
[全般 - 表示] カテゴリ	表示に関連した設定を行います。
[全般 - 外部テキスト・エディタ] カテゴリ	外部テキスト・エディタに関連した設定を行います。



カテゴリ	設定内容	
[全般 - フォントと色] カテゴリ	各パネルで表示するフォントと色に関連した設定を行います。	
[全般 - 外部ツール]カテゴリ	外部ツールを起動する際の設定を行います。	
[全般 - ビルド/デバッグ] カテゴリ	ビルド、またはデバッグに関連した設定を行います。	
[全般 - Python コンソール]カテゴリ	Python コンソールに関連した設定を行います。	
[全般 - テキスト・エディタ] カテゴリ	テキスト・エディタに関連した設定を行います。	
[全般 - アップデート]カテゴリ	アップデートに関連した設定を行います。	
[その他 - ユーザ情報]カテゴリ	ユーザ情報に関連した設定を行います。	

備考 [全般 - フォントと色] / [全般 - ビルド/デバッグ] 以外のカテゴリについては,「CubeSuite+ 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)設定エリア

選択したカテゴリに対して、各種オプションを設定するエリアです。 各カテゴリの設定方法についての詳細は、該当するカテゴリの項を参照してください。

ボタン	機能	
すべて初期値に戻す	このダイアログのすべての設定項目をデフォルトの状態に戻します。 ただし、「全般 - 外部ツール]カテゴリでは、新規登録した内容の削除は行いません。	
ОК	変更した設定内容を適用し、このダイアログをクローズします。	
キャンセル	変更した設定内容を無効とし、このダイアログをクローズします。	
適用	変更した設定内容を適用します(このダイアログをクローズしません)。	
ヘルプ	このダイアログのヘルプを表示します。	



[全般-フォントと色] カテゴリ

全般に関わる設定のうち、各パネルで表示するフォントと色に関連した設定を行います。

1) 小部方本入ト・エディタ 分部ツール ・1) 指定する園川(5): (1) アオスト・エディタ アリカラブート マホート 日 デキスト・エディタ アリフブート マホート 日 アリントント マホート 日 アリントント マホート 日 アリントント マホート 日 アリント アリント 日 アリント アリント マリント アリント アリント マリンド アリント </th <th>オプション 全般 </th> <th>全般 - フォントと色</th> <th></th>	オプション 全般 	全般 - フォントと色	
2) 	 小部デキスト・エティート フォントと臣 ハボントンニート ハボントンニート ウルドンデバッグ ビルドンデバッグ アキスト・エディタ テキスト・エディタ アップデート その他 ユーザ情報 	 	 ▲ 「 標準の文字色を使用する(C) ☆字色(丁) ✓ 標準の背景色を使用する(D) 背景色(B)
3) テキスト・エディタのフォント設定 フォンド(E) 4) インボード(E) ギ初期値に戻す(L)	2)	(描変更 → 表示例(<u>M</u>): Aað	▼ 色のリセット(R) あぁアァ亜宇
	4)	→ テキスト・エディタのフォント設定 → インボート(<u>P</u>) エクスボート(<u>E</u>)	フォンド(F) 初期値に戻す(L)

図 A—67 オプション ダイアログ([全般 - フォントと色]カテゴリ)

- 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]

- [ツール] メニュー→ [オプション…] を選択

[各エリアの説明]

(1) 色の設定エリア

色の設定を行うエリアです。

(a) [指定する箇所] エリア

表示色を指定したい箇所をリストー覧から選択します。 表示されるリストー覧の項目と、デフォルトで設定されている色の関係は次のとおりです。

RENESAS

項目	表示例			説明	
_{標準} 注	AaBbCc	文字色	黒	すべてのウインドウ/パネルにおいて、標準となる表	
		背景色	白	示色	
警告	AaBbCc	文字色	青	出力 パネルにおいて、警告メッセージの表示色、お	
		背景色	標準色	よびプロジェクト・ツリー パネルにおける "警告あ	
				り"ファイル名の表示色	
エラー	AaBbCc	文字色	赤	出力 パネルにおいて、エラー・メッセージの表示色	
		背景色	薄グレー	およびプロジェクト・ツリー パネルにおける"エ	
				ラーあり"ファイル名の表示色	
予約語	AaBbCc	文字色	茶	エディタ パネルにおいて、使用するコンパイラ/ア	
		背景色	標準色	センフラの予約語の表示色	
コメント	AaBbCc	文字色	緑	エディタ パネルにおいて、コメント部(C ソース・	
		背景色	標準色	ファイルの場合, "/*~*/")の表示色	
制御文字	AaBbCc	文字色	青緑	出力 パネルにおいて、制御文字の表示色	
		背景色	標準色		
文字列定数	AaBbCc	文字色	グレー	エディタ パネルにおいて、文字列定数の表示色	
		背景色	標準色		
強調	AaBbCc	文字色	白	プラグイン製品などにおいて、強調箇所の表示色	
		背景色	赤紫		
值変更	AaBbCc	文字色	薄茶	メモリ パネル /CPU レジスタ パネル/ローカル変数	
		背景色	クリーム	パネル /IOR パネル/ウォッチ パネルにおいて、プロ	
				グラムの実行により値が変更した箇所の表示色	
值編集	AaBbCc	文字色	青	メモリ パネル <i>I</i> CPU レジスタ パネル/ローカル変数	
		背景色	標準色	パネル /IOR パネル/ウォッチ パネルにおいて、ユー	
				サが強制的に値を変更した箇所の表示色	
PC 位置	AaBbCc	文字色	黒	エディタ パネル/逆アセンフル パネルにおいて、カ	
		背景色	山吹	レントPC位直のめる行の表示已	
ブレークポイ	AaBbCc	文字色	黒	エディタ パネル/逆アセンブル パネルにおいて、ブ	
ント		背景色	サーモンピンク	レークボイントが設定されている行の表示色	
リアルタイム	AaBbCc	文字色	ピンク	メモリ パネル/ウォッチ パネルにおいて、リアルタイ	
更新中		背景色	標準色	ム表示更新に設定されている領域の表示色	
リード/	AaBbCc	文字色	標準色	メモリ パネル/トレース パネルにおいて、リード、	
フェッチ		背景色	薄緑	またはフェッチされた箇所の表示色	
ライト	AaBbCc	文字色	標準色	メモリ パネル/トレース パネルにおいて、ライトされ	
		背景色	オレンジ	た箇所の表示色	
リード&ライ	AaBbCc	文字色	標準色	メモリ パネル/トレース パネルにおいて、リードと	
٢		背景色	薄青	ライトされた箇所の表示色	
カバレッジ	AaBbCc	文字色	標準色	エディタ パネル/逆アセンブル パネルにおいて	
100%		背景色	ライトグリーン	コード・カバレッジ率 100% の行の表示色	



項目	表示例			説明
カバレッジ1	AaBbCc	文字色	標準色	エディタ パネル/逆アセンブル パネルにおいて
~ 99%		背景色	ライトピンク	コード・カバレッジ率1~ 99% の行の表示色
カバレッジ	AaBbCc	文字色	標準色	エディタ パネル/逆アセンブル パネルにおいて
0%		背景色	ライトグレー	コード・カバレッジ率 0%(未実行)の行の表示色
無効	AaBbCc	文字色	グレー	メモリ パネルにおいて、メモリ・マッピングされて
		背景色	標準色	いない領域,およびプロジェクト・ツリー パネル上
				で実際存在しないファイル名の表示色

- 注 [標準]の文字色/背景色は、使用するホスト・マシンにおける Windows の設定に依存します。ここでは、Windows のデフォルト設定である"文字色:黒"、"背景色:白"を表記しています。
- (b) [標準の文字色を使用する]

>	[指定する箇所]エリアで選択している項目を、標準の文字色を使用して表示します。
	[指定する箇所] エリアで選択している項目の文字色を,任意に指定します。
	[文字色 …]ボタンが有効となります。

(c) [標準の背景色を使用する]

>	[指定する箇所]エリアで選択している項目を 標準の背景色を使用して表示します。
	[指定する箇所]エリアで選択している項目の背景色を、任意に指定します。
	[背景色] ボタンが有効となります。

(d)ボタン

ボタン	機能
文字色	色の設定 ダイアログがオープンし、[指定する箇所]エリアで選択している項目の文字色
	を指定します。
	ただし、[標準の文字色を使用する]をチェックしている場合は、無効となります。
背景色	色の設定 ダイアログがオープンし、[指定する箇所]エリアで選択している項目の背景色
	を指定します。
	ただし、[標準の背景色を使用する]をチェックしている場合は、無効となります。
色のリセット	[指定する箇所] エリアで選択している項目の色情報をリセットし、デフォルトの設定に
	戻します。



図 A---68 色の設定 ダイアログ

色の設定	28
基本色(B):	
作成した色(<u>C</u>):	
色の作成	(D) >>
OK \$++>>t	2/1/

(2) [表示例] エリア

色の設定エリアで指定した色とフォントの表示例を示します。

デフォルトでは、文字列"Aa あぁアァ亜宇"を表示しますが、テキスト・ボックスに任意の文字列を直接入 力することができます。

(3) [テキスト・エディタのフォント設定] エリア

[フォント...] ボタンを押下することにより、次のフォント ダイアログをオープンし、使用するテキスト・ エディタで使用するフォントを設定します。

フォント			2 🛛
フォント名(<u>F</u>): MS ゴシック	スタイル(Y): 標準	サイズ(<u>S</u>): 10	ОК
MSゴシック かMS 明朝 のMV Boli かNSimSun かOCRB かOMEJDIC のPalatino Linotype	▲	10 11 12 14 16 18 20	キャンセル
文字師り □ 取り消し線(½) □ 下線(凹)	サンブル Aaあぁ	アァ亜宇	
	文字セット(<u>R</u>): 日本語	~	•

図 A---69 フォント ダイアログ

(4) ボタン・エリア

インポート	ファイルに保存した設定内容をこのカテゴリに反映するためのオプション設定ファイ
	ルを開く ダイアログをオープンします。

エクスポート	このカテゴリの設定内容をファイルに保存するためのオプション設定ファイルを保存
	ダイアログをオープンします。
初期値に戻す	現在表示している項目の指定をすべてデフォルトに戻します。



[全般 - ビルド/デバッグ] カテゴリ

全般に関わる設定のうち、ビルド、またはデバッグに関連した設定を行います。



	オプション		
	 □ 全般 □ 起動と終了 □ まー 	全般 - ビルド/デバッグ	
(1) —	る 水 外部テキスト・エディク マーントレク	▶ 🔽 ラピッド・ビルドを行う(<u>R</u>) 📃 登録されたフ	ァイルの変更を監視する(<u>O</u>)
		ラビッド・ビルドを有効にすると、編集したソースファイルを信 集と平行してビルドが行われます。ラビッド・ビルドを行う場 Ctrl+Sでこまめに保存することをお勧めします。	呆存した時に、ビルドを開始し、編 殆合、一つのファイルの編集後に
(2)	□ □ その他	▶ 🔲 プロジェクト・ツリーに依存関係ファイルを表示する(P)	
(3) —		▶ 🗌 ビルド成功時に品質記録情報をファイル出力する(@)	
(4) —		▶ 🔲 エラーが上限に達したらビルドを停止する(₩)	上限数(山): 100 🛫
(5) —		★ 依存関係の更新タイミング(E):	初回ビルド時のみ 🔜
(6)		▶ 🔲 ダウンロードしているロード・モジュール・ファイルの変更を監視	見する(<u>D</u>)
(7)		▶ 🔽 ソース・ファイルを自動追加する(デバッグ専用プロジェクトの)	み)(<u>S)</u>
(8) —		▶ 🔲 ブレーク時にビープ音を鳴らす(<u>B</u>)	
(9)			→初期値に戻す①
		OK キャンセル	通用(A) ヘルプ(H)

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]

- [ツール] メニュー→ [オプション…] を選択

[各エリアの説明]

(1) [ラピッド・ビルドを行う]

>	ラピッド・ビルド機能 ^注 を有効にします(デフォルト)。
	ラピッド・ビルド機能を使用しません。

注 編集したソース・ファイルの保存時に、ビルドを自動で開始する機能です。
 この機能を有効にすることにより、ソース・ファイルの編集と同時にビルドを行うことができます。
 なお、この機能を使用する場合、ソース・ファイル編集後、こまめに上書き保存することを推奨します。

RENESAS

(a) [登録されたファイルの変更を監視する]

<	プロジェクトに登録されたソースファイルの変更を監視し、外部エディタなどで編集/保存された
	ときに、ラピッド・ビルドを開始します。
	プロジェクトに登録されたソースファイルの変更を監視し、外部エディタなどで編集/保存された
	ときに、ラピッド・ビルドを開始しません(デフォルト)。

- 備考 この項目は、[ラピッド・ビルドを行う]チェック・ボックスにチェックが付いている場合のみ有 効です。
- 注意1. この項目をチェックし、かつ、ラピッド・ビルドの対象となったファイルをビルド前に実行す るコマンド、ビルド後に実行するコマンドなどで自動で編集/上書きするように登録した場 合、ラピッド・ビルドが終了しなくなります。 ラピッド・ビルドが終了しなくなった場合は、本項目のチェックを外して、ラピッド・ビルド を停止してください。
 - この項目をチェックし、かつ、プロジェクトに登録されたソース・ファイルで存在しないファ イル(プロジェクト・ツリーでグレー表示されたファイル)がある場合、エクスプローラなど でファイルを再登録しても、監視状態にはなりません。監視状態にするためには、プロジェク ト・ファイルを読み込み直すか、またはこの項目のチェックを一旦外してダイアログを閉じた 後、再度この項目をチェックしてください。
- (2) [プロジェクト・ツリーに依存関係ファイルを表示する]

 マース・ファイルが依存しているファイル群をプロジェクト・ツリーに表示します。

 ソース・ファイルが依存しているファイル群をプロジェクト・ツリーに表示しません(デフォルト)。

(3) [ビルド成功時に品質記録情報をファイル出力する]

>	ビルド成功時に品質記録情報ファイルを出力します。
	ビルド成功時に品質記録情報ファイルを出力しません(デフォルト)。

- 備考1. 品質記録情報ファイルは、ラピッド・ビルドを行う場合、デバッグ専用プロジェクトをビルドする 場合、ファイル単位でコンパイル/アセンブルする場合は出力しません。
 - 2. 品質記録情報ファイルには、以下の情報を出力します。
 - ファイルの作成日時
 - ビルド結果のログ
 - ビルド中に使用したコマンド・ファイルの情報
 - 本製品の詳細バージョンや現在のプロジェクトの情報
 - 品質記録情報ファイルは、各プロジェクトのプロジェクト・フォルダに "品質記録情報(プロジェクト名,ビルド・モード名).txt"というファイル名で出力します。
 同名のファイルが存在する場合は上書きします。

また、プロジェクト・ツリーのビルド・ツール生成ファイル・ノードにも表示します。

(4) [エラーが上限に達したらビルドを停止する]

>	エラー数が[上限数]で指定した個数に達したときにビルドを停止します。
	エラー数が[上限数]で指定した個数に達してもビルドを停止しません(デフォルト)。

(a) [上限数]

エラーの上限数を指定します。

キーボードより1~10000の数値を直接入力するか(半角数字のみ), または 🤹 ボタンにより数値を 選択することで指定します。デフォルトでは100が指定されています。

- **備考** この項目は、[エラーが上限に達したらビルドを停止する] チェック・ボックスにチェックが付い ている場合のみ有効です。
- (5) [依存関係の更新タイミング]

依存関係の更新を自動で行うタイミングを、以下の中からドロップダウン・リストにより選択します。

初回ビルド時のみ	プロジェクトを開いたあと、初回のビルドを実行する直前に依存関係を更新します(デフォ
	ルト)。
ビルドごと	ビルドを実行する直前に依存関係を更新します。

(6) [ダウンロードしているロード・モジュール・ファイルの変更を監視する]

>	デバッグ・ツールにダウンロードしているロード・モジュール・ファイルの変更を監視し、変更があった 場合は、ダウンロードの実行を確認するメッセージ ダイアログを表示します。
	デバッグ・ツールにダウンロードしているロード・モジュール・ファイルの変更の監視を行いません(デ フォルト)。

(7) [ソース・ファイルを自動追加する (デバッグ専用プロジェクトのみ)]

>	デバッグ専用プロジェクトにおいて、デバッグ・ツールにロード・モジュール・ファイルをダウンロード
	する際、プロジェクト・ツリーにソース・ファイルを自動追加します(デフォルト)。
	デバッグ専用プロジェクトにおいて、デバッグ・ツールにロード・モジュール・ファイルをダウンロード
	する際、プロジェクト・ツリーへのソース・ファイルの自動追加を行いません。

- 注意 この機能は、プロジェクト・ツリーのダウンロード・ファイル・ノードにロード・モジュール・ファ イルを追加した場合のみ有効となります。デバッグ・ツールのプロパティ パネルの [ダウンロード・ ファイル設定] タブにてロード・モジュール・ファイルを追加した場合は、プロジェクト・ツリーに ソース・ファイルは追加されません。
- (8) [ブレーク時にビープ音を鳴らす]

プログラムの実行が、ブレーク・イベント(ハードウエア・ブレーク/ソフトウエア・ブレーク)により 停止した際、ビープ音を鳴らします。

~

RENESAS

「 プ 停

プログラムの実行が、ブレーク・イベント(ハードウエア・ブレーク/ソフトウエア・ブレーク)により 停止した際、ビープ音を鳴らしません(デフォルト)。

(9)ボタン・エリア

初期値に戻す	現在表示している項目をすべてデフォルトに戻します。



ダウンロードするファイルを選択 ダイアログ

ダウンロードするファイルの選択を行います。

図 A—71 タワンロートするファイルを選択 タイアロ

	ダウンロードするファイル	ルを選択						2
(1)_	ファイルの場所①:	🚞 sample			• ()	1 10	•	
	していた 最近使ったファイル							
	び デスクトップ							
2)—	אנאנדא <i>א</i> ד							
	ער דא 😨							
L	र् रा २७२७-७					-		
3)—		·ファイル名(<u>N</u>):	10.000		*	L	開く(C	2
4)		ファイルの種類(1):	[ロード・モジュール]	・ファイル (*.abs)	*	L	キャンセ	<i>w</i>

- 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]

- ダウンロード・ファイル ダイアログの [ダウンロード・ファイルのプロパティ] エリアにおいて, [ファイル] プロパティ内の [...] ボタンのクリック

[各エリアの説明]

(1) [ファイルの場所] エリア

ダウンロードするファイルの存在するフォルダを選択します。

(2) ファイル一覧エリア

[ファイルの場所],および [ファイルの種類] で選択された条件に合致するファイルの一覧を表示します。

(3) [ファイル名] エリア

ダウンロードするファイル名を指定します。

(4) [ファイルの種類] エリア

ダウンロードするファイルの種類(ファイル・タイプ)を選択します。

ロード・モジュール・ファイル (*.abs)	ロード・モジュール・フォーマット(デフォルト)
ヘキサ・ファイル (*.hex)	インテル拡張ヘキサ・フォーマット
S レコード・ファイル (*.mot)	モトローラ・Sタイプ・フォーマット
バイナリ・データ・ファイル (*.bin)	バイナリ・フォーマット
すべてのファイル (*.*)	すべてのファイル・フォーマット

ボタン	機能
開く	指定したファイルをダウンロード・ファイル ダイアログに追加します。
キャンセル	このダイアログをクローズします。



ウォッチ式データ・ファイルを開く ダイアログ

ウォッチ パネルにウォッチ式をインポートするファイルの選択を行います。

図 A—72	ウォッチ式デ	ータ・ファ	イルを開く	ダイアログ
--------	--------	-------	-------	-------

 (1) 「ファイルの場所の: image: sample image: sample		ウォッチ式データ・ファ	イルを開く					2 🔀
(2) $V_{2}^{(2)} = V_{2}^{(2)} V_{2}^{(2)} = V_{2}^{(2)} = V_{2}^{(2)} = V_{2}^{(2)} = V_{2}^{(2)} V_{2}^{(2)} = V_{2}^{(2)} $	(1)	ファイルの場所型:	🚞 sample		v ()	1 🖻		
$(2) - \begin{cases} \overrightarrow{r}, 2\beta + y\beta \\ \overrightarrow{r}, \gamma + \gamma + \gamma + \gamma \\ \overrightarrow{r}, \gamma + \gamma + \gamma - \beta \\ \overrightarrow{r}, \gamma + \gamma + \gamma - \beta \end{cases}$ $(3) - 7rf \mu A (W): \qquad \qquad$		<u> し</u> 最近使ったファイル						
マイコンピュータ マイコンピュータ マイネットワーク (3) ファイル名(W): 開((0))	(2)—	デスクトップ						
(3) マイネットワーク ファイル名(W): 開(()) (3)		ערביר אד						
	(3)	र्ग २७१७-७	ファイル名(<u>N</u>):		~		開(⊙)	

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]

- 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

[各エリアの説明]

(1) [ファイルの場所] エリア

インポートするファイルの存在するフォルダを選択します。

(2) ファイル一覧エリア

[ファイルの場所],および[ファイルの種類]で選択された条件に合致するファイルの一覧を表示します。

(3) [ファイル名] エリア

インポートするファイル名を指定します。

(4) [ファイルの種類] エリア

次のファイルの種類(ファイル・タイプ)が表示されます。

インボート可能 CSV(カンマ区切り) (*.csv) インポートが可能な CSV 形式	インポート可能 CSV(カンマ区切り)(*.csv)	インポートが可能な CSV 形式
--	----------------------------	------------------

ボタン	機能
開く	指定したファイルをウォッチ パネルにインポートします。
キャンセル	このダイアログをクローズします。



ファイルを開く ダイアログ

オープンするファイルの選択を行います。

図 A---73 ファイルを開く ダイアログ

	ファイルを開く					? 🔀
(1) -	ファイルの場所型:	🚞 sample		G Ø	• 🖽 🍤	
Ē	最近使ったファイル	🗀 DefaultBuild 📾 sample.mtpj				
	デスクトップ					
(2) -	71 F#1X2F					
	71 JUL1-9					
	الم جز جهال	- (1.5.0)				
(3)(4)	······	ファイル名(型): ファイルの種類(団):	プロジェクト・ファイル(*.mtpj)	~	「爾N型」 キャンセル

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]
- [機能ボタン]

[オープン方法]

- [ファイル] メニュー→ [ファイルを開く ...], または [エンコードを指定して開く ...]を選択

[各エリアの説明]

(1) [ファイルの場所] エリア

オープンするファイルが存在するフォルダを選択します。

はじめて選択する場合は"C: ¥ Documents and Settings ¥*ユーザ名*¥ My Documents", 2回目以降は前回選択し たフォルダが、デフォルトで選択されます。

(2) ファイル一覧エリア

[ファイルの場所],および [ファイルの種類] で選択した条件に合致するファイルの一覧を表示します。

(3) [ファイル名] エリア

オープンするファイルの名前を指定します。

(4)[ファイルの種類] エリア

オープンするファイルの種類(ファイル・タイプ)を選択します。

すべてのファイル (*.*)	すべての形式
プロジェクト・ファイル (*.mtpj)	プロジェクト・ファイル
e2 studio 用プロジェクト・ファイル (*.rcpc)	e ² studio 用プロジェクト・ファイル
CubeSuite 用プロジェクト・ファイル (*.cspj)	CubeSuite 用プロジェクト・ファイル
HEW 用ワークスペース・ファイル (*.hws)	HEW 用ワークスペース・ファイル
HEW 用プロジェクト・ファイル (*.hwp)	HEW 用プロジェクト・ファイル
PM+ 用ワークスペース・ファイル (*.prw)	PM+ 用ワークスペース・ファイル
PM+ 用プロジェクト・ファイル (*.prj)	PM+ 用プロジェクト・ファイル
C ソース・ファイル (*.c)	Cソース・ファイル
ヘッダ・ファイル (*.h; *.inc)	ヘッダ・ファイル
アセンブリ・ソース・ファイル (*.asm; *.s; *fsy)	アセンブリ・ソース・ファイル
リンク・マップ・ファイル (*.map; *.lbp)	リンク・マップ・ファイル
スタック情報ファイル (*.sni)	スタック情報ファイル
インテル拡張へキサ・ファイル (*.hex)	インテル拡張へキサ・ファイル
モトローラ・Sタイプ・ファイル(*.mot)	モトローラ・Sタイプ・ファイル
テキスト・ファイル (*.txt)	テキスト形式

ボタン	機能
開く	- [ファイル]メニュー→[ファイルを開く …]からオープンした場合 指定」たファイルをオープン」ます
	「コンビスティルショーンシビステ。 - [ファイル]メニュー→ [エンコードを指定して開く] からオープンした場合 ファイル・エンコードの選択 ダイアログをオープンします。
キャンセル	このダイアログをクローズします。



名前を付けて保存 ダイアログ

編集中のファイル、または各パネルの内容を名前を付けてファイルに保存します。

図 A—74 名前を付けて保存 ダイアログ

	名前を付けて保存					2 🛛
(1)-	保存する場所①:	🚞 sample	 ~	0 🗊	• 🛄	
ſ	して 最近使ったファイル					
	ごう デスクトップ					
2)-	کر ۲۲ ۴キ۱/۲۶					
	ער גאר אד אי-גאעב אד					
	マイ ネットワーク				<u></u>	
,) —		ファイル治(N):		~	1未仔	

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 ファイル名を保存 ...]を選択

- CPU レジスタ 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 CPU レジスタ・データ を保存 ...]を選択
- 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
- IOR 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 IOR データ を保存 ...]を選択
- コール・スタック 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 コール・スタック・ データ を保存...]を選択
- ローカル変数 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 ローカル変数データ を 保存...]を選択
- 出力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 タブ名を保存…]を選択

[各エリアの説明]

- (1) [保存する場所] エリア ファイルを保存するフォルダを選択します。
- (2) ファイル一覧エリア

[保存する場所]エリア,および[ファイルの種類]エリアで選択された条件に合致するファイルの一覧を表示します。

- (3) [ファイル名] エリア 保存する際のファイル名を指定します。
- (4) [ファイルの種類] エリア
 - (a) エディタパネルの場合

編集中のファイルの種類に依存して、次のファイルの種類(ファイル・タイプ)を表示します。

備考次の表示文字列はプロジェクト・ツリーに登録しているファイルのみが対象です。

C ソース・ファイル (*.c)	C ソース・ファイル
ヘッダ・ファイル (*.h;*.inc)	ヘッダ・ファイル
アセンブリ・ソース・ファイル (*.asm; *.s; *fsy)	アセンブリ・ソース・ファイル
リンク順指定ファイル (*.mtls)	リンク順指定ファイル
リンク・マップ・ファイル (*.map; *.lbp)	リンク・マップ・ファイル
インテル拡張ヘキサ・ファイル (*.hex)	インテル拡張ヘキサ・ファイル
アセンブル・リスト・ファイル (*.lst)	アセンブル・リスト・ファイル
スタック情報ファイル (*.sni)	スタック情報ファイル
モトローラ・S タイプ・ファイル (*.mot)	モトローラ・Sタイプ・ファイル
テキスト・ファイル (*.txt)	テキスト形式
CSV(カンマ区切り)(*.csv) ^{注1}	CSV 形式 ^{注 2}

- 注1. エディタ パネルが混合表示モードの状態で、このダイアログを呼び出した場合のみ表示されます。
 - 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)で括ります。また、アドレス/命令コード情報には"0x"を付与します。
- (b) CPU レジスタ パネル/ウォッチ パネル/IOR パネル/コール・スタック パネル/ローカル変数 パネルの
 - 場合

次のファイルの種類(ファイル・タイプ)を表示します。

ドロップダウン・リストより選択したファイル形式でパネルの内容をファイルに保存します。

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1}
インポート可能 CSV(カンマ区切り)(*.csv) 注2	インポートが可能な CSV 形式 ^{注 1}

注1. 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。

2. ウォッチ パネルからこのダイアログを呼び出した場合のみ表示されます。

(c)出力パネルの場合

次のファイルの種類(ファイル・タイプ)を表示します。

テキスト形式でのみ保存することができます。

テキスト・ファイル (*.txt)	テキスト形式(デフォルト)
-------------------	---------------

ボタン	機能	
保存	指定したファイル名でファイルを保存します。	
キャンセル	このダイアログをクローズします。	



データ保存ファイルを選択 ダイアログ

データを保存するファイルの選択を行います。

(1)- $7rf\mu0場所 @: sample ③ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @$	_	データ保存ファイルを読	業択				? 🔀
(2) $-\begin{bmatrix} & & & & & & \\ & & & & & & \\ & & & & & &$	(1)-	ファイルの場所①:	🚞 sample	 *	G	🌶 🖻 🛄	
$(2) - \begin{cases} \vec{r} \cdot \lambda p \cdot y \cdot p \\ \vec{r} \cdot \vec{r} \cdot$		しています。 最近使ったファイル					
マイコンピュータ マイコンピュータ マイネットワーク (3) ファイル名(い): 開(〇)	(2)-	デスクトップ					
マイネットワーク (3)		ער בארב אד א- בארב אד					
	(3)	२१ २७४७-७	・ファイル名(<u>N</u>):		~	開K (2

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]

- データ保存 ダイアログの [ファイル名] エリアにおいて […] ボタンをクリック

[各エリアの説明]

(1)[ファイルの場所] エリア

保存するファイルの存在するフォルダを選択します。

(2) ファイル一覧エリア

[ファイルの場所],および[ファイルの種類]で選択された条件に合致するファイルの一覧を表示します。

(3) [ファイル名] エリア

保存するファイル名を指定します。

(4) [ファイルの種類] エリア

保存するファイルの種類(ファイル・タイプ)を選択します。 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

インテル・ヘキサ・フォーマット (*.hex)	インテル拡張ヘキサ・フォーマット
モトローラSフォーマット (*.mot)	モトローラ・Sタイプ・フォーマット
バイナリ・データ (*.bin)	バイナリ・フォーマット

備考 アップロードについての詳細は、「2.5.3 アップロードを実行する」を参照してください。

ボタン	機能
開く	指定したファイルをデータ保存 ダイアログに指定します。
キャンセル	このダイアログをクローズします。



オプション設定ファイルを開く ダイアログ

オプションダイアログの [全般-フォントと色] カテゴリにインポートするオプション設定ファイルを選択します。

	オプション設定ファイ	ルを開く					2 🛛	
(1)	ファイルの場所(1):	📋 マイ ドキュメント		~	G 🗊	📂 🛄•		
	していたファイル 最近使ったファイル	🕮 マイ ピクチャ 🛃 マイ ミュージック						
	デスクトップ							
(2)-	المربح (المربح) المربح المربح							
	ערב אד קר באניב אד							
		- (1.50)				(_
(3)		·ファイル名(N):	フォントと色.mtpu		~	開	KW J	□ ↓ ± ± ± ± ± ± ± ± ± ± ± ± ± ± ± ± ± ±
(4)		ファイルの種類(工):	オプション設定ファイル (*.mtpu)		*	++	ンセル	

図 A---76 オプション設定ファイルを開く ダイアログ

- 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]
 - [機能ボタン]

[オープン方法]

- オプション ダイアログの [全般 - フォントと色] カテゴリにおいて, [インポート ...] ボタンをクリック

[各エリアの説明]

(1) [ファイルの場所] エリア

オプション設定ファイルが存在するフォルダを選択します。 はじめて選択する場合は "C: ¥ Documents and Settings ¥*ユーザ名*¥ My Documents", 2 回目以降は前回選択し たフォルダが, デフォルトで選択されます。

(2) ファイルの一覧エリア

[ファイルの場所],および [ファイルの種類] で選択された条件に合致するファイルの一覧を表示します。

(3) [ファイル名] エリア

オプション設定ファイルのファイル名を指定します。

(4) [ファイルの種類] エリア

次のファイルの種類(ファイル・タイプ)が表示されます。

オプション設定ファイル (*.mtpu)	オプション設定ファイル
----------------------	-------------

ボタン	機能
開く	指定したファイルをオプション ダイアログの [全般 - フォントと色] カテゴリにイン ポートします。
キャンセル	このダイアログをクローズします。



オプション設定ファイルを保存 ダイアログ

オプション ダイアログの [全般 - フォントと色] カテゴリの設定内容をオプション設定ファイルに保存します。

	オプション設定ファイ	ルを保存					2 🖬
1)	保存する場所(1):	📋 マイ ドキュメント		~	0 0	• 🖽	
2)-	よび使ったファイル ほび使ったファイル デスクトップ マイドキュメント マイドキュメント マイニンピュータ	ের্বা ৫০ 5 २ র্বা ≷⊥ − উ%০					
(3)	२२ २००००००	ファイル名(<u>N</u>):	フォントと色.mtpu		~	保存	s
4)	•	ファイルの種類(工):	オプション設定ファイル (*.mtpu)		*	キャン	セル

図 A---77 オプション設定ファイルを保存 ダイアログ

- 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]
 - [機能ボタン]

[オープン方法]

- オプション ダイアログの [全般 - フォントと色] カテゴリにおいて, [エクスポート …] ボタンをクリック

[各エリアの説明]

(1) [保存する場所] エリア

オプション設定ファイルを保存するフォルダを選択します。

はじめて選択する場合は"C: ¥ Documents and Settings ¥*ユーザ名*¥ My Documents", 2 回目以降は前回選択し たフォルダが、デフォルトで選択されます。

(2) ファイルの一覧エリア

[保存する場所],および[ファイルの種類]で選択された条件に合致するファイルの一覧を表示します。

(3) [ファイル名] エリア

オプション設定ファイルのファイル名を指定します。

(4) [ファイルの種類] エリア

次のファイルの種類(ファイル・タイプ)が表示されます。

オプション設定ファイル (*.mtpu) オプション設定ファィ

ボタン	機能		
保存	指定したファイル名でオプション設定ファイルを保存します。		
キャンセル	このダイアログをクローズします。		



付録 B 索 引

[A]

Auto 変数 … 267

【**C】** CPU レジスタ パネル … 255

【D】 DMM 機能 … 99

【I】 ID 認証用のキーコード … 25,33 IOR パネル … 259

【L】 Low pin debug interface … 12, 13 LPD 通信方式 … 12, 13

【P】 PC をここに設定 … 81 Print Preview ウインドウ … 343 [Printf イベント] タブ … 325 Printf イベント … 154, 301, 302, 325

【R】
RAM モニタ機能 … 98
Register 変数 … 267
RRM 機能 … 98
Run-Beak 時間 … 145
Run-Break タイマ … 145, 301, 302

【あ行】 アクション・イベント … 154 アクション・イベント ダイアログ … 323 [Printf イベント] タブ … 325 アップロード … 45,55 アップロード可能なファイル形式 … 55 アドレス・オフセット設定 ダイアログ … 335 イベント・エリア … 214,246 イベント種別 … 301

イベントの管理 … 157 イベントのアドレスにジャンプする … 159 イベントを削除する … 159 実行中に設定/削除可能なイベント種別 … 160 設定したイベントを削除する … 159 設定状態(有効/無効)を変更する … 158 特定のイベント種別のみ表示する … 158 有効イベント数の制限 … 160 イベントの設定状態 … 158 イベントパネル … 299 イベント・マーク … 300 印刷アドレス範囲設定 ダイアログ … 341 インテル拡張ヘキサ・ファイル … 51 ウインドウ・リファレンス … 172 ウォッチ式 … 112,272 ウォッチ式データ・ファイルを開く ダイアログ … 378 ウォッチ式として登録可能な対象 … 275 ウォッチ式の登録方法 … 113 ウォッチ式の表示形式 … 279 ウォッチ パネル … 272 エディタパネル … 209 エンコード … 329 オプション設定ファイルを開く ダイアログ … 387 オプション設定ファイルを保存 ダイアログ … 389 オプションダイアログ … 365 「全般-ビルド/デバッグ]カテゴリ … 372 [全般 - フォントと色] カテゴリ … 367 オプション・バイト … 319 オフセット値 … 248,293 オンチップ・デバッグ・セキュリティ ID … 25,33

【か行】

改行コード … 329 カバレッジ測定 … 150 カバレッジ測定結果を表示する … 151 カバレッジ測定の設定をする … 150 カレントPC 位置 … 247



付録B 索 引

カレント PC マーク … 215, 247	指定アドレスへ移動 … 69
関数のエピローグ … 267	指定位置へ移動 ダイアログ … 359
関数のプロローグ … 267	指定行へ移動 … 63
関数ヘジャンプ ダイアログ … 357	指定行へのジャンプ ダイアログ … 355
逆アセンブル結果の表示 … 67	出力パネル … 309
逆アセンブル・テキスト … 67, 244	初期化データ … 337
逆アセンブル パネル … 244	処理中表示 ダイアログ … 364
逆アセンブル表示モード … 68, 136, 294	シンボル定義箇所へ移動 … 69
キャッシュ・ヒット率 … 310	スクロール範囲設定 ダイアログ … 353
強制ブレーク機能 … 84	スコープの指定 … 110
共用体 … 267, 274	スタックからの関数呼び出し情報の表示 … 120
グローバル変数 … 110	コール・スタック情報を表示する … 120
クロックの設定 … 15, 22, 30, 38	コール・スタック情報を保存する … 121
構造体 … 267, 274	ステータスバー … 179
コード … 248	ステップ・イン実行 … 82, 178
コード・カバレッジ測定 … 150	ステップ・オーバ実行 … 82, 178
コール・スタック情報 … 286	ステップ実行 … 81
コール・スタック パネル … 286	スマート・エディット機能 … 218
ここまで実行 … 81	セキュリティ ID … 192
混合表示モード … 59, 68, 136, 294	接続方法 … 43
コンフィギュレーション … 39	切断方法 … 43
	[全般 - ビルド/デバッグ]カテゴリ … 372
【さ行】	[全般 - フォントと色]カテゴリ … 367
システム・レジスタ … 104, 255	ソース表示モード … 136, 294
実行時間の計測 … 145	[ソース・レベル] タブ … 350
実行開始から停止までの実行時間を計測する …	ソース・レベル・デバッグ … 57, 209
145	ソフトウエア・ブレーク … 86, 301, 302
測定可能時間の範囲 … 149	ソフトウエア・ブレーク機能 … 85
任意区間の実行時間を計測する … 146	
実行履歴の収集 … 122	【た行】
実行開始から停止までの実行履歴を収集する …	ターゲット・ボード … 16
128	タイマ … 41
実行履歴の収集を停止/再開する … 134	タイマ開始 … 301
実行履歴の表示内容を保存する … 143	タイマ開始イベント … 146, 302
実行履歴を表示する … 134	タイマ計測 … 301,302
条件を満たしたときのみの実行履歴を収集する …	タイマ計測イベント … 146
132	タイマ終了 … 301
トレース・データを検索する … 137	タイマ終了イベント … 146,302
トレース動作の設定をする … 122	ダウンロード … 45,204
トレース・メモリをクリアする … 137	ダウンロード可能なファイル … 47
任意区間の実行履歴を収集する … 129	ダウンロード条件 … 50, 314



ダウンロードするファイルを選択 ダイアログ … 376	表示桁数設定 ダイアログ … 333
ダウンロード・ファイル ダイアログ … 314	ビルトイン・イベント … 128, 302
タグ・ジャンプ … 65, 218, 310	ファイル・エンコードの選択 ダイアログ … 328
通常表示モード … 58	ファイルの監視機能 … 218
データ・カバレッジ測定 … 150	ファイルの保存設定 ダイアログ … 329
データ保存 ダイアログ … 361	ファイルを開く ダイアログ … 380
データ保存ファイルを選択 ダイアログ … 385	フック処理 … 207
テキスト編集 ダイアログ … 322	フック処理を設定する … 163
デバッグ情報 … 205	ブックマーク ダイアログ … 331
デバッグ専用プロジェクト … 45	フラッシュ・オプションの設定 ダイアログ … 318
デバッグ・ツールとの接続/切断 … 43	ブレーク … 27, 35
デバッグ・ツールバー … 177	ブレーク・イベント … 88,90
デバッグ・マネージャ パネル … 182	ブレークポイント … 86
動作環境設定 … 14	ブレークポイントの削除 … 88,90
特長 … 7	ブレークポイントの種別 … 197
トレース・イベント … 129	ブレーク要因 … 93, 294
トレース開始 … 301	プログラム内にアクションを挿入する … 154
トレース開始イベント … 129, 302	printf を挿入する … 154
トレース検索 ダイアログ … 345	プログラムの実行 … 79
[ソース・レベル] タブ … 350	プログラムを実行する … 80
[命令レベル]タブ … 347	プログラムをステップ実行する … 81
トレース終了 … 301	プログラムの停止 … 84
トレース終了イベント … 129, 302	任意の場所で停止する … 86
トレース動作の設定 … 122	任意の場所で停止する(ブレーク・イベント) …
トレース パネル … 290	88
トレース番号 … 291	プログラムの実行を手動で停止する … 85,86
トレース・フレーム … 128	変数へのアクセスで停止する … 90
トレース・メモリ … 123, 126, 199	プログラムの表示と変更 … 57
トレース・タイム・タグ … 199	逆アセンブルを表示する … 67
	ソース・ファイルを表示する … 57
【な行】	他の処理と平行してビルドを実行する … 71
内部スタティック変数 … 267	ライン・アセンブルを行う … 72
名前を付けて保存 ダイアログ … 382	プログラム・レジスタ … 104, 255
入力信号のマスク … 20,27,35	プロジェクト・ツリー パネル … 184
【什行】	プロセッサ・エレメント … 10, 76, 182
いードウェア・ブレーク … 86 301	プロパティ パネル … 186
ハードウェア・ブレーク機能 … 84	[接続用設定] タブ … 189
バイナリ・ファイル … 51	[ダウンロード・ファイル設定]タブ … 204
パイプライン … 292	[デバッグ・ツール] タブ … 194
ホーファーン 232 配列 … 267 274	[フック処理設定] タブ … 207
	[フラッシュ・オプション設定]タブ … 206

RENESAS

CubeSuite+ V2.01.00

付録B 索 引

リセット … 79

ローカル変数 … 110

ローカル変数 パネル … 266

リターン・アウト実行 … 83,178

ロード・モジュール・ファイル … 45

ベリファイ … 196 変数値のポップアップ表示 … 223 変数へのアクセス … 90 ポインタ型変数 … 267,274 ポイント・トレース … 302 ポイント・トレース・イベント … 132,234,252

【ま行】

無条件トレース … 302 無条件トレース・イベント … 128, 158 [命令レベル] タブ … 347 命令レベル・デバッグ … 57,209 メイン・ウインドウ … 174 メニューバー … 174 メモリ・アクセス … 19, 26, 34, 40 メモリ検索 ダイアログ … 339 メモリ初期化 ダイアログ … 337 メモリの設定 … 18,40 メモリ・マッピング … 18, 25, 33, 40 メモリ・マッピング ダイアログ … 312 メモリ、レジスタ、変数の表示/変更 … 94 CPU レジスタを表示/変更する … 104 IOR を表示/変更する … 106 ウォッチ式を表示/変更する … 112 グローバル変数・スタティック変数を表示/変更す る … 110 メモリを表示/変更する … 94 ローカル変数を表示/変更する … 110

モトローラ・Sタイプ・ファイル … 51

【や行】

有効イベント数の制限 … 160 優先するブート・ローダ・プロジェクトの選択 ダイアロ グ … 320 読み込み保護対象 … 262

【ら行】

ライン・アセンブル … 72,248
ラピッド・ビルド機能 … 71
ラベル名 … 248,293
リアルタイム表示更新機能 … 97,117
リサイクル・モード … 60



改訂記録

Rev	改行口	改訂内容		
itev.	光 打口	ページ	ポイント	
1.00	2013.09.01	_	初版発行	

CubeSuite+ V2.01.00 ユーザーズマニュアル RH850 デバッグ編

- 発行年月日 2013年9月1日 Rev.1.00
- 発行 ルネサス エレクトロニクス株式会社〒211-8668 神奈川県川崎市中原区下沼部 1753
RENESAS

ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

http://www.renesas.com

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。 ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2(日本ビル) (03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。 総合お問合せ窓口:http://japan.renesas.com/contact/

© 2013 Renesas Electronics Corporation and Renesas Solutions Corp. Colophon 1.1

CubeSuite+ V2.01.00

