

CubeSuite+ V2.00.00

統合開発環境

ユーザーズマニュアル V850 デバッグ編

対象デバイス

V850 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。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
 2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
 3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
 4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
 5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、
家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、
防災・防犯装置、各種安全装置等
- 当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
 7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
 8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
 9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事情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と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、V850 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、V850 ファミリの統合開発環境（ソフトウェア開発における、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 ウィンドウ・リファレンス](#)

[付録B ユーザ・オープン・インタフェース](#)

[付録C 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 ユーザーズ・マニュアル	起動編	R20UT2444J	R20UT2444E
	V850 設計編	R20UT2134J	R20UT2134E
	RL78 設計編	R20UT2136J	R20UT2136E
	78K0R 設計編	R20UT2137J	R20UT2137E
	78K0 設計編	R20UT2138J	R20UT2138E
	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)	R20UT2139J	R20UT2139E
	RL78, 78K0R コーディング編	R20UT2140J	R20UT2140E
	78K0 コーディング編	R20UT2141J	R20UT2141E
	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
	RX デバッグ編	R20UT2350J	R20UT2350E
	V850 デバッグ編	このマニュアル	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	R20UT2447J	R20UT2447E	
メッセージ編	R20UT2448J	R20UT2448E	

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 … 8

- 1.1 概 要 … 8
- 1.2 特 長 … 8

第2章 機 能 … 10

- 2.1 概 要 … 10
- 2.2 デバッグを始める前の準備 … 13
 - 2.2.1 ホスト・マシンとの接続を確認する … 13
- 2.3 デバッグ・ツールの動作環境設定 … 17
 - 2.3.1 使用するデバッグ・ツールを選択する … 17
 - 2.3.2 【IECUBE】 / 【IECUBE2】 の場合 … 18
 - 2.3.3 【MINICUBE】 / 【MINICUBE2】 の場合 … 39
 - 2.3.4 【E1】 の場合 … 55
 - 2.3.5 【E20】 の場合 … 71
 - 2.3.6 【EZ Emulator】 の場合 … 87
 - 2.3.7 【シミュレータ】 の場合 … 96
- 2.4 デバッグ・ツールとの接続／切断 … 106
 - 2.4.1 デバッグ・ツールを接続する … 106
 - 2.4.2 デバッグ・ツールを切断する … 106
 - 2.4.3 デバッグ・ツールをホット・プラグイン接続する【E1/E20(LPD)/(JTAG)【V850E2】】
… 107
- 2.5 ダウンロード／アップロード … 109
 - 2.5.1 ダウンロードを実行する … 109
 - 2.5.2 応用的なダウンロード方法 … 113
 - 2.5.3 アップロードを実行する … 123
- 2.6 プログラムの表示と変更 … 125
 - 2.6.1 ソース・ファイルを表示する … 125
 - 2.6.2 逆アセンブル結果を表示する … 134
 - 2.6.3 他の処理と平行してビルドを実行する … 139
 - 2.6.4 ライン・アセンブルを行う … 140
- 2.7 プログラムの実行 … 143
 - 2.7.1 マイクロコントローラ（CPU）をリセットする … 144
 - 2.7.2 プログラムを実行する … 144
 - 2.7.3 プログラムをステップ実行する … 146
- 2.8 プログラムの停止（ブレーク） … 149
 - 2.8.1 プログラムの実行を手動で停止する … 149
 - 2.8.2 任意の場所で停止する（ブレークポイント） … 149
 - 2.8.3 任意の場所で停止する（ブレーク・イベント） … 153
 - 2.8.4 変数 I/O レジスタへのアクセスで停止する … 155
 - 2.8.5 不正な実行を検出して停止する【IECUBE】 … 159
 - 2.8.6 その他のブレーク要因 … 160

2.9	メモリ、レジスタ、変数の表示／変更	…	161
2.9.1	メモリを表示／変更する	…	161
2.9.2	CPUレジスタを表示／変更する	…	176
2.9.3	I/Oレジスタを表示／変更する	…	178
2.9.4	グローバル変数／スタティック変数を表示／変更する	…	181
2.9.5	ローカル変数を表示／変更する	…	181
2.9.6	ウォッチ式を表示／変更する	…	183
2.10	スタックからの関数呼び出し情報の表示	…	191
2.10.1	コール・スタック情報を表示する	…	191
2.11	実行履歴の収集【IECUBE】【IECUBE2】【シミュレータ】	…	193
2.11.1	トレース動作の設定をする	…	193
2.11.2	実行停止までの実行履歴を収集する	…	201
2.11.3	任意区間の実行履歴を収集する	…	202
2.11.4	条件を満たしたときのみの実行履歴を収集する	…	205
2.11.5	実行履歴の収集を停止／再開する	…	207
2.11.6	実行履歴を表示する	…	208
2.11.7	トレース・メモリをクリアする	…	210
2.11.8	トレース・データを検索する	…	210
2.11.9	実行履歴の表示内容を保存する	…	216
2.12	実行時間の計測	…	218
2.12.1	実行停止までの実行時間を計測する	…	218
2.12.2	任意区間の実行時間を計測する	…	219
2.12.3	測定可能時間の範囲	…	222
2.13	カバレッジの測定【IECUBE】【IECUBE2】【シミュレータ】	…	224
2.13.1	カバレッジ測定の設定をする	…	224
2.13.2	カバレッジ測定結果を表示する	…	227
2.14	プログラム内へのアクションの設定	…	230
2.14.1	printfを挿入する	…	230
2.15	イベントの管理	…	233
2.15.1	設定状態（有効／無効）を変更する	…	234
2.15.2	特定のイベント種別のみ表示する	…	234
2.15.3	イベントのアドレスにジャンプする	…	235
2.15.4	イベントを削除する	…	235
2.15.5	イベントにコメントを入力する	…	235
2.15.6	イベント設定に関する留意事項	…	236
2.16	フック処理を設定する	…	242
2.17	シミュレータ GUIの使用【シミュレータ】	…	244
2.17.1	マイコンの入出力波形を確認する	…	245
2.17.2	端子へ信号を入力する	…	246
2.17.3	シリアル通信を行う	…	247
2.17.4	ボタン/LED/レベル・ゲージなどの部品を使用する	…	248
2.18	入力値について	…	249
2.18.1	入力規約	…	249
2.18.2	シンボル名の入力補完機能	…	250
2.18.3	入力不備箇所に対するアイコン表示	…	251

付録 A ウィンドウ・リファレンス … 252

A.1	説明	…	252
-----	----	---	-----

付録 B ユーザ・オープン・インタフェース … 608

- B.1 概 要 … 608
 - B.1.1 インタフェース関数の種類 … 609
 - B.1.2 インタフェース方式 … 609
 - B.1.3 開発環境 … 610
- B.2 ユーザ・モデルの作成 … 611
 - B.2.1 プログラム構成 … 611
 - B.2.2 ユーザ・モデルのプログラミング … 612
 - B.2.3 プログラム・ファイル (UserModel.c) の記述例 … 614
 - B.2.4 コンパイルとリンク … 615
- B.3 ユーザ・モデルの組み込み … 616
 - B.3.1 シミュレータ・コンフィギュレーション・ファイルへの記述 … 616
 - B.3.2 シミュレータ・コンフィギュレーション・ファイルの記述例 … 619
- B.4 提供インタフェース関数 … 620
 - B.4.1 概 要 … 620
 - B.4.2 基本インタフェース関数 … 622
 - B.4.3 時間インタフェース関数 … 626
 - B.4.4 端子インタフェース関数 … 635
 - B.4.5 外部バス・インタフェース関数 … 649
 - B.4.6 シリアル・インタフェース関数 … 657
 - B.4.7 信号出力器インタフェース関数 … 682
 - B.4.8 エラー番号一覧 … 690
- B.5 ユーザ定義関数 … 692
- B.6 サンプル・プログラム (Timer モデル) … 706
 - B.6.1 概 要 … 706
 - B.6.2 構 成 … 706
 - B.6.3 動 作 … 706
 - B.6.4 プロジェクト・ファイル … 707
 - B.6.5 プログラム詳細 … 708

付録 C 索 引 … 712

第1章 概 説

CubeSuite+ は、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、V850 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。

- 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 (IECUBE/IECUBE2)、オンチップ・デバッグ・エミュレータ (MINICUBE/MINICUBE2/E1/E20/EZ Emulator)、およびシミュレータと組み合わせて使用することにより、より快適な開発環境を実現できます。

- C 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 (コード・フラッシュ) の対応

【V850E1】【V850ES】

IECUBE と接続することにより、フラッシュ・セルフ・プログラミング・エミュレーションを行うことができます。

【V850E2】

接続可能なすべてのデバッグ・ツールで、フラッシュ・セルフ・プログラミング・エミュレーション (オプション・バイトを含む) を行うことができます。

- データフラッシュ・メモリへの書き込み

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

- リアルタイム表示更新機能

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

- デバッグ環境の保存／復元

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

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 CA850/CX 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 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 (IECUBE/IECUBE2/MINICUBE/MINICUBE2/E1/E20/EZ Emulator/ シミュレータ) を接続します。

(5) 使用するデバッグ・ツールを選択する

プロジェクトで使用するデバッグ・ツールを選択します。

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

(6) デバッグ・ツールの動作環境設定を行う

(5) で選択したデバッグ・ツールの動作環境を設定します。

- [【IECUBE】](#) / [【IECUBE2】](#) の場合

- 【MINICUBE】 / 【MINICUBE2】 の場合
- 【E1】 の場合
- 【E20】 の場合
- 【EZ Emulator】 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・ツールへダウンロードします。

(9) ソース・ファイルを表示する

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

(10) プログラムを実行する

目的に応じた実行方法により、プログラムを実行します。

なお、実行したプログラムを任意の箇所で停止する場合は、あらかじめブレークポイント／ブレーク・イベント^注を設定しておきます（「2.8.2 任意の場所で停止する（ブレークポイント）」 / 「2.8.3 任意の場所で停止する（ブレーク・イベント）」 / 「2.8.4 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には、「2.15.6 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 【V850E2】

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 *n*）をデバッグ・マネージャパネル【V850E2】上で指定してください。

(11) プログラムの実行を手動で停止する

実行したプログラムを停止します。

ただし、(10) 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【IECUBE】【IECUBE2】【シミュレータ】^注
- 実行時間の計測^注
- カバレッジの測定【IECUBE】【IECUBE2】【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には、「2.15.6 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
- シミュレータ GUI の使用【シミュレータ】

2. 取得した各種情報をファイルに保存することができます。

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

(13) アップロードを実行する

必要に応じ、プログラム（メモリ内容）を任意のファイル形式（ヘキサ・フォーマット／バイナリ・データ・フォーマットなど）で保存します。

(14) 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。

なお、プロジェクトで選択しているマイクロコントローラと接続可能なデバッグ・ツールの関係は次の表のとおりです。

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

表2—1 マイクロコントローラの種類と接続可能なデバッグ・ツールの関係

マイクロコントローラ	デバッグ・ツール						
	IECUBE	IECUBE2	MINICUBE E1/E20(JTAG)	MINICUBE2 E1/E20(Serial)	E1/E20(LPD)	シミュレータ	EZ Emulator
V850E1	○	—	○	○	—	○	○
V850ES	○	—	○	○	—	○	○
V850E2 (シングルコア)	—	○注	○	○注	○注	○	—
V850E2 (マルチコア)	—	—	○	—	—	—	—

注 【V850E2】

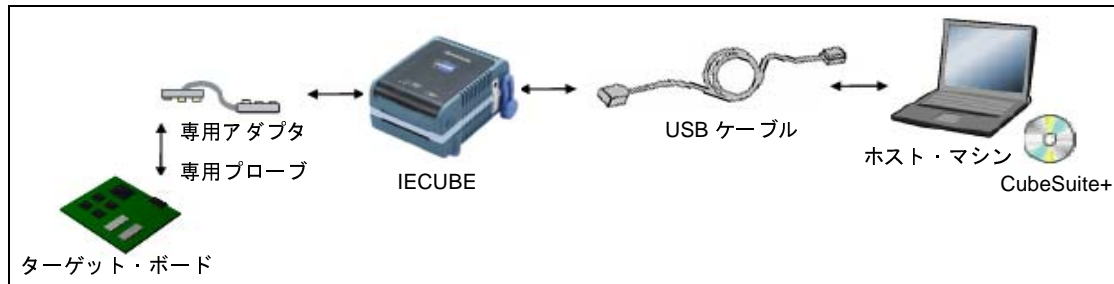
選択するマイクロコントローラの種類により、未サポートの場合があります。

- (1) 【IECUBE】の場合
- (2) 【IECUBE2】の場合
- (3) 【MINICUBE】の場合
- (4) 【MINICUBE2】の場合
- (5) 【E1】の場合
- (6) 【E20】の場合
- (7) 【EZ Emulator】の場合
- (8) 【シミュレータ】の場合

(1) 【IECUBE】の場合

ホスト・マシン、IECUBE、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、IECUBE のユーザーズ・マニュアルを参照してください。

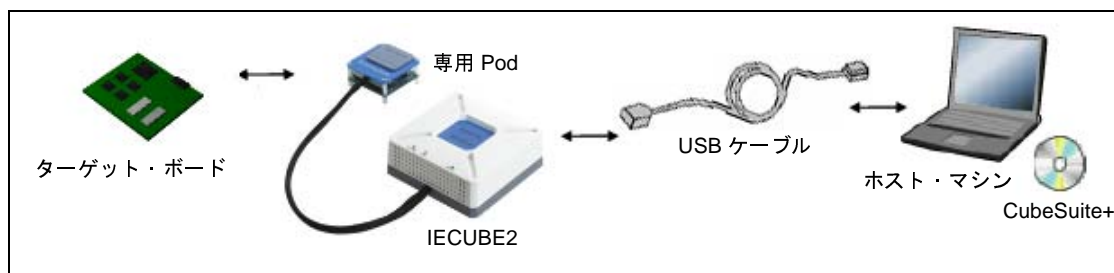
図 2—1 ホスト・マシンとデバッグ・ツールとの接続例【IECUBE】



(2) 【IECUBE2】の場合

ホスト・マシン、IECUBE2、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、IECUBE2 のユーザーズ・マニュアルを参照してください。

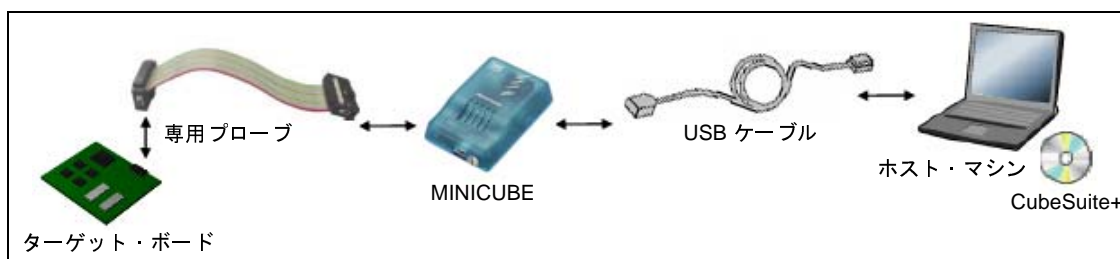
図 2—2 ホスト・マシンとデバッグ・ツールとの接続例【IECUBE2】



(3) 【MINICUBE】の場合

ホスト・マシン、MINICUBE、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、MINICUBE のユーザーズ・マニュアルを参照してください。

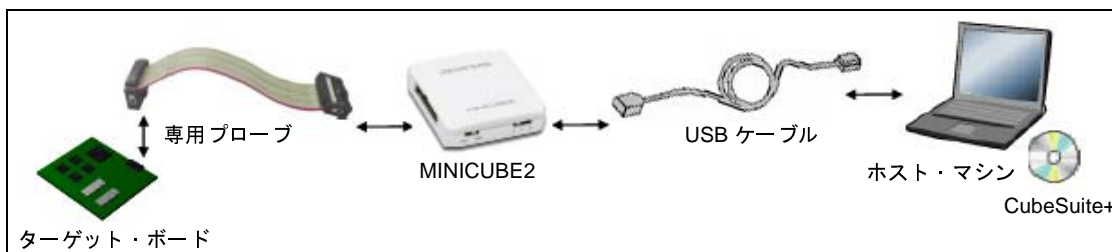
図 2—3 ホスト・マシンとデバッグ・ツールとの接続例【MINICUBE】



(4) 【MINICUBE2】の場合

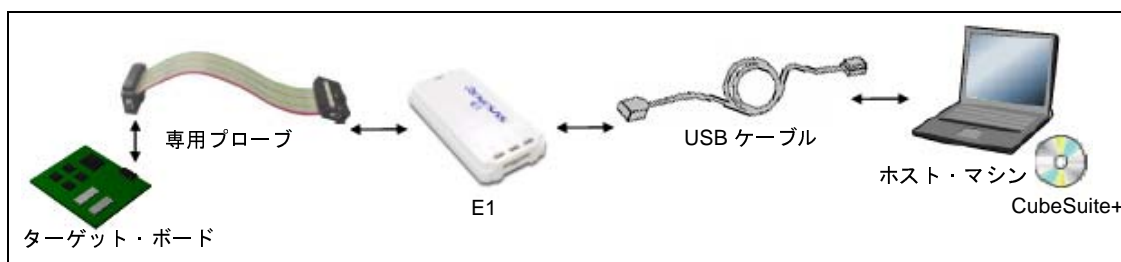
ホスト・マシン、MINICUBE2、および必要に応じてターゲット・ボードを接続します。
接続方法についての詳細は、MINICUBE2 のユーザーズ・マニュアルを参照してください。

図 2—4 ホスト・マシンとデバッグ・ツールとの接続例【MINICUBE2】

**(5) 【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—5 ホスト・マシンとデバッグ・ツールとの接続例【E1】

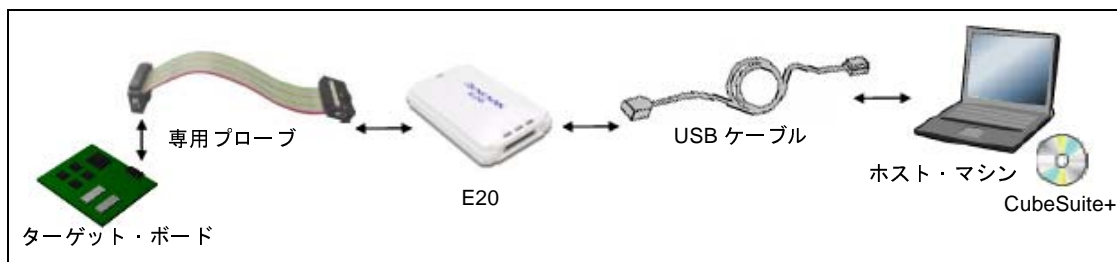


備考 ターゲット・ボードとの通信方式として、シリアル通信【E1(Serial)】、LPD 通信【E1(LPD)】、または JTAG 通信【E1(JTAG)】を選択することができます（「[2.3.1 使用するデバッグ・ツールを選択する](#)」参照）。

(6) 【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—6 ホスト・マシンとデバッグ・ツールとの接続例【E20】



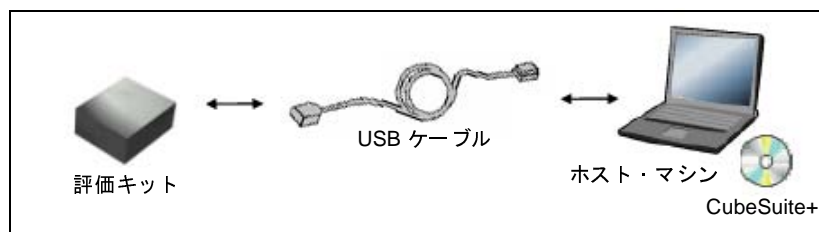
備考 ターゲット・ボードとの通信方式として、シリアル通信【E1(Serial)】、LPD 通信【E1(LPD)】、または JTAG 通信【E1(JTAG)】を選択することができます（「[2.3.1 使用するデバッグ・ツールを選択する](#)」参照）。

(7) 【EZ Emulator】の場合

ホスト・マシン、評価キットなどを接続します。

接続方法についての詳細は、EZ Emulator のユーザーズ・マニュアルを参照してください。

図 2—7 ホスト・マシンとデバッグ・ツールとの接続例【EZ Emulator】



(8) 【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—8 ホスト・マシンとデバッグ・ツールとの接続例【シミュレータ】



2.3 デバッグ・ツールの動作環境設定

この節では、各デバッグ・ツールの動作環境の設定方法について説明します。

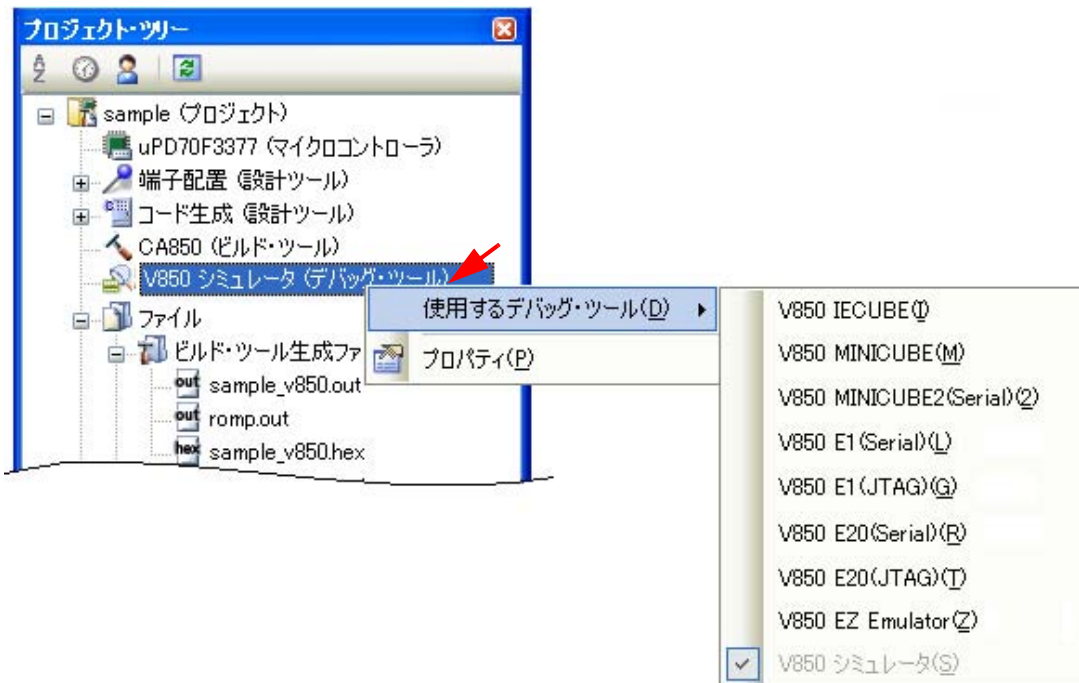
2.3.1 使用するデバッグ・ツールを選択する

デバッグ・ツールの動作環境設定は、使用するデバッグ・ツールに対応したプロパティパネルで行います。

そのため、まず、プロジェクト内で使用するデバッグ・ツールを選択します（使用するデバッグ・ツールはプロジェクトごとに選択可）。

使用するデバッグ・ツールの選択/切り替えは、プロジェクト・ツリーパネル上の [品種名 デバッグ・ツール名 (デバッグ・ツール)] ノードを右クリックすることで表示されるコンテキスト・メニューから行ってください。

図 2—9 使用するデバッグ・ツールの選択/切り替え



注意 表示されるコンテキスト・メニューは、プロジェクトで選択しているマイクロコントローラの種類により異なります（「表 2—1 マイクロコントローラの種類と接続可能なデバッグ・ツールの関係」参照）。

備考 【E1】【E20】

E1/E20 とターゲット・ボードとの通信方式として、シリアル通信を使用する場合は [品種名 E1(Serial)] / [品種名 E20(Serial)] を、LPD 通信を使用する場合は [品種名 E1(LPD)] / [品種名 E20(LPD)] を、また JTAG 通信を使用する場合は [品種名 E1(JTAG)] / [品種名 E20(JTAG)] を選択してください。

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。

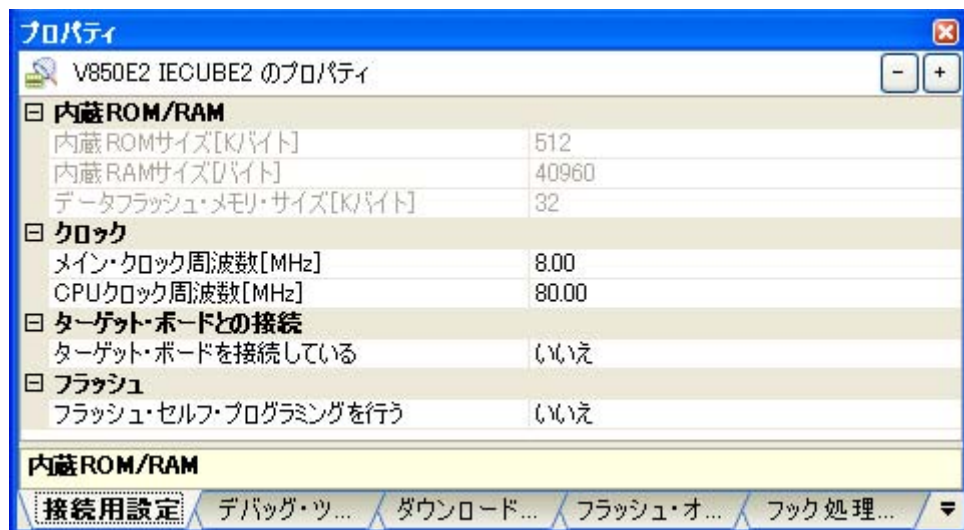
2.3.2 【IECUBE】 / 【IECUBE2】 の場合

IECUBE/IECUBE2 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—10 動作環境設定【IECUBE】（プロパティ パネルの例）



図 2—11 動作環境設定【IECUBE2】（プロパティ パネルの例）



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) [フラッシュ・セルフ・エミュレーション設定] タブ【IECUBE】
- (4) [データフラッシュ・エミュレーション設定] タブ【IECUBE】
- (5) [ダウンロード・ファイル設定] タブ

(6) [フラッシュ・オプション設定] タブ【IECUBE2】

(7) [フック処理設定] タブ

(1) [接続用設定] タブ

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

- (a) [内蔵 ROM/RAM]
- (b) [データフラッシュ]【IECUBE】
- (c) [クロック]
- (d) [ターゲット・ボードとの接続]
- (e) [プログラマブル I/O]【IECUBE】
- (f) [フラッシュ]【IECUBE2】

(a) [内蔵 ROM/RAM]

このカテゴリでは、内蔵 ROM/RAM に関する設定を行います。

デフォルトで、選択しているマイクロコントローラの内蔵 ROM/RAM サイズが設定されます（選択しているマイクロコントローラが V850E1 コアで、VSB Flash ROM/VSB RAM 搭載品の場合は、そのサイズを加算した値が設定されます）。

備考 選択しているマイクロコントローラと同様のメモリ・マッピングでデバッグを行う場合は、このカテゴリ内の設定を変更する必要はありません。

図 2—12 [内蔵 ROM/RAM] カテゴリ【IECUBE】

内蔵ROM/RAM	
内蔵ROMサイズ[K/バイト]	256
内蔵RAMサイズ[B/バイト]	16384

図 2—13 [内蔵 ROM/RAM] カテゴリ【IECUBE2】

内蔵ROM/RAM	
内蔵ROMサイズ[K/バイト]	512
内蔵RAMサイズ[B/バイト]	40960
データフラッシュ・メモリ・サイズ[K/バイト]	32

- [内蔵 ROM サイズ [K バイト]]

- 【IECUBE】

エミュレーションする内蔵 ROM サイズを指定します（単位：K バイト）。


IECUBE のメモリ資源を利用して、メモリ・マッピングを変更後にデバッグを行う場合は、ドロップダウン・リストにより指定してください。

ただし、選択しているマイクロコントローラが ROM レス品の場合は、[0] の値を変更することはできません。

- 【IECUBE2】
エミュレーションする内蔵 ROM サイズを表示します（単位：Kバイト）。
このプロパティ値を変更することはできません。
- [内蔵 RAM サイズ [バイト]]
 - 【IECUBE】
エミュレーションする内蔵 RAM サイズを指定します（単位：バイト）。
IECUBE のメモリ資源を利用して、メモリ・マッピングを変更後にデバッグを行う場合は、ドロップダウン・リストにより指定してください。
 - 【IECUBE2】
エミュレーションする内蔵 RAM サイズを表示します（単位：バイト）。
このプロパティ値を変更することはできません。
- [データフラッシュ・メモリ・サイズ [Kバイト]] 【IECUBE2】
選択しているマイクロコントローラのデータフラッシュ・メモリ領域のサイズを表示します（単位：Kバイト）。
このプロパティ値を変更することはできません。

注意 【IECUBE】

設定を変更する際は、他のメモリ・マッピング領域と重複しないよう注意が必要です。

なお、設定欄の  マークは、このプロパティ値の設定を変更すると、自動的に CPU リセットを生じることが示します。

(b) [データフラッシュ] 【IECUBE】

このカテゴリでは、データフラッシュ・メモリ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データフラッシュ・メモリ内蔵品の場合のみ表示されます。

図 2—14 [データフラッシュ] カテゴリ

☐ データフラッシュ	
データフラッシュ・メモリ領域を使用する	はい
データフラッシュ・メモリ・サイズ [Kバイト]	32
チップ・セレクト	CS0

- [データフラッシュ・メモリ領域を使用する]
データフラッシュ・メモリを使用するか否かをドロップダウン・リストにより指定します。
データフラッシュ・メモリを使用する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

- [データフラッシュ・メモリ・サイズ [K バイト]]

このプロパティは、[データフラッシュ・メモリ領域を使用する] 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。

選択しているマイクロコントローラのデータフラッシュ・メモリ領域のサイズを表示します（単位：K バイト）。

このプロパティ値を変更することはできません。

- [チップ・セレクト]

このプロパティは、[データフラッシュ・メモリ領域を使用する] プロパティにおいて、[はい] を指定した場合のみ表示されます。

データフラッシュ・メモリをマッピングする際のチップ・セレクトをドロップ・ダウンリストにより指定します。

なお、指定できるチップ・セレクトは、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、使用するマイクロコントローラに V850E1 コアを選択している場合、デフォルトで表示しているチップ・セレクトを変更することはできません。

(c) [クロック]

このカテゴリでは、クロックに関する設定を行います。

図 2—15 [クロック] カテゴリ [IECUBE]

目 クロック	
メイン・クロック周波数 [MHz]	4.00
メイン・クロック通倍	1
サブ・クロック周波数 [kHz]	32.765
モータ・クロック	システム

図 2—16 [クロック] カテゴリ [IECUBE2]

目 クロック	
メイン・クロック周波数 [MHz]	8.00
CPUクロック周波数 [MHz]	80.00

- [メイン・クロック周波数 [MHz]]

メイン・クロック周波数（通倍前）を指定します。

ドロップダウン・リストによる選択か、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。

ドロップダウン・リストには、次の周波数が表示されます（単位：MHz）。

- 【V850E1】【V850ES】

2.50, 4.00（デフォルト）, 5.00, 6.00, 8.00, 13.50, 18.00

- 【V850E2】

4.00, 5.00, 7.20, 8.00（デフォルト）, 9.60, 10.00, 13.50, 16.00

- [メイン・クロック通倍] 【IECUBE】

メイン・クロック周波数に対する通倍数を指定します。

ドロップダウン・リストによる選択か、または直接入力により、1～99の範囲の値を指定します。

ドロップダウン・リストには、次の通倍数が表示されます。

1 (デフォルト), 2, 3, 4, 5, 6, 7, 8, 9, 10

- [サブ・クロック周波数 [kHz]] 【IECUBE】

このプロパティは、選択しているマイクロコントローラがサブ・クロックをサポートしている場合にのみ表示されます。

サブ・クロック周波数を指定します。

ドロップダウン・リストによる選択か、または直接入力により、0.001～999.999 (単位: kHz) の範囲の周波数を指定します。

ドロップダウン・リストには、現在選択しているマイクロコントローラが使用可能なサブ・クロック周波数が表示されます (選択できる周波数が固定の場合はドロップダウン・リストは表示されません)。

デフォルトでは、使用可能なサブ・クロック周波数の最小値が指定されます。

- [モニタ・クロック] 【IECUBE】

このプロパティは、選択しているマイクロコントローラがサブ・クロックをサポートしている場合にのみ表示されます。

プログラム停止中にモニタ・プログラムが動作するクロックを次のドロップダウン・リストより指定します。

システム	メイン・クロックで動作します (デフォルト)。
ユーザ	プログラムで設定されているクロックで動作します。

- [CPU クロック周波数 [MHz]] 【IECUBE2】

トレースの時間表示のための CPU クロック周波数を指定します。

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 (デフォルトでは [80.00] が指定されます)。

(d) [ターゲット・ボードとの接続]

このカテゴリでは、ターゲット・ボードとの接続に関する設定を行います。

図 2—17 [ターゲット・ボードとの接続] カテゴリ 【IECUBE】 【IECUBE2】

☐ 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

- [ターゲット・ボードを接続している]

IECUBE/IECUBE2 にターゲット・ボードを接続しているか否かをドロップダウン・リストにより指定します。

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（デフォルトでは [いいえ] が指定されます）。

注意 IECUBE が CubeSuite+ に接続している場合、このプロパティ値を変更することはできません。

(e) [プログラマブル I/O] [IECUBE]

このカテゴリでは、プログラマブル I/O 領域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プログラマブル I/O 領域をサポートしている場合のみ指定可能です。

備考 デバッグを行う際に、一時的に値を変更する場合にのみ設定を変更してください。

プロジェクトとして共通の値を設定する場合は、使用するビルド・ツールのプロパティ パネルの [共通オプション] タブ上の [デバイス] カテゴリ内 [プログラマブル I/O 領域開始アドレス] プロパティで設定を行ってください。

図 2—18 [プログラマブル I/O] カテゴリ

プログラマブル I/O	
プログラマブル I/O 領域を使用する	はい
プログラマブル I/O 領域開始アドレス	HEX 0

- [プログラマブル I/O 領域を使用する]

プログラマブル I/O 領域を使用するか否かをドロップダウン・リストにより指定します。

プログラマブル I/O 領域を使用する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

- [プログラマブル I/O 領域開始アドレス]

このプロパティは、[プログラマブル I/O 領域を使用する] 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。

プログラマブル I/O 領域の開始アドレスを指定します。

直接入力により、プログラマブル I/O 領域内のアドレスを指定してください（デフォルトでは [0] が指定されます）。

なお、アドレスは、16 K バイトでアラインされます。

(f) [フラッシュ] [IECUBE2]

このカテゴリでは、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。

図 2—19 [フラッシュ] カテゴリ

フラッシュ	
フラッシュ・セルフ・プログラミングを行う	いいえ

- [フラッシュ・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、内蔵フラッシュ ROM の書き換えを行うか否かをドロップダウン・リストにより指定します。

内蔵フラッシュ ROM の書き換えを行う場合は、[はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

なお、[はい] を指定した場合、内蔵フラッシュ ROM のキャッシュは行われません。

注意 IECUBE2 が CubeSuite+ に接続している場合、このプロパティ値を変更することはできません。

(2) [デバッグ・ツール設定] タブ

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

- (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) [フェイルセーフ・ブレーク] [IECUBE]
- (f) [トレース]
- (g) [タイマ]
- (h) [カバレッジ]
- (i) [入力信号のマスク]
- (j) [外部フラッシュ・メモリ・ダウンロード]

(a) [メモリ]

このカテゴリでは、メモリに関する設定を行います。

図 2—20 [メモリ] カテゴリ [IECUBE] [IECUBE2]

☐ メモリ	
☐ メモリ・マッピング	[4]
☐ [0]	内蔵ROM領域
☐ [1]	ノン・マップ領域
☐ [2]	内蔵RAM領域
☐ [3]	I/Oレジスタ領域
メモリ書き込み時にベリファイを行う	はい

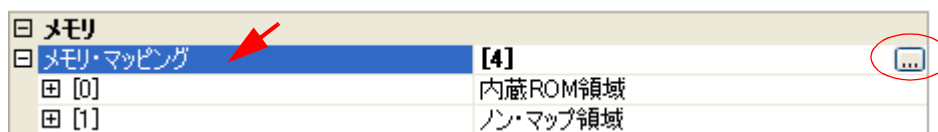
- [メモリ・マッピング]

現在のメモリ・マッピングの状況が、メモリ領域の種別ごとに詳細表示されます。

このパネル上でマッピング値を変更することはできません。メモリ・マッピングを追加する必要がある場合は、[メモリ・マッピング] プロパティを選択することで設定欄右端に表示される [...] ボタンのクリックによりオープンする **メモリ・マッピング ダイアログ** で行います。

設定方法についての詳細は、**メモリ・マッピング ダイアログ** の項を参照してください。

図 2—21 メモリ・マッピング ダイアログのオープン



注意 デバッグ・ツールと未接続の場合、ユーザにより追加されたメモリ・マッピング領域のみが表示対象となります。

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

- [メモリ書き込み時にベリファイを行う]

メモリ値の初期化を行う際に、ベリファイを行うか否かをドロップダウン・リストにより指定します。ベリファイを行う場合は「はい」を選択してください（デフォルト）。

(b) [実行中のメモリ・アクセス]

このカテゴリでは、プログラム実行中におけるメモリ・アクセス（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—22 [実行中のメモリ・アクセス] カテゴリ【IECUBE】

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

図 2—23 [実行中のメモリ・アクセス] カテゴリ【IECUBE2】

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

- [実行中にアクセスする]【IECUBE2】

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ドロップダウン・リストにより指定します。なお、[フラッシュ] カテゴリ内 [フラッシュ・セルフ・プログラミングを行う] プロパティにおいて [いいえ] を選択している場合は、内蔵 ROM 領域も対象となります。

アクセスを許可する場合は「はい」を選択してください（デフォルトでは「いいえ」が指定されます）。

- [実行を一瞬停止してアクセスする]

このプロパティは、[実行中にアクセスする] [IECUBE2] プロパティにおいて [いいえ] を指定した場合は表示されません。

プログラム実行中にはアクセスできないメモリ領域（ターゲット・メモリ領域 /I/O レジスタ領域 /CPU レジスタなど）に対して、実行を一瞬停止し、アクセスを許可するか否かをドロップダウン・リストにより指定します。

なお、[フラッシュ] カテゴリ内 [フラッシュ・セルフ・プログラミングを行う] プロパティにおいて [いいえ] を選択している場合は、内蔵 ROM 領域も内蔵 RAM 領域と同様の扱いとなります。

アクセスを許可する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されません）。

備考 【V850E2】

内蔵 RAM 領域に対しては、このプロパティの指定にかかわらず、一瞬停止することなくアクセスします。

- [HALT を解除してアクセスする]

このプロパティは、[実行を一瞬停止してアクセスする] 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。

HALT モードを解除して、内蔵 RAM 領域以外のメモリ（CPU レジスタ /I/O レジスタ）にアクセスするか否かをドロップダウン・リストにより指定します（内蔵 RAM 領域へのアクセスに対しては HALT モードを解除しません）。

なお、[フラッシュ] カテゴリ内 [フラッシュ・セルフ・プログラミングを行う] プロパティにおいて [いいえ] を選択している場合は、内蔵 ROM 領域も内蔵 RAM 領域と同様の扱いとなります。

HALT モードを解除してアクセスする場合は [はい] を選択してください（デフォルト）。

- [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**の表示内容を自動的に更新するか否かをドロップダウン・リストにより指定します。

表示内容の更新を行う場合は [はい] を選択してください（デフォルト）。

注意 【IECUBE】

[トレース] カテゴリ内 [トレース・データの用途] プロパティにおいて、[トレース] を指定している場合、このプロパティは無効となります（「[2.11.1](#) **トレース動作の設定をする**」参照）。

- [表示更新間隔 [ms]]

このプロパティは、[実行中に表示更新を行う] プロパティにおいて [はい] を指定した場合のみ表示されます。

プログラム実行中に、**ウォッチパネル**/**メモリパネル**の表示内容を自動的に更新する間隔を 100 ms 単位で指定します。

直接入力により、100 ~ 65500 の整数（100 ms 未満の端数切り上げ）を指定してください（デフォルトでは [500] が指定されます）。

- [リアルタイム表示更新を自動設定する]【IECUBE】

このプロパティは、[実行中に表示更新を行う] プロパティにおいて [はい] を指定した場合のみ表示されます。

ウォッチパネル／メモリパネルで表示している領域を、IECUBE で可能な限り自動的にリアルタイム表示更新機能の対象領域に設定し、プログラム実行中に表示内容を更新する場合は [はい] を選択してください（デフォルト）。

(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—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ドロップダウン・リストにより指定します。

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（デフォルトでは [いいえ] が指定されます）。

(d) [ブレーク]

このカテゴリでは、ブレーク機能に関する設定を行います。

図 2—25 [ブレーク] カテゴリ【IECUBE】【IECUBE2】

ブレーク	
優先的に使用するブレークポイントの種類	ソフトウェア・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
オープン・ブレーク機能を使用する	いいえ (Output signal)

- [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、選択しているマイクロコントローラが使用可能なブレークポイントの種類が1つのみの場合は表示されません。

エディタパネル／逆アセンブルパネルにおいて、マウスのワンクリック操作でブレークポイントを設定する際に使用するブレークポイントの種別を次のドロップダウン・リストにより指定します。

なお、ブレークポイントについての詳細は、「2.8.2 任意の場所で停止する (ブレークポイント)」を参照してください。

ソフトウェア・ブレイク	ソフトウェア・ブレイクポイントを優先的に設定します（デフォルト）。
ハードウェア・ブレイク	ハードウェア・ブレイ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ションを停止する]
プログラム実行停止時に、エミュレータの周辺エミュレーション機能を停止するか否かをドロップダウン・リストにより指定します。
停止する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。
- [オープン・ブレイク機能を使用する]
このプロパティは、選択しているマイクロコントローラがオープン・ブレイク機能をサポートしている場合のみ表示されます。
オープン・ブレイク機能を使用するか否かを次のドロップダウン・リストにより指定します。

はい (Hi-Z)	CPU 停止時、オープン・ブレイクの対象端子は Hi-Z になります。
いいえ (Output signal)	CPU 停止時、オープン・ブレイクの対象端子は通常の状態のまま動作します（デフォルト）。

(e) [フェイルセーフ・ブレイク] 【IECUBE】

このカテゴリでは、フェイルセーフ・ブレイク機能に関する設定を行います。

フェイルセーフ・ブレイク機能、およびこのカテゴリ内の設定についての詳細は、「[2.8.5 不正な実行を検出して停止する【IECUBE】](#)」を参照してください。

(f) [トレース]

このカテゴリでは、トレース機能に関する設定を行います。

トレース機能、およびこのカテゴリ内の設定についての詳細は、「[2.11 実行履歴の収集【IECUBE】](#)」
【IECUBE2】【シミュレータ】」を参照してください。

(g) [タイマ]

このカテゴリでは、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、「[2.12 実行時間の計測](#)」を参照してください。

注意 【IECUBE2】

[トレース] カテゴリ内 [トレース・データの用途] プロパティにおいて、[タイマ] 以外を指定している場合、このカテゴリは無効となります。

図 2—26 [タイマ] カテゴリ 【IECUBE】

☐ タイマ	
タイマの分周率	1/1(20ns/1.4min)

図 2—27 [タイマ] カテゴリ【IECUBE2】

目 録	
タイマの分周率	1/1(5ns/21.4s)

- [タイマの分周率]

タイマ計測に使用するタイマ・カウンタ（33 ビット，50 MHz【IECUBE】/32 ビット，200 MHz【IECUBE2】）の分周率を，ドロップダウン・リストにより指定します。

ドロップダウン・リストには，次の分周率が表示されます（“()”内は分解能/最大測定時間を示す）。

- 【IECUBE】

1/1(20ns/1.4min) (デフォルト),	1/2(40ns/2.8min),	1/4(80ns/5.7min),
1/8(160ns/11.4min),	1/16(320ns/22.8min),	1/32(640ns/45.6min),
1/64(1280ns/1.5h),	1/128(2560ns/3.0h),	1/256(5120ns/6.0h),
1/512(10240ns/12.2h),	1/1024(20480ns/24.4h),	1/2048(40960ns/48.8h),
1/4096(81920ns/97.6h)		

- 【IECUBE2】

1/1(5ns/42.9s) (デフォルト),	1/2(10ns/1.4min),	1/4(20ns/2.8min),
1/8(40ns/5.71min),	1/16(80ns/11.42min),	1/32(160ns/22.8min),
1/64(320ns/45.6min),	1/128(640ns/1.5h),	1/256(1280ns/3.0h),
1/512(2560ns/6.0h),	1/1024(5120ns/12.2h),	1/2048(10240ns/24.4h),
1/4096(20480ns/48.8h),	1/8192(40960ns/97.6h),	1/16384(81920ns/195.2h)

(h) [カバレッジ]

このカテゴリでは，カバレッジ機能に関する設定を行います。

カバレッジ機能，およびこのカテゴリ内の設定についての詳細は，「[2.13 カバレッジの測定【IECUBE】【IECUBE2】【シミュレータ】](#)」を参照してください。

(i) [入力信号のマスク]

このカテゴリでは，入力信号のマスクに関する設定を行います。

図 2—28 [入力信号のマスク] カテゴリ【IECUBE】

目 録	
NMI0 信号をマスクする	いいえ
NMI1 信号をマスクする	いいえ
NMI2 信号をマスクする	いいえ
HLDRQ 信号をマスクする	いいえ
RESET 信号をマスクする	いいえ
STOP 信号をマスクする	いいえ
WAIT 信号をマスクする	いいえ

図 2—29 「入力信号のマスク」カテゴリ【IECUBE2】

日 入力信号のマスク	
HLDQR 信号をマスクする	いいえ
STOP 信号をマスクする	いいえ
WAIT 信号をマスクする	いいえ
RESET 信号のマスク	TARGET RESET信号をマスクする

次に示す各プロパティの設定において、該当する信号をマスクする場合は「はい」を、マスクしない場合は「いいえ」をドロップダウン・リストにより指定してください。

- 【IECUBE】
 - [NMI n 信号をマスクする]
 - [HLDQR 信号をマスクする] 注1
 - [RESET 信号をマスクする]
 - [STOP 信号をマスクする]
 - [WAIT 信号をマスクする] 注1
- 【IECUBE2】
 - [HLDQR 信号をマスクする] 注1
 - [STOP 信号をマスクする]
 - [WAIT 信号をマスクする] 注1
 - [RESET 信号のマスク] 注2

注1. [接続設定用] タブ上の「ターゲット・ボードとの接続」プロパティにおいて「いいえ」を指定している場合、デバッグ・ツールと接続時に、自動的に「はい」に固定となります（変更不可）。

2. マスクする場合は、[TARGET RESET 信号をマスクする]（デフォルト）、または [TARGET RESET 信号と INTERNAL RESET 信号をマスクする] のいずれかを指定します。

なお、マスクしない場合は、「マスクしない」を指定してください（[接続設定用] タブ上の「ターゲット・ボードとの接続」プロパティにおいて「はい」を指定している場合のみ）。

(j) 「外部フラッシュ・メモリ・ダウンロード」

このカテゴリでは、外部フラッシュ・メモリへのダウンロードに関する設定を行います。

外部フラッシュ・メモリへのダウンロード方法、およびこのカテゴリ内の設定についての詳細は、「(5) 外部フラッシュ・メモリへダウンロードする【IECUBE】【IECUBE2】【MINICUBE】【E1/E20(LPD)/ (JTAG)】」を参照してください。

(3) 「フラッシュ・セルフ・エミュレーション設定」タブ【IECUBE】

「フラッシュ・セルフ・エミュレーション設定」タブ【IECUBE】では、次に示すカテゴリごとに、フラッシュ・セルフ・プログラミング・エミュレーション（コード・フラッシュ）の設定を行います。

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・メモリ内蔵品の場合のみ表示されます。

なお、フラッシュ関数のエミュレーション可否一覧、および注意事項に関しては、[\[フラッシュ・セルフ・プログラミング・エミュレーション特記事項\]](#)を参照してください。

- (a) [\[フラッシュ・セルフ・エミュレーション\]](#)
- (b) [\[マクロ・サービス・エラー\]](#)
- (c) [\[セキュリティ・フラグ・エミュレーション設定\]](#)
- (d) [\[リセット・ベクタ/ブート・スワップ\]](#)

(a) [\[フラッシュ・セルフ・エミュレーション\]](#)

このカテゴリでは、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に関する設定を行います。

図 2—30 [\[フラッシュ・セルフ・エミュレーション\]](#) カテゴリ

<input type="checkbox"/> フラッシュ・セルフ・エミュレーション	
フラッシュ・セルフ・プログラミング・エミュレーションを行う	いいえ
フラッシュ・マクロ・サービス仕様	フラッシュ・セルフ・プログラミング Ver.2.00 以前

- [\[フラッシュ・セルフ・プログラミング・エミュレーションを行う\]](#)
 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を使用するか否かをドロップダウン・リストにより指定します。
 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を使用する場合は「はい」を選択してください（デフォルトでは「いいえ」が指定されます）。
- [\[フラッシュ・マクロ・サービス仕様\]](#)
 このプロパティは、使用するフラッシュ・メモリのプロセスが、Type01 の場合のみ表示されます^注。
 フラッシュ・マクロ・サービスの仕様を次のドロップダウン・リストにより指定します。

フラッシュ・セルフ・プログラミング Ver.2.00 以前（デフォルト）
フラッシュ・セルフ・プログラミング Ver.3.00 以降

注 デバッグ・ツールと接続時に使用するフラッシュ・メモリのプロセスの判定を行います。
 したがって、デバッグ・ツールと切断時は非表示とはなりません。

(b) [\[マクロ・サービス・エラー\]](#)

このカテゴリでは、フラッシュ・セルフ・プログラミングを行う際のフラッシュ・マクロ・サービスに関する設定として、セルフ・プログラミング・ライブラリのフラッシュ関数の動作を設定します。

なお、フラッシュ関数についての詳細は、「V850 マイクロコントローラ フラッシュ・メモリ・セルフ・プログラミング・ライブラリ Type0x」を参照してください。

図 2—31 [マクロ・サービス・エラー] カテゴリ

目 マクロ・サービス・エラー	
FlashBlockErase 関数でエラーを返す	いいえ
FlashBlockIVerify 関数でエラーを返す	いいえ
FlashWordWrite 関数でエラーを返す	いいえ
FlashBlockBlankCheck 関数でエラーを返す	いいえ
FlashSetInfo 関数でエラーを返す	いいえ
FlashFLMDCheck 関数でエラーを返す	いいえ

次に示す各プロパティの設定において、フラッシュ・メモリがダメージを受けた場合に返るエラー値を強制的に返すか否かを指定します（フラッシュ・メモリがダメージを受けた場合に返るエラーは、通常のエミュレーションでは発生しません）。

エラー値を強制的に返す場合は、ドロップダウン・リストにより [はい] を指定してください（デフォルトでは、すべてに [いいえ] が設定されます）。

- [FlashBlockErase 関数でエラーを返す]
- [FlashBlockIVerify 関数でエラーを返す]
- [FlashWordWrite 関数でエラーを返す]
- [FlashBlockBlankCheck 関数でエラーを返す]
- [FlashSetInfo 関数でエラーを返す]
- [FlashFLMDCheck 関数でエラーを返す]

(c) [セキュリティ・フラグ・エミュレーション設定]

このカテゴリでは、セキュリティ・フラグ・エミュレーション機能に関する設定を行います。

フラッシュ・メモリにセキュリティを設定した場合の、セキュリティ・フラグの初期値をエミュレーションします。

図 2—32 [セキュリティ・フラグ・エミュレーション設定] カテゴリ

目 セキュリティ・フラグ・エミュレーション設定	
フラッシュROMの消去を禁止する	いいえ
ブロック消去を禁止する	いいえ
リードを禁止する	いいえ
ライトを禁止する	いいえ
ブート領域書き換えを禁止する	いいえ

- [フラッシュ ROM の消去を禁止する]

このプロパティは、使用するフラッシュ・メモリがこの機能をサポートしている場合のみ表示されます。フラッシュ ROM の消去禁止のエミュレーションを行うか否かをドロップダウン・リストにより指定します。

フラッシュ ROM の消去禁止のエミュレーションを行う場合は [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）。

- [ブロック消去を禁止する]

ブロック消去禁止のエミュレーションを行うか否かをドロップダウン・リストにより指定します。ブロック消去禁止のエミュレーションを行う場合は [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）。

- [リードを禁止する]

このプロパティは、使用するフラッシュ・メモリのプロセスが Type01 でフラッシュ・セルフ・プログラミング Ver2.00 以前の場合は表示されません^注。

リード禁止のエミュレーションを行うか否かをドロップダウン・リストにより指定します。リード禁止のエミュレーションを行う場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

注 デバッグ・ツールと接続時に使用するフラッシュ・メモリのプロセスの判定を行います。したがって、デバッグ・ツールと切断時は非表示とはなりません。

- [ライトを禁止する]

ライト禁止のエミュレーションを行うか否かをドロップダウン・リストにより指定します。ライト禁止のエミュレーションを行う場合は [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）。

- [ブート領域書き換えを禁止する]

このプロパティは、使用するフラッシュ・メモリのプロセスが Type02, Type03, または Type04 の場合のみ表示されます^注。

ブート領域書き換え禁止のエミュレーションを行うか否かをドロップダウン・リストにより指定します。

ブート領域書き換え禁止のエミュレーションを行う場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

注 デバッグ・ツールと接続時に使用するフラッシュ・メモリのプロセスの判定を行います。したがって、デバッグ・ツールと切断時は非表示とはなりません。

(d) [リセット・ベクタ/ブート・スワップ]

このカテゴリでは、リセット・ベクタ、およびブート・スワップ機能に関する設定を行います。

図 2—33 [リセット・ベクタ/ブート・スワップ] カテゴリ

リセット・ベクタ/ブート・スワップ	
リセット・ベクタ・アドレス	HEX 0
ブート・スワップ・クラス	0

- [リセット・ベクタ・アドレス]

このプロパティは、使用するフラッシュ・メモリのプロセスが Type01 でフラッシュ・セルフ・プログラミング Ver3.00 以降の場合のみ表示されます^注。

リセット・ベクタ・アドレスを指定します。

直接入力により、0x0 ~ 0xFFFFFFFF の範囲の 16 進数で指定します（デフォルトでは [0] が指定されます）。

注 デバッグ・ツールと接続時に使用するフラッシュ・メモリのプロセスの判定を行います。

したがって、デバッグ・ツールと切断時は非表示とはなりません。

- [ブート・スワップ・クラスタ]

このプロパティは、使用するフラッシュ・メモリのプロセスが Type04 の場合のみ表示されます^注。

ブート・スワップ・クラスタの対象領域を指定します。

直接入力により、0 ~ 127 の範囲の 10 進数で指定します（デフォルトでは [0] が指定されます）。

なお、ここでの設定値により、ブート・ブロック・クラスタの対象領域も自動的に変更されます（ブート・ブロック・クラスタの対象領域は、マイクロコントローラのセキュリティ機能により、ブート領域書き換え禁止設定が可能な領域となります）。

ここでの設定値とブート・ブロック・クラスタの対象領域の関係は、[表 2-2 ブート・スワップ・クラスタとブート・ブロック・クラスタの関係](#)のとおりです。

注 デバッグ・ツールと接続時に使用するフラッシュ・メモリのプロセスの判定を行います。

したがって、デバッグ・ツールと切断時は非表示とはなりません。

表 2-2 ブート・スワップ・クラスタとブート・ブロック・クラスタの関係

設定値	フラッシュ・メモリ・サイズが 256k バイト以下の製品の場合		フラッシュ・メモリ・サイズが 384k バイト以上の製品の場合	
	ブート・ブロック ・クラスタ	ブート・スワップ ・クラスタ	ブート・ブロック ・クラスタ	ブート・スワップ ・クラスタ
0	00000H - 007FFH	00000H - 01FFFH	00000H - 007FFH	RESV - 03FFFH
1	RESV - 00FFFH		RESV - 01FFFH	
2	RESV - 017FFH		RESV - 02FFFH	
3	RESV - 01FFFH		RESV - 03FFFH	
4	RESV - 027FFH	00000H - 03FFFH	RESV - 04FFFH	RESV - 07FFFH
:	:		:	
7	RESV - 03FFFH		RESV - 07FFFH	
8	RESV - 047FFFH	00000H - 07FFFH	RESV - 08FFFH	RESV - 0FFFFH
:	:		:	
15	RESV - 07FFFH		RESV - 0FFFFH	

設定値	フラッシュ・メモリ・サイズが 256k バイト以下の製品の場合		フラッシュ・メモリ・サイズが 384k バイト以上の製品の場合	
	ブート・ブロック ・クラスタ	ブート・スワップ ・クラスタ	ブート・ブロック ・クラスタ	ブート・スワップ ・クラスタ
16	RESV - 087FFH	00000H - 0FFFFH	RESV - 10FFFFH	RESV - 1FFFFH
:	:		:	
31	RESV - 0FFFFH		RESV - 1FFFFH	
32	RESV - 107FFH		RESV - 20FFFFH	
:	:		:	
127	RESV - 3FFFFH		RESV - 7FFFFH	

(4) [データフラッシュ・エミュレーション設定] タブ [IECUBE]

[データフラッシュ・エミュレーション設定] タブ [IECUBE] では、次に示すカテゴリごとに、データフラッシュ・エミュレーションの設定を行います。

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データフラッシュ・メモリ内蔵品の場合のみ表示されます。

- (a) [データフラッシュ・エミュレーション]
- (b) [EEPROM ライブラリ・エラー]
- (c) [書き込み/消去時間設定]

(a) [データフラッシュ・エミュレーション]

このカテゴリでは、データフラッシュ・エミュレーション機能に関する設定を行います。

図 2—34 [データフラッシュ・エミュレーション] カテゴリ

☐ データフラッシュ・エミュレーション	
データフラッシュ・エミュレーションを行う	いいえ

- [データフラッシュ・エミュレーションを行う]

データフラッシュ・エミュレーション機能を使用するか否かをドロップダウン・リストにより指定します。

データフラッシュ・エミュレーション機能を使用する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

(b) [EEPROM ライブラリ・エラー]

このカテゴリでは、EEPROM エミュレーション・ライブラリに関する設定として、EEPROM エミュレーション・ライブラリ関数の動作を設定します。

なお、EEPROM エミュレーション・ライブラリ関数についての詳細は、「V850 マイクロコントローラフラッシュ・メモリ・セルフ・プログラミング・ライブラリ Type0x」を参照してください。

図 2—35 [EEPROM ライブラリ・エラー] カテゴリ

EEPROMライブラリ・エラー	
WordWrite関数でエラーを返す	はい
WordWrite関数エラーの発生アドレス	HEX 0
BlockErase関数でエラーを返す	はい
BlockErase関数エラーの発生アドレス	HEX 0
BlockIVerify関数でエラーを返す	はい
BlockIVerify関数エラーの発生アドレス	HEX 0
BlockBlankCheck関数でエラーを返す	はい
BlockBlankCheck関数エラーの発生アドレス	HEX 0

次に示す各プロパティにおいて、該当する関数で発生するエラー値を強制的に返すか否かをドロップダウン・リストにより指定します（関数で発生するエラー値は、通常のエミュレーションでは返ることがありません）。デフォルトでは、すべてに [いいえ] が設定されています。

エラー値を強制的に返す場合は、該当するプロパティを [はい] に指定したのち、下段に表示される [xxx 関数エラーの発生アドレス] プロパティにおいて、エラーを発生させるデータフラッシュ・メモリ領域内のアドレスを、0x0 ~ 0xFFFFFFFF の範囲の 16 進数で直接入力により指定してください（デフォルトでは [0] が指定されます）。

- [WordWrite 関数でエラーを返す]
- [BlockErase 関数でエラーを返す]
- [BlockIVerify 関数でエラーを返す]
- [BlockBlankCheck 関数でエラーを返す]

(c) [書き込み／消去時間設定]

このカテゴリでは、データフラッシュ・メモリへの書き込み、または消去時間に関する設定を行います。

図 2—36 [書き込み／消去時間設定] カテゴリ

書き込み／消去時間設定	
データフラッシュ・メモリへの書き込み時間	フラッシュ・マクロ・スペック上の想定回数
データフラッシュ・メモリの消去時間	フラッシュ・マクロ・スペック上の想定回数

- [データフラッシュ・メモリへの書き込み時間]

データフラッシュ・メモリへ書き込む際の遅延時間をシミュレートします。

シミュレート値を次のドロップダウン・リストにより指定してください。

リトライなし	リトライ回数を 0 とし、遅延時間 = 0（書き込み時間最速）とします。
フラッシュ・マクロ・スペック上の想定回数	使用するフラッシュ・マクロのスペック上の想定回数を遅延時間とします（デフォルト）。
フラッシュ・マクロ・スペック上の想定最大回数	使用するフラッシュ・マクロのスペック上の想定最大回数を遅延時間とします。
リトライ回数最大	リトライ回数を最大とし、遅延時間を最大（書き込み時間最長）とします。

- [データフラッシュ・メモリの消去時間]

データフラッシュ・メモリの消去を行う際の遅延時間をシミュレートします。

シミュレート値を次のドロップダウン・リストにより指定してください。

リトライなし	リトライ回数を0とし、遅延時間=0（消去時間最速）とします。
フラッシュ・マクロ・スペック上の想定回数	使用するフラッシュ・マクロのスペック上の想定回数を遅延時間とします（デフォルト）。
フラッシュ・マクロ・スペック上の想定最大回数	使用するフラッシュ・マクロのスペック上の想定最大回数を遅延時間とします。
リトライ回数最大	リトライ回数を最大とし、遅延時間を最大（消去時間最長）とします。

(5) [ダウンロード・ファイル設定] タブ

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

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

(6) [フラッシュ・オプション設定] タブ [IECUBE2]

[フラッシュ・オプション設定] タブ [V850E2] では、V850E2 コアの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 \[V850E2\]](#) 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）。

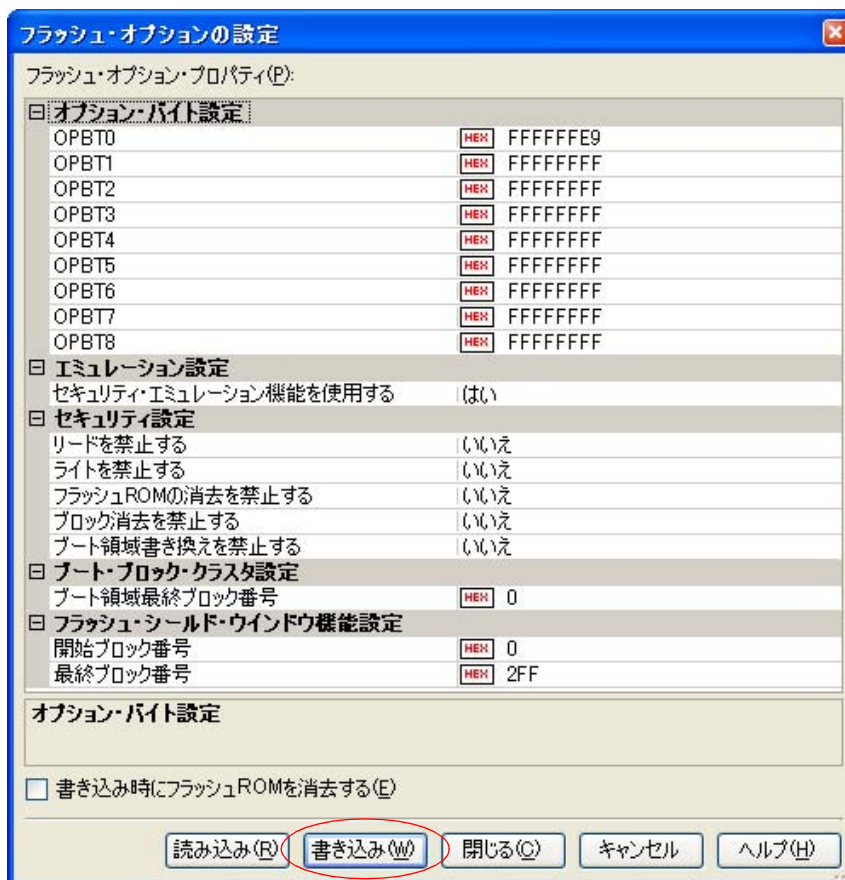
このダイアログにおいて、各項目の設定を完了したのち、[書き込み] ボタンをクリックしてください。

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 \[V850E2\]](#) を参照してください。

図 2—37 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—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 [IECUBE2])



(7) [フック処理設定] タブ

[フック処理設定] タブでは、デバッグ・ツールにフック処理の設定を行います。

フック処理、および各カテゴリ内の設定についての詳細は、「2.16 フック処理を設定する」を参照してください。

2.3.3 【MINICUBE】 / 【MINICUBE2】 の場合

MINICUBE/MINICUBE2 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【MINICUBE【V850E1】【V850ES】】（プロパティパネルの例）

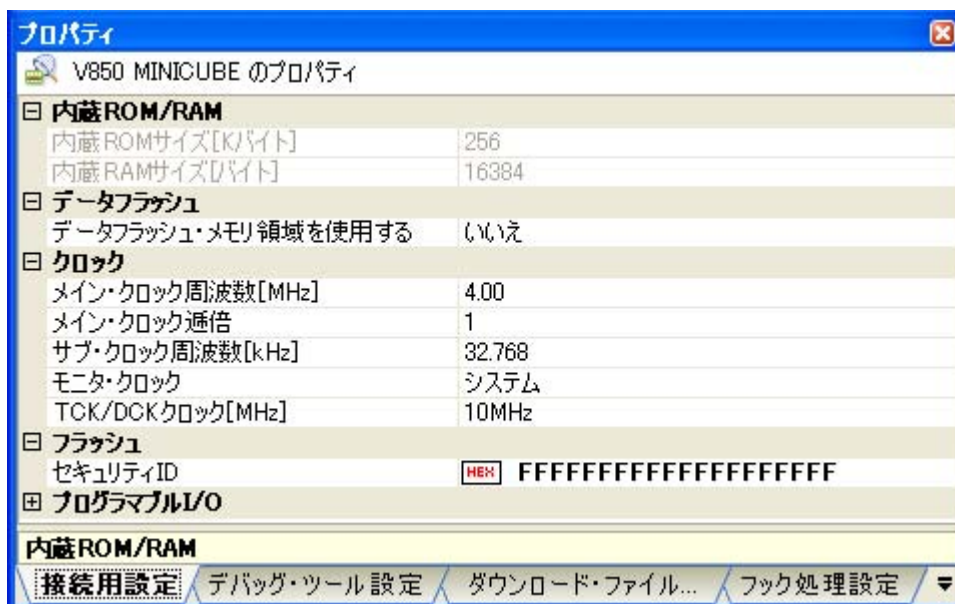


図 2—40 動作環境設定【MINICUBE【V850E2】】（プロパティパネルの例）

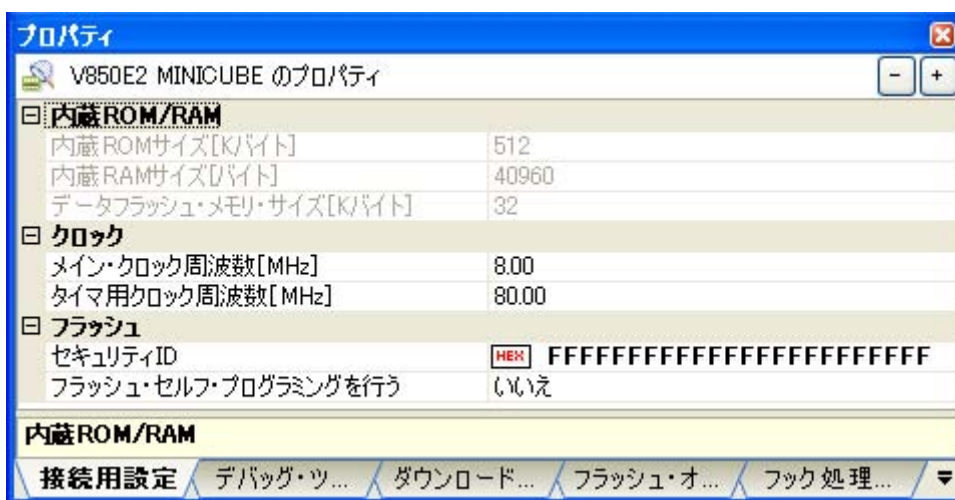


図 2—41 動作環境設定【MINICUBE2【V850E1】【V850ES】】（プロパティ パネルの例）

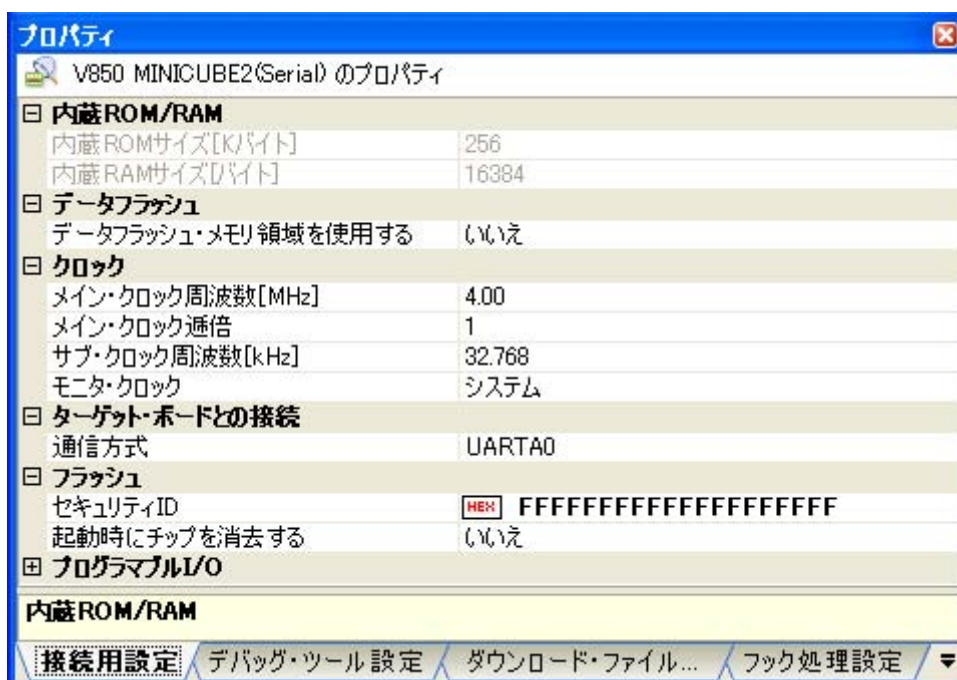


図 2—42 動作環境設定【MINICUBE2【V850E2】】（プロパティ パネルの例）



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【V850E2】
- (5) [フック処理設定] タブ

(1) [接続用設定] タブ

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

- (a) [内蔵 ROM/RAM]
- (b) [データフラッシュ] 【V850E1】 【V850ES】
- (c) [クロック]
- (d) [ターゲット・ボードとの接続] 【MINICUBE2】
- (e) [フラッシュ]
- (f) [プログラマブル I/O] 【V850E1】 【V850ES】

(a) [内蔵 ROM/RAM]

このカテゴリでは、内蔵 ROM/RAM に関する設定を表示します。

図 2—43 [内蔵 ROM/RAM] カテゴリ 【MINICUBE 【V850E1】 【V850ES】】 【MINICUBE2】

内蔵ROM/RAM	
内蔵ROMサイズ[K/バイト]	256
内蔵RAMサイズ[B/バイト]	16384

図 2—44 [内蔵 ROM/RAM] カテゴリ 【MINICUBE 【V850E2】】

内蔵ROM/RAM	
内蔵ROMサイズ[K/バイト]	512
内蔵RAMサイズ[B/バイト]	40960
データフラッシュ・メモリ・サイズ[K/バイト]	32

- [内蔵 ROM サイズ [K バイト]]

エミュレーションする内蔵 ROM サイズを表示します (単位 : K バイト)。

なお、選択しているマイクロコントローラが V850E1 コアで、VSB Flash ROM 搭載品の場合では、VSB Flash ROM サイズを加えた値を表示します。

このプロパティ値を変更することはできません。

- [内蔵 RAM サイズ [バイト]]

エミュレーションする内蔵 RAM サイズを表示します (単位 : バイト)。

なお、選択しているマイクロコントローラが V850E1 コアで、VSB RAM 搭載品の場合では、VSB RAM サイズを加えた値を表示します。

また、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、“合計 RAM サイズ (コア 1 つ分の RAM サイズ x コアの数)” 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。

- [データフラッシュ・メモリ・サイズ [K バイト]] 【MINICUBE 【V850E2】】

選択しているマイクロコントローラのデータフラッシュ・メモリ領域のサイズを表示します (単位 : K バイト)。

選択しているマイクロコントローラがデータフラッシュ・メモリ非搭載品の場合は、[0] を表示します。
このプロパティ値を変更することはできません。

(b) [データフラッシュ] [V850E1] [V850ES]

このカテゴリでは、データフラッシュ・メモリに関する設定を行います。
ただし、このカテゴリは、データフラッシュ・メモリ内蔵品の場合のみ表示されます。

図 2—45 [データフラッシュ] カテゴリ

データフラッシュ	
データフラッシュ・メモリ領域を使用する	はい
データフラッシュ・メモリ・サイズ[Kバイト]	32
チップ・セレクト	CS0

- [データフラッシュ・メモリ領域を使用する]

データフラッシュ・メモリを使用するか否かをドロップダウン・リストにより指定します。
データフラッシュ・メモリを使用する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

- [データフラッシュ・メモリ・サイズ[Kバイト]]

このプロパティは、[\[データフラッシュ・メモリ領域を使用する\]](#) 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。

選択しているマイクロコントローラのデータフラッシュ・メモリ領域のサイズを表示します（単位：Kバイト）。

このプロパティ値を変更することはできません。

- [チップ・セレクト]

このプロパティは、[\[データフラッシュ・メモリ領域を使用する\]](#) プロパティにおいて、[はい] を指定した場合のみ表示されます。

データフラッシュ・メモリをマッピングする際のチップ・セレクトをドロップ・ダウンリストにより指定します。

なお、指定できるチップ・セレクトは、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、使用するマイクロコントローラに V850ES コア以外を選択している場合、デフォルトで表示しているチップ・セレクトを変更することはできません。

(c) [クロック]

このカテゴリでは、クロックに関する設定を行います。

図 2—46 [クロック] カテゴリ【MINICUBE【V850E1】【V850ES】】

クロック	
メイン・クロック周波数[MHz]	4.00
メイン・クロック通倍	1
サブ・クロック周波数[kHz]	32.765
モニタ・クロック	システム
TCK/DCKクロック[MHz]	10MHz

図 2—47 [クロック] カテゴリ【MINICUBE2【V850E1】【V850ES】】

クロック	
メイン・クロック周波数[MHz]	4.00
メイン・クロック通倍	1
サブ・クロック周波数[kHz]	32.765
モニタ・クロック	システム

図 2—48 [クロック] カテゴリ【MINICUBE【V850E2】】【MINICUBE2【V850E2】】

クロック	
メイン・クロック周波数[MHz]	8.00
タイマ用クロック周波数[MHz]	80.00

図 2—49 [クロック] カテゴリ【MINICUBE】【MINICUBE2】(V850E2の一部製品)

クロック	
CPUクロック周波数[MHz]	80.00
タイマ用クロック周波数[MHz]	80.00

- [メイン・クロック周波数 [MHz]]

メイン・クロック周波数（通倍前）を指定します。

ドロップダウン・リストによる選択か、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。

ドロップダウン・リストには、次の周波数が表示されます（単位：MHz）。

- 【V850E1】【V850ES】

2.50, 4.00（デフォルト）, 5.00, 6.00, 8.00, 13.50, 18.00

- 【V850E2】

4.00, 5.00, 7.20, 8.00（デフォルト）, 9.60, 10.00, 13.50, 16.00

注意 【V850E2】

このプロパティは、[\[CPU クロック周波数 \[MHz\]\]](#) [【V850E2】](#) プロパティが表示されている場合は非表示となります。

- [メイン・クロック通倍] 【V850E1】 【V850ES】

メイン・クロック周波数に対する通倍数を指定します。

ドロップダウン・リストによる選択か、または直接入力により、1～99の範囲の値を指定します。

ドロップダウン・リストには、次の通倍数が表示されます。

1 (デフォルト), 2, 3, 4, 5, 6, 7, 8, 9, 10

- [サブ・クロック周波数 [kHz]] 【V850E1】 【V850ES】

このプロパティは、選択しているマイクロコントローラがサブ・クロックをサポートしている場合のみ表示されます。

サブ・クロック周波数を指定します。

ドロップダウン・リストによる選択か、または直接入力により、0.001～999.999 (単位: kHz) の範囲の数値を指定します。

ドロップダウン・リストには、現在選択しているマイクロコントローラが使用可能なサブ・クロック周波数が表示されます。

デフォルトでは、使用可能なサブ・クロック周波数の最小値が指定されます。

- [モニタ・クロック] 【V850E1】 【V850ES】

このプロパティは、選択しているマイクロコントローラがサブ・クロックをサポートしている場合のみ表示されます。

プログラム停止中に使用するクロックを指定します。

次のドロップダウン・リストにより指定します。

システム	メイン・クロックで動作します (デフォルト)。
ユーザ	プログラムで設定されているクロックで動作します。

- [TCK/DCK クロック [MHz]] 【MINICUBE 【V850E1】 【V850ES】】

デバッグ・コントロール・ユニット (DCU) へ供給するクロックをドロップダウン・リストにより指定します。

ドロップダウン・リストには、次の周波数が表示されます。

10MHz (デフォルト), 20MHz

注意 1. 通常は、[10MHz] の指定を変更しないでください。

[20MHz] を指定した場合、デバッグ・ツールと接続できない可能性があります。

2. MINICUBE が CubeSuite+ に接続している場合、このプロパティ値を変更することはできません。

- [タイマ用クロック周波数 [MHz]] 【V850E2】

タイマ用のクロック周波数を指定します。

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 (デフォルトでは [80.00] が指定されます)。

ただし、[CPUクロック周波数 [MHz]] 【V850E2】 プロパティを表示している場合は、このプロパティ値を変更することはできません（[CPUクロック周波数 [MHz]] プロパティと同値に固定されます）。

- [CPUクロック周波数 [MHz]] 【V850E2】

このプロパティは、選択しているマイクロコントローラが V850E2 の一部製品でのみ表示されます。CPUクロック周波数（逡倍後）を指定します。

ドロップダウン・リストによる選択か、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。

ドロップダウン・リストには、次の周波数が表示されます（単位：MHz）。

80.00（デフォルト）、100.00

(d) [ターゲット・ボードとの接続] 【MINICUBE2】

このカテゴリでは、MINICUBE2 とターゲット・ボードとの接続に関する設定を行います。

図 2—50 [ターゲット・ボードとの接続] カテゴリ 【MINICUBE2】

<input type="checkbox"/> ターゲット・ボードとの接続	
通信方式	UARTA0

- [通信方式]

MINICUBE2 がターゲット・ボード上のマイクロコントローラとシリアル通信を行う際に接続するポートをドロップダウン・リストにより指定します。

ドロップダウン・リスト内に表示されるポートの種類は、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、選択しているマイクロコントローラにより、シリアル通信の際に使用できるポートが限定される場合は、デフォルトで表示している通信方式を変更することはできません（デフォルトの指定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）。

注意 MINICUBE2 が CubeSuite+ に接続している場合、このプロパティ値を変更することはできません。

(e) [フラッシュ]

このカテゴリでは、フラッシュ書き換えに関する設定を行います。

注意 MINICUBE/MINICUBE2 が CubeSuite+ に接続している場合、このカテゴリ内のプロパティ値を変更することはできません。

図 2—51 [フラッシュ] カテゴリ 【MINICUBE 【V850E1】 【V850ES】】

<input type="checkbox"/> フラッシュ	
セキュリティID	<input type="text" value="HEX FFFFFFFFFFFFFFFFFF"/>

図 2—52 【フラッシュ】カテゴリ【MINICUBE【V850E2】】

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

図 2—53 【フラッシュ】カテゴリ【MINICUBE2【V850E1】【V850ES】】

□ フラッシュ	
セキュリティID	HEX FFFFFFFFFFFFFFFFFFFFFFFF
起動時にフラッシュROMを消去する	いいえ

図 2—54 【フラッシュ】カテゴリ【MINICUBE2【V850E2】】

□ フラッシュ	
セキュリティID	HEX FFFFFFFFFFFFFFFFFFFFFFFF
起動時にフラッシュROMを消去する	いいえ

- 【セキュリティID】

このプロパティは、選択しているマイクロコントローラが、フラッシュ・メモリのROMセキュリティ機能（オンチップ・デバッグ・セキュリティID）をサポートしている場合のみ表示されます。

内蔵ROM、または内蔵フラッシュ・メモリ上のコードを読み出す際のセキュリティIDを指定します。

なお、オンチップ・デバッグ・セキュリティIDについての詳細は、MINICUBE/MINICUBE2のユーザーズ・マニュアルを参照してください。

- 【V850E1】【V850ES】

直接入力により、20桁の16進数（10バイト）で指定します（デフォルトでは [FFFFFFFFFFFFFFFFFFFFFF] が指定されます）。

- 【V850E2】

直接入力により、24桁の16進数（12バイト）で指定します（デフォルトでは [FFFFFFFFFFFFFFFFFFFFFF] が指定されます）。

備考 【V850E1】【V850ES】

このプロパティの設定を変更すると、ビルド・ツールのセキュリティIDを変更するか否かを確認するメッセージを表示します。

- 【フラッシュ・セルフ・プログラミングを行う】【MINICUBE【V850E2】】

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、内蔵フラッシュROMの書き換えを行うか否かをドロップダウン・リストにより指定します。

内蔵フラッシュROMの書き換えを行う場合は、[はい]を選択してください（デフォルトでは [いいえ] が指定されます）。

なお、[はい]を指定した場合、内蔵フラッシュROMのキャッシュは行われません。

- [起動時にフラッシュ ROM を消去する]【MINICUBE2】

デバッグ・ツールと接続する際に、フラッシュ ROM の消去を行うか否かをドロップダウン・リストにより指定します。

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）。

なお、このプロパティは、デバッグ・ツールと接続後、自動的に [いいえ] に設定されます。

(f) [プログラマブル I/O]【V850E1】【V850ES】

このカテゴリでは、プログラマブル I/O 領域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プログラマブル I/O 領域をサポートしている場合のみ指定可能です。

備考 デバッグを行う際に、一時的に値を変更する場合にのみ設定を変更してください。

プロジェクトとして共通の値を設定する場合は、使用するビルド・ツールのプロパティパネルの [共通オプション] タブ上の [デバイス] カテゴリ内 [プログラマブル I/O 領域開始アドレス] プロパティで設定を行ってください。

図 2—55 [プログラマブル I/O] カテゴリ

☐ プログラマブル I/O	
プログラマブル I/O 領域を使用する	はい
プログラマブル I/O 領域開始アドレス	HEX 0

- [プログラマブル I/O 領域を使用する]

プログラマブル I/O 領域を使用するか否かをドロップダウン・リストにより指定します。

プログラマブル I/O 領域を使用する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

- [プログラマブル I/O 領域開始アドレス]

このプロパティは、[プログラマブル I/O 領域を使用する] 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。

プログラマブル I/O 領域の開始アドレスを指定します。

直接入力により、プログラマブル I/O 領域内のアドレスを指定してください（デフォルトでは [0] が指定されます）。

なお、アドレスは、16 K バイトでアラインされます。

(2) [デバッグ・ツール設定] タブ

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

- (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레이크]
- (e) [入力信号のマスク]
- (f) [外部フラッシュ・メモリ・ダウンロード] 【MINICUBE】

(a) [メモリ]

このカテゴリでは、メモリに関する設定を行います。

図 2—56 [メモリ] カテゴリ 【MINICUBE】 【MINICUBE2】

☐ メモリ	
☐ メモリ・マッピング	[4]
田 [0]	内蔵ROM領域
田 [1]	ノン・マップ領域
田 [2]	内蔵RAM領域
田 [3]	I/Oレジスタ領域
メモリ書き込み時にベリファイを行う	はい

- [メモリ・マッピング]

現在のメモリ・マッピングの状況が、メモリ領域の種別ごとに詳細表示されます。

このパネル上でマッピング値を変更することはできません。メモリ・マッピングを追加する必要がある場合は、[メモリ・マッピング] プロパティを選択することで設定欄右端に表示される [...] ボタンのクリックによりオープンする**メモリ・マッピング ダイアログ**で行います。

設定方法についての詳細は、**メモリ・マッピング ダイアログ**の項を参照してください。

図 2—57 メモリ・マッピング ダイアログのオープン

☐ メモリ	
☐ メモリ・マッピング	[4]
田 [0]	内蔵ROM領域
田 [1]	ノン・マップ領域

注意 デバッグ・ツールと未接続の場合、ユーザにより追加されたメモリ・マッピング領域のみが表示対象となります。

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

- [メモリ書き込み時にベリファイを行う]

メモリ値の初期化を行う際に、ベリファイを行うか否かをドロップダウン・リストにより指定します。

ベリファイを行う場合は [はい] を選択してください (デフォルト)。

(b) [実行中のメモリ・アクセス]

このカテゴリでは、プログラム実行中におけるメモリ・アクセス (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—58 [実行中のメモリ・アクセス] カテゴリ【MINICUBE【V850E1】【V850ES】】

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

図 2—59 [実行中のメモリ・アクセス] カテゴリ【MINICUBE【V850E2】】

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

図 2—60 [実行中のメモリ・アクセス] カテゴリ【MINICUBE2】

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

- [実行中にアクセスする]【MINICUBE【V850E2】】

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ドロップダウン・リストにより指定します。なお、[フラッシュ] カテゴリ内 [フラッシュ・セルフ・プログラミングを行う]【MINICUBE【V850E2】】プロパティにおいて [いいえ] を選択している場合は、内蔵 ROM 領域も対象となります。アクセスを許可する場合は [はい] を選択してください (デフォルトでは [いいえ] が指定されます)。

- [実行を一瞬停止してアクセスする]

このプロパティは、選択しているマイクロコントローラがマルチコア版の場合、または [実行中にアクセスする]【MINICUBE【V850E2】】プロパティにおいて [いいえ] を指定した場合は表示されません。プログラム実行中にはアクセスできないメモリ領域 (ターゲット・メモリ領域 /I/O レジスタ領域 /CPU レジスタなど) に対して、実行を一瞬停止し、アクセスを許可するか否かをドロップダウン・リストにより指定します。

なお、[フラッシュ] カテゴリ内 [フラッシュ・セルフ・プログラミングを行う] 【MINICUBE 【V850E2】】 プロパティにおいて [いいえ] を選択している場合は、内蔵 ROM 領域も内蔵 RAM 領域と同様の扱いとなります。

アクセスを許可する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

備考 【V850E2】

内蔵 RAM 領域に対しては、このプロパティの指定にかかわらず、一瞬停止することなくアクセスします。

- [実行中に表示更新を行う]

プログラム実行中に、**ウォッチパネル/メモリパネル**の表示内容を自動的に更新するか否かをドロップダウン・リストにより指定します。

表示内容の更新を行う場合は [はい] を選択してください（デフォルト）。

- [表示更新間隔 [ms]]

このプロパティは、[実行中に表示更新を行う] プロパティにおいて [はい] を指定した場合にのみ有効となります。

プログラム実行中に、**ウォッチパネル/メモリパネル**の表示内容を自動的に更新する間隔を 100 ms 単位で指定します。

直接入力により、100 ~ 65500 の範囲の整数（100 ms 未満の端数切り上げ）を指定してください（デフォルトでは [500] が指定されます）。

- [HALT を解除してアクセスする] 【MINICUBE】

このプロパティは、[実行を一瞬停止してアクセスする] 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。

HALT モードを解除して、内蔵 RAM 領域以外のメモリ（CPU レジスタ /I/O レジスタ）にアクセスするか否かをドロップダウン・リストにより指定します（内蔵 RAM 領域へのアクセスに対しては HALT モードを解除しません）。

なお、[フラッシュ] カテゴリ内 [フラッシュ・セルフ・プログラミングを行う] 【MINICUBE 【V850E2】】 プロパティにおいて [いいえ] を選択している場合は、内蔵 ROM 領域も内蔵 RAM 領域と同様の扱いとなります。

HALT モードを解除してアクセスする場合は [はい] を選択してください（デフォルト）。

(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—61 [実行中のイベント設定] カテゴリ

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

- [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ドロップダウン・リストにより指定します。

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（デフォルトでは「いいえ」が指定されます）。

(d) [ブレーク]

このカテゴリでは、ブレーク機能に関する設定を行います。

図 2—62 [ブレーク] カテゴリ【MINICUBE】

日 ブレーク	
優先的に使用するブレークポイントの種類	ソフトウェア・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
オープン・ブレーク機能を使用する	いいえ (Output signal)

図 2—63 [ブレーク] カテゴリ【MINICUBE2】

日 ブレーク	
優先的に使用するブレークポイントの種類	ソフトウェア・ブレーク

- [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、選択しているマイクロコントローラが使用可能なブレークポイントの種類が1つのみの場合は表示されません。

[エディタ パネル/逆アセンブル パネル](#)において、マウスのワンクリック操作でブレークポイントを設定する際に使用するブレークポイントの種別を次のドロップダウン・リストにより指定します。

なお、ブレークポイントについての詳細は、「[2.8.2 任意の場所で停止する \(ブレークポイント\)](#)」を参照してください。

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

- [停止時に周辺エミュレーションを停止する]【MINICUBE】

プログラム実行停止時に、エミュレータの周辺エミュレーション機能を停止するか否かをドロップダウン・リストにより指定します。

停止する場合は「はい」を選択してください（デフォルトでは「いいえ」が指定されます）。

- [オープン・ブレーク機能を使用する]【MINICUBE】

このプロパティは、選択しているマイクロコントローラがオープン・ブレーク機能をサポートしている場合にのみ表示されます。

オープン・ブレイク機能を使用するか否かを次のドロップダウン・リストにより指定します。

はい (Hi-Z)	CPU 停止時、オープン・ブレイクの対象端子は Hi-Z になります。
いいえ (Output signal)	CPU 停止時、オープン・ブレイクの対象端子は通常の状態のまま動作します。

(e) [入力信号のマスク]

このカテゴリでは、入力信号のマスクに関する設定を行います。

図 2—64 [入力信号のマスク] カテゴリ【MINICUBE【V850E1】【V850ES】】

□ 入力信号のマスク	
NMI0 信号をマスクする	いいえ
NMI1 信号をマスクする	いいえ
NMI2 信号をマスクする	いいえ
HLDQR 信号をマスクする	いいえ
RESET 信号をマスクする	いいえ
STOP 信号をマスクする	いいえ
WAIT 信号をマスクする	いいえ
DBINT 信号をマスクする	いいえ

図 2—65 [入力信号のマスク] カテゴリ【MINICUBE【V850E2】】

□ 入力信号のマスク	
HLDQR 信号をマスクする	いいえ
RESET 信号をマスクする	いいえ
STOP 信号をマスクする	いいえ
WAIT 信号をマスクする	いいえ

図 2—66 [入力信号のマスク] カテゴリ【MINICUBE2】

□ 入力信号のマスク	
RESET 信号をマスクする	いいえ

次に示す各プロパティの設定において、該当する信号をマスクする場合は [はい] を、マスクしない場合は [いいえ] をドロップダウン・リストにより指定してください（デフォルトでは、すべてのプロパティに [いいえ] が指定されます）。

- [NMI n 信号をマスクする]【MINICUBE【V850E1】【V850ES】】
- [HLDQR 信号をマスクする]【MINICUBE】
- [RESET 信号をマスクする]
- [STOP 信号をマスクする]【MINICUBE】
- [WAIT 信号をマスクする]【MINICUBE】
- [DBINT 信号をマスクする]【MINICUBE【V850E1】【V850ES】】

(f) [外部フラッシュ・メモリ・ダウンロード]【MINICUBE】

このカテゴリでは、外部フラッシュ・メモリへのダウンロードに関する設定を行います。

外部フラッシュ・メモリへのダウンロード方法、およびこのカテゴリ内の設定についての詳細は、「(5) 外部フラッシュ・メモリへダウンロードする【IECUBE】【IECUBE2】【MINICUBE】【E1/E20(LPD)/ (JTAG)】」を参照してください。

(3) [ダウンロード・ファイル設定] タブ

[ダウンロード・ファイル設定] タブでは、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【V850E2】

[フラッシュ・オプション設定] タブ【V850E2】では、V850E2 コアの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【V850E2】](#)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）。

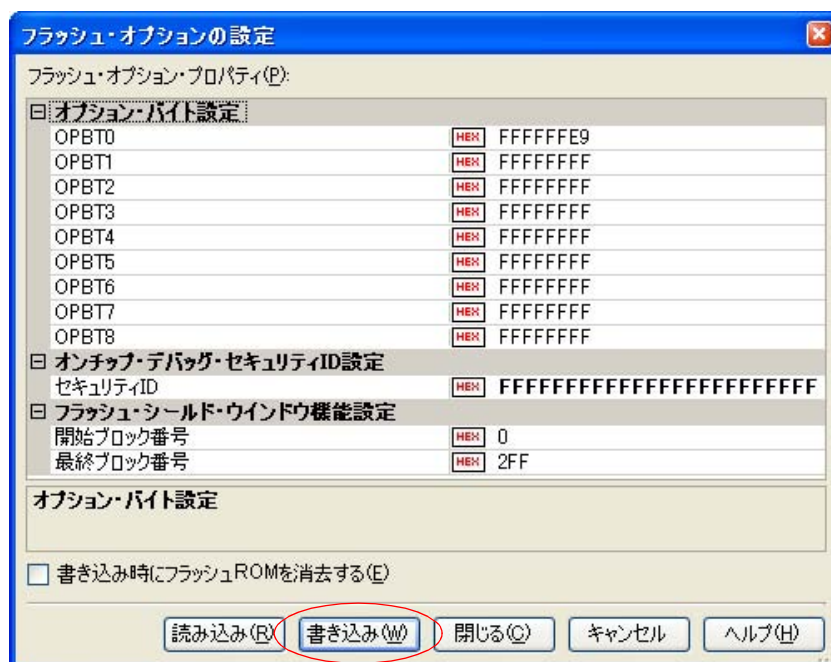
このダイアログにおいて、各項目の設定を完了したのち、[書き込み] ボタンをクリックしてください。

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【V850E2】](#)を参照してください。

図 2—67 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—68 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【MINICUBE】【MINICUBE2】)



(5) [フック処理設定] タブ

[フック処理設定] タブでは、デバッグ・ツールにフック処理の設定を行います。

フック処理、および各カテゴリ内の設定についての詳細は、「[2.16 フック処理を設定する](#)」を参照してください。

2.3.4 【E1】の場合

E1 を使用する場合の動作環境の設定を次のプロパティパネルで行います。

なお、ターゲット・ボードとの通信方式（シリアル通信【E1(Serial)】/LPD 通信【E1(LPD)】/JTAG 通信【E1(JTAG)】）により、プロパティパネルの内容は異なります。

図 2—69 動作環境設定【E1(Serial)【V850E1】【V850ES】】（プロパティパネルの例）



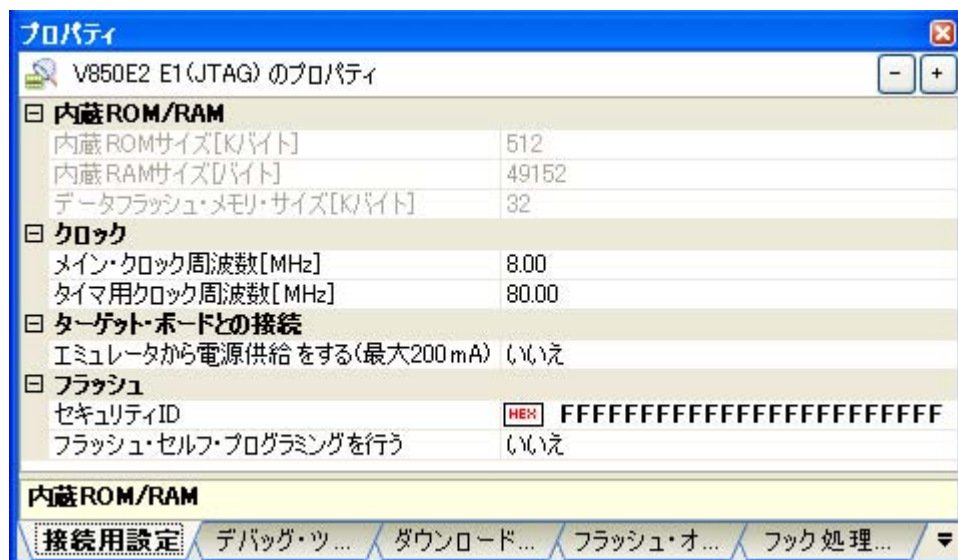
図 2—70 動作環境設定【E1(Serial)【V850E2】】（プロパティパネルの例）



図 2—71 動作環境設定【E1(JTAG)【V850E1】【V850ES】】(プロパティ パネルの例)



図 2—72 動作環境設定【E1(LPD)/(JTAG)【V850E2】】(プロパティ パネルの例)



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【V850E2】
- (5) [フック処理設定] タブ

(1) [接続用設定] タブ

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

- (a) [内蔵 ROM/RAM]
- (b) [データフラッシュ] 【V850E1】 【V850ES】
- (c) [クロック]
- (d) [ターゲット・ボードとの接続]
- (e) [フラッシュ]
- (f) [プログラマブル I/O] 【V850E1】 【V850ES】

(a) [内蔵 ROM/RAM]

このカテゴリでは、内蔵 ROM/RAM に関する設定を表示します。

図 2—73 [内蔵 ROM/RAM] カテゴリ 【E1 【V850E1】 【V850ES】】

内蔵ROM/RAM	
内蔵ROMサイズ[K/バイト]	256
内蔵RAMサイズ[B/バイト]	16384

図 2—74 [内蔵 ROM/RAM] カテゴリ 【E1 【V850E2】】

内蔵ROM/RAM	
内蔵ROMサイズ[K/バイト]	512
内蔵RAMサイズ[B/バイト]	40960
データフラッシュ・メモリ・サイズ[K/バイト]	32

- [内蔵 ROM サイズ [K バイト]]

エミュレーションする内蔵 ROM サイズを表示します (単位: K バイト)。

なお、選択しているマイクロコントローラが V850E1 コアで、VSB Flash ROM 搭載品の場合では、VSB Flash ROM サイズを加えた値を表示します。

このプロパティ値を変更することはできません。

- [内蔵 RAM サイズ [バイト]]

エミュレーションする内蔵 RAM サイズを表示します (単位: バイト)。

なお、選択しているマイクロコントローラが V850E1 コアで、VSB RAM 搭載品の場合では、VSB RAM サイズを加えた値を表示します。

また、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、“合計 RAM サイズ”(コア 1 つ分の RAM サイズ × コアの数) 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。

- [データフラッシュ・メモリ・サイズ [K バイト]] 【V850E2】

選択しているマイクロコントローラのデータフラッシュ・メモリ領域のサイズを表示します (単位: K バイト)。

選択しているマイクロコントローラがデータフラッシュ・メモリ非搭載品の場合は、[0] を表示します。

このプロパティ値を変更することはできません。

(b) [データフラッシュ] [V850E1] [V850ES]

このカテゴリでは、データフラッシュ・メモリに関する設定を行います。

ただし、このカテゴリは、データフラッシュ・メモリ内蔵品の場合のみ表示されます。

図 2—75 [データフラッシュ] カテゴリ

日 データフラッシュ	
データフラッシュ・メモリ領域を使用する	はい
データフラッシュ・メモリ・サイズ[Kバイト]	32
チップ・セレクト	CS0

- [データフラッシュ・メモリ領域を使用する]

データフラッシュ・メモリを使用するか否かをドロップダウン・リストにより指定します。

データフラッシュ・メモリを使用する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

- [データフラッシュ・メモリ・サイズ[Kバイト]]

このプロパティは、[データフラッシュ・メモリ領域を使用する] 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。

選択しているマイクロコントローラのデータフラッシュ・メモリ領域のサイズを表示します（単位：Kバイト）。

このプロパティ値を変更することはできません。

- [チップ・セレクト]

このプロパティは、[データフラッシュ・メモリ領域を使用する] プロパティにおいて、[はい] を指定した場合のみ表示されます。

データフラッシュ・メモリをマッピングする際のチップ・セレクトをドロップ・ダウンリストにより指定します。

なお、指定できるチップ・セレクトは、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、使用するマイクロコントローラに V850ES コア以外を選択している場合、デフォルトで表示しているチップ・セレクトを変更することはできません。

(c) [クロック]

このカテゴリでは、クロックに関する設定を行います。

図 2—76 [クロック] カテゴリ【E1(Serial)【V850E1】【V850ES】】

目 クロック	
メイン・クロック周波数[MHz]	4.00
メイン・クロック通倍	1
サブ・クロック周波数[kHz]	32.765
モニタ・クロック	システム

図 2—77 [クロック] カテゴリ【E1(JTAG)【V850E1】【V850ES】】

目 クロック	
メイン・クロック周波数[MHz]	4.00
メイン・クロック通倍	1
サブ・クロック周波数[kHz]	32.765
モニタ・クロック	システム
TCK/DCKクロック[MHz]	10MHz

図 2—78 [クロック] カテゴリ【E1【V850E2】】

目 クロック	
メイン・クロック周波数[MHz]	8.00
タイマ用クロック周波数[MHz]	80.00

図 2—79 [クロック] カテゴリ【E1】(V850E2の一部製品)

目 クロック	
CPUクロック周波数[MHz]	80.00
タイマ用クロック周波数[MHz]	80.00

- [メイン・クロック周波数 [MHz]]

メイン・クロック周波数（通倍前）を指定します。

ドロップダウン・リストによる選択か、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。

ドロップダウン・リストには、次の周波数が表示されます（単位：MHz）。

- 【V850E1】【V850ES】

2.50, 4.00（デフォルト）, 5.00, 6.00, 8.00, 13.50, 18.00

- 【V850E2】

4.00, 5.00, 7.20, 8.00（デフォルト）, 9.60, 10.00, 13.50, 16.00

注意 【V850E2】

このプロパティは、**[CPUクロック周波数 [MHz]]** **[V850E2]** プロパティが表示されている場合は非表示となります。

- [メイン・クロック通倍] 【V850E1】 【V850ES】

メイン・クロック周波数に対する通倍数を指定します。

ドロップダウン・リストによる選択か、または直接入力により、1～99の範囲の値を指定します。

ドロップダウン・リストには、次の通倍数が表示されます。

1 (デフォルト), 2, 3, 4, 5, 6, 7, 8, 9, 10

- [サブ・クロック周波数 [kHz]] 【V850E1】 【V850ES】

このプロパティは、選択しているマイクロコントローラがサブ・クロックをサポートしている場合のみ表示されます。

サブ・クロック周波数を指定します。

ドロップダウン・リストによる選択か、または直接入力により、0.001～999.999 (単位: kHz) の範囲の数値を指定します。

ドロップダウン・リストには、現在選択しているマイクロコントローラが使用可能なサブ・クロック周波数が表示されます。

デフォルトでは、使用可能なサブ・クロック周波数の最小値が指定されます。

- [モニタ・クロック] 【V850E1】 【V850ES】

このプロパティは、選択しているマイクロコントローラがサブ・クロックをサポートしている場合のみ表示されます。

プログラム停止中に使用するクロックを指定します。

次のドロップダウン・リストにより指定します。

システム	メイン・クロックで動作します (デフォルト)。
ユーザ	プログラムで設定されているクロックで動作します。

- [TCK/DCK クロック [MHz]] 【E1(JTAG)】 【V850E1】 【V850ES】】

デバッグ・コントロール・ユニット (DCU) へ供給するクロックをドロップダウン・リストにより指定します。

ドロップダウン・リストには、次の周波数が表示されます。

10MHz (デフォルト), 20MHz

注意 1. 通常は、[10MHz] の指定を変更しないでください。

[20MHz] を指定した場合、デバッグ・ツールと接続できない可能性があります。

2. E1(JTAG) が CubeSuite+ に接続している場合、このプロパティ値を変更することはできません。

- [タイマ用クロック周波数 [MHz]] 【V850E2】

タイマ用のクロック周波数を指定します。

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 (デフォルトでは [80.00] が指定されます)。

ただし、[CPU クロック周波数 [MHz]] 【V850E2】 プロパティを表示している場合は、このプロパティ値を変更することはできません ([CPU クロック周波数 [MHz]] プロパティと同値に固定されます)。

- [CPUクロック周波数 [MHz]] 【V850E2】

このプロパティは、選択しているマイクロコントローラが V850E2 の一部製品でのみ表示されます。
CPU クロック周波数（逡倍後）を指定します。

ドロップダウン・リストによる選択か、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。

ドロップダウン・リストには、次の周波数が表示されます（単位：MHz）。

80.00（デフォルト）、100.00

(d) [ターゲット・ボードとの接続]

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

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値を変更することはできません。

図 2—80 [ターゲット・ボードとの接続] カテゴリ 【E1(Serial)】

☐ ターゲット・ボードとの接続	
通信方式	UARTA0
エミュレータから電源供給をする(最大200mA)	はい
供給電圧	3.3V

図 2—81 [ターゲット・ボードとの接続] カテゴリ 【E1(LPD)/(JTAG)】

☐ ターゲット・ボードとの接続	
エミュレータから電源供給をする(最大200mA)	はい
供給電圧	3.3V

- [通信方式] 【E1(Serial)】

E1(Serial) がターゲット・ボード上のマイクロコントローラとシリアル通信を行う際に接続するポートをドロップダウン・リストにより指定します。

ドロップダウン・リスト内に表示されるポートの種類は、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、選択しているマイクロコントローラにより、シリアル通信の際に使用できるポートが限定される場合は、デフォルトで表示している通信方式を変更することはできません（デフォルトの指定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）。

- [エミュレータから電源供給をする (最大 200mA)]

E1 からターゲット・ボードへ電源を供給するかどうかをドロップダウン・リストにより指定します。

電源を供給する場合は [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）。

- [供給電圧]

このプロパティは、[\[エミュレータから電源供給をする \(最大 200mA\)\]](#) プロパティが表示されている際に [はい] を指定した場合のみ表示されます。

ターゲット・ボードへ供給する電圧を次のドロップダウン・リストにより指定します。

3.3V (デフォルト), 5.0V

(e) [フラッシュ]

このカテゴリでは、フラッシュ書き換えに関する設定を行います。

注意 E1 が CubeSuite+ に接続している場合、このカテゴリ内のプロパティ値を変更することはできません。

図 2—82 [フラッシュ] カテゴリ【E1(Serial)】

□ フラッシュ	
セキュリティID	HEX FFFFFFFFFFFFFFFFFFFFFFFF
起動時にフラッシュROMを消去する	いいえ

図 2—83 [フラッシュ] カテゴリ【E1(JTAG)【V850E1】【V850ES】】

□ フラッシュ	
セキュリティID	HEX FFFFFFFFFFFFFFFFFFFFFFFF

図 2—84 [フラッシュ] カテゴリ【E1(LPD)/(JTAG)【V850E2】】

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

- [セキュリティ ID]

このプロパティは、選択しているマイクロコントローラが、フラッシュ・メモリの ROM セキュリティ機能 (オンチップ・デバッグ・セキュリティ ID) をサポートしている場合のみ表示されます。

内蔵 ROM, または内蔵フラッシュ・メモリ上のコードを読み出す際のセキュリティ ID を指定します。

なお、オンチップ・デバッグ・セキュリティ ID 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。

- 【V850E1】【V850ES】

直接入力により、20 桁の 16 進数 (10 バイト) で指定します (デフォルトでは [FFFFFFFFFFFFFFFFFFFFFF] が指定されます)。

- 【V850E2】

直接入力により、24 桁の 16 進数 (12 バイト) で指定します (デフォルトでは [FFFFFFFFFFFFFFFFFFFFFF] が指定されます)。

備考 【V850E1】【V850ES】

このプロパティの設定を変更すると、ビルド・ツールのセキュリティ ID を変更するか否かを確認するメッセージを表示します。

- [起動時にフラッシュ ROM を消去する] 【E1(Serial)】

デバッグ・ツールと接続する際に、フラッシュ ROM の消去を行うか否かをドロップダウン・リストにより指定します。

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）。

なお、このプロパティは、デバッグ・ツールと接続後、自動的に [いいえ] に設定されます。

- [フラッシュ・セルフ・プログラミングを行う] 【E1(LPD)/(JTAG) 【V850E2】】

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、内蔵フラッシュ ROM の書き換えを行うか否かをドロップダウン・リストにより指定します。

ただし、ホット・プラグイン接続時は、この指定は無視されます。

内蔵フラッシュ ROM の書き換えを行う場合は、[はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

なお、[はい] を指定した場合、内蔵フラッシュ ROM のキャッシュは行われません。

(f) [プログラマブル I/O] 【V850E1】 【V850ES】

このカテゴリでは、プログラマブル I/O 領域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プログラマブル I/O 領域をサポートしている場合のみ指定可能です。

備考 デバッグを行う際に、一時的に値を変更する場合にのみ設定を変更してください。

プロジェクトとして共通の値を設定する場合は、使用するビルド・ツールのプロパティパネルの [共通オプション] タブ上の [デバイス] カテゴリ内 [プログラマブル I/O 領域開始アドレス] プロパティで設定を行ってください。

図 2—85 [プログラマブル I/O] カテゴリ

プログラマブル I/O	
プログラマブル I/O 領域を使用する	はい
プログラマブル I/O 領域開始アドレス	HEX 0

- [プログラマブル I/O 領域を使用する]

プログラマブル I/O 領域を使用するか否かをドロップダウン・リストにより指定します。

プログラマブル I/O 領域を使用する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

- [プログラマブル I/O 領域開始アドレス]

このプロパティは、[プログラマブル I/O 領域を使用する] 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。

プログラマブル I/O 領域の開始アドレスを指定します。

直接入力により、プログラマブル I/O 領域内のアドレスを指定してください（デフォルトでは [0] が指定されます）。

なお、アドレスは、16 K バイトでアラインされます。

(2) [デバッグ・ツール設定] タブ

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

- (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레이크]
- (e) [入力信号のマスク]
- (f) [外部フラッシュ・メモリ・ダウンロード] **[E1(LPD)/(JTAG)]**

(a) [メモリ]

このカテゴリでは、メモリに関する設定を行います。

図 2—86 [メモリ] カテゴリ **[E1]**

☐ メモリ	
☐ メモリ・マッピング	[4]
田 [0]	内蔵ROM領域
田 [1]	ノン・マップ領域
田 [2]	内蔵RAM領域
田 [3]	I/Oレジスタ領域
メモリ書き込み時にベリファイを行う	はい


- [メモリ・マッピング]

現在のメモリ・マッピングの状況が、メモリ領域の種別ごとに詳細表示されます。

このパネル上でマッピング値を変更することはできません。メモリ・マッピングを追加する必要がある場合は、[メモリ・マッピング] プロパティを選択することで設定欄右端に表示される [...] ボタンのクリックによりオープンする**メモリ・マッピング ダイアログ**で行います。

設定方法についての詳細は、**メモリ・マッピング ダイアログ**の項を参照してください。

図 2—87 メモリ・マッピング ダイアログのオープン

☐ メモリ	
☐ メモリ・マッピング	[4] 
田 [0]	内蔵ROM領域
田 [1]	ノン・マップ領域

注意 デバッグ・ツールと未接続の場合、ユーザにより追加されたメモリ・マッピング領域のみが表示対象となります。

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

- [メモリ書き込み時にベリファイを行う]

メモリ値の初期化を行う際に、ベリファイを行うか否かをドロップダウン・リストにより指定します。ベリファイを行う場合は [はい] を選択してください (デフォルト)。

(b) [実行中のメモリ・アクセス]

このカテゴリでは、プログラム実行中におけるメモリ・アクセス (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—88 [実行中のメモリ・アクセス] カテゴリ 【E1(Serial)】

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

図 2—89 [実行中のメモリ・アクセス] カテゴリ 【E1(JTAG) 【V850E1】 【V850ES】】

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

図 2—90 [実行中のメモリ・アクセス] カテゴリ 【E1(LPD)/(JTAG) 【V850E2】】

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

- [実行中にアクセスする] 【E1(LPD)/(JTAG) 【V850E2】】

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ドロップダウン・リストにより指定します。なお、[フラッシュ] カテゴリ内 [フラッシュ・セルフ・プログラミングを行う] 【E1(LPD)/(JTAG) 【V850E2】】 プロパティにおいて [いいえ] を選択している場合は、内蔵 ROM 領域も対象となります。アクセスを許可する場合は [はい] を選択してください (デフォルトでは [いいえ] が指定されます)。

- [実行を一瞬停止してアクセスする]

このプロパティは、選択しているマイクロコントローラがマルチコア版の場合、または [実行中にアクセスする] 【E1(LPD)/(JTAG) 【V850E2】】 プロパティにおいて [いいえ] を指定した場合は表示されません。

プログラム実行中にはアクセスできないメモリ領域（ターゲット・メモリ領域 /I/O レジスタ領域 /CPU レジスタなど）に対して、実行を一瞬停止し、アクセスを許可するか否かをドロップダウン・リストにより指定します。

なお、[フラッシュ] カテゴリ内 [フラッシュ・セルフ・プログラミングを行う] 【E1(LPD)/(JTAG) 【V850E2】】 プロパティにおいて [いいえ] を選択している場合は、内蔵 ROM 領域も内蔵 RAM 領域と同様の扱いとなります。

アクセスを許可する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されません）。

備考 【V850E2】

内蔵 RAM 領域に対しては、このプロパティの指定にかかわらず、一瞬停止することなくアクセスします。

- [実行中に表示更新を行う]

プログラム実行中に、**ウォッチパネル**/**メモリパネル**の表示内容を自動的に更新するか否かをドロップダウン・リストにより指定します。

表示内容の更新を行う場合は [はい] を選択してください（デフォルト）。

- [表示更新間隔 [ms]]

このプロパティは、**[実行中に表示更新を行う]** プロパティにおいて [はい] を指定した場合にのみ有効となります。

プログラム実行中に、**ウォッチパネル**/**メモリパネル**の表示内容を自動的に更新する間隔を 100 ms 単位で指定します。

直接入力により、100 ~ 65500 の範囲の整数（100 ms 未満の端数切り上げ）を指定してください（デフォルトでは [500] が指定されます）。

- [HALT を解除してアクセスする] 【E1(LPD)/(JTAG)】

このプロパティは、**[実行を一瞬停止してアクセスする]** 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。

HALT モードを解除して、内蔵 RAM 領域以外のメモリ（CPU レジスタ /I/O レジスタ）にアクセスするか否かをドロップダウン・リストにより指定します（内蔵 RAM 領域へのアクセスに対しては HALT モードを解除しません）。

なお、[フラッシュ] カテゴリ内 [フラッシュ・セルフ・プログラミングを行う] 【E1(LPD)/(JTAG) 【V850E2】】 プロパティにおいて [いいえ] を選択している場合は、内蔵 ROM 領域も内蔵 RAM 領域と同様の扱いとなります。

HALT モードを解除してアクセスする場合は [はい] を選択してください（デフォルト）。

(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—91 [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ドロップダウン・リストにより指定します。

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（デフォルトでは「いいえ」が指定されます）。

(d) [ブレーク]

このカテゴリでは、ブレーク機能に関する設定を行います。

図 2—92 [ブレーク] カテゴリ【E1(Serial)】

ブレーク	
優先的に使用するブレークポイントの種類	ソフトウェア・ブレーク

図 2—93 [ブレーク] カテゴリ【E1(LPD)/(JTAG)】

ブレーク	
優先的に使用するブレークポイントの種類	ソフトウェア・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
オープン・ブレーク機能を使用する	いいえ (Output signal)

- [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、選択しているマイクロコントローラが使用可能なブレークポイントの種類が1つのみの場合は表示されません。

[エディタパネル／逆アセンブルパネル](#)において、マウスのワンクリック操作でブレークポイントを設定する際に使用するブレークポイントの種別を次のドロップダウン・リストにより指定します。

なお、ブレークポイントについての詳細は、「[2.8.2 任意の場所で停止する（ブレークポイント）](#)」を参照してください。

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

- [停止時に周辺エミュレーションを停止する] 【E1(LPD)/(JTAG)】

プログラム実行停止時に、エミュレータの周辺エミュレーション機能を停止するか否かをドロップダウン・リストにより指定します。

停止する場合は [はい] を選択してください (デフォルトでは [いいえ] が指定されます)。

- [オープン・ブ레이크機能を使用する] 【E1(LPD)/(JTAG)】

このプロパティは、選択しているマイクロコントローラがオープン・ブ레이크機能をサポートしている場合にのみ表示されます。

オープン・ブ레이크機能を使用するか否かを次のドロップダウン・リストにより指定します。

はい (Hi-Z)	CPU 停止時、オープン・ブ레이크の対象端子は Hi-Z になります。
いいえ (Output signal)	CPU 停止時、オープン・ブ레이크の対象端子は通常の状態のまま動作します。

(e) [入力信号のマスク]

このカテゴリでは、入力信号のマスクに関する設定を行います。

図 2—94 [入力信号のマスク] カテゴリ 【E1(Serial)】

☐ 入力信号のマスク	
RESET 信号をマスクする	いいえ

図 2—95 [入力信号のマスク] カテゴリ 【E1(JTAG) 【V850E1】 【V850ES】】

☐ 入力信号のマスク	
NMI0 信号をマスクする	いいえ
NMI1 信号をマスクする	いいえ
NMI2 信号をマスクする	いいえ
HLDREQ 信号をマスクする	いいえ
RESET 信号をマスクする	いいえ
STOP 信号をマスクする	いいえ
WAIT 信号をマスクする	いいえ
DBINT 信号をマスクする	いいえ

図 2—96 [入力信号のマスク] カテゴリ 【E1(LPD)/(JTAG) 【V850E2】】

☐ 入力信号のマスク	
HLDREQ信号をマスクする	いいえ
RESET信号をマスクする	いいえ
STOP信号をマスクする	いいえ
WAIT信号をマスクする	いいえ

次に示す各プロパティの設定において、該当する信号をマスクする場合は [はい] を、マスクしない場合は [いいえ] をドロップダウン・リストにより指定してください (デフォルトでは、すべてのプロパティに [いいえ] が指定されます)。

- [NMI n 信号をマスクする] 【E1(JTAG) 【V850E1】 【V850ES】】
- [HLDREQ 信号をマスクする] 【E1(LPD)/(JTAG)】

- [RESET 信号をマスクする]
- [STOP 信号をマスクする] 【E1(LPD)/(JTAG)】
- [WAIT 信号をマスクする] 【E1(LPD)/(JTAG)】
- [DBINT 信号をマスクする] 【E1(JTAG) 【V850E1】 【V850ES】】

(f) [外部フラッシュ・メモリ・ダウンロード] 【E1(LPD)/(JTAG)】

このカテゴリでは、外部フラッシュ・メモリへのダウンロードに関する設定を行います。

外部フラッシュ・メモリへのダウンロード方法、およびこのカテゴリ内の設定についての詳細は、「(5) 外部フラッシュ・メモリへダウンロードする 【IECUBE】 【IECUBE2】 【MINICUBE】 【E1/E20(LPD)/(JTAG)】」を参照してください。

(3) [ダウンロード・ファイル設定] タブ

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

(4) [フラッシュ・オプション設定] タブ 【V850E2】

[フラッシュ・オプション設定] タブ 【V850E2】 では、V850E2 コアの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 【V850E2】](#)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)。

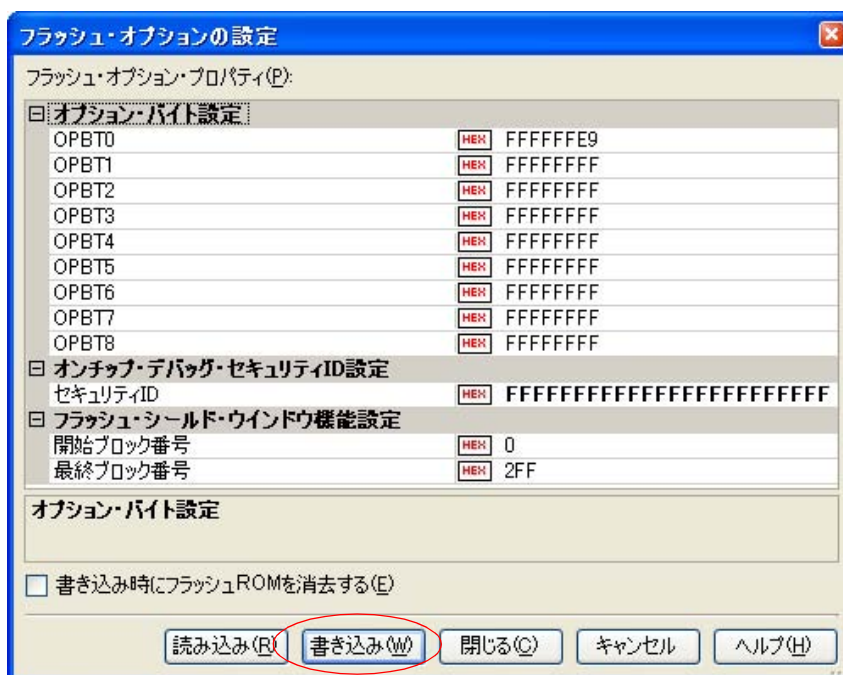
このダイアログにおいて、各項目の設定を完了したのち、[書き込み] ボタンをクリックしてください。

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 【V850E2】](#)を参照してください。

図 2—97 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—98 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【E1】）



(5) [フック処理設定] タブ

[フック処理設定] タブでは、デバッグ・ツールにフック処理の設定を行います。

フック処理、および各カテゴリ内の設定についての詳細は、「[2.16 フック処理を設定する](#)」を参照してください。

2.3.5 【E20】の場合

E20 を使用する場合の動作環境の設定を次の**プロパティ パネル**で行います。

なお、ターゲット・ボードとの通信方式（シリアル通信【E20(Serial)】/LPD 通信【E20(LPD)】/JTAG 通信【E20(JTAG)】）により、**プロパティ パネル**の内容は異なります。

図 2—99 動作環境設定【E20(Serial)【V850E1】【V850ES】】（プロパティ パネルの例）



図 2—100 動作環境設定【E20(Serial)【V850E2】】（プロパティ パネルの例）

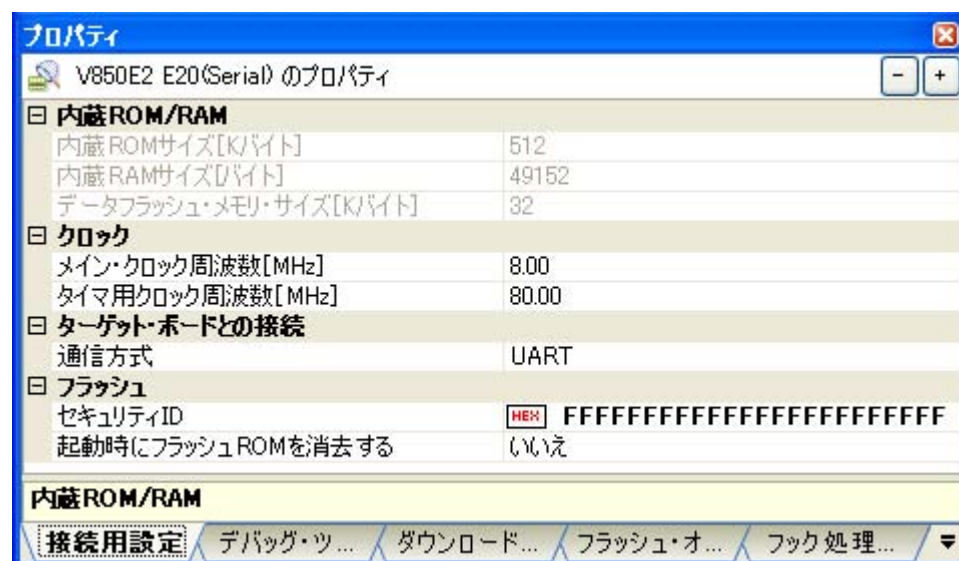
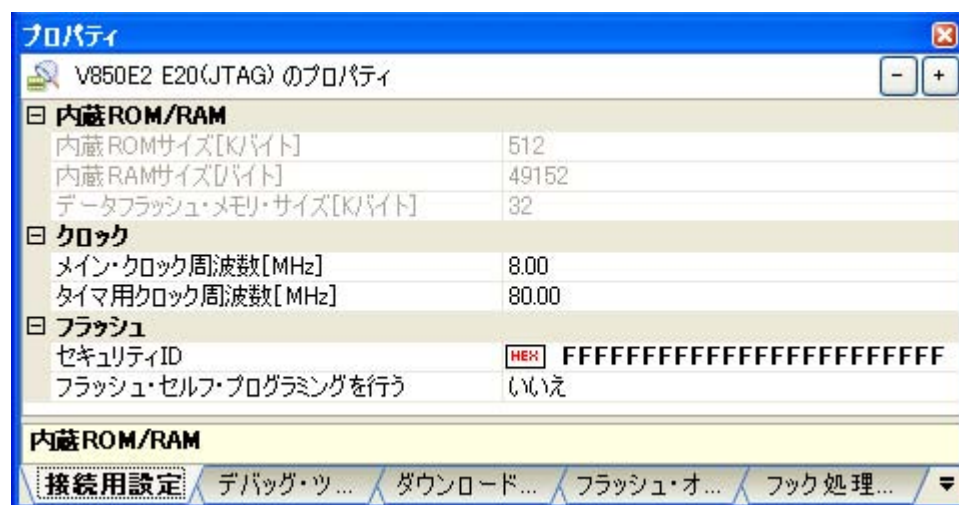


図 2—101 動作環境設定【E20(JTAG)【V850E1】【V850ES】】(プロパティパネルの例)



図 2—102 動作環境設定【E20(LPD)/(JTAG)【V850E2】】(プロパティパネルの例)



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【V850E2】
- (5) [フック処理設定] タブ

(1) [接続用設定] タブ

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

- (a) [内蔵 ROM/RAM]
- (b) [データフラッシュ] 【V850E1】 【V850ES】
- (c) [クロック]
- (d) [ターゲット・ボードとの接続] 【E20(Serial)】
- (e) [フラッシュ]
- (f) [プログラマブル I/O] 【V850E1】 【V850ES】

(a) [内蔵 ROM/RAM]

このカテゴリでは、内蔵 ROM/RAM に関する設定を表示します。

図 2—103 [内蔵 ROM/RAM] カテゴリ 【E20 【V850E1】 【V850ES】】

内蔵ROM/RAM	
内蔵ROMサイズ[K/バイト]	256
内蔵RAMサイズ[B/バイト]	16384

図 2—104 [内蔵 ROM/RAM] カテゴリ 【E20 【V850E2】】

内蔵ROM/RAM	
内蔵ROMサイズ[K/バイト]	512
内蔵RAMサイズ[B/バイト]	40960
データフラッシュ・メモリ・サイズ[K/バイト]	32

- [内蔵 ROM サイズ [K バイト]]

エミュレーションする内蔵 ROM サイズを表示します (単位: K バイト)。

なお、選択しているマイクロコントローラが V850E1 コアで、VSB Flash ROM 搭載品の場合では、VSB Flash ROM サイズを加えた値を表示します。

このプロパティ値を変更することはできません。

- [内蔵 RAM サイズ [バイト]]

エミュレーションする内蔵 RAM サイズを表示します (単位: バイト)。

なお、選択しているマイクロコントローラが V850E1 コアで、VSB RAM 搭載品の場合では、VSB RAM サイズを加えた値を表示します。

また、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、“合計 RAM サイズ”(コア 1 つ分の RAM サイズ × コアの数) 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。

- [データフラッシュ・メモリ・サイズ [K バイト]] 【E20(LPD)/(JTAG) 【V850E2】】

選択しているマイクロコントローラのデータフラッシュ・メモリ領域のサイズを表示します (単位: K バイト)。

選択しているマイクロコントローラがデータフラッシュ・メモリ非搭載品の場合は、[0] を表示します。

このプロパティ値を変更することはできません。

(b) [データフラッシュ] [V850E1] [V850ES]

このカテゴリでは、データフラッシュ・メモリに関する設定を行います。

ただし、このカテゴリは、データフラッシュ・メモリ内蔵品の場合のみ表示されます。

図 2—105 [データフラッシュ] カテゴリ

☐ データフラッシュ	
データフラッシュ・メモリ領域を使用する	はい
データフラッシュ・メモリ・サイズ[Kバイト]	32
チップ・セレクト	CS0

- [データフラッシュ・メモリ領域を使用する]

データフラッシュ・メモリを使用するか否かをドロップダウン・リストにより指定します。

データフラッシュ・メモリを使用する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

- [データフラッシュ・メモリ・サイズ[Kバイト]]

このプロパティは、[データフラッシュ・メモリ領域を使用する] 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。

選択しているマイクロコントローラのデータフラッシュ・メモリ領域のサイズを表示します（単位：Kバイト）。

このプロパティ値を変更することはできません。

- [チップ・セレクト]

このプロパティは、[データフラッシュ・メモリ領域を使用する] プロパティにおいて、[はい] を指定した場合のみ表示されます。

データフラッシュ・メモリをマッピングする際のチップ・セレクトをドロップ・ダウンリストにより指定します。

なお、指定できるチップ・セレクトは、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、使用するマイクロコントローラに V850ES コア以外を選択している場合、デフォルトで表示しているチップ・セレクトを変更することはできません。

(c) [クロック]

このカテゴリでは、クロックに関する設定を行います。

図 2—106 [クロック] カテゴリ【E20(Serial)【V850E1】【V850ES】】

クロック	
メイン・クロック周波数[MHz]	4.00
メイン・クロック通倍	1
サブ・クロック周波数[kHz]	32.765
モニタ・クロック	システム

図 2—107 [クロック] カテゴリ【E20(JTAG)【V850E1】【V850ES】】

クロック	
メイン・クロック周波数[MHz]	4.00
メイン・クロック通倍	1
サブ・クロック周波数[kHz]	32.765
モニタ・クロック	システム
TCK/DCKクロック[MHz]	10MHz

図 2—108 [クロック] カテゴリ【E20【V850E2】】

クロック	
メイン・クロック周波数[MHz]	8.00
タイマ用クロック周波数[MHz]	80.00

図 2—109 [クロック] カテゴリ【E20】(V850E2の一部製品)

クロック	
CPUクロック周波数[MHz]	80.00
タイマ用クロック周波数[MHz]	80.00

- [メイン・クロック周波数 [MHz]]

メイン・クロック周波数（通倍前）を指定します。

ドロップダウン・リストによる選択か、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。

ドロップダウン・リストには、次の周波数が表示されます（単位：MHz）。

- 【V850E1】【V850ES】

2.50, 4.00（デフォルト）, 5.00, 6.00, 8.00, 13.50, 18.00

- 【V850E2】

4.00, 5.00, 7.20, 8.00（デフォルト）, 9.60, 10.00, 13.50, 16.00

注意 【V850E2】

このプロパティは、**[CPU クロック周波数 [MHz]]** **[V850E2]** プロパティが表示されている場合は非表示となります。

- [メイン・クロック通倍] 【V850E1】 【V850ES】

メイン・クロック周波数に対する通倍数を指定します。

ドロップダウン・リストによる選択か、または直接入力により、1～99の範囲の値を指定します。

ドロップダウン・リストには、次の通倍数が表示されます。

1 (デフォルト), 2, 3, 4, 5, 6, 7, 8, 9, 10

- [サブ・クロック周波数 [kHz]] 【V850E1】 【V850ES】

このプロパティは、選択しているマイクロコントローラがサブ・クロックをサポートしている場合のみ表示されます。

サブ・クロック周波数を指定します。

ドロップダウン・リストによる選択か、または直接入力により、0.001～999.999 (単位: kHz) の範囲の数値を指定します。

ドロップダウン・リストには、現在選択しているマイクロコントローラが使用可能なサブ・クロック周波数が表示されます。

デフォルトでは、使用可能なサブ・クロック周波数の最小値が指定されます。

- [モニタ・クロック] 【V850E1】 【V850ES】

このプロパティは、選択しているマイクロコントローラがサブ・クロックをサポートしている場合のみ表示されます。

プログラム停止中に使用するクロックを指定します。

次のドロップダウン・リストにより指定します。

システム	メイン・クロックで動作します (デフォルト)。
ユーザ	プログラムで設定されているクロックで動作します。

- [TCK/DCK クロック [MHz]] 【E20(JTAG)】 【V850E1】 【V850ES】】

デバッグ・コントロール・ユニット (DCU) へ供給するクロックをドロップダウン・リストにより指定します。

ドロップダウン・リストには、次の周波数が表示されます。

10MHz (デフォルト), 20MHz

注意 1. 通常は、[10MHz] の指定を変更しないでください。

[20MHz] を指定した場合、デバッグ・ツールと接続できない可能性があります。

2. E20(JTAG) が CubeSuite+ に接続している場合、このプロパティ値を変更することはできません。

- [タイマ用クロック周波数 [MHz]] 【V850E2】

タイマ用のクロック周波数を指定します。

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 (デフォルトでは [80.00] が指定されます)。

ただし、[CPUクロック周波数 [MHz]] 【V850E2】 プロパティを表示している場合は、このプロパティ値を変更することはできません（[CPUクロック周波数 [MHz]] プロパティと同値に固定されます）。

- [CPUクロック周波数 [MHz]] 【V850E2】

このプロパティは、選択しているマイクロコントローラが V850E2 の一部製品でのみ表示されます。ドロップダウン・リストによる選択か、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。

ドロップダウン・リストには、次の周波数が表示されます（単位：MHz）。

80.00（デフォルト）、100.00

(d) [ターゲット・ボードとの接続] 【E20(Serial)】

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

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値を変更することはできません。

図 2—110 [ターゲット・ボードとの接続] カテゴリ 【E20(Serial)】

<div style="background-color: #e0e0e0; padding: 2px;"> 日 ターゲット・ボードとの接続 </div>	
通信方式	UART0

- [通信方式]

E20(Serial) がターゲット・ボード上のマイクロコントローラとシリアル通信を行う際に接続するポートをドロップダウン・リストにより指定します。

ドロップダウン・リスト内に表示されるポートの種類は、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、選択しているマイクロコントローラにより、シリアル通信の際に使用できるポートが限定される場合は、デフォルトで表示している通信方式を変更することはできません（デフォルトの指定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）。

(e) [フラッシュ]

このカテゴリでは、フラッシュ書き換えに関する設定を行います。

注意 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値を変更することはできません。

図 2—111 [フラッシュ] カテゴリ 【E20(Serial)】

<div style="background-color: #e0e0e0; padding: 2px;"> 日 フラッシュ </div>	
セキュリティID	<div style="border: 1px solid red; display: inline-block; padding: 0 2px;">HEX</div> FFFFFFFFFFFFFFFFFF
起動時にフラッシュROMを消去する	いいえ

図 2—112 [フラッシュ] カテゴリ【E20(JTAG)【V850E1】【V850ES】】

□ フラッシュ	
セキュリティID	HEX FFFFFFFFFFFFFFFFFF

図 2—113 [フラッシュ] カテゴリ【E20(LPD)/(JTAG)【V850E2】】

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

- [セキュリティ ID]

このプロパティは、選択しているマイクロコントローラが、フラッシュ・メモリの ROM セキュリティ機能（オンチップ・デバッグ・セキュリティ ID）をサポートしている場合のみ表示されます。

内蔵 ROM、または内蔵フラッシュ・メモリ上のコードを読み出す際のセキュリティ ID を指定します。

なお、オンチップ・デバッグ・セキュリティ ID 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。

- 【V850E1】【V850ES】

直接入力により、20 桁の 16 進数（10 バイト）で指定します（デフォルトでは [FFFFFFFFFFFFFFFF] が指定されます）。

- 【V850E2】

直接入力により、24 桁の 16 進数（12 バイト）で指定します（デフォルトでは [FFFFFFFFFFFFFFFF] が指定されます）。

備考 【V850E1】【V850ES】

このプロパティの設定を変更すると、ビルド・ツールのセキュリティ ID を変更するか否かを確認するメッセージを表示します。

- [起動時にフラッシュ ROM を消去する]【E20(Serial)】

デバッグ・ツールと接続する際に、フラッシュ ROM の消去を行うか否かをドロップダウン・リストにより指定します。

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）。

なお、このプロパティは、デバッグ・ツールと接続後、自動的に [いいえ] に設定されます。

- [フラッシュ・セルフ・プログラミングを行う]【E20(LPD)/(JTAG)【V850E2】】

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、内蔵フラッシュ ROM の書き換えを行うか否かをドロップダウン・リストにより指定します。

ただし、ホット・プラグイン接続時は、この指定は無視されます。

内蔵フラッシュ ROM の書き換えを行う場合は、[はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

なお、[はい] を指定した場合、内蔵フラッシュ ROM のキャッシュは行われません。

(f) [プログラマブル I/O] [V850E1] [V850ES]

このカテゴリでは、プログラマブル I/O 領域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プログラマブル I/O 領域をサポートしている場合のみ指定可能です。

備考 デバッグを行う際に、一時的に値を変更する場合にのみ設定を変更してください。

プロジェクトとして共通の値を設定する場合は、使用するビルド・ツールのプロパティパネルの [共通オプション] タブ上の [デバイス] カテゴリ内 [プログラマブル I/O 領域開始アドレス] プロパティで設定を行ってください。

図 2—114 [プログラマブル I/O] カテゴリ

プログラマブル I/O	
プログラマブル I/O 領域を使用する	はい
プログラマブル I/O 領域開始アドレス	HEX 0

- [プログラマブル I/O 領域を使用する]

プログラマブル I/O 領域を使用するか否かをドロップダウン・リストにより指定します。

プログラマブル I/O 領域を使用する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

- [プログラマブル I/O 領域開始アドレス]

このプロパティは、[プログラマブル I/O 領域を使用する] 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。

プログラマブル I/O 領域の開始アドレスを指定します。

直接入力により、プログラマブル I/O 領域内のアドレスを指定してください（デフォルトでは [0] が指定されます）。

なお、アドレスは、16 K バイトでアラインされます。

(2) [デバッグ・ツール設定] タブ

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

(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) [入力信号のマスク]

(f) [外部フラッシュ・メモリ・ダウンロード] [E20(LPD)/(JTAG)]

(a) [メモリ]

このカテゴリでは、メモリに関する設定を行います。

図 2—115 [メモリ] カテゴリ [E20]

☐ メモリ	
☐ メモリ・マッピング	[4]
田 [0]	内蔵ROM領域
田 [1]	ノン・マップ領域
田 [2]	内蔵RAM領域
田 [3]	I/Oレジスタ領域
メモリ書き込み時にベリファイを行う	はい


- [メモリ・マッピング]

現在のメモリ・マッピングの状況が、メモリ領域の種別ごとに詳細表示されます。

このパネル上でマッピング値を変更することはできません。メモリ・マッピングを追加する必要がある場合は、[メモリ・マッピング] プロパティを選択することで設定欄右端に表示される [...] ボタンのクリックによりオープンする**メモリ・マッピング ダイアログ**で行います。

設定方法についての詳細は、[メモリ・マッピング ダイアログ](#)の項を参照してください。

図 2—116 メモリ・マッピング ダイアログのオープン

☐ メモリ	
☐ メモリ・マッピング	[4] 
田 [0]	内蔵ROM領域
田 [1]	ノン・マップ領域

注意 デバッグ・ツールと未接続の場合、ユーザにより追加されたメモリ・マッピング領域のみが表示対象となります。

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

- [メモリ書き込み時にベリファイを行う]

メモリ値の初期化を行う際に、ベリファイを行うか否かをドロップダウン・リストにより指定します。ベリファイを行う場合は [はい] を選択してください（デフォルト）。

(b) [実行中のメモリ・アクセス]

このカテゴリでは、プログラム実行中におけるメモリ・アクセス（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—117 【実行中のメモリ・アクセス】カテゴリ【E20(Serial)】

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

図 2—118 【実行中のメモリ・アクセス】カテゴリ【E20(JTAG)】【V850E1】【V850ES】

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

図 2—119 【実行中のメモリ・アクセス】カテゴリ【E20(LPD)/(JTAG)】【V850E2】

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

- 【実行中にアクセスする】【E20(LPD)/(JTAG)】【V850E2】

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ドロップダウン・リストにより指定します。なお、[フラッシュ] カテゴリ内 [フラッシュ・セルフ・プログラミングを行う]【E20(LPD)/(JTAG)】【V850E2】プロパティにおいて [いいえ] を選択している場合は、内蔵 ROM 領域も対象となります。アクセスを許可する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

- 【実行を一瞬停止してアクセスする】

このプロパティは、選択しているマイクロコントローラがマルチコア版の場合、または [実行中にアクセスする]【E20(LPD)/(JTAG)】【V850E2】プロパティにおいて [いいえ] を指定した場合は表示されません。

プログラム実行中にはアクセスできないメモリ領域（ターゲット・メモリ領域 /I/O レジスタ領域 /CPU レジスタなど）に対して、実行を一瞬停止し、アクセスを許可するか否かをドロップダウン・リストにより指定します。

なお、[フラッシュ] カテゴリ内 [フラッシュ・セルフ・プログラミングを行う]【E20(LPD)/(JTAG)】【V850E2】プロパティにおいて [いいえ] を選択している場合は、内蔵 ROM 領域も内蔵 RAM 領域と同様の扱いとなります。

アクセスを許可する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

備考 【V850E2】

内蔵 RAM 領域に対しては、このプロパティの指定にかかわらず、一瞬停止することなくアクセスします。

- [実行中に表示更新を行う]

プログラム実行中に、**ウォッチパネル／メモリパネル**の表示内容を自動的に更新するか否かをドロップダウン・リストにより指定します。

表示内容の更新を行う場合は [はい] を選択してください (デフォルト)。

- [表示更新間隔 [ms]]

このプロパティは、[実行中に表示更新を行う] プロパティにおいて [はい] を指定した場合にのみ有効となります。

プログラム実行中に、**ウォッチパネル／メモリパネル**の表示内容を自動的に更新する間隔を 100 ms 単位で指定します。

直接入力により、100 ~ 65500 の範囲の整数 (100 ms 未満の端数切り上げ) を指定してください (デフォルトでは [500] が指定されます)。

- [HALT を解除してアクセスする] 【E20(LPD)/(JTAG)】

このプロパティは、[実行を一瞬停止してアクセスする] 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。

HALT モードを解除して、内蔵 RAM 領域以外のメモリ (CPU レジスタ //I/O レジスタ) にアクセスするか否かをドロップダウン・リストにより指定します (内蔵 RAM 領域へのアクセスに対しては HALT モードを解除しません)。

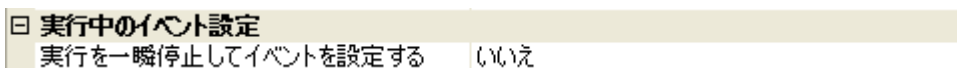
なお、[フラッシュ] カテゴリ内 [フラッシュ・セルフ・プログラミングを行う] 【E20(LPD)/(JTAG) 【V850E2】】 プロパティにおいて [いいえ] を選択している場合は、内蔵 ROM 領域も内蔵 RAM 領域と同様の扱いとなります。

HALT モードを解除してアクセスする場合は [はい] を選択してください (デフォルト)。

(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—120 [実行中のイベント設定] カテゴリ



- [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ドロップダウン・リストにより指定します。

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（デフォルトでは「いいえ」が指定されます）。

(d) [ブレーク]

このカテゴリでは、ブレーク機能に関する設定を行います。

図 2—121 [ブレーク] カテゴリ【E20(Serial)】

☐ ブレーク	
優先的に使用するブレークポイントの種類	ソフトウェア・ブレーク

図 2—122 [ブレーク] カテゴリ【E20(LPD)/(JTAG)】

☐ ブレーク	
優先的に使用するブレークポイントの種類	ソフトウェア・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
オープン・ブレーク機能を使用する	いいえ (Output signal)

- [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、選択しているマイクロコントローラが使用可能なブレークポイントの種類が1つのみの場合は表示されません。

[エディタ パネル／逆アセンブル パネル](#)において、マウスのワンクリック操作でブレークポイントを設定する際に使用するブレークポイントの種別を次のドロップダウン・リストにより指定します。

なお、ブレークポイントについての詳細は、「[2.8.2 任意の場所で停止する \(ブレークポイント\)](#)」を参照してください。

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

- [停止時に周辺エミュレーションを停止する]【E20(LPD)/(JTAG)】

プログラム実行停止時に、エミュレータの周辺エミュレーション機能を停止するか否かをドロップダウン・リストにより指定します。

停止する場合は「はい」を選択してください（デフォルトでは「いいえ」が指定されます）。

- [オープン・ブレーク機能を使用する]【E20(LPD)/(JTAG)】

このプロパティは、選択しているマイクロコントローラがオープン・ブレーク機能をサポートしている場合にのみ表示されます。オープン・ブレーク機能を使用するか否かを次のドロップダウン・リストにより指定します。

はい (Hi-Z)	CPU 停止時、オープン・ブレークの対象端子は Hi-Z になります。
いいえ (Output signal)	CPU 停止時、オープン・ブレークの対象端子は通常の状態のまま動作します。

(e) [入力信号のマスク]

このカテゴリでは、入力信号のマスクに関する設定を行います。

図 2—123 [入力信号のマスク] カテゴリ【E20(Serial)】

□ 入力信号のマスク	
RESET 信号をマスクする	いいえ

図 2—124 [入力信号のマスク] カテゴリ【E20(JTAG)【V850E1】【V850ES】】

□ 入力信号のマスク	
NMI0 信号をマスクする	いいえ
NMI1 信号をマスクする	いいえ
NMI2 信号をマスクする	いいえ
HLDRQ 信号をマスクする	いいえ
RESET 信号をマスクする	いいえ
STOP 信号をマスクする	いいえ
WAIT 信号をマスクする	いいえ
DBINT 信号をマスクする	いいえ

図 2—125 [入力信号のマスク] カテゴリ【E20(LPD)/(JTAG)【V850E2】】

□ 入力信号のマスク	
HLDRQ 信号をマスクする	いいえ
RESET 信号をマスクする	いいえ
STOP 信号をマスクする	いいえ
WAIT 信号をマスクする	いいえ

次に示す各プロパティの設定において、該当する信号をマスクする場合は [はい] を、マスクしない場合は [いいえ] をドロップダウン・リストにより指定してください（デフォルトでは、すべてのプロパティに [いいえ] が指定されます）。

- [NMI n 信号をマスクする]【E20(JTAG)【V850E1】【V850ES】】
- [HLDRQ 信号をマスクする]【E20(LPD)/(JTAG)】
- [RESET 信号をマスクする]
- [STOP 信号をマスクする]【E20(LPD)/(JTAG)】
- [WAIT 信号をマスクする]【E20(LPD)/(JTAG)】
- [DBINT 信号をマスクする]【E20(JTAG)【V850E1】【V850ES】】

(f) [外部フラッシュ・メモリ・ダウンロード]【E20(LPD)/(JTAG)】

このカテゴリでは、外部フラッシュ・メモリへのダウンロードに関する設定を行います。

外部フラッシュ・メモリへのダウンロード方法、およびこのカテゴリ内の設定についての詳細は、「(5) 外部フラッシュ・メモリへダウンロードする【IECUBE】【IECUBE2】【MINICUBE】【E1/E20(LPD)/(JTAG)】」を参照してください。

(3) [ダウンロード・ファイル設定] タブ

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

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

(4) [フラッシュ・オプション設定] タブ【V850E2】

[フラッシュ・オプション設定] タブ【V850E2】では、V850E2 コアの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【V850E2】](#)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）。

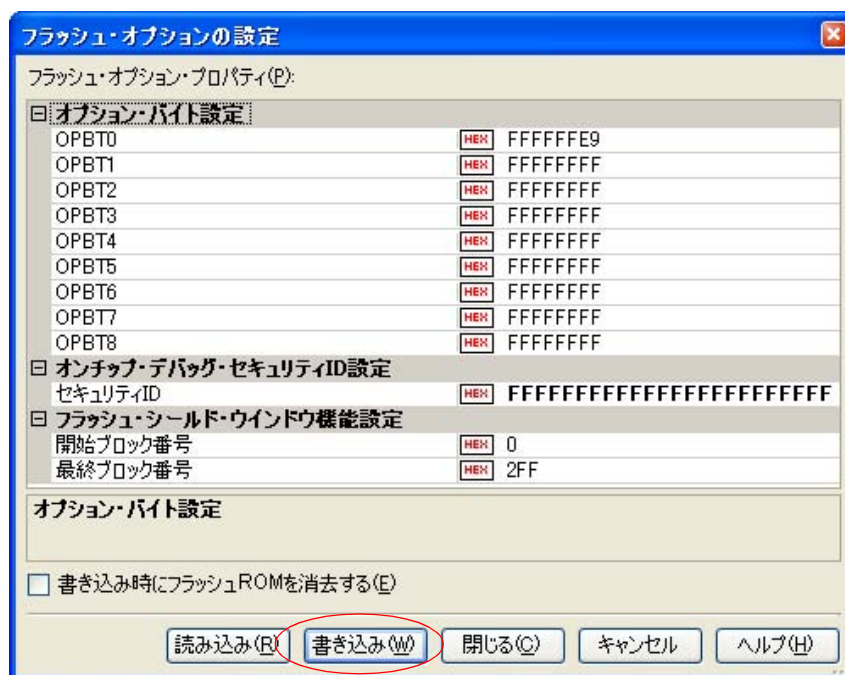
このダイアログにおいて、各項目の設定を完了したのち、[書き込み] ボタンをクリックしてください。

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【V850E2】](#)を参照してください。

図 2—126 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—127 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【E20】）



(5) [フック処理設定] タブ

[フック処理設定] タブでは、デバッグ・ツールにフック処理の設定を行います。

フック処理、および各カテゴリ内の設定についての詳細は、「[2.16 フック処理を設定する](#)」を参照してください。

2.3.6 【EZ Emulator】の場合

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

図 2—128 動作環境設定【EZ Emulator】（プロパティ パネルの例）



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

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

- (a) [内蔵 ROM/RAM]
- (b) [データフラッシュ]
- (c) [クロック]
- (d) [ターゲット・ボードとの接続]
- (e) [フラッシュ]
- (f) [プログラマブル I/O]

(a) [内蔵 ROM/RAM]

このカテゴリでは、内蔵 ROM/RAM に関する設定を表示します。

図 2—129 [内蔵 ROM/RAM] カテゴリ【EZ Emulator】

内蔵ROM/RAM	
内蔵ROMサイズ[Kバイト]	256
内蔵RAMサイズ[バイト]	16384

- [内蔵 ROM サイズ [K バイト]]

エミュレーションする内蔵 ROM サイズを表示します (単位 : K バイト)。

なお、選択しているマイクロコントローラが V850E1 コアで、VSB Flash ROM 搭載品の場合では、VSB Flash ROM サイズを加えた値を表示します。

このプロパティ値を変更することはできません。

- [内蔵 RAM サイズ [バイト]]

エミュレーションする内蔵 RAM サイズを表示します (単位 : バイト)。

なお、選択しているマイクロコントローラが V850E1 コアで、VSB RAM 搭載品の場合では、VSB RAM サイズを加えた値を表示します。

また、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、“合計 RAM サイズ”(コア 1 つ分の RAM サイズ x コアの数) 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。

(b) [データフラッシュ]

このカテゴリでは、データフラッシュ・メモリに関する設定を行います。

ただし、このカテゴリは、データフラッシュ・メモリ内蔵品の場合のみ表示されます。

図 2—130 [データフラッシュ] カテゴリ

データフラッシュ	
データフラッシュ・メモリ領域を使用する	はい
データフラッシュ・メモリ・サイズ[Kバイト]	32
チップ・セレクト	CS0

- [データフラッシュ・メモリ領域を使用する]

データフラッシュ・メモリを使用するか否かをドロップダウン・リストにより指定します。

データフラッシュ・メモリを使用する場合は [はい] を選択してください (デフォルトでは [いいえ] が指定されます)。

- [データフラッシュ・メモリ・サイズ [K バイト]]

このプロパティは、[データフラッシュ・メモリ領域を使用する] 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。

選択しているマイクロコントローラのデータフラッシュ・メモリ領域のサイズを表示します（単位：Kバイト）。

このプロパティ値を変更することはできません。

- [チップ・セレクト]

このプロパティは、**[データフラッシュ・メモリ領域を使用する]** プロパティにおいて、**[はい]** を指定した場合のみ表示されます。

データフラッシュ・メモリをマッピングする際のチップ・セレクトをドロップ・ダウンリストにより指定します。

なお、指定できるチップ・セレクトは、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、使用するマイクロコントローラに V850ES コア以外を選択している場合、デフォルトで表示しているチップ・セレクトを変更することはできません。

(c) [クロック]

このカテゴリでは、クロックに関する設定を行います。

図 2—131 [クロック] カテゴリ【EZ Emulator】

クロック	
メイン・クロック周波数 [MHz]	4.00
メイン・クロック逡倍	1
サブ・クロック周波数 [kHz]	32.765
モニタ・クロック	システム

- [メイン・クロック周波数 [MHz]]

メイン・クロック周波数（逡倍前）を指定します。

ドロップダウン・リストによる選択か、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。

ドロップダウン・リストには、次の周波数が表示されます（単位：MHz）。

2.50, 4.00（デフォルト）、5.00, 6.00, 8.00, 13.50, 18.00

- [メイン・クロック逡倍]

メイン・クロック周波数に対する逡倍数を指定します。

ドロップダウン・リストによる選択か、または直接入力により、1 ~ 99 の範囲の値を指定します。

ドロップダウン・リストには、次の逡倍数が表示されます。

1（デフォルト）、2, 3, 4, 5, 6, 7, 8, 9, 10

- [サブ・クロック周波数 [kHz]]

このプロパティは、選択しているマイクロコントローラがサブ・クロックをサポートしている場合のみ表示されます。

サブ・クロック周波数を指定します。

ドロップダウン・リストによる選択か、または直接入力により、0.001 ~ 999.999（単位：kHz）の範囲の数値を指定します。

ドロップダウン・リストには、現在選択しているマイクロコントローラが使用可能なサブ・クロック周波数が表示されます。

デフォルトでは、使用可能なサブ・クロック周波数の最小値が指定されます。

- [モニタ・クロック]

このプロパティは、選択しているマイクロコントローラがサブ・クロックをサポートしている場合のみ表示されます。

プログラム停止中に使用するクロックを指定します。

次のドロップダウン・リストにより指定します。

システム	メイン・クロックで動作します（デフォルト）。
ユーザ	プログラムで設定されているクロックで動作します。

(d) [ターゲット・ボードとの接続]

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

図 2—132 [ターゲット・ボードとの接続] カテゴリ【EZ Emulator】

☐ ターゲット・ボードとの接続	
通信方式	UARTA0

- [通信方式]

EZ Emulator がターゲット・ボード上のマイクロコントローラとシリアル通信を行う際に接続するポートをドロップダウン・リストにより指定します。

ドロップダウン・リスト内に表示されるポートの種類は、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、選択しているマイクロコントローラにより、シリアル通信の際に使用できるポートが限定される場合は、デフォルトで表示している通信方式を変更することはできません（デフォルトの指定は選択しているマイクロコントローラに依存します）。

注意 EZ Emulator が CubeSuite+ に接続している場合、このプロパティ値を変更することはできません。

(e) [フラッシュ]

このカテゴリでは、フラッシュ書き換えに関する設定を行います。

図 2—133 [フラッシュ] カテゴリ【EZ Emulator】

☐ フラッシュ	
セキュリティID	HEX FFFFFFFFFFFFFFFFFFFFFFFF
起動時にフラッシュROMを消去する	いいえ

- [セキュリティ ID]

このプロパティは、選択しているマイクロコントローラが、フラッシュ・メモリのROMセキュリティ機能（オンチップ・デバッグ・セキュリティ ID）をサポートしている場合のみ表示されます。

内蔵 ROM、または内蔵フラッシュ・メモリ上のコードを読み出す際のセキュリティ ID を指定します。

直接入力により、20 桁の 16 進数（10 バイト）で指定します（デフォルトでは

[FFFFFFFFFFFFFFFFFFFFFF] が指定されます）。

なお、オンチップ・デバッグ・セキュリティ ID についての詳細は、EZ Emulator のユーザーズ・マニュアルを参照してください。

注意 EZ Emulator が CubeSuite+ に接続している場合、このプロパティ値を変更することはできません。

備考 このプロパティの設定を変更すると、ビルド・ツールのセキュリティ ID を変更するか否かを確認するメッセージを表示します。

- [起動時にフラッシュ ROM を消去する]

デバッグ・ツールと接続する際に、フラッシュ ROM の消去を行うか否かをドロップダウン・リストにより指定します。

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）。

なお、このプロパティは、デバッグ・ツールと接続後、自動的に [いいえ] に設定されます。

注意 EZ Emulator が CubeSuite+ に接続している場合、このプロパティ値を変更することはできません。

(f) [プログラマブル I/O]

このカテゴリでは、プログラマブル I/O 領域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プログラマブル I/O 領域をサポートしている場合のみ指定可能です。

備考 デバッグを行う際に、一時的に値を変更する場合にのみ設定を変更してください。

プロジェクトとして共通の値を設定する場合は、使用するビルド・ツールのプロパティ パネルの [共通オプション] タブ上の [デバイス] カテゴリ内 [プログラマブル I/O 領域開始アドレス] プロパティで設定を行ってください。

図 2—134 [プログラマブル I/O] カテゴリ

プログラマブル I/O	
プログラマブル I/O 領域を使用する	はい
プログラマブル I/O 領域開始アドレス	HEX 0

- [プログラマブル I/O 領域を使用する]

プログラマブル I/O 領域を使用するか否かをドロップダウン・リストにより指定します。

プログラマブル I/O 領域を使用する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

- [プログラマブル I/O 領域開始アドレス]

このプロパティは、[プログラマブル I/O 領域を使用する] 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。

プログラマブル I/O 領域の開始アドレスを指定します。

直接入力により、プログラマブル I/O 領域内のアドレスを指定してください（デフォルトでは [0] が指定されます）。

なお、アドレスは、16 K バイトでアラインされます。

(2) [デバッグ・ツール設定] タブ

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

- (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) [入力信号のマスク]

(a) [メモリ]

このカテゴリでは、メモリに関する設定を行います。

図 2—135 [メモリ] カテゴリ [EZ Emulator]

☐ メモリ	
☐ メモリ・マッピング	[4]
田 [0]	内蔵ROM領域
田 [1]	ノン・マップ領域
田 [2]	内蔵RAM領域
田 [3]	I/Oレジスタ領域
メモリ書き込み時にベリファイを行う	はい

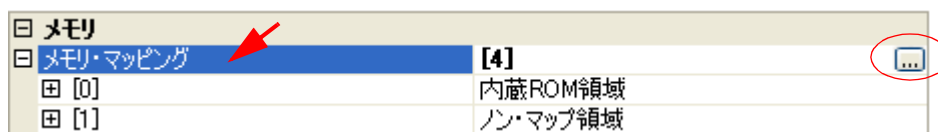
- [メモリ・マッピング]

現在のメモリ・マッピングの状況が、メモリ領域の種別ごとに詳細表示されます。

このパネル上でマッピング値を変更することはできません。メモリ・マッピングを追加する必要がある場合は、[メモリ・マッピング] プロパティを選択することで設定欄右端に表示される [...] ボタンのクリックによりオープンする **メモリ・マッピング ダイアログ** で行います。

設定方法についての詳細は、**メモリ・マッピング ダイアログ** の項を参照してください。

図 2—136 メモリ・マッピング ダイアログのオープン



注意 デバッグ・ツールと未接続の場合、ユーザにより追加されたメモリ・マッピング領域のみが表示対象となります。

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

- [メモリ書き込み時にベリファイを行う]
メモリ値の初期化を行う際に、ベリファイを行うか否かをドロップダウン・リストにより指定します。ベリファイを行う場合は「はい」を選択してください（デフォルト）。

(b) [実行中のメモリ・アクセス]

このカテゴリでは、プログラム実行中におけるメモリ・アクセス（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—137 [実行中のメモリ・アクセス] カテゴリ【EZ Emulator】

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

- [実行を一瞬停止してアクセスする]
プログラム実行中にはアクセスできないメモリ領域（ターゲット・メモリ領域 / I/O レジスタ領域 / CPU レジスタなど）に対して、実行を一瞬停止し、アクセスを許可するか否かをドロップダウン・リストにより指定します。
- [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**の表示内容を自動的に更新するか否かをドロップダウン・リストにより指定します。
表示内容の更新を行う場合は「はい」を選択してください（デフォルト）。
- [表示更新間隔 [ms]]
このプロパティは、**[実行中に表示更新を行う]** プロパティにおいて「はい」を指定した場合にのみ有効となります。

プログラム実行中に、[ウォッチパネル](#)/[メモリパネル](#)の表示内容を自動的に更新する間隔を 100 ms 単位で指定します。

直接入力により、100 ~ 65500 の範囲の整数（100 ms 未満の端数切り上げ）を指定してください（デフォルトでは [500] が指定されます）。

(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—138 [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ドロップダウン・リストにより指定します。

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（デフォルトでは [いいえ] が指定されます）。

(d) [ブレーク]

このカテゴリでは、ブレーク機能に関する設定を行います。

図 2—139 [ブレーク] カテゴリ【EZ Emulator】

ブレーク	
優先的に使用するブレークポイントの種類	ソフトウェア・ブレーク

- [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、選択しているマイクロコントローラが使用可能なブレークポイントの種類が 1 つのみの場合は表示されません。

[エディタパネル](#)/[逆アセンブルパネル](#)において、マウスのワンクリック操作でブレークポイントを設定する際に使用するブレークポイントの種別を次のドロップダウン・リストにより指定します。

なお、ブレークポイントについての詳細は、「[2.8.2 任意の場所で停止する（ブレークポイント）](#)」を参照してください。

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

(e) [入力信号のマスク]

このカテゴリでは、入力信号のマスクに関する設定を行います。

図 2—140 [入力信号のマスク] カテゴリ【EZ Emulator】

入力信号のマスク	
RESET 信号をマスクする	いいえ

次に示す各プロパティの設定において、該当する信号をマスクする場合は [はい] を、マスクしない場合は [いいえ] をドロップダウン・リストにより指定してください（デフォルトでは、すべてのプロパティに [いいえ] が指定されます）。

- [RESET 信号をマスクする]

(3) [ダウンロード・ファイル設定] タブ

[ダウンロード・ファイル設定] タブでは、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

[フック処理設定] タブでは、デバッグ・ツールにフック処理の設定を行います。

フック処理、および各カテゴリ内の設定についての詳細は、「[2.16 フック処理を設定する](#)」を参照してください。

2.3.7 【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—141 動作環境設定【シミュレータ【V850E1】【V850ES】】（プロパティパネルの例）



図 2—142 動作環境設定【シミュレータ【V850E2】】（プロパティパネルの例）



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【V850E2】
- (5) [フック処理設定] タブ

注意 1. 【V850E1】【V850ES】

命令シミュレータとして使用する場合、クロック・ジェネレータのシミュレーションは行われません。このため、CPU の動作クロックは、常に**プロパティ パネルの [接続用設定] タブ上の [クロック]** カテゴリ内 [メイン・クロック周波数 [MHz]] プロパティで設定した値となります（クロック・ジェネレータが持つ内蔵周辺 I/O レジスタを操作しても CPU の動作クロックは変化しません）。

2. 【V850E2】

命令シミュレータとして使用する場合、次の機能をシミュレーションします（これ以外の機能はシミュレーションしません）。

- CPU 命令
- 例外
- システム・レジスタ保護
- メモリ保護
- タイミング監視機能
- 浮動小数点演算機能

また、次の注意が必要です。

- 外部メモリ領域へアクセスはできません。
- 浮動小数点ユニット（FPU）のシミュレーションの結果は、実デバイスと誤差が生じます。シミュレータは、Visual C++ の浮動小数点ライブラリを用いて、80 ビットで計算した結果をレジスタに格納します。
- システム・エラー例外／メモリ・エラー例外はサポートしていません。
- キャッシュ・メモリのシミュレーションはサポートしていません。
- SYNCE, SYNCM, および SYNCP の 3 命令はサポートしていません。実行した場合は、NOP と同様の動作になります。
- CPU の動作クロックは 4MHz 固定となります。プロパティ パネルの [接続用設定] タブ上の [クロック] カテゴリ内 [メイン・クロック周波数 [MHz]] プロパティを変更しても CPU の動作クロックに反映されません。
- データフラッシュ領域にアクセスできません。アクセスした場合、エラー発生後ブレイクします。
- オプション・バイト格納レジスタ OPBT0 の値は常に 0 となります。
- EH_RESET レジスタの機能はサポートしていません。CPU リセットが発生した時のリセット・アドレスは 0x0 に固定しています。
- 各命令の実行クロック数は、命令実行直後に他の命令を実行する場合の実行クロック数となります。

備考 使用する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る場合、シミュレータ GUI を使用することができます。シミュレータ GUI についての詳細は、「[2.17 シミュレータ GUI の使用【シミュレータ】](#)」を参照してください。

(1) [接続用設定] タブ

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

- (a) [内蔵 ROM/RAM]
- (b) [クロック]
- (c) [コンフィギュレーション]
- (d) [プログラマブル I/O] 【V850E1】 【V850ES】

(a) [内蔵 ROM/RAM]

このカテゴリでは、内蔵 ROM/RAM に関する設定を行います。

デフォルトで、選択しているマイクロコントローラの内蔵 ROM/RAM サイズが設定されます（選択しているマイクロコントローラが V850E1 コアで、VSB Flash ROM/VSB RAM 搭載品の場合は、そのサイズを加算した値が設定されます）。

選択しているマイクロコントローラと同様のメモリ・マッピングでデバッグを行う場合は、このカテゴリ内の設定を変更する必要はありません。

図 2—143 [内蔵 ROM/RAM] カテゴリ 【シミュレータ 【V850E1】 【V850ES】】

内蔵ROM/RAM	
内蔵ROMサイズ[Kバイト]	256
内蔵RAMサイズ[バイト]	16384

図 2—144 [内蔵 ROM/RAM] カテゴリ 【シミュレータ 【V850E2】】

内蔵ROM/RAM	
内蔵ROMサイズ[Kバイト]	2048
内蔵RAMサイズ[バイト]	147456

- [内蔵 ROM サイズ[Kバイト]]

- 【V850E1】 【V850ES】

シミュレーションする内蔵 ROM サイズを指定します（単位：Kバイト）。

メモリ・マッピングを変更後にデバッグを行う場合は、ドロップダウン・リストにより指定してください。

ただし、選択しているマイクロコントローラが ROM レス品の場合は、[0] の値を変更することはできません。

- 【V850E2】

シミュレーションする内蔵 ROM サイズを表示します（単位：Kバイト）。

このプロパティ値を変更することはできません。

- [内蔵 RAM サイズ[バイト]]

- 【V850E1】 【V850ES】

シミュレーションする内蔵 RAM サイズを指定します（単位：バイト）。

メモリ・マッピングを変更後にデバッグを行う場合は、ドロップダウン・リストにより指定してください。

- 【V850E2】

シミュレーションする内蔵 RAM サイズを表示します（単位：バイト）。

このプロパティ値を変更することはできません。

(b) 【クロック】

このカテゴリでは、クロックに関する設定を行います。

図 2—145 【クロック】 カテゴリ【シミュレータ】

目 クロック	
メイン・クロック周波数 [MHz]	4.00
サブ・クロック周波数 [kHz]	32.768
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・リストによる選択か、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。

ドロップダウン・リストには、次の周波数が表示されます（単位：MHz）。

- 【V850E1】【V850ES】

1.00, 2.00, 3.00, 3.57, 4.00（デフォルト）, 4.19, 4.91, 5.00, 6.00, 8.00, 8.38, 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

- 【V850E2】

1.00, 2.00, 3.00, 3.57, 4.00, 4.19, 4.91, 5.00, 6.00, 7.20, 8.00（デフォルト）, 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

- 【サブ・クロック周波数 [kHz]】

このプロパティは、選択しているマイクロコントローラがサブ・クロックをサポートしている場合のみ表示されます。

サブ・クロック周波数を指定します。

ドロップダウン・リストによる選択か、または直接入力により、0.001 ~ 999.999（単位：kHz）の範囲の数値を指定します。

ドロップダウン・リストには、次の周波数が表示されます（単位：kHz）。

32.768（デフォルト）

- 【タイマ/トレース用クロック周波数の選択】

タイマ/トレース機能を使用する際のクロック周波数を指定します。

次のドロップダウン・リストより選択します。

CPU クロック周波数	CPU クロック周波数を使用します (デフォルト)。
周波数の指定	周波数を任意に指定します (下段のプロパティ項目が有効となります)。

- [タイマ/トレース用クロック周波数の単位]

このプロパティは、[タイマ/トレース用クロック周波数の選択] プロパティにおいて [周波数の指定] を指定した場合のみ指定可能です。

タイマ/トレース用のクロック周波数の単位を指定します。

次のドロップダウン・リストより選択します。

MHz	周波数の単位を MHz とします (デフォルト)。
KHz	周波数の単位を kHz とします。

- [タイマ/トレース用クロック周波数]

このプロパティは、[タイマ/トレース用クロック周波数の選択] プロパティでの指定により、次のように動作が異なります。

- [周波数の指定] を指定した場合

タイマ/トレース用クロックの周波数を指定します。

直接入力により、1 kHz ~ 999.999 MHz の範囲となるよう数値を指定してください (デフォルトでは [4.00] [V850E1] [V850ES] / [8.00] [V850E2] が指定されます)。

なお、周波数の単位は、[タイマ/トレース用クロック周波数の単位] プロパティでの指定に依存します。

- [CPU クロック周波数] を指定した場合

デバッグ・ツールと切断時は [---] を、デバッグ・ツールと接続時は [CPU クロック周波数] を表示します (変更不可)。

(c) [コンフィギュレーション]

このカテゴリでは、シミュレータのカスタマイズに関する設定を行います。

注意 シミュレータが CubeSuite+ に接続している場合、このカテゴリ内のプロパティ値を変更することはできません。

図 2—146 [コンフィギュレーション] カテゴリ

目 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	はい
シミュレータ・コンフィギュレーション・ファイル	

- [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]

シミュレータに対して、ユーザ・カスタマイズ (ユーザ・モデルの追加) を行うためのシミュレータ・コンフィギュレーション・ファイルを使用するか否かをドロップダウン・リストにより指定します。

使用する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

- [シミュレータ・コンフィギュレーション・ファイル]

このプロパティは、[\[シミュレータ・コンフィギュレーション・ファイルを使用する\]](#) 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。

使用するシミュレータ・コンフィギュレーション・ファイルを指定します。

ファイル名を直接入力するか、このプロパティを選択することで設定欄右端に表示される [...] ボタンのクリックによりオープンする[シミュレータ・コンフィギュレーション・ファイルを選択 ダイアログ](#) [【シミュレータ】](#)によりファイルを選択することで指定します。

(d) [プログラマブル I/O] [【V850E1】](#) [【V850ES】](#)

このカテゴリでは、プログラマブル I/O 領域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プログラマブル I/O 領域をサポートしている場合のみ指定可能です。

備考 デバッグを行う際に、一時的に値を変更する場合にのみ設定を変更してください。

プロジェクトとして共通の値を設定する場合は、使用するビルド・ツールのプロパティ パネルの [共通オプション] タブ上の [デバイス] カテゴリ内 [\[プログラマブル I/O 領域開始アドレス\]](#) プロパティで設定を行ってください。

図 2—147 [プログラマブル I/O] カテゴリ

☐ プログラマブル I/O	
プログラマブル I/O 領域を使用する	はい
プログラマブル I/O 領域開始アドレス	HEX 0

- [プログラマブル I/O 領域を使用する]

プログラマブル I/O 領域を使用するか否かをドロップダウン・リストにより指定します。

プログラマブル I/O 領域を使用する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

- [プログラマブル I/O 領域開始アドレス]

このプロパティは、[\[プログラマブル I/O 領域を使用する\]](#) 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。

プログラマブル I/O 領域の開始アドレスを指定します。

または直接入力により、プログラマブル I/O 領域内のアドレスを指定してください（デフォルトでは [0] が指定されます）。

なお、アドレスは、16 K バイトでアラインされます。

(2) [デバッグ・ツール設定] タブ

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

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

(a) [メモリ]

このカテゴリでは、メモリに関する設定を行います。

図 2—148 [メモリ] カテゴリ【シミュレータ】

☐ メモリ	
☐ メモリ・マッピング	[4]
田 [0]	内蔵ROM領域
田 [1]	ノン・マップ領域
田 [2]	内蔵RAM領域
田 [3]	I/Oレジスタ領域


- [メモリ・マッピング]

現在のメモリ・マッピングの状況が、メモリ領域の種別ごとに詳細表示されます。

このパネル上でマッピング値を変更することはできません。メモリ・マッピングを追加する必要がある場合は、[メモリ・マッピング] プロパティを選択することで設定欄右端に表示される [...] ボタンのクリックによりオープンする**メモリ・マッピング ダイアログ**で行います。

設定方法についての詳細は、**メモリ・マッピング ダイアログ**の項を参照してください。

図 2—149 メモリ・マッピング ダイアログのオープン

☐ メモリ	
☐ メモリ・マッピング	[4] 
田 [0]	内蔵ROM領域
田 [1]	ノン・マップ領域

注意 デバッグ・ツールと未接続の場合、ユーザにより追加されたメモリ・マッピング領域のみが表示対象となります。

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

(b) [実行中のメモリ・アクセス]

このカテゴリでは、プログラム実行中におけるメモリ・アクセス（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—150 [実行中のメモリ・アクセス] カテゴリ【シミュレータ】

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

- [実行中に表示更新を行う]

プログラム実行中に、**ウォッチパネル／メモリパネル**の表示内容を自動的に更新するか否かをドロップダウン・リストにより指定します。

表示内容の更新を行う場合は「はい」を選択してください（デフォルト）。

- [表示更新間隔[ms]]

このプロパティは、「**実行中に表示更新を行う**」プロパティにおいて「はい」を指定した場合にのみ有効となります。

プログラム実行中に、**ウォッチパネル／メモリパネル**の表示内容を自動的に更新する間隔を 100 ms 単位で指定します。

直接入力により、100 ~ 65500 の整数（100 ms 未満の端数切り上げ）を指定してください（デフォルトでは「500」が指定されます）。

(c) [ブレーク]

このカテゴリでは、ブレーク機能に関する設定を行います。

図 2—151 [ブレーク] カテゴリ【シミュレータ】

ブレーク	
停止時にブレーク位置の命令を実行	いいえ

- [停止時にブレーク位置の命令を実行]

ブレークポイントによるプログラム実行停止のタイミングを、ブレークポイントが設定されている位置の命令実行後とするか、または命令実行前とするかを指定します。

ドロップダウン・リストにより、命令実行後に停止する場合は「はい」を選択してください（デフォルトでは「いいえ」が指定されます）。

なお、ブレークポイントについての詳細は、「2.8.2 任意の場所で停止する（ブレークポイント）」を参照してください。

(d) [トレース]

このカテゴリでは、トレース機能に関する設定を行います。

トレース機能、およびこのカテゴリ内の設定についての詳細は、「[2.11 実行履歴の収集【IECUBE】](#)」
「[IECUBE2【シミュレータ】](#)」を参照してください。

(e) [タイマ]

このカテゴリでは、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、「[2.12 実行時間の計測](#)」を参照してください。

図 2—152 [タイマ] カテゴリ【シミュレータ】

☐ 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ドロップダウン・リストにより指定します。

タイマ機能を使用する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

(f) [カバレッジ]

このカテゴリでは、カバレッジ機能に関する設定を行います。

カバレッジ機能、およびこのカテゴリ内の設定についての詳細は、「[2.13 カバレッジの測定](#)」
「[IECUBE【IECUBE2】【シミュレータ】](#)」を参照してください。

(g) [シミュレータ GUI]

このカテゴリでは、シミュレータ GUIに関する設定を行います。

シミュレータ GUI の機能、およびこのカテゴリ内の設定についての詳細は、「[2.17 シミュレータ GUI の使用【シミュレータ】](#)」を参照してください。

(3) [ダウンロード・ファイル設定] タブ

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

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

(4) [フラッシュ・オプション設定] タブ【V850E2】

[\[フラッシュ・オプション設定\] タブ【V850E2】](#)では、V850E2 コアの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【V850E2】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【V850E2】を参照してください。

図 2—153 フラッシュ・オプションの設定 ダイアログのオープン

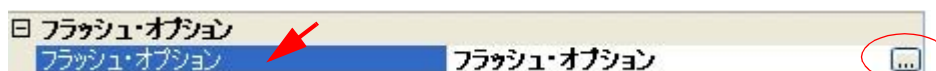


図 2—154 フラッシュ・オプションの設定（フラッシュ・オプションの設定 ダイアログ【シミュレータ】）



(5) [フック処理設定] タブ

[フック処理設定] タブでは、デバッグ・ツールにフック処理の設定を行います。

フック処理、および各カテゴリ内の設定についての詳細は、「2.16 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

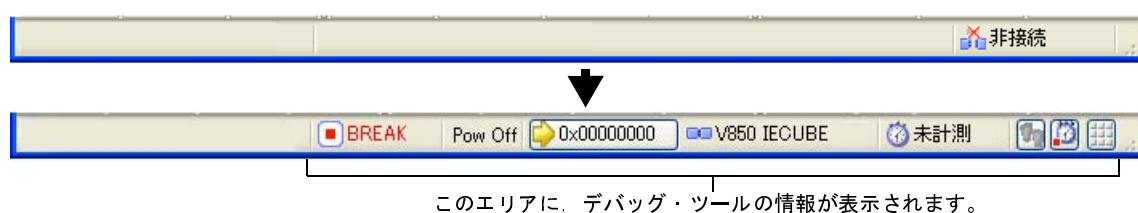
この節では、CubeSuite+ とデバッグ・ツールの接続方法、および切断方法について説明します。

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・ウィ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・ウィンドウ**を参照してください。

図 2—155 デバッグ・ツールとの接続に成功したステータスバー




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 1. 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. 【シミュレータ】

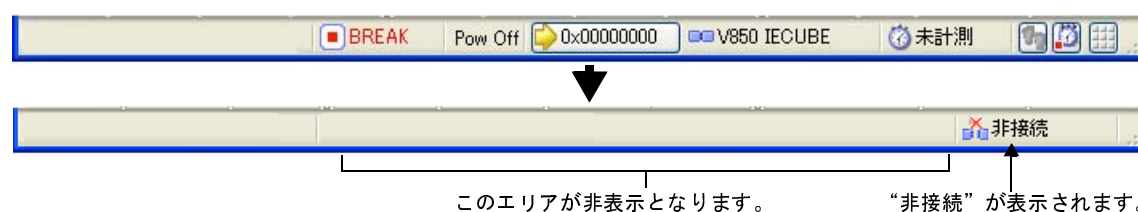
選択しているマイクロコントローラのシミュレータが周辺機能シミュレーションをサポートしている場合、デバッグ・ツールと接続すると、**シミュレータ GUI ウィンドウ**がオープンします（デフォルト）。

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  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・ツールとの通信を切断すると、**メイン・ウィンドウのステータスバー**が、次のように変化します。

図 2—156 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。

2.4.3 デバッグ・ツールをホット・プラグイン接続する【E1/E20(LPD)/(JTAG)【V850E2】】

ホット・プラグイン接続とは、プログラム実行中のターゲット・ボードに対してデバッグ・ツールを接続し、実行中のプログラムをデバッグすることができる接続方式です。

ホット・プラグイン接続の手順は、次のとおりです。

注意 1. 【E1(JTAG)【V850E2】】

ホット・プラグイン接続を行うには、ターゲット・ボードと E1 間に、E1 用ホットプラグアダプタが必要となります。

2. ホット・プラグイン接続を行った場合、フラッシュ・セルフ・ライブラリを使用した内蔵フラッシュ ROM の自動書き換えは行いません。

(1) プログラムを実行する

エミュレータと接続していない状態で、あらかじめターゲット・ボード上のマイクロコントローラにダウンロードしたプログラムを実行します。

(2) デバッグ・ツールを選択する

アクティブ・プロジェクトでホット・プラグイン接続可能なデバッグ・ツール (E1/E20(LPD)/(JTAG)) を選択します。

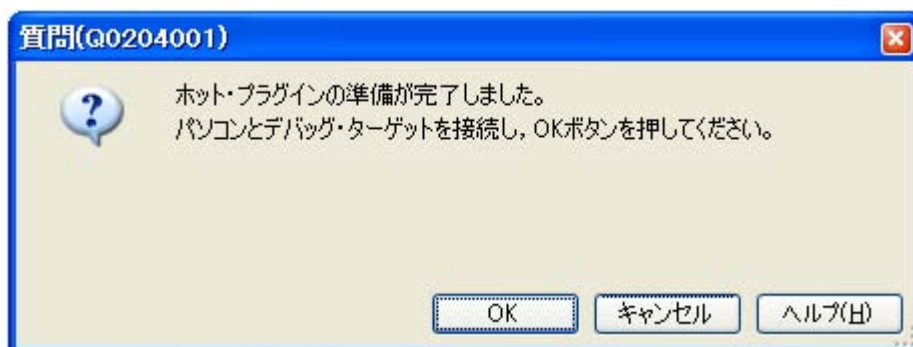
(3) CubeSuite+ にデバッグ・ツールをホット・プラグイン接続する

[デバッグ] メニュー → [ホット・プラグイン] を選択することにより、ホット・プラグイン接続の準備を開始します。

(4) ターゲット・ボードを接続する

ホット・プラグイン接続の準備が完了すると、ターゲット・ボードの接続を促す次のメッセージが表示されます。ターゲット・ボードとエミュレータを接続したのち、[OK] ボタンをクリックすると、アクティブ・プロジェクトで選択しているデバッグ・ツールとの通信を開始します。

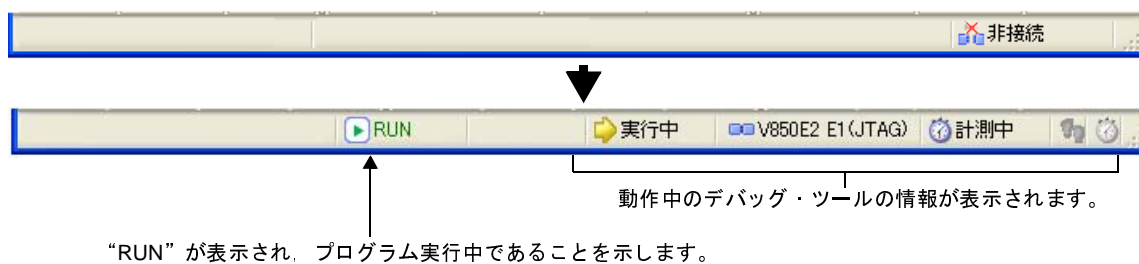
図 2—157 ホット・プラグイン接続準備完了メッセージ



(5) ホット・プラグイン接続の完了

デバッグ・ツールとの接続に成功すると、[メイン・ウィンドウ](#)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・ウィンドウ](#)を参照してください。

図 2—158 デバッグ・ツールとのホット・プラグイン接続に成功したステータスバー



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+ へ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

図 2—159 【ダウンロード】カテゴリ

☐ ダウンロード	
☐ ダウンロードするファイル	【1】
☐ [0]	Debug_Build#v850esfx3.out
ファイル	Debug_Build#v850esfx3.out
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・モードの選択	スピード優先
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
予約領域の上書きをチェックする	はい

注意 プロジェクトで使用するデバッグ・ツールにより、表示されるプロパティ項目は異なります。

(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をリセットするかどうかをドロップダウン・リストにより指定します。
CPUをリセットする場合は「はい」を選択してください（デフォルト）。

(c) [ダウンロード・モードの選択]（【シミュレータ】以外）

フラッシュROMへダウンロードを行う際のダウンロード・モードを指定します。
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。

スピード優先	先頭データと最終データ間の空き領域はFFHで補完されます（先頭データ以前と最終データ以降の空き領域はダウンロード前の値が保持されます）。 書き込みデータの量を減らすため、ダウンロード速度は速くなります（デフォルト）。
データ優先	空き領域はダウンロード前の値が保持されます。 空き領域のデータを一度読み込むため、ダウンロード速度は非常に遅くなります。

(d) [ダウンロード前にフラッシュROMを消去する]

このプロパティは、**[ダウンロード・モードの選択]（【シミュレータ】以外）** プロパティで**[スピード優先]**を指定した場合（デフォルト）のみ有効となります。

ダウンロード実行前にフラッシュROMを消去するかどうかをドロップダウン・リストにより指定します。

フラッシュROMを消去する場合は「はい」を選択してください（デフォルトでは「いいえ」が指定されます）。

(e) [イベント設定位置の自動変更方法]

デバッグ作業を進めることにより、変更を加えたプログラムを再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。

(f) [予約領域の上書きをチェックする]【MINICUBE2】【E1/E20(Serial)】【EZ Emulator】

ダウンロードの際に、エミュレータが使用する予約領域に対して上書きした場合に、メッセージを出力するかどうかをドロップダウン・リストにより指定します。

メッセージを出力する場合は「はい」を選択してください（デフォルト）。

(2) [デバッグ情報] カテゴリの設定

図 2—160 [デバッグ情報] カテゴリ

☐ 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
スタートアップ開始シンボル	__start
スタートアップ終了シンボル	__startend

(a) [CPU リセット後に指定シンボル位置まで実行する]

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

プログラムを指定シンボル位置まで実行する場合は [はい] を選択してください（デフォルト）。

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

(b) [指定シンボル]

このプロパティは、[CPUリセット後に指定シンボル位置まで実行する] プロパティにおいて [はい] を指定した場合のみ表示されます。

CPUリセット後にプログラムを実行して停止する位置を指定します。

直接入力により、0 ~ “アドレス空間の終アドレス” 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。

アセンブラ・ソースの場合： メイン関数に相当する先頭ラベル

Cソースの場合： メイン関数名の先頭に付与したシンボル

(c) [スタートアップ開始シンボル]

スタートアップ・ルーチンのテキスト領域（コード領域）の開始シンボルを指定します。

直接入力により、0 ~ “アドレス空間の終アドレス” 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 [__start] が指定されます）。

なお、アセンブラ・ソースの場合は、ここでの設定は不要となります。

(d) [スタートアップ終了シンボル]

スタートアップ・ルーチンのテキスト領域（コード領域）の終了シンボルを指定します。

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 [__startend] が指定されます）。

なお、アセンブラ・ソースの場合は、ここでの設定は不要となります。

注意 1. ダウンロード直後に自動的にソース・テキストを表示するためには、スタートアップ・シンボルが正しく指定されている必要があります。

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

(3) ダウンロードの実行

デバッグ・ツールバーの  ボタンをクリックします。

なお、デバッグ・ツールと切断時にこの操作が行われた場合は、自動的にデバッグ・ツールと接続したのち、ダウンロードを実行します。

備考 デバッグ作業を進めることにより、変更を加えたプログラムを再度ダウンロードする場合は、[メイン・ウィンドウ](#)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（[2.16 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，次のファイル形式をダウンロードすることができます。

表 2—3 ダウンロード可能なファイル形式

ファイルの種類	ファイル形式	拡張子
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	.lmf 【CX】 .out 【CA850】 .拡張子 【GHS】 注1
ヘキサ・ファイル	インテル・ヘキサ・フォーマット（標準）	.hex
	インテル・ヘキサ・フォーマット（拡張）	
	インテル・ヘキサ・フォーマット（拡張リニア）	
	モトローラ S タイプ・ヘキサ・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)	
	拡張テクトロニクス・ヘキサ・フォーマット	
ヘキサ・ファイル [ID タグ付] 注2 (【シミュレータ】以外)	ID タグ付きのインテル・ヘキサ・フォーマット（拡張）	.hex
	ID タグ付きのインテル・ヘキサ・フォーマット（拡張リニア）	
	ID タグ付きのモトローラ S タイプ・ヘキサ・フォーマット - (S0, S2, S8-24 ビット) - (S0, S3, S7-32 ビット)	
	ID タグ付きの拡張テクトロニクス・ヘキサ・フォーマット	
ヘキサ・ファイル (フラッシュプログラム形式) 注2 (【シミュレータ】以外)	フラッシュ・プログラマ形式のインテル・ヘキサ・フォーマット（拡張）	.hex
	フラッシュ・プログラマ形式のインテル・ヘキサ・フォーマット（拡張リニア）	
	フラッシュ・プログラマ形式のモトローラ S タイプ・ヘキサ・フォーマット - (S0, S2, S8-24 ビット) - (S0, S3, S7-32 ビット)	
	フラッシュ・プログラマ形式のモトローラ S タイプ・ヘキサ・フォーマット - (S0, S2, S8-24 ビット) - (S0, S3, S7-32 ビット)	
バイナリ・データ・ファイル	バイナリ・データ・フォーマット	.bin

注 1. GHS コンパイラ（米国 Green Hills Software, Inc. 製）使用上の注意点

なお，GHS コンパイラを使用する場合，デバッグ専用プロジェクトを作成する必要があります。

- 対応バージョン

- MULTI (Ver.5.x)

- 対応オプション

- デバッグ・オプション：-G -dual_debug（推奨）

- 最適化オプション：-Ospace, -O, -Ospeed, -Onone

- その他 : -prepare_dispose, -call
- C++ 言語, および GNU C 拡張仕様はサポートしていません。
- デバッグ上の注意点
 - アセンブラ・ソース・レベルのデバッグはできません。
 - #include 内で定義された関数にブレークポイントの設定はできません。
 - デバッグ情報に変数の生存期間情報がないため, 変数のウォッチ式の値が不正となる場合があります (変数が最適化されて削除されたり, 値が一時的にレジスタに割り当てられた場合など)。
 - 浮動小数点数の即値の扱いにより, コンパイラとデバッガで値丸めなどの誤差が生じます。
 - コール・スタック情報を取得できないため, **コール・スタック パネル**ではカレント関数のみの表示となります。
 - #include 文, setjmp 関数, および longjmp 関数のステップ実行はできません。
 - 可変長引数の関数からのリターン・アウト実行は失敗する場合があります。
 - 【IECUBE】
 - フラッシュ・ライブラリを ROM 化した場合, 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を使用することはできません。
- 2. 選択しているマイクロコントローラがデータフラッシュ・メモリ内蔵品の場合のみ有効となります。

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**【...】**ボタンをクリックするとオープンします。

図 2—161 ダウンロード・ファイル ダイアログのオープン

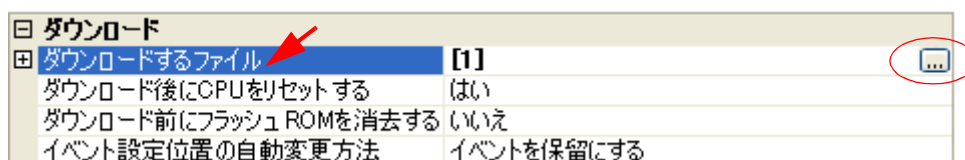
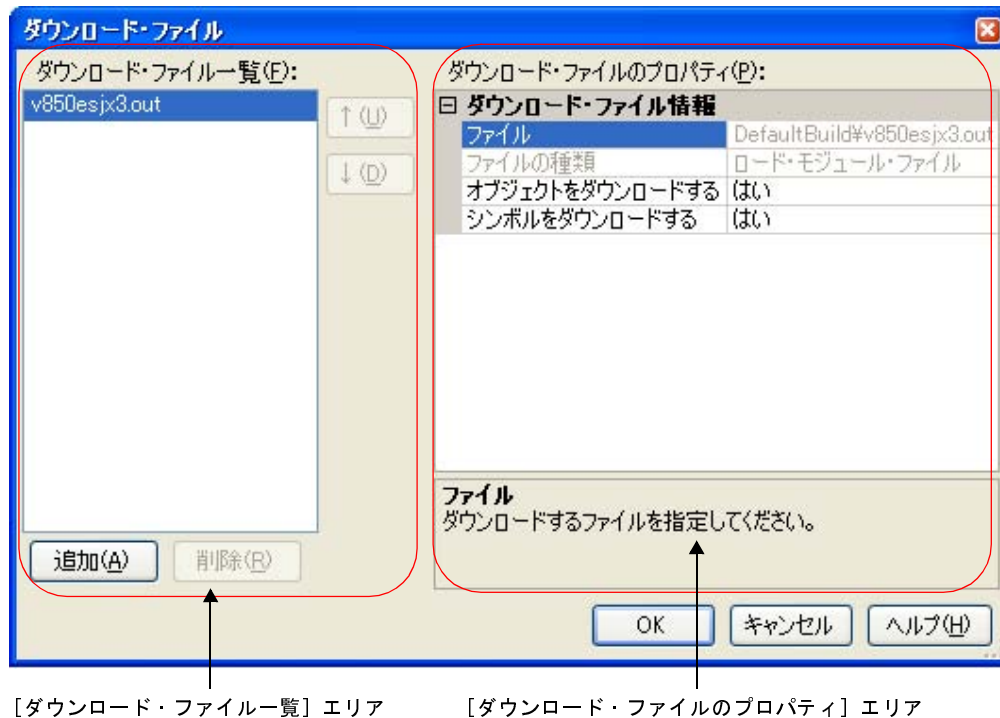


図 2—162 応用的なダウンロード方法（ダウンロード・ファイル ダイアログ）



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。

なお、各エリアの見方、および機能についての詳細は、ダウンロード・ファイル ダイアログの項を参照してください。

- (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/*.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) ヘキサ／バイナリ・データ・フォーマットのファイルでソース・レベル・デバッグを行う
- (5) 外部フラッシュ・メモリへダウンロードする【IECUBE】【IECUBE2】【MINICUBE】【E1/E20(LPD)/JTAG】

(1) ロード・モジュール・ファイルのダウンロード条件を変更する

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

(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・ファイルの現在のダウンロード条件が表示されます。

表示される各項目において、設定の変更を行います。

オブジェクトをダウンロードする	指定したファイルからオブジェクト情報をダウンロードするかを指定します。		
	デフォルト	はい	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	オブジェクト情報をダウンロードします。
		いいえ	オブジェクト情報をダウンロードしません。
シンボルをダウンロードする	指定したファイルからシンボル情報をダウンロードするかを指定します ^注 。		
	デフォルト	はい	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	シンボル情報をダウンロードします。
		いいえ	シンボル情報をダウンロードしません。

注 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。

(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/*.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（ヘキサ・ファイル (*.hex) / バイナリ・データ・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・ファイル一覧] 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（ヘキサ・ファイル（*.hex）／バイナリ・データ・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
	変更方法	キーボードからの直接入力、またはこのプロパティを選択すると右端に表示される [...] ボタンのクリックによりオープンする ダウンロードするファイルを選択 ダイアログ による指定
	指定可能値	「 表 2—3 ダウンロード可能なファイル形式 」参照
ファイルの種類	追加するダウンロード・ファイルのファイル形式を指定します。 ここでは、[ヘキサ・ファイル]、または[バイナリ・データ・ファイル]を選択します。	
	デフォルト	ロード・モジュール・ファイル
	変更方法	ドロップダウン・リストによる選択
	指定可能値	次のいずれか - ロード・モジュール・ファイル - ヘキサ・ファイル - ヘキサ・ファイル [ID タグ付] ^注 - バイナリ・データ・ファイル
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]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
	指定可能値	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
	指定可能値	0x0 ~ 0xFFFFFFFF の 16 進数

注 【シミュレータ】以外

選択しているマイクロコントローラがデータフラッシュ・メモリ内蔵品で、データフラッシュ・メモリ機能を有効にしている場合（[プロパティ パネル](#)の[\[接続用設定\]](#)タブ上の[\[データフラッシュ\]](#)カテゴリ内[\[データフラッシュ・メモリ領域を使用する\]](#)プロパティを[はい]に設定）のみ表示されます。

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どうかの指定は、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・ファイル一覧] 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・ファイル一覧] エリア内の空欄の項目に、ここで指定したファイル名が反映されます。

ファイル	追加するロード・モジュール・ファイルを指定します（最大指定文字数：259 文字）。	
	デフォルト	空欄
	変更方法	キーボードからの直接入力。またはこのプロパティを選択すると右端に表示される [...] ボタンのクリックによりオープンする ダウンロードするファイルを選択 ダイアログ による指定
	指定可能値	「表 2—3 ダウンロード可能なファイル形式」参照
ファイルの種類	追加するダウンロード・ファイルのファイル形式を指定します。 ここでは、[ロード・モジュール・ファイル] を選択します（デフォルト）。	
オブジェクトをダウンロードする	指定したファイルからオブジェクト情報をダウンロードするか否かを指定します。	
	デフォルト	はい
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい
	いいえ	オブジェクト情報をダウンロードしません。
シンボルをダウンロードする	指定したファイルからシンボル情報をダウンロードするか否かを指定します ^注 。	
	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
	いいえ	シンボル情報をダウンロードしません。

注 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。

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

ヘキサ・ファイル (*.hex)、またはバイナリ・データ・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・ファイル一覧] 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	ダウンロードするヘキサ・ファイル (*.hex)、またはバイナリ・データ・ファイル (*.bin) を作成する元となったロード・モジュール・ファイルを指定します。 キーボードからの直接入力、または [...] ボタンのクリックによりオープンする ダウンロードするファイルを選択 ダイアログ により指定してください。
ファイルの種類	[ロード・モジュール・ファイル] を指定します（デフォルト）。
オブジェクトをダウンロードする	[いいえ] を指定します。
シンボルをダウンロードする	[はい] を指定します（デフォルト）。

(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）。

(5) 外部フラッシュ・メモリへダウンロードする【IECUBE】【IECUBE2】【MINICUBE】【E1/E20(LPD)/(JTAG)】

ダウンロード・ファイルダイアログで指定したダウンロード・ファイルを、使用するマイクロコントローラの外部バスに接続されたフラッシュ・メモリにダウンロードすることができます。

この場合は、次の手順の操作を行ってください。

(a) 外部フラッシュ・メモリ情報の設定

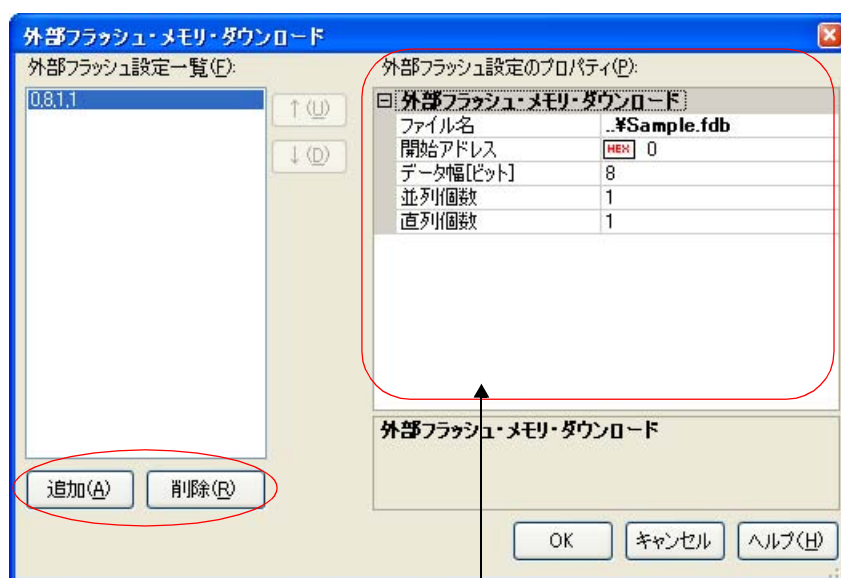
外部フラッシュ・メモリにダウンロードする際の外部フラッシュ・メモリ情報ファイル (*.fdb) の選択、およびダウンロード条件（外部フラッシュ・メモリのための設定）の設定を行います。

これらの設定は、プロパティパネルの【デバッグ・ツール設定】タブタブ上の【外部フラッシュ・メモリ・ダウンロード】カテゴリ内【外部フラッシュ・メモリ・ダウンロード設定】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する外部フラッシュ・メモリ・ダウンロードダイアログにおいて行います。

図 2—163 【外部フラッシュ・メモリ・ダウンロード】カテゴリ



図 2—164 外部フラッシュ・メモリへのダウンロード（外部フラッシュ・メモリ・ダウンロードダイアログ）



【外部フラッシュ設定のプロパティ】エリア

上記外部フラッシュ・メモリ・ダウンロードダイアログにおいて、使用する外部フラッシュ・メモリ情報ファイルを新規に登録するために、[追加] ボタンをクリックしたのち、[外部フラッシュ設定のプロパティ] エリア内の次のプロパティを設定します。

- 【ファイル名】

外部フラッシュ・メモリ情報ファイル (*.fdb) を指定します。

ファイル名を直接入力するか（相対パスによる指定の場合はプロジェクト・フォルダ基点）、このプロパティを選択することで設定欄右端に表示される [...] ボタンのクリックによりオープンする外部フラッシュ・メモリ情報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で指定します。
 なお、外部フラッシュ・メモリ情報ファイルは、弊社 Web サイト（バージョンアップ・サービス）よりダウンロードしてください。

- [開始アドレス]

外部フラッシュ・メモリの開始アドレスを指定します。

0x0 ~ 0xFFFFFFFF の範囲のアドレスを直接入力で指定してください（デフォルトでは [0] が指定されます）。

- [データ幅]

外部フラッシュ・メモリのデータ幅（アクセス・サイズ）をドロップダウン・リストより指定します（「表 2—4 パラメータ設定値一覧（外部フラッシュ・メモリ）」参照）。

ドロップダウン・リストには、次のデータ幅が表示されます（単位：ビット）。

8（デフォルト）、16、32

- [並列個数]

外部フラッシュ・メモリの並列個数を指定します（「表 2—4 パラメータ設定値一覧（外部フラッシュ・メモリ）」参照）。

ドロップダウン・リストによる選択か、または 1 ~ 99 の範囲の整数値を直接入力で指定します。

ドロップダウン・リストには、次の数値が表示されます。

1（デフォルト）、2、4

- [直列個数]

外部フラッシュ・メモリの直列個数を指定します（「表 2—4 パラメータ設定値一覧（外部フラッシュ・メモリ）」参照）。

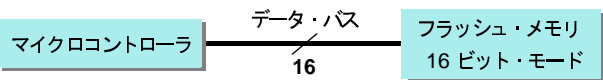
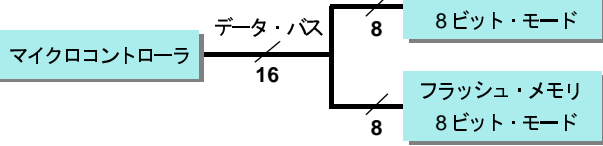
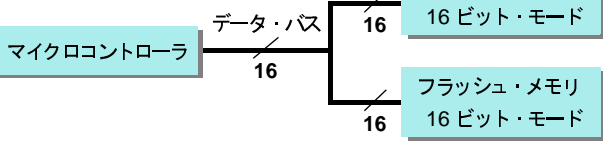
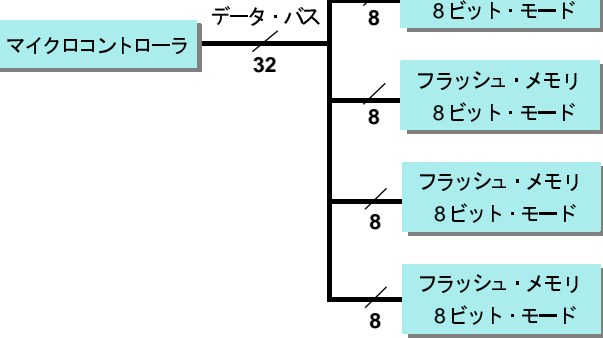
ドロップダウン・リストによる選択か、または 1 ~ 99 の範囲の整数値を直接入力で指定します。

ドロップダウン・リストには、次の数値が表示されます。

1（デフォルト）、2

表 2—4 パラメータ設定値一覧（外部フラッシュ・メモリ）

フラッシュ・メモリとの接続	データ幅	並列個数	直列個数
<p>データ・バス 8</p> <p>マイクロコントローラ</p> <p>フラッシュ・メモリ 8ビット・モード</p>	8	1	1
<p>データ・バス 8</p> <p>マイクロコントローラ</p> <p>フラッシュ・メモリ 8ビット・モード</p> <p>フラッシュ・メモリ 8ビット・モード</p>	8	1	2

フラッシュ・メモリとの接続		データ幅	並列個数	直列個数
	データ・バス 16	16	1	1
	データ・バス 16	16	2	1
	データ・バス 16	16	1	2
	データ・バス 32	32	4	1

(b) 外部バスの設定

マイクロコントローラからフラッシュ・メモリへのアクセスを可能にするため、周辺 I/O レジスタを外部バスの設定に変更します。

周辺 I/O レジスタの設定変更は、[IOR パネル](#)より行うこともできますが、各レジスタを個々に変更する必要が生じるため、フック処理機能を使用することにより、ダウンロード直前に一括処理を指定することをお勧めします（[「2.16 フック処理を設定する」](#)参照）。

(c) ダウンロードの実行

ダウンロードを実行します（[「\(3\) ダウンロードの実行」](#)参照）。

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—165 メモリ内容のアップロード（データ保存 ダイアログ）



(1) [ファイル名] の指定

保存するファイル名を指定します。

テキスト・ボックスに直接入力するか（最大指定文字数：259 文字）、またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。

また、[...] ボタンをクリックすることでオープンするデータ保存ファイルを選択 ダイアログにより、ファイルを選択することもできます。

(2) [ファイルの種類] の指定

保存するファイルの形式を次のドロップダウン・リストにより選択します。

選択できるファイルの形式は次のとおりです。

表 2—5 アップロード可能なファイル形式

[ファイルの種類] のリスト表示
インテル・ヘキサ・フォーマット (*.hex) ^注
モトローラ・ヘキサ・フォーマット (S0, S3, S7 - 32 ビット) (*.hex)
拡張テクトロニクス・ヘキサ・フォーマット (*.hex)
インテル・ヘキサ・フォーマット [ID タグ付] (*.hex)
モトローラ・ヘキサ・フォーマット (S0, S3, S7 - 32 ビット) [ID タグ付] (*.hex)
拡張テクトロニクス・ヘキサ・フォーマット [ID タグ付] (*.hex)
インテル・ヘキサ・フォーマット (フラッシュプログラマ形式) (*.hex)

[ファイルの種類] のリスト表示
モトローラ・ヘキサ・フォーマット (S0, S3, S7 - 32 ビット) (フラッシュ プログラム形式) (*.hex)
バイナリ・データ (*.bin)

注 インテル・ヘキサ・フォーマットでは、**[保存範囲 アドレス / シンボル]** の指定により、保存範囲が 1M 空間を越える場合は、“拡張リニア (32 ビット・アドレス)” で保存され、1M 空間以内の場合は、“拡張 (20 ビット・アドレス)” で保存されます。

備考 【EZ Emulator】【シミュレータ】以外

“[ID タグ付]”，または“(フラッシュプログラム形式)”を含む項目は、選択しているマイクロコントローラがデータフラッシュ・メモリ内蔵品の場合のみ表示されます。

“[ID タグ付]”を選択することにより、ID タグの情報も保存することができます。

また、“(フラッシュプログラム形式)”を選択すると、フラッシュ・プログラム形式で保存します。

(3) [保存範囲 アドレス / シンボル] の指定

ファイルに保存する範囲を“開始アドレス”と“終了アドレス”で指定します。

それぞれのテキスト・ボックスに 16 進数の数値 / アドレス式を直接入力するか、またはドロップダウン・リストより入力履歴項目を選択します (最大履歴数 : 10 個)。

備考 このテキスト・ボックスで [Ctrl] + [Space] キーを押下することにより、現在のcaret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（「[2.7.3 プログラムをステップ実行する](#)」参照）、およびコード・カバレッジ測定結果の表示（「[2.13.2 カバレッジ測定結果を表示する](#)」参照）を行います。

- 逆アセンブル パネル

ダウンロードしたプログラム（メモリ内容）の逆アセンブル結果の表示／編集（ライン・アセンブル）を行うほか、命令レベル・デバッグ（「[2.7.3 プログラムをステップ実行する](#)」参照）、およびコード・カバレッジ測定結果の表示（「[2.13.2 カバレッジ測定結果を表示する](#)」参照）を行います。

なお、このパネルでは、逆アセンブル結果の表示とともに、対応するソース・テキストも表示することができます（デフォルト）。

備考 通常、ソース・レベル・デバッグを行うためには、デバッグ情報を持つロード・モジュール・ファイルをダウンロードする必要がありますが、ヘキサ・ファイル (*.hex)、またはバイナリ・データ・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（「[\(4\) ヘキサ／バイナリ・データ・フォーマットのファイルでソース・レベル・デバッグを行う](#)」参照）。

2.6.1 ソース・ファイルを表示する

ソース・ファイルの表示は、次の[エディタ パネル](#)で行います。

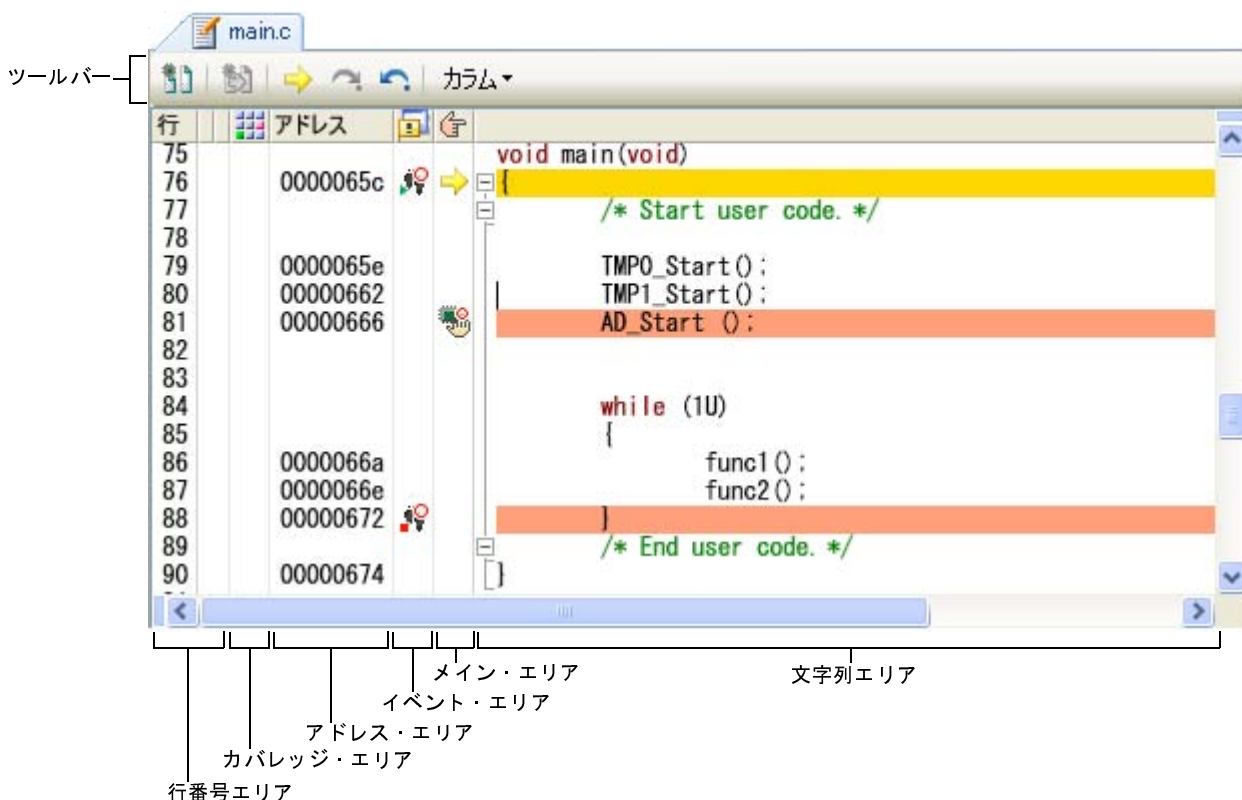
エディタ パネルは、ロード・モジュール・ファイルが正常にダウンロードされると、指定された位置（「[2.5.1 ダウンロードを実行する](#)」参照）のソース・テキストを表示した状態で自動的にオープンします。

手でエディタ 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 1.** [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。
- 2.** このパネルの表示は、[Ctrl] キー+マウス・ホイールにより、拡大／縮小することができます（「[\(m\) 表示の拡大／縮小](#)」参照）。

図 2—166 ソース・ファイルの表示（エディタ 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) 関数へジャンプする
- (8) タグ・ジャンプする

(1) 表示モードを変更する


デバッグ・ツールと接続中で、ダウンロードしたソース・ファイルを表示している場合、ツールバーの  ボタン（トグル）のクリックにより、各ソース・コードに対応した、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（混合表示モード）。

図 2-167 エディタ パネル (通常表示モード)

75			<code>void main(void)</code>
76	0000065c		<code>/* Start user code.</code>
77			
78			
79	0000065e		<code>TMPO_Start();</code>
80	00000662		<code>TMP1_Start();</code>
81	00000666		<code>AD_Start ();</code>
82			

図 2-168 エディタ パネル (混合表示モード)

75			<code>void main(void)</code>
76	0000065c	d50d	<code>br _main+0x1a</code>
77			<code>/* Start user code.</code>
78			
79	0000065e	80ffd201	<code>jarl _TMPO_Start, lp</code>
80	00000662	80ffa202	<code>TMP1_Start();</code>
81	00000666	80ffaa00	<code>_TMP1_Start, lp</code>
			<code>AD_Start ();</code>
			<code>jarl _AD_Start, lp</code>

命令コード

ラベル

逆アセンブル・テキスト

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・ファイルを表示している場合のみ有効となる機能です。

2. 混合表示モードでは、ソース・ファイルの編集、および保存はできません。

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...] / [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。

(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、この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。

カラム	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・ファイルがロードモジュールより新しいことを示す表示	行番号エリアにおいて、ダウンロードされたロード・モジュールの更新状態を示すマークを表示します。 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。
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。 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—169 通常の動作

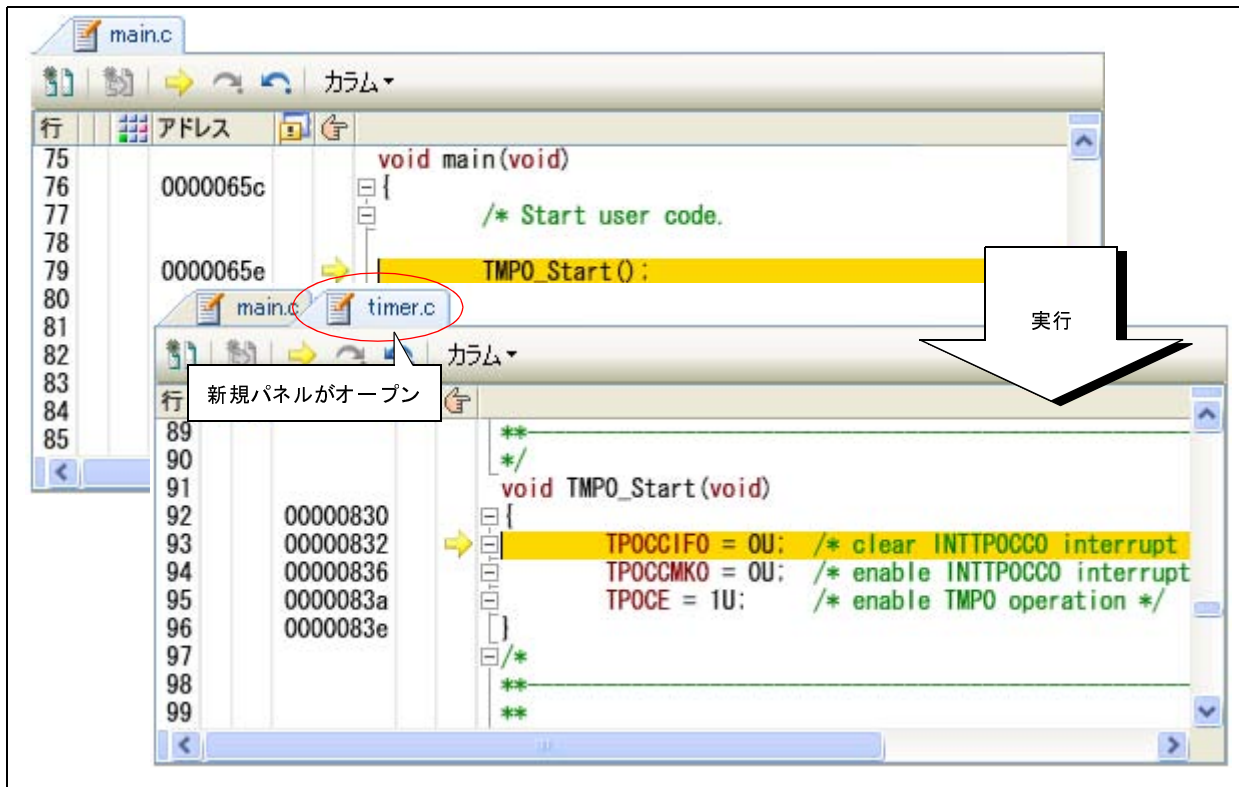
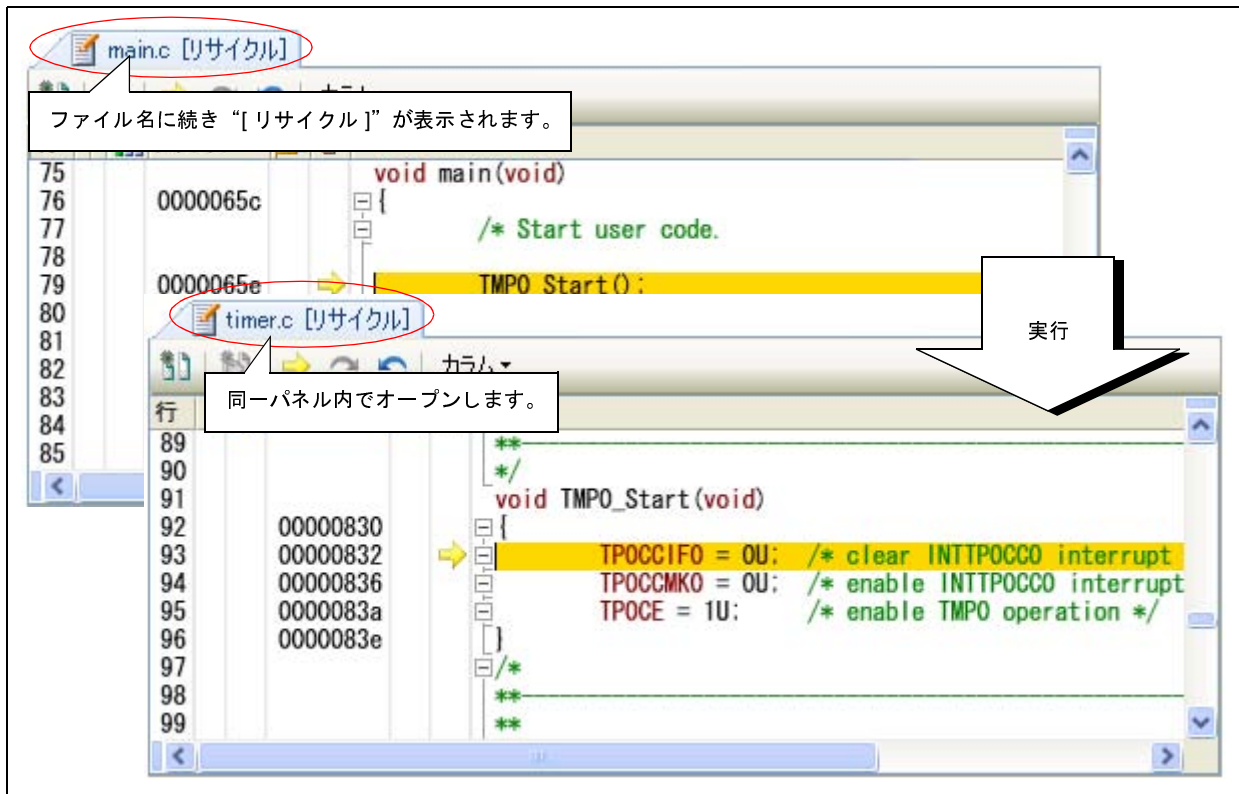


図 2—170 リサイクル・モードの動作



- 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・ファイルを表示している場合のみ有効となる機能です。
2. 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。

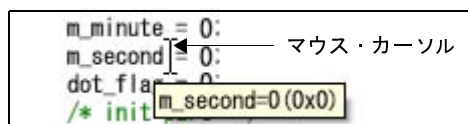
備考 すでに該当するソース・ファイルを表示している**エディタパネル**がオープンしている場合、リサイクル・モードのパネルには表示せず、すでにオープンしているエディタパネルを表示します。

(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—17 ウォッチ式の表示形式（デフォルト）」と同等）。

図 2—171 マウス・カーソルによる変数のポップアップ表示（エディタパネル）



(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  ボタンをクリック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—172 ソース・テキスト内の文字列検索（検索・置換ダイアログ）



(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—173 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)

**(a) [行番号 (有効な行の範囲)] の指定**

“(有効な行の範囲)”には、現在のファイルの有効な行の範囲が表示されます。

キャレットを移動したい行番号を 10 進数で直接入力により指定します。

または、シンボルを入力することも可能です。

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。

(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。

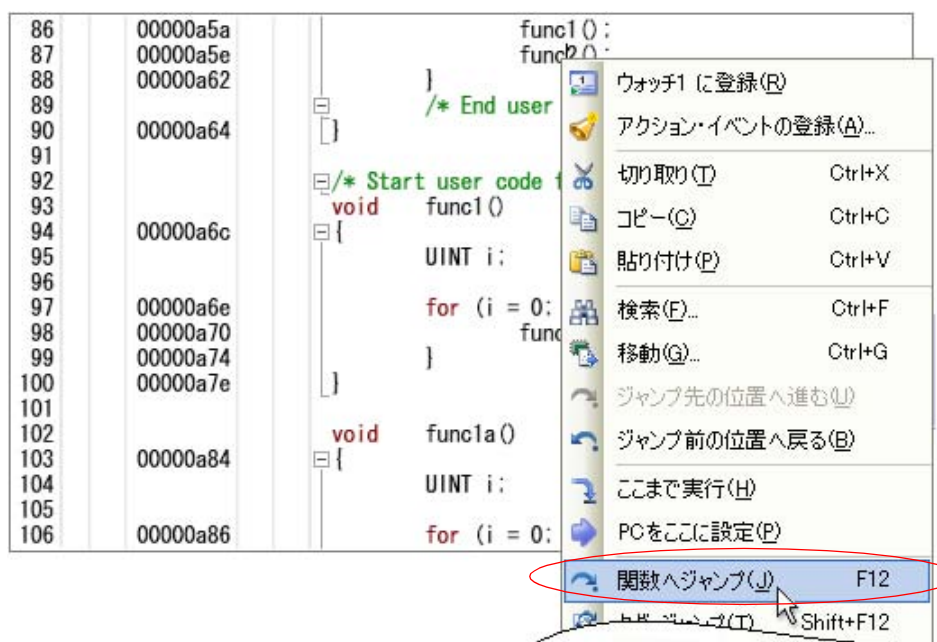
(7) 関数へジャンプする

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、または変数名【CX】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、対象関数にカー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—174 関数へジャンプ



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) CA850 の場合

- 対象がアクティブ・プロジェクト内の関数である。
 - 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。
 - 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。
- 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（ヘキサ・ファイルの場合、シンボル情報をダウンロードする設定が必要（「(4) ヘキサ／バイナリ・データ・フォーマットのファイルでソース・レベル・デバッグを行う」参照））。

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。

(b) CX 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。

(c) 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。
- 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（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（「[\(4\)ヘキサ／バイナリ・データ・フォーマットのファイルでソース・レベル・デバッグを行う](#)」参照））。

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

現在カー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）。

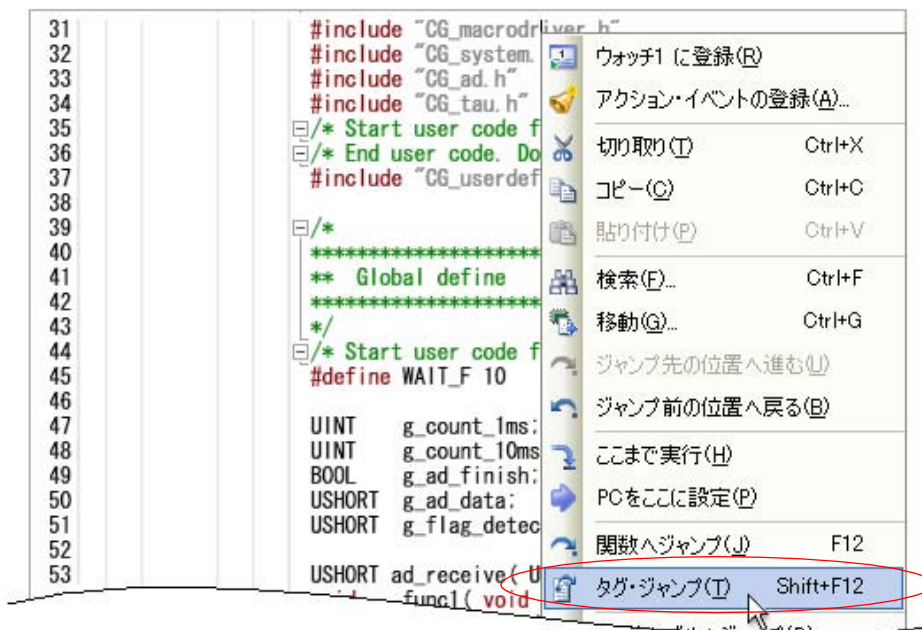
ソース・テキスト上で、対象行にカーレットを移動したのち、コンテキスト・メニューの [\[タグ・ジャンプ\]](#) を選択してください。

タグ・ジャンプの動作は次のとおりです。

表 2—6 タグ・ジャンプの動作

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

図 2—175 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います。
3. パス指定（パス／ファイル名）にスペースを含む場合は、" " で括られている必要があります。

2.6.2 逆アセンブル結果を表示する

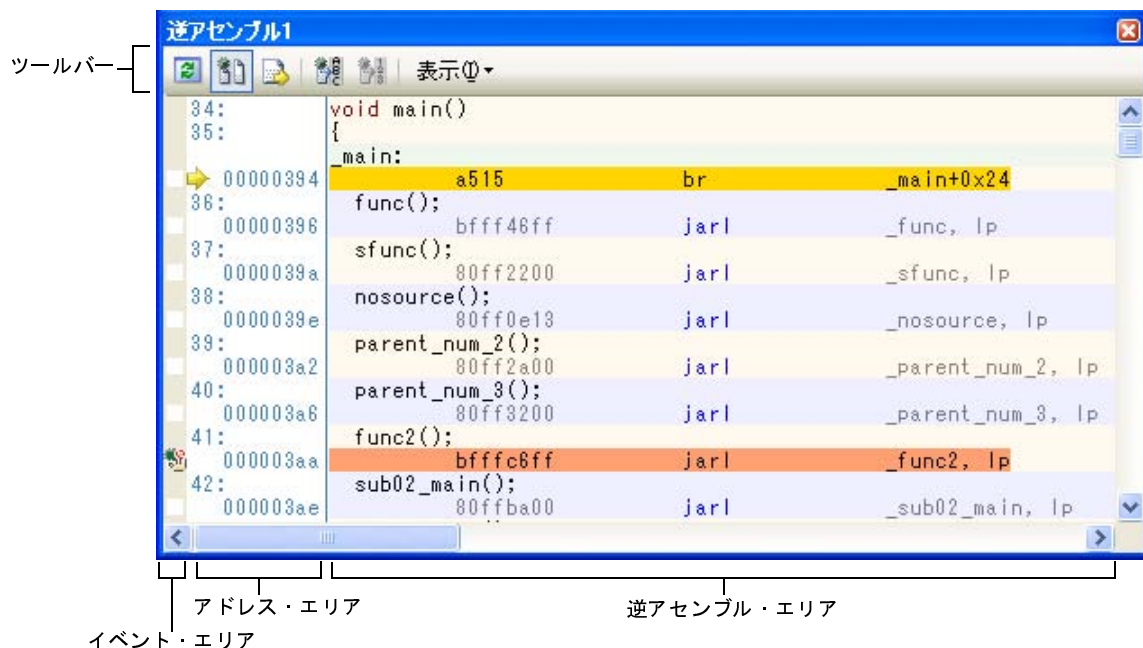
ダウンロードしたプログラム（メモリ内容）の逆アセンブル結果（逆アセンブル・テキスト）の表示は、次の逆アセンブルパネルで行います。


[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。

図 2—176 逆アセンブル結果の表示（逆アセンブルパネル）



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

ここでは、次の操作方法について説明します。

- (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

逆アセンブル結果は、デフォルトで混合表示モード（逆アセンブル・テキストとソース・テキストを混合表示）で表示されます。


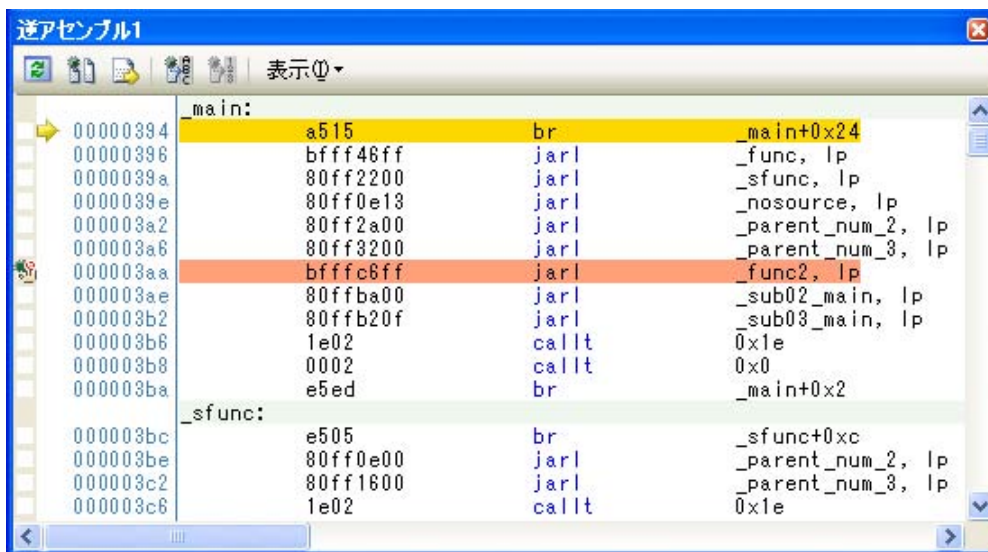
ツールバーの  ボタン（トグル）のクリックにより、ソース・テキストを表示／非表示にすることができます。

図 2—177 ソース・テキストを非表示にした場合の表示例



(2) 表示形式を変更する

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

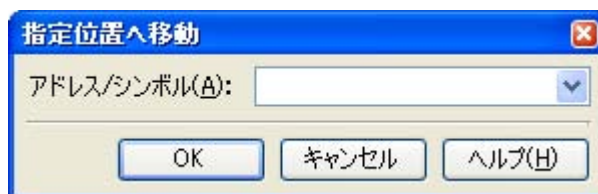
表示	表示形式を変更する次のボタンを表示します。
	ラベルのオフセット値を表示します。アドレスにラベルが定義されていない場合、一番近いラベルからのオフセット値を表示します。
	アドレス値を“シンボル + オフセット値”で表示します (デフォルト)。ただし、アドレス値にシンボルが定義されている場合は、シンボルのみを表示します。
	レジスタ名を機能名称で表示します (デフォルト)。
	レジスタ名を絶対名称で表示します。

(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—178 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)



(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

指定したアドレスへ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—179 逆アセンブル・データの保存（データ保存 ダイアログ）



(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。

図 2—180 逆アセンブル・データ保存の際の出カイメージ

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

- 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）。

(1) デバッグ専用プロジェクト以外の場合

- プロジェクトに追加している C ソース・ファイル/アセンブラ・ソース・ファイル/ヘッダ・ファイル/リンク・ディレクティブ・ファイル/シンボル情報ファイル/オブジェクト・モジュール・ファイル/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

(2) デバッグ専用プロジェクトの場合

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

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

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

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

- 備考 1. ソース・ファイル編集後、[Ctrl] + [S] キーの押下により、こまめに上書き保存することを推奨します。
2. ラピッド・ビルドの有効/無効は、プロジェクト全体（メイン・プロジェクト、およびサブプロジェクト）に対して設定されます。
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SEMBルを行う

逆アSEMB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SEMB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SEMB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SEMBル結果を別の命令で上書きすることによりメモリに空きが生じた場合、次の例のように自動的に nop 命令でバイト数を補います。

例 1. 3 行目の 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

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

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	0xffff, 0x20

2. 命令コードが“0x90, 0x07, 0xbb, 0xaa 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, 0xffff

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, 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, 0xffff

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caret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 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 1. 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。

2. 【V850E1】

選択しているマイクロコントローラが V850E/MA3 などの V850E コアの場合、内蔵 RAM 領域（アドレス 0x0FFF0000 ~ 0x0FFFEFFF）におけるプログラム実行に関して次の注意が必要です。

- 内蔵 RAM 内でブレイクした場合、[逆アセンブルパネル](#)では、アドレス 0x03FF0000 ~ 0x03FFFEFFF が表示されます。
- 内蔵 RAM 内の関数へステップ・インした場合、ステップ・オーバの動作となります。

備考 【V850E2】

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、プログラムの実行制御は、すべてのコアに対して同時に行われます（ただし、ステップ実行は、現在選択しているコア（PE n ）でのみ行われます）。

なお、デバッグ対象のコア（PE n ）の選択、およびその状態の表示は[デバッグ・マネージャパネル](#)【V850E2】で行ってください。


図 2—181 デバッグ・ツールバー



図 2—182 [デバッグ] メニュー



2.7.1 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（「[2.16 フック処理を設定する](#)」参照）。

2.7.2 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、「[2.8 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（「[2.16 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  ボタンをクリックします。

この操作によりプログラムの実行を開始した場合、次のいずれかの状態までその実行を続けます。

-  ボタンのクリック (「2.8.1 プログラムの実行を手動で停止する」参照)
- PC がブレークポイントに到達 (「2.8.2 任意の場所で停止する (ブレークポイント)」参照)
- ブレーク・イベント条件の成立 (「2.8.3 任意の場所で停止する (ブレーク・イベント)」 / 「2.8.4 変数 I/O レジスタへのアクセスで停止する」参照)
- フェイルセーフ・ブレークの発生 (「2.8.5 不正な実行を検出して停止する【IECUBE】」参照)
- その他のブレーク要因の発生

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


(2) 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。

-  ボタンのクリック (「2.8.1 プログラムの実行を手動で停止する」参照)
- PC がブレークポイントに到達 (「2.8.2 任意の場所で停止する (ブレークポイント)」参照)
- ブレーク・イベント条件の成立 (「2.8.3 任意の場所で停止する (ブレーク・イベント)」 / 「2.8.4 変数 I/O レジスタへのアクセスで停止する」参照)
- フェイルセーフ・ブレークの発生 (「2.8.5 不正な実行を検出して停止する【IECUBE】」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 (「2.8.1 プログラムの実行を手動で停止する」参照)
- フェイルセーフ・ブレークの発生 (「2.8.5 不正な実行を検出して停止する【IECUBE】」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/命令にカー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がカーレット位置のアドレスに到達
-  ボタンのクリック（「2.8.1 プログラムの実行を手動で停止する」参照）
- フェイルセーフ・ブレークの発生（「2.8.5 不正な実行を検出して停止する【IECUBE】」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/命令にカー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値が現在カーレットのある行/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。


2.7.3 プログラムをステップ実行する

次のいずれかの操作を行うと、現在のアドレス（カレントPC値で示されるアドレス）から、ソース・レベル単位（ソース・テキスト1行分）、または命令レベル単位（1命令分）でプログラムをステップ実行したのち、自動的に停止します。

プログラムの停止後は逐一各パネルの内容が自動的に更新されるため、ステップ実行は、プログラムの実行遷移をソース・レベル単位/命令単位でデバッグする場合に有効な実行方法です。

なお、ステップ実行を行う際の実行単位は、次に示すようにエディタパネルにおける設定に依存します。

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

備考  ボタンは、エディタパネルを混合表示モードに設定（「(1) 表示モードを変更する」参照）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. ステップ実行でスタンバイ・モード (HALT/STOP/IDLE) に移行する命令を実行した場合、スタンバイ・モード命令以降の次命令で停止します。

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。

- 【シミュレータ】以外

ステップ実行ではスタンバイ・モードに移行しません。

- 【シミュレータ】

ステップ実行でスタンバイ・モードに移行します。

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。

4. 【シミュレータ】以外


- ステップ実行中は、割り込みが禁止されます。また、フェイルセーフ・ブレーク【IECUBE】を発生しません。

5. 【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。

(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ま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  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。

2.8 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。

- 【シミュレータ】以外

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。


2. 【E1】【E20】【EZ Emulator】

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

2.8.1 プログラムの実行を手動で停止する

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

2.8.2 任意の場所で停止する（ブレークポイント）

ブレークポイントを設定することにより、任意の箇所でプログラムの実行を容易に停止させることができます。ブレークポイントは、マウスのワン・クリックで設定することができます。

ブレークポイントを設定するためには、あらかじめ使用するブレークポイントの種別、および動作の設定を行う必要があります。

ここでは、次の操作方法について説明します。

- (1) 使用するブレークポイントの種別／動作を設定する
- (2) ブレークポイントを設定する
- (3) ブレークポイントを削除する

(1) 使用するブレークポイントの種別／動作を設定する

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

なお、設定内容は、使用するデバッグ・ツールにより異なります。

- (a) 【シミュレータ】以外の場合
- (b) 【シミュレータ】の場合

(a) 【シミュレータ】以外の場合

図 2—183 【ブレーク】カテゴリ【IECUBE】【IECUBE2】【MINICUBE】【E1/E20(LPD)/(JTAG)】

日 ブレーク	
優先的に使用するブレークポイントの種類	ソフトウェア・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
オープン・ブレーク機能を使用する	いいえ (Output signal)

図 2—184 【ブレーク】カテゴリ【MINICUBE2】【E1/E20(Serial)】【EZ Emulator】

日 ブレーク	
優先的に使用するブレークポイントの種類	ソフトウェア・ブレーク

- 【優先的に使用するブレークポイントの種類】

マウスのワンクリック操作で設定するブレークポイントの種別を指定します。

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。

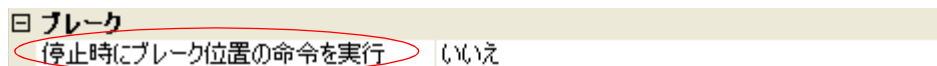
ハードウェア・ブレーク	デバッグ・ツールが、プログラム実行中にブレーク条件を逐次確認し、条件を満たした際にプログラムを停止させます ^注 。 設定すると、ハードウェア・ブレーク・イベント（実行系）として扱われます。
ソフトウェア・ブレーク	指定したアドレスの命令コードを一時的にブレーク用の命令に書き換え、その命令を実行した際にプログラムを停止させます（デフォルト）。 設定すると、ソフトウェア・ブレーク・イベントとして扱われます。

注 ハードウェア・ブレーク（実行系）には、指定したアドレスの命令実行前にブレークする“実行前ブレーク”と、命令実行後にブレークする“実行後ブレーク”があり、これらはそれぞれデバッグ・ツールの資源を用いて実現している機能です。CubeSuite+ では、ハードウェア・ブレークのブレークポイントを設定する際、まず“実行前ブレーク”の資源を使用し、資源がなくなり次第、“実行後ブレーク”の資源を使用します（「(1) 有効イベント数の制限」参照）。そのため、ユーザが実行前／実行後のどちらかを選択することはできません。

- 注意 1. 選択しているマイクロコントローラが使用可能なブレークポイントの種類が1つのみの場合、【優先的に使用するブレークポイントの種類】プロパティは表示されません。
2. 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。

(b) 【シミュレータ】の場合

図 2—185 【ブレーク】カテゴリ【シミュレータ】



- 【停止時にブレーク位置の命令を実行】

ブレークポイントによるプログラム実行停止のタイミングを、指定したアドレスの命令実行前（実行前ブレーク）とするか、または命令実行後（実行後ブレーク）とするかを指定します。

命令実行前にブレークする場合は【いいえ】を、命令実行後にブレークする場合は【はい】を選択してください（デフォルトでは【いいえ】が指定されます）。

なお、設定したブレークポイントは、すべてハードウェア・ブレーク・イベントとして扱われます。

注意 【はい】を指定した場合、現在設定されているアクション・イベントは、すべてハードウェア・ブレーク・イベントとして動作します（「2.14 プログラム内へのアクションの設定」参照）。

(2) ブレークポイントを設定する

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

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

ブレークポイントは、クリックした行に対応する先頭アドレスの命令に設定されます。

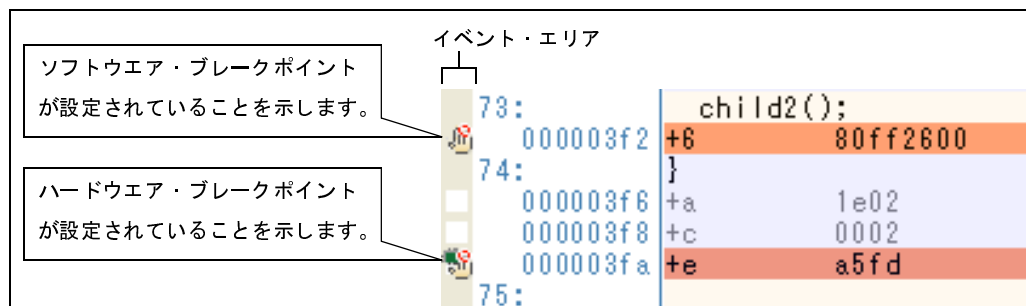
ブレークポイントが設定されると、設定した箇所に次のイベント・マークが表示され、ソース・テキスト行／逆アセンブル・テキスト行が強調表示されます。

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（「2.15 イベントの管理」参照）。

表 2—7 ブレークポイントのイベント・マーク

デバッグ・ツール	イベント種別	【優先的に使用するブレークポイントの種類】の設定	イベント・マーク
シミュレータ以外	ブレーク	ハードウェア・ブレーク	
		ソフトウェア・ブレーク	
シミュレータ	ブレーク	—	

図 2—186 ブレークポイントの設定例（逆アセンブルパネルの場合）




注意 1. ブレークポイントはブレーク・イベントとして設定され、イベントとして管理されるため、設定数に制限があります。ブレークポイントの設定に関しては（有効イベント数の制限など）、[「2. 15. 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. 【V850E1】【V850ES】：【IECUBE】【MINICUBE】【E1/E20(JTAG)】

- 疑似 RRM 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）とソフトウェア・ブレークは排他使用の機能です。そのため、[プロパティパネルの「デバッグ・ツール設定」タブの「実行中のメモリ・アクセス」](#)カテゴリ内「実行を一瞬停止してアクセスする」プロパティを「はい」に指定している場合、ソフトウェア・ブレークを設定することはできません。
- ハードウェア・ブレークとソフトウェア・ブレークが競合した場合、PC 値が不正な値となる場合があります（メッセージが出力されます）。この場合、ブレーク後の再実行で不正な命令を実行してしまう可能性があるため、CPU リセットを行ってください。

3. データフラッシュ・メモリ領域にソフトウェア・ブレークを設定することはできません。
4. 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。

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. 15. 1 設定状態（有効／無効）を変更する」](#)参照）。

また、すでにイベントが設定されている箇所で、新たにイベントを設定した場合は、複数のイベントが設定されていることを示すイベント・マーク（)が表示されます。

2. 【シミュレータ】

設定できるブレークポイントは、ハードウェア・ブレークポイント固定です。

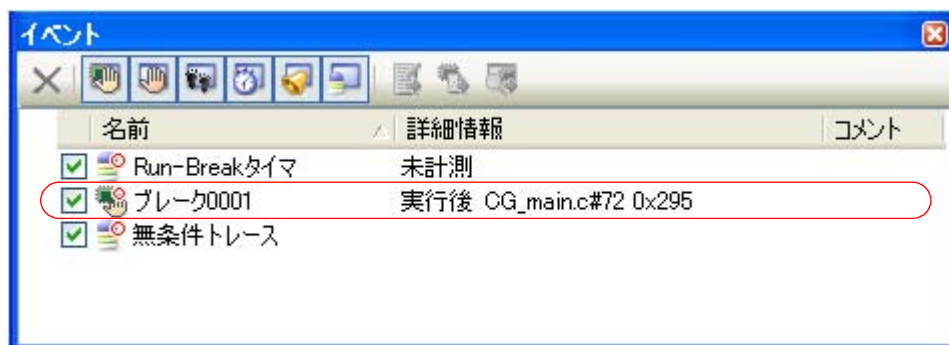
3. 【シミュレータ】以外

次に示す操作により、「[\(1\) 使用するブレークポイントの種別／動作を設定する](#)」の指定に依存することなく、ハードウェア・ブレークポイント／ソフトウェア・ブレークポイントを設定することができます。

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

注 “操作方法 1” は、[逆アセンブルパネル](#)でのみ有効です。

図 2—187 イベントパネルのブレイクポイントの設定例



(3) ブレイクポイントを削除する

設定したブレイクポイントを削除するには、[エディタパネル](#)/[逆アセンブルパネル](#)上において、表示されているイベント・マークを再度クリックします（イベント・マークが消失します）。

2.8.3 任意の場所で停止する（ブレイク・イベント）

ブレイク・イベント（実行系）を設定することにより、任意の箇所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) ブレイク・イベント（実行系）を設定する
- (2) ブレイク・イベント（実行系）を削除する

(1) ブレイク・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
ハードウェア・ブレイク	[ブレイクの設定] → [ハード・ブレイクの設定] を選択	デバッグ・ツールが、プログラム実行中にブレイク条件を逐次確認し、条件を満たした際にプログラムを停止させます ^注 。
ソフトウェア・ブレイク （【シミュレータ】以外）	[ブレイクの設定] → [ソフト・ブレイクの設定] を選択	指定したアドレスの命令コードを一時的にブレイク用の命令に書き換え、その命令を実行した際にプログラムを停止させます。



注 ハードウェア・ブレイク（実行系）には、指定したアドレスの命令実行前にブレイクする“実行前ブレイク”と、命令実行後にブレイクする“実行後ブレイク”があり、これらはそれぞれデバッグ・ツールの資源を用いて実現している機能です。CubeSuite+では、ハードウェア・ブレイクを設定する際、まず“実行前ブレイク”の資源を使用し、資源がなくなり次第、“実行後ブレイク”の資源を使用します

(「(1) 有効イベント数の制限」参照)。そのため、ユーザが実行前／実行後のどちらかを選択することはできません。

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対して設定されます。ブレーク・イベント（実行系）が設定されると、設定した箇所に次のイベント・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（「[2.15 イベントの管理](#)」参照）。

表 2—8 ブレーク・イベントのイベント・マーク

イベント種別	イベント・マーク
ハードウェア・ブレーク	
ソフトウェア・ブレーク （【シミュレータ】以外）	

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）、「[2.15.6 イベント設定に関する留意事項](#)」も参照してください。

2. **【V850E1】【V850ES】：【IECUBE】【MINICUBE】【E1/E20(JTAG)】**

- 疑似 RRM 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）とソフトウェア・ブレークは排他使用の機能です。そのため、[プロパティパネルの \[デバッグ・ツール設定\] タブ](#)の [実行中のメモリ・アクセス] カテゴリ内 [実行を一瞬停止してアクセスする] プロパティを [はい] に指定している場合、ソフトウェア・ブレークを設定することはできません。
- ハードウェア・ブレークとソフトウェア・ブレークが競合した場合、PC 値が不正な値となる場合があります（メッセージが出力されます）。この場合、ブレーク後の再実行で不正な命令を実行してしまう可能性があるため、CPU リセットを行ってください。

3. データフラッシュ・メモリ領域にソフトウェア・ブレークを設定することはできません。

備考 イベントの設定状態によりイベント・マークは異なります（「[2.15.1 設定状態（有効／無効）を変更する](#)」参照）。


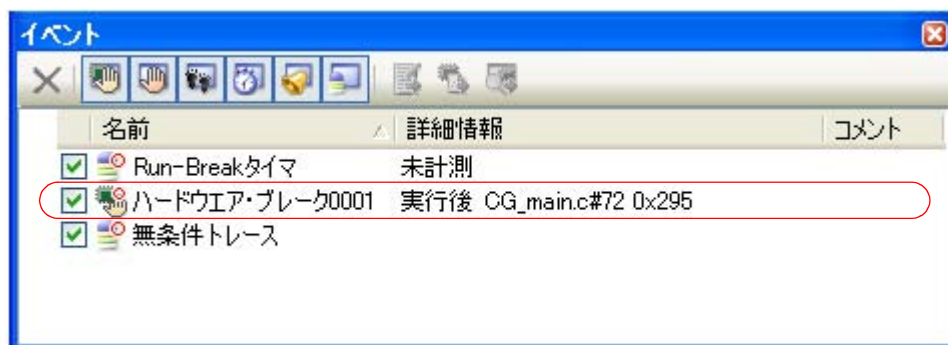

また、すでにイベントが設定されている箇所で、新たにイベントを設定した場合は、複数のイベントが設定されていることを示すイベント・マーク（)が表示されます。

図 2—188 イベントパネルのハードウェア・ブレイク・イベント（実行系）の設定例



(2) ブレイク・イベント（実行系）を削除する

設定したブレイク・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、対象となるハードウェア・ブレイク・イベント／ソフトウェア・ブレイク・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（「2.15.4 イベントを削除する」参照）。

2.8.4 変数 I/O レジスタへのアクセスで停止する

ブレイク・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。

アクセス系のブレイク・イベントで指定できるアクセス種別は次のとおりです。

表 2—9 変数へのアクセス種別

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

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) ブレイク・イベント（アクセス系）を設定する
- (2) ブレイク・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（アクセス系）の設定は、次のいずれかの操作により行います。

注意 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）、[「2.15.6 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 I/O レジスタにブレーク・イベント（アクセス系）を設定する場合

操作は、ソース・テキスト／逆アセンブル・テキストを表示している [エディタパネル／逆アセンブルパネル](#) 上で行います。

ソース・テキスト／逆アセンブル・テキスト上の任意の変数、または I/O レジスタを選択したのち、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。

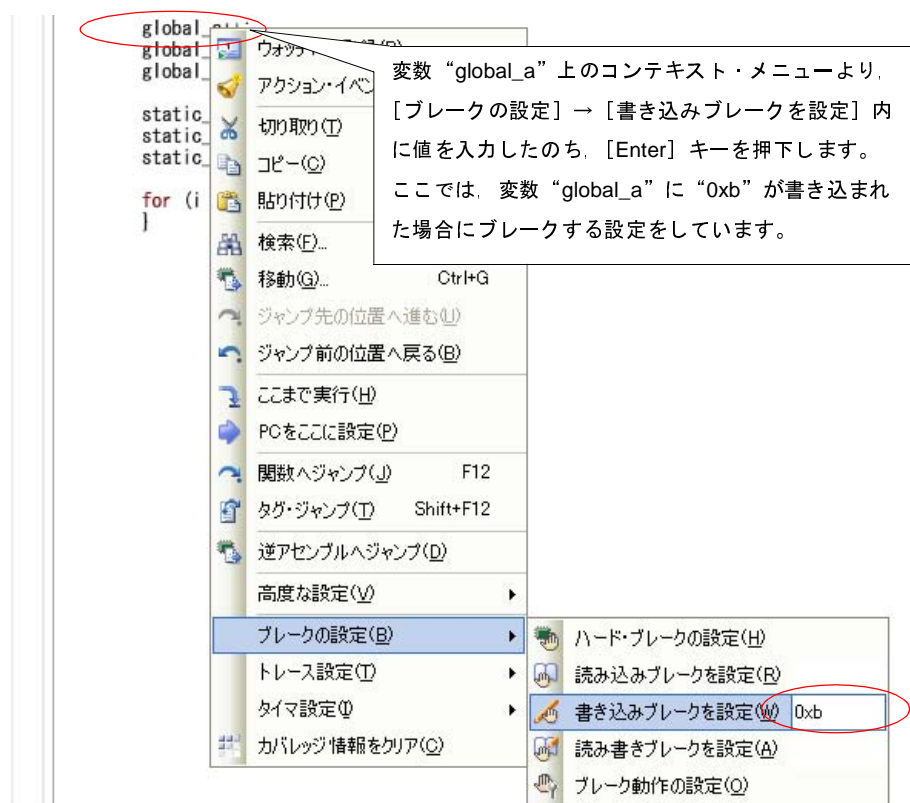
アクセス種別	操作方法
リード	[ブレークの設定] → [読み込みブレークを設定] を選択したのち、[Enter] キーを押下
ライト	[ブレークの設定] → [書き込みブレークを設定] を選択したのち、[Enter] キーを押下
リード／ライト	[ブレークの設定] → [読み書きブレークを設定] を選択したのち、[Enter] キーを押下

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 I/O レジスタを選択しても設定することはできません。

図 2—189 エディタ パネル上の変数に対するハードウェア・ブ레이크・イベント（アクセス系）の設定例



(b) 登録したウォッチ式にブ레이크・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 /I/O レジスタのみとなります。

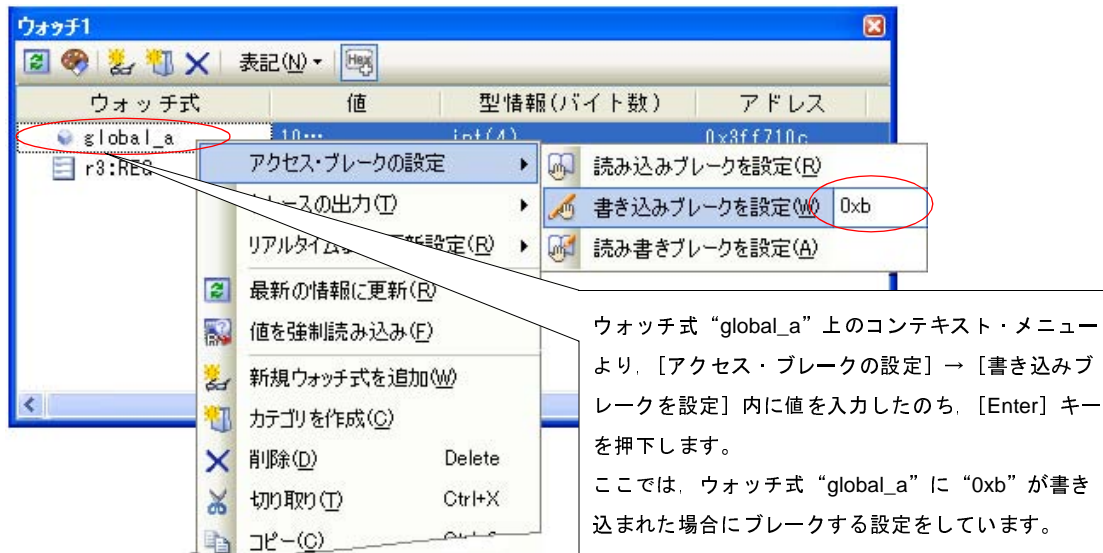
アクセス種別	操作方法
リード	[アクセス・ブ레이크の設定] → [読み込みブ레이크を設定] を選択したのち、[Enter] キーを押下
ライト	[アクセス・ブ레이크の設定] → [書き込みブ레이크を設定] を選択したのち、[Enter] キーを押下
リード／ライト	[アクセス・ブ레이크の設定] → [読み書きブ레이크を設定] を選択したのち、[Enter] キーを押下

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레이크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레이크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。

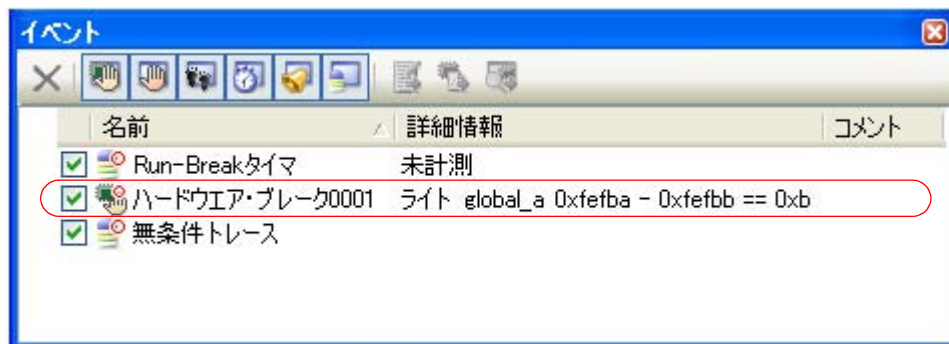
カレント・スコープ外のウォッチ式を対象とする場合は、スコープ指定したウォッチ式を選択してください。

図 2—190 ウォッチ式に対するハードウェア・ブレイク・イベント（アクセス系）の設定例




以上の操作を行うことにより、ブレイク・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・ブレイク・イベント（アクセス系）として管理されます（「2.15 イベントの管理」参照）。

図 2—191 イベントパネルのハードウェア・ブレイク・イベント（アクセス系）の設定例



(2) ブレイク・イベント（アクセス系）を削除する

設定したブレイク・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・ブレイク・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（「2.15.4 イベントを削除する」参照）。





2.8.5 不正な実行を検出して停止する【IECUBE】

内蔵 ROM/内蔵 RAM/I/O レジスタ/外部メモリなどへの不正なアクセスによる暴走の可能性を検出し、プログラムの実行を強制的にブレークさせることができます（フェイルセーフ・ブレーク機能）。

この機能には、多彩なブレーク条件が用意されており、それぞれのブレーク条件の有効/無効の設定は、プロパティパネルの [デバッグ・ツール設定] タブ上の [フェイルセーフ・ブレーク] カテゴリ内の各プロパティにより個別に行います。


注意 ステップ実行中は、フェイルセーフ・ブレーク機能は無効となります。

図 2—192 [フェイルセーフ・ブレーク] カテゴリ

日 フェイルセーフ・ブレーク		
内蔵 ROMへの書き込み直後に停止する		はい
内蔵 ROMの非メモリ・マッピング領域へのアクセス直後に停止する		はい
内蔵 RAMの非メモリ・マッピング領域へのアクセス直後に停止する		はい
内蔵 RAMの書き込み禁止領域への書き込みを検出する		はい
I/Oレジスタ領域内の読み込み禁止領域からの読み込み直後に停止する		はい
I/Oレジスタ領域内の書き込み禁止領域への書き込み直後に停止する		はい
I/Oレジスタ領域内の非マッピング領域にアクセス直後に停止する		はい
書き込み禁止外部メモリへの書き込み直後に停止する		はい
外部メモリの非メモリ・マッピング領域へのアクセス直後に停止する		はい

次に示す各プロパティの設定において、有効とする場合は [はい] を、無効とする場合は [いいえ] をドロップダウン・リストにより指定してください。

デフォルトでは、すべてのプロパティに [はい] が指定されます。

なお、設定欄の  マークは、そのプロパティ値の設定を変更することにより、自動的に CPU リセットを生じること示します。

- [内蔵 ROM への書き込み直後に停止する] 注1
- [内蔵 ROM の非メモリ・マッピング領域へのアクセス直後に停止する] 注1
- [内蔵 RAM の非メモリ・マッピング領域へのアクセス直後に停止する]
- [内蔵 RAM の書き込み禁止領域への書き込みを検出する] 注2
- [I/O レジスタ領域内の読み込み禁止領域からの読み込み直後に停止する]
- [I/O レジスタ領域内の書き込み禁止領域への書き込み直後に停止する]
- [I/O レジスタ領域内の非マッピング領域にアクセス直後に停止する]
- [書き込み禁止外部メモリへの書き込み直後に停止する]
- [外部メモリの非メモリ・マッピング領域へのアクセス直後に停止する]

注1. 選択しているマイクロコントローラが ROM レス品の場合は、無効となります。

2. プログラム停止時に、内蔵 RAM の書き込み禁止領域への書き込みを、ベリファイ・チェックすることにより検出するか否かを指定します。

2.8.6 その他のブレイク要因

上記のほか、プログラムの実行が停止する原因（ブレイク要因）には次のものがあります。

なお、ブレイク要因は、プログラム停止時に、[メイン・ウィンドウ](#)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。

表 2—10 その他のブレイク要因

要因	使用するデバッグ・ツール				
	IECUBE	IECUBE2	MINICUBE E1/E20(LPD) E1/E20(JTAG)	MINICUBE2 E1/E20(Serial) EZ Emulator	シミュレータ
トレース・メモリを使い切った ^{注1}	○	○	—	—	○
実行時間オーバの検出	○	○	○ ^{注2}	—	—
ノン・マップ領域へのアクセス	○	—	—	—	○
書き込み禁止領域への書き込み	○	—	—	—	○
フラッシュ・マクロ・サービス中	—	○	○	○	—
テンポラリ・ブレイクの発生 ^{注3}	○	○	○	○	○
エスケープ・ブレイクの発生	○	—	○	○	—
IRAM ガード領域への書き込み	○	—	—	—	—
不正命令例外の発生	○	○	○	—	—
実行の失敗、または不明な原因	○	○	○	○	—

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. V850E2 コアのみ

3. CubeSuite+ 内部でのみ使用するブレイク（ユーザは使用不可）

2.9 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。

備考【V850E2】

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、各情報の扱いは次のとおりです。

- メモリ範囲と値
現在選択しているコア（PE n ）にかかわらず、同じ値を表示します。
- レジスタ（IOR/PC を含む）の値
現在選択しているコア（PE n ）の値を取得し表示／設定します。
- シンボル（ウォッチ式／変数名を含む）
現在選択しているコア（PE n ）の PC 値を基にアドレスと値を決定します（たとえば、シンボルが特定のコアのみで有効であった場合でも、現在選択しているコアを基にアドレスと値を決定します）。

なお、デバッグ対象のコアの選択、およびその状態の表示は [デバッグ・マネージャパネル【V850E2】](#)で行ってください。

2.9.1 メモリを表示／変更する

メモリの内容の表示、および値の変更は、次の [メモリパネル](#)で行います。

[表示]メニュー→[メモリ]→[メモリ1～4]を選択してください。

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


なお、各エリアの見方、および機能についての詳細は、[メモリパネル](#)の項を参照してください。

図 2—193 メモリの内容の表示（データフラッシュ・メモリ非内蔵品の場合のメモリパネル）



図 2—194 メモリの内容の表示（データフラッシュ・メモリ内蔵品の場合のメモリパネル）



備考 1. 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。

2. 【シミュレータ】以外

選択しているマイクロコントローラがデータフラッシュ・メモリ内蔵品の場合、データフラッシュ・メモリのIDタグのメモリ値も表示／変更することができます。

詳細については、メモリパネルの項を参照してください。

ここでは、次の操作方法について説明します。

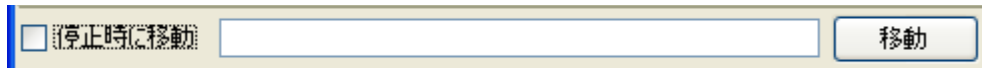
- (1) 表示位置を指定する
- (2) 値の表示形式を変更する
- (3) メモリの内容を変更する
- (4) プログラム実行中にメモリの内容を表示／変更する
- (5) メモリの内容を検索する
- (6) メモリの内容を一括して変更（初期化）する
- (7) メモリを表示内容を保存する

(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—195 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ます。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caret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]	<input checked="" type="checkbox"/>	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caretが移動します。
	<input type="checkbox"/>	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caret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。

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。
	メモリ値を 16 進数で表示します（デフォルト）。
	メモリ値を符号付き 10 進数で表示します。
	メモリ値を符号なし 10 進数で表示します。
	メモリ値を 8 進数で表示します。
	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。
	メモリ値を4ビット幅で表示します。
	メモリ値を8ビット幅で表示します (デフォルト)。
	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。
	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。
	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。
	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。
	文字列を単精度浮動小数点数値 ^注 で表示します。
	文字列を倍精度浮動小数点数値で表示します。
	文字列を単精度浮動小数点数の複素数で表示します。
	文字列を倍精度浮動小数点数の複素数で表示します。
	文字列を単精度浮動小数点数の虚数で表示します。
	文字列を倍精度浮動小数点数の虚数で表示します。
表示	表示形式を変更する次のボタンを表示します。
	スクロール範囲を設定するための 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・オフセット値を設定	アドレス・エリアに表示するアドレスのオフセット値を設定するため、 アドレス・オフセット設定 ダイアログ をオープンします。

注 浮動小数点数値表示についての詳細は、[メモリ パネル](#)の項を参照してください。

(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。

2. 数値の途中に“-”を入力した場合

符号あり 10進数 16ビット表示において、表示値“32768”を“32-68”と編集した場合、“3”と“2”が空白に変わり、変更値は“-68”となります。

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。

4. 同一の値を入力した場合

現在のメモリ値と同一の値を指定した場合でも、指定した値をメモリに書き込みます。

(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、デバッグ・ツールが持つ RRM 機能（読み込み）【IECUBE】【シミュレータ】、疑似 RRM 機能（読み込み）（【シミュレータ】以外）、DMM 機能／疑似 DMM 機能（書き込み）により実現され、各機能ごとに読み込み／書き込みが可能な対象領域は異なります。

まず、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。

表 2—11 リアルタイム表示更新機能の基本設定

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]

注意 ローカル変数は、リアルタイム表示更新機能の対象外です。

備考 メモリパネル／ウォッチパネルにおける値の書き換え方法についての詳細は、「(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）【IECUBE】【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。

注意 1. 【IECUBE】

RRM 機能の対象領域のバンク（256 バイト境界）をまたぐように配置された変数式の値を読み込むと、上位アドレスが不正値となり、値をエラー色で表示します。

2. 【IECUBE2】

RRM 機能はサポートしていません。

表 2—12 RRM 機能の対象領域

対象領域	IECUBE 注1	シミュレータ	
		V850E1/V850ES	V850E2
内蔵 ROM	○	○	○
内蔵 RAM	○	○	○
H バス共有メモリ領域	—	—	○
バックアップ RAM 領域	—	—	○
データフラッシュ (ID タグを除く)	○	—	—
エミュレーション・メモリ	○	○	○
ターゲット・メモリ	○	○	○
CPU レジスタ	—	○	○注2
I/O レジスタ	—	○	○

注 1. 【IECUBE】

対象領域は、合計して 8 箇所／最大 2048 バイトに限定されます。

そのため、CubeSuite+ では、次に示す決定規則（優先順位）に従い、上記制限内によるリアルタイム表示更新の対象を自動的に決定します（プログラム実行直前において、前面に表示しているパネルのみが対象となります）。

- (1) **ウォッチパネル**に表示しているウォッチ式を上から順に設定（複数のウォッチパネルをオープンしている場合、パネルの番号の小さい順）
- (2) **メモリパネル**に表示しているメモリをアドレスの小さい順に設定（複数のメモリパネルをオープンしている場合、パネルの番号の小さい順）

なお、対象領域にリード禁止領域が含まれている場合は、その領域の表示更新は行いません。

2. トレーサ／タイマ動作中は不可

ただし、RRM 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	プロパティ	設定値
IECUBE	[実行中のメモリ・アクセス] カテゴリ → [リアルタイム表示更新を自動設定する]	[はい]
	[実行中のメモリ・アクセス] カテゴリ → [実行を一瞬停止してアクセスする]	[はい] 注
	[トレース] カテゴリ → [トレース・データの用途]	[RRM]
シミュレータ	設定不要	

注 【V850E1】 【V850ES】 : 【IECUBE】 【MINICUBE】 【E1/E20(JTAG)】

ソフトウェア・ブレークポイントを有効状態で設定している場合は、それらを無効状態にする必要があります。また、このプロパティを [はい] に指定したのちは、ソフトウェア・ブレークポイントを設定することはできません。

(b) 疑似 RRM 機能（読み込み）（【シミュレータ】以外）

プログラムの実行を一瞬停止して、ソフトウェア・エミュレーションによりメモリ/ウォッチ式の内容を読み込む機能です。

疑似 RRM 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・エラーが発生します。

ただし、[実行中のメモリ・アクセス] カテゴリ内の [HALT を解除してアクセスする] プロパティにおいて [はい] を指定している場合、HALT モードに移行した場合のみ、HALT 状態を解除して読み込みを行います（【IECUBE】 【IECUBE2】 【MINICUBE】 【E1/E20(LPD)/(JTAG)】）。

表 2—13 疑似 RRM 機能の対象領域 【V850E1】 【V850ES】

領域名	IECUBE	MINICUBE E1/E20(JTAG)	MINICUBE2 E1/E20(Serial) EZ Emulator
内蔵 ROM	○	○	—
内蔵 RAM	○	○	○注
データフラッシュ (ID タグを除く)	○	○	○
エミュレーション・メモリ	○	—	—
ターゲット・メモリ	○	○	—
CPU レジスタ	○	○	—
I/O レジスタ	○	○	—

注 割り込み禁止 (DI) 期間は不可

表 2—14 疑似 RRM 機能の対象領域【V850E2】

領域名	IECUBE2		MINICUBE E1/E20(LPD)/(JTAG)		MINICUBE2 E1/E20(Serial)
	設定 A	設定 B	設定 A	設定 B	
内蔵 ROM ^{注1}	○ ^{注3}	○ ^{注3}	○ ^{注4}	○ ^{注4}	—
内蔵 RAM ^{注2}	○	○	○	○	○ ^{注5}
Hバス共有メモリ領域	—	○	—	○	—
バックアップRAM領域	—	○	—	○	○ ^{注5}
データフラッシュ (IDタグを除く)	—	○	—	○ ^{注4}	—
ターゲット・メモリ	—	○	○	○ ^{注4}	—
CPUレジスタ	—	○	○	○ ^{注4}	—
I/Oレジスタ	—	○	○	○ ^{注4}	—

注 1. 実行前のキャッシュ・データのため、リアルタイムな値ではない

2. 実行を一瞬停止することなく読み込み可 (MINICUBE2/E1(Serial)/E20(Serial)を除く)

3. プロパティパネルの [接続用設定] タブ上の [フラッシュ] カテゴリ内 [フラッシュ・セルフ・プログラミングを行う] プロパティが [いいえ] の場合のみ可

4. 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のみ可

5. 割り込み禁止 (DI) 期間は不可

ただし、疑似 RRM 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	プロパティ		設定値
【V850E2】 IECUBE2 MINICUBE E1/E20(LPD) E1/E20(JTAG)	設定 A	[実行中のメモリ・アクセス] カテゴリ → [実行中にアクセスする]	[はい]
	設定 B	[実行中のメモリ・アクセス] カテゴリ → [実行中にアクセスする]	[はい]
		[実行中のメモリ・アクセス] カテゴリ → [実行を一瞬停止してアクセスする]	[はい]
上記以外	[実行中のメモリ・アクセス] カテゴリ → [実行を一瞬停止してアクセスする]		[はい] ^注

注 【V850E1】【V850ES】: 【IECUBE】【MINICUBE】【E1/E20(JTAG)】

ソフトウェア・ブレークポイントを有効状態で設定している場合は、それらを無効状態にする必要があります。また、このプロパティを [はい] に指定したのちは、ソフトウェア・ブレークポイントを設定することはできません。

(c) DMM 機能／疑似 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／疑似 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 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・エラーが発生します。

ただし、[実行中のメモリ・アクセス] カテゴリ内の [HALT を解除してアクセスする] プロパティにおいて [はい] を指定している場合、HALT モードに移行した場合のみ、HALT 状態を解除して書き込みを行います（[IECUBE] [IECUBE2] [MINICUBE] [E1/E20(LPD)/(JTAG)]）。

表 2—15 DMM 機能／疑似 DMM 機能の対象領域【V850E1】【V850ES】

領域名	IECUBE	MINICUBE E1/E20(JTAG)	MINICUBE2 E1/E20(Serial) EZ Emulator	シミュレータ
内蔵 ROM	—	—	—	○
内蔵 RAM	●	●	●注 1	○
エミュレーション・メモリ	●	—	—	○
ターゲット・メモリ	●	●	—	○
CPU レジスタ	●	●	●注 1	○注 2
I/O レジスタ (標準 I/O レジスタのみ)	●	●	●	○

表 2—16 DMM 機能／疑似 DMM 機能の対象領域【V850E2】

領域名	IECUBE2		MINICUBE E1/E20(JTAG)/(LPD)		MINICUBE2 E1(Serial) E20(Serial)	シミュレータ
	設定 A	設定 B	設定 A	設定 B		
内蔵 ROM	—	—	—	—	—	○
内蔵 RAM	○	○	○	○	●注 1	○
H バス 共有メモリ領域	—	●	—	●注 3	—	○
バックアップ RAM 領域	—	●	—	●注 3	●注 1	○
エミュレーション・メモリ	—	—	—	—	—	○
ターゲット・メモリ	—	●	—	●注 3	—	○
CPU レジスタ	—	●	—	●注 3	●注 1	○注 2
I/O レジスタ	—	●	—	●注 3	●注 1	○注 2

●： 実行を一瞬停止して可能

○： 実行を一瞬停止せずに可能

注 1. 割り込み禁止（DI）期間は不可

2. トレーサ／タイマ動作中は不可

3. マルチコア対応版は不可

ただし、DMM 機能／疑似 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	プロパティ		設定値
【V850E2】 IECUBE2 MINICUBE E1/E20(LPD) E1/E20(JTAG)	設定 A	[実行中のメモリ・アクセス] カテゴリ → [実行中にアクセスする]	[はい]
	設定 B	[実行中のメモリ・アクセス] カテゴリ → [実行中にアクセスする]	[はい]
		[実行中のメモリ・アクセス] カテゴリ → [実行を一瞬停止してアクセスする]	[はい]
シミュレータ	設定不要		
上記以外	[実行中のメモリ・アクセス] カテゴリ → [実行を一瞬停止してアクセスする]		[はい] 注

注 【V850E1】【V850ES】：【IECUBE】【MINICUBE】【E1/E20(JTAG)】

ソフトウェア・ブレークポイントを有効状態で設定している場合は、それらを無効状態にする必要があります。また、このプロパティを [はい] に指定したのは、ソフトウェア・ブレークポイントを設定することはできません。

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。

また、さらに、IECUBE を使用して RRM 機能によるリアルタイム表示更新を行っているメモリ値は、[メモリパネル](#)において、アクセス状態に従って背景色が次のように変化します（表示の際の文字色／背景色は、[オプションダイアログ](#)における [全般 - フォントと色] カテゴリの設定に依存します）。

アクセス状態	表示例
リード／フェッチ	00 00 00 00
ライト	00 00 00 00
リードとライト	00 00 00 00

注意 RRM 機能や疑似 RRM 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を読み出すため、実際には代入していない値を表示することがあるため注意が必要です。

図 2—196 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）



図 2—197 RRM 機能による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）【IECUBE】



(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—198 メモリ内容の検索（メモリ検索 ダイアログ）



- 注意 1. プログラム実行中に、メモリの内容を検索することはできません。
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。
3. データフラッシュ・メモリ用 ID タグの値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	<p>選択したメモリ・マッピング範囲内で検索を行います。</p> <p>このリスト項目は、メモリ・マッピング ダイアログで表示しているメモリ・マッピングを個々に表示します（ノン・マップ領域を除く）。</p> <p>表示形式：<メモリ種別> <アドレス範囲> <サイズ></p>

(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（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、現在のカー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。

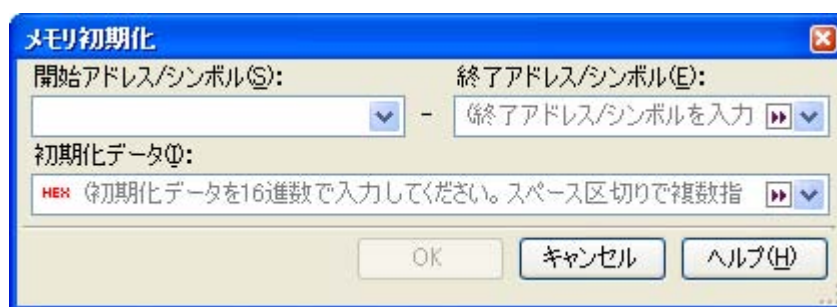
(6) メモリの内容を一括して変更（初期化）する

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—199 メモリ内容の一括変更（メモリ初期化ダイアログ）



注意 この操作は、データフラッシュ・メモリ用 ID タグの値に対して行うことはできません。

(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 (初期化データ)	書き込みイメージ (バイト単位)	
	リトル・エンディアン	ビッグ・エンディアン
1	01	01
0 12	00 12	00 12
00 012 345	00 12 00 45 03	00 00 12 03 45
000 12 000345	00 00 12 45 03 00	00 00 12 00 03 45

(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。

図 2—200 メモリ・データの保存（データ保存ダイアログ）



(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—201 メモリ・データ保存の際の出カイメージ

【テキスト・ファイル (*.txt) で保存】

(16 進表記 /8 ビット幅 /ASCII コードの場合の例)

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+a	+b	+c	+d	+e	+f
00000000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000010	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11

【CSV ファイル (*.csv) で保存】

(16 進表記 /8 ビット幅 /ASCII コードの場合の例)

00000000,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00000010,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,

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。

2.9.2 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—202 CPU レジスタの内容の表示 (CPU レジスタパネル)



ここでは、次の操作方法について説明します。











- (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。

表記	値の表示形式を変更する次のボタンを表示します。
	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）。
	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。
	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。
	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。
	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。
	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。
	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。 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。
	選択している項目を Double で表示します。 ただし、8 バイト・データ以外の場合は、規定値で表示します。
	値表示の末尾に、その値の 16 進数表記を “()” で囲んで併記します。

(2) 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、「[2.9.6 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—203 CPU レジスタ保存の際の出カイメージ

レジスタ名	値

カテゴリ名	
- レジスタ名	値
:	:

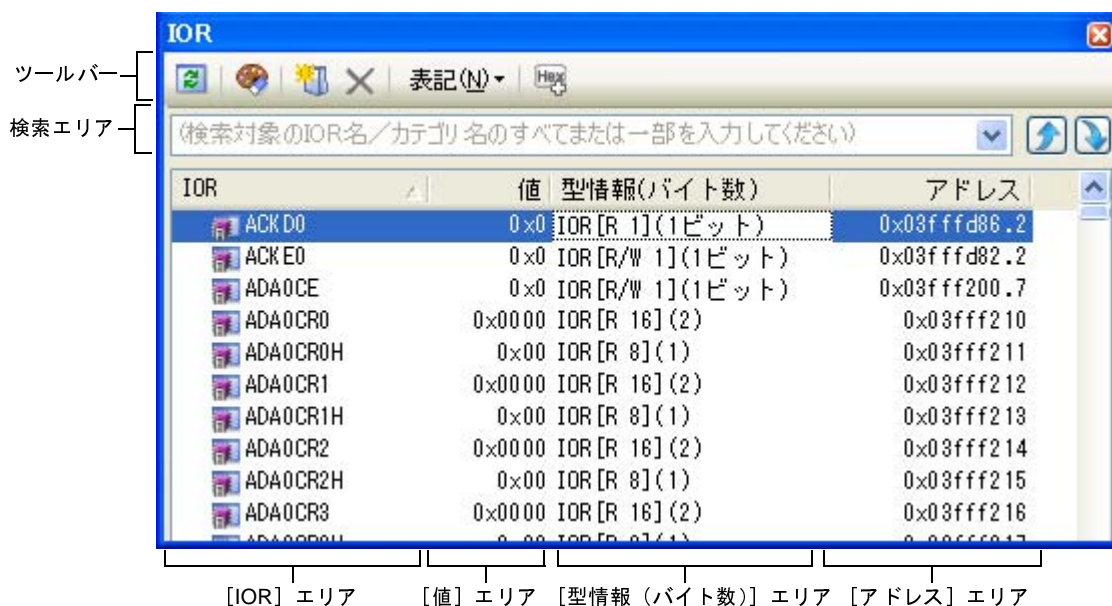
2.9.3 I/O レジスタを表示／変更する

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—204 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。

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

備考 1. カテゴリ（フォルダ）により分類されて非表示の状態の I/O レジスタ名も検索します（展開して選択状態となります）。

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  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。
	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）。
	選択している項目の値を符号付き10進数で表示します。
	選択している項目の値を符号なし10進数で表示します。
	選択している項目の値を8進数で表示します。
	選択している項目の値を2進数で表示します。
	選択している項目の値をASCIIコードで表示します。
	選択している項目の値表示の末尾に、その値の16進数表記を“()”で囲んで併記します。

(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“WTM”の値に“0x41”を書き込んだ場合

→ WTM に、“0x41”が書き込まれます。

- I/O レジスタ名“WTM”の値に ASCII 文字“A”を書き込んだ場合

→ WTM 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、「[2.9.6 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—205 I/O レジスタ保存の際の出カイメージ

IOR 名	値	型情報 (バイト数)	アドレス
カテゴリ名	値	型情報 (バイト数)	アドレス
-IOR 名	:	:	:
:	:	:	:

2.9.4 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, 「[2.9.6 ウォッチ式を表示/変更する](#)」を参照してください。

2.9.5 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。

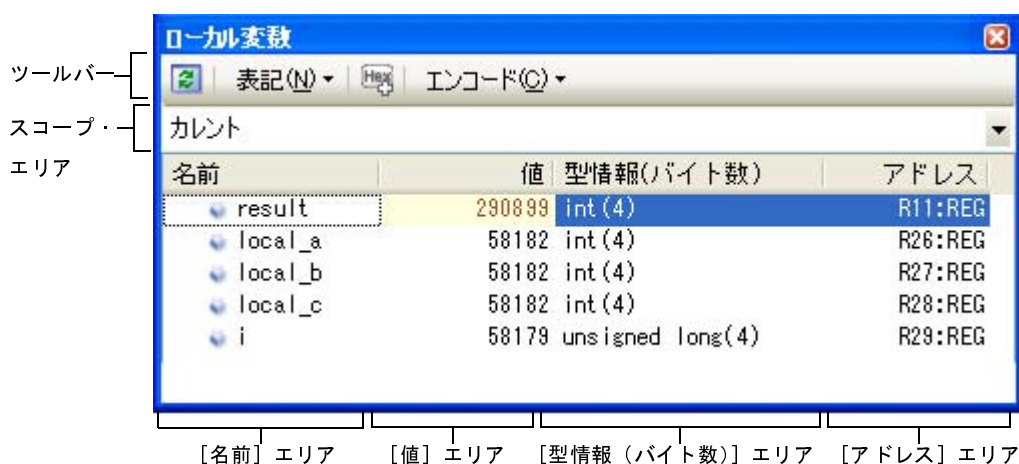
ローカル変数パネルでは, ローカル変数名や関数名を表示します。また, 関数の引数もローカル変数として表示します。

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—206 ローカル変数の内容の表示 (ローカル変数パネル)


















ここでは, 次の操作方法について説明します。

- (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。

表記	値の表示形式を変更する次のボタンを表示します。
	このパネル上の値の表記を変数ごとの規定値で表示します（デフォルト）。
	このパネル上の値を 16 進数で表示します。
	このパネル上の値を 10 進数で表示します。
	このパネル上の値を 8 進数で表示します。
	このパネル上の値を 2 進数で表示します。
	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）。
	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。
	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。
	このパネル上の値を Double で表示します。 ただし、8 バイト・データ以外、または型情報を持つ場合は、規定値で表示します。
	値表示の末尾に、その値の 16 進数表記を “()” で囲んで併記します。
エンコード	文字列変数のエンコードを変更する次のボタンを表示します。
	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。
	文字列変数を EUC-JP コードで表示します。
	文字列変数を UTF-8 コードで表示します。
	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。
- 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.** ローカル変数の値には、次のように ASCII 文字による入力も可能です。
- ASCII 文字による入力の場合
 - 変数 “ch” の [値] エリアに “A” を入力
 - “ch” が割り当てられているメモリ領域に “0x41” を書き込む

- 数値による入力の場合
変数“ch”の[値]エリアに“0x41”を入力
→ “ch”が割り当てられているメモリ領域に“0x41”を書き込む
- 文字列（ASCII）による入力の場合
文字配列“str”の表示形式をASCIIに設定し、[値]エリアに“ABC”を入力
→ “str”が割り当てられているメモリ領域に“0x41, 0x42, 0x43, 0x00”を書き込む

(3) ローカル変数の表示内容を保存する

[ファイル]メニュー→[名前を付けてローカル変数データを保存...]を選択することにより、名前を付けて保存ダイアログをオープンし、ローカル変数のすべての内容をテキスト・ファイル(*.txt)/CSVファイル(*.csv)に保存することができます。

ファイルに保存する際は、デバッグ・ツールから最新の情報を取得します。

なお、配列、ポインタ型変数、構造体/共用体、CPUレジスタ（部分を表す名前が付与されているもののみ）を展開表示している場合では、各展開要素の値も保存されます。展開表示していない場合は、先頭に“+”マークが付与され、値は空欄となります。

図 2—207 ローカル変数保存の際の出カイメージ

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

2.9.6 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—208 ウォッチ式の内容の表示（ウォッチパネル）



ここでは、次の操作方法について説明します。

- (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128個まで登録することができます（上限値を越えて登録しようとした場合、メッセージを表示します）。
- 2.** コンパイラによる最適化のため、対象となる変数を使用していないブロックでは変数データがスタック／レジスタに存在しない場合があります。この場合、対象となる変数をウォッチ式として登録しても値の表示は“?”のままとなります。

- 備考 1.** 各ウォッチパネル（ウォッチ1～ウォッチ4）上で登録したウォッチ式は、それぞれ個別に管理され、プロジェクトのユーザ情報として保存されます。
- 2.** ウォッチ式は、同名を複数登録することができます。

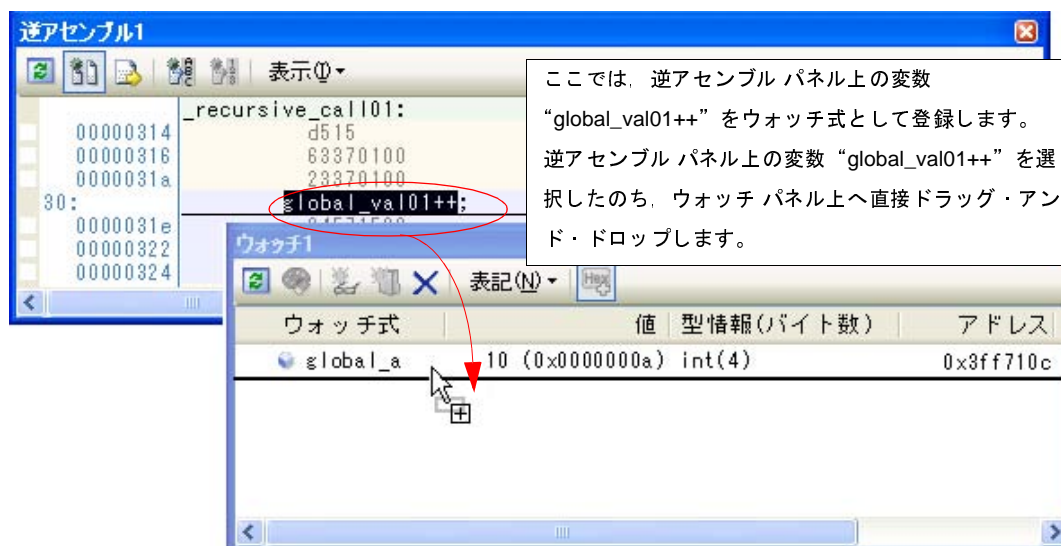
(a) 他のパネルから登録する場合

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。

なお、この操作が可能なパネルと、ウォッチ式として登録可能な対象との関係についての詳細は、「表 A—12 各パネルとウォッチ式として登録可能な対象の関係」を参照してください。

図 2—209 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


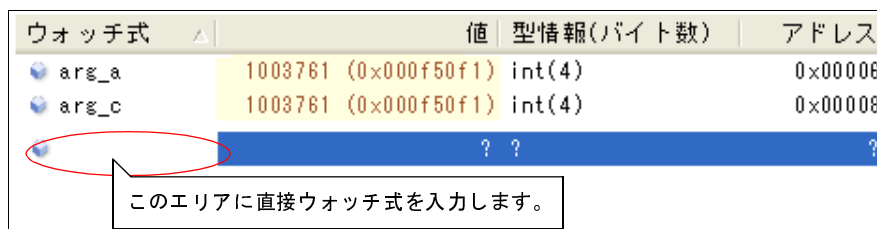
任意のウォッチパネル（ウォッチ 1～ウォッチ 4）において、ツールバーの  ボタンをクリックすると、[ウォッチ式] エリアに次のエントリ・ボックスが表示されます。

図 2—210 ウォッチ式のエントリ・ボックス



エントリ・ボックス内に、キーボードより直接ウォッチ式を入力したのち、[Enter] キーを押下します。なお、この際のウォッチ式の入力形式については、次の表を参照してください。

- 「表 A—13 ウォッチ式の入力形式」
- 「表 A—14 C 言語変数をスコープ指定してウォッチ登録した場合の扱い」
- 「表 A—15 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—16 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。

2. 1つのウォッチパネルにおいて、カテゴリは64個まで作成することができます（上限値を越えて作成しようとした場合、メッセージを表示します）。


(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  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  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  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。
	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—17 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）。
	選択している項目の値を 16 進数で表示します。
	選択している項目の値を符号付き 10 進数で表示します。
	選択している項目の値を符号なし 10 進数で表示します。
	選択している項目の値を 8 進数で表示します。
	選択している項目の値を 2 進数で表示します。
	選択している項目の値を ASCII コードで表示します。
	選択している項目の値を 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。
	選択している項目の値を Double で表示します。 ただし、選択しているウォッチ式が 8 バイト・データの場合のみ有効となります。
	選択している項目の値表示の末尾に、その値の 16 進数表記を “()” で囲んで併記します。 ただし、16 進数表記をしている場合は併記しません。

(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。

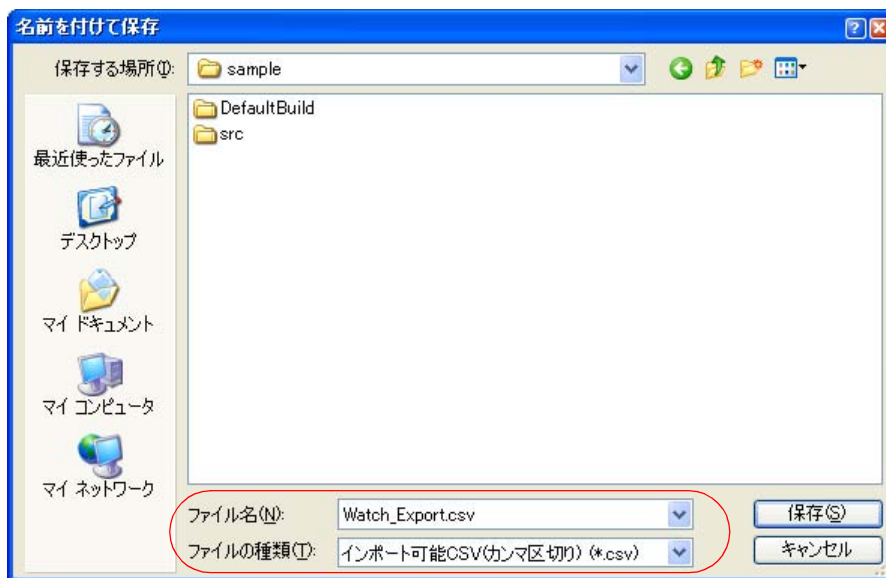
- [ファイル名] : 保存するファイル名 (*.csv) を指定します。

- [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—211 ウォッチ式のエクスポート



(b) ウォッチ式をインポートする

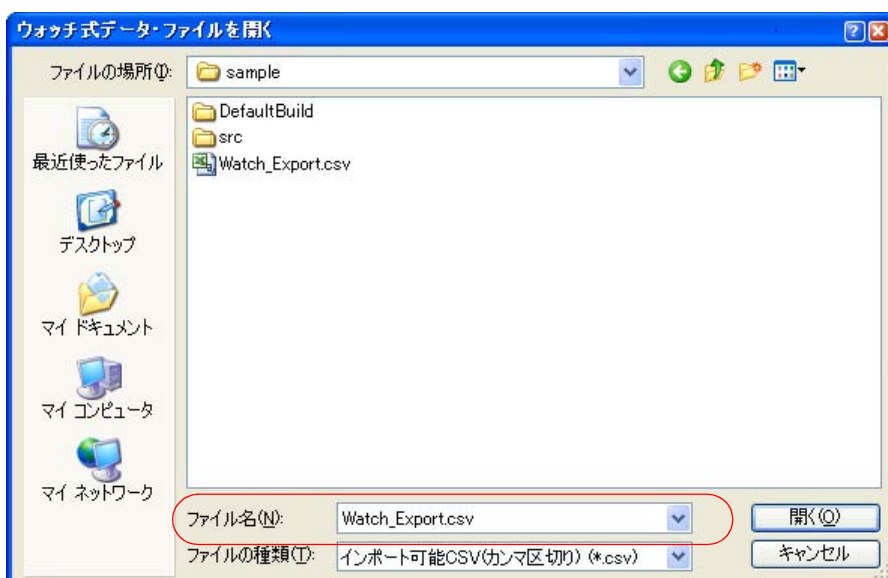
(a) でエクスポートしたファイルを、ウォッチパネルにインポートします。

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。

図 2—212 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—213 ウォッチ・データ保存の際の出カイメージ

ウォッチ式	値	型情報 (バイト数)	アドレス	メモ
変数式	値	型情報 (バイト数)	アドレス	メモ
- カテゴリ名				
変数式	値	型情報 (バイト数)	アドレス	メモ
⋮	⋮	⋮	⋮	⋮

備考 [ファイル]メニュー→[ウォッチ・データを保存]の選択によりパネルの内容を上書き保存した場合、ウォッチパネル（ウォッチ1～4）はそれぞれ個別に扱われます。

2.10 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 (CA850/CX) は、ANSI 規格に沿って関数呼び出し情報をスタックに積んでいます。この関数呼び出し情報 (以降、コール・スタック情報と呼びます) を解析することで、関数の呼び出しの深さ、呼び出し元位置、および引数などを知ることができます。

2.10.1 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—214 コール・スタック情報の表示 (コール・スタック パネル)













ここでは、次の操作方法について説明します。

- (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。

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。
	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）。
	このパネル上の値を 16 進数で表示します。
	このパネル上の値を 10 進数で表示します。
	このパネル上の値を 8 進数で表示します。
	このパネル上の値を 2 進数で表示します。
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。
	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。
	このパネル上の文字列変数を EUC-JP コードで表示します。
	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。

(2) ソース行へジャンプする

行をダブルクリックすることにより、選択行が示す関数呼び出し元のソース行にカーレットを移動した状態で **エディタ パネル** がオープンします（すでにオープンしている場合は、エディタ パネルにジャンプ）。

備考 コンテキスト・メニューの [逆アセンブルへジャンプ] を選択することにより、現在選択している行が示す関数呼び出し元のアドレスにカー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—215 コール・スタック情報保存の際の出カイメージ

深さ	コール・スタック
0	コール・スタック情報
1	コール・スタック情報
:	:

2.11 実行履歴の収集【IECUBE】【IECUBE2】【シミュレータ】

この節では、プログラムの実行履歴の収集方法について説明します。

一般的に、プログラムの実行履歴をトレースと呼び、以降の記述で使用します。プログラムが暴走した場合、暴走後のメモリ内容やスタック情報などから原因を探ることは非常に困難ですが、収集したトレース・データの内容を解析することにより、暴走するまでの過程を直接探ることができ、プログラムの潜在的バグを発見するために有効です。

注意 1. 【IECUBE】

トレース機能／リアルタイム表示更新機能（RRM 機能）／カバレッジ機能は一部排他使用の機能です。

2. 【IECUBE2】

トレース機能／タイマ機能（Run-Break タイマを除く）／カバレッジ機能は一部排他使用の機能です。

3. 【MINICUBE】【MINICUBE2】【E1】【E20】【EZ Emulator】

トレース機能はサポートしていません。

2.11.1 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) 【IECUBE】の場合

(2) 【IECUBE2】の場合

(3) 【シミュレータ】の場合

(1) 【IECUBE】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—216 [トレース] カテゴリ【IECUBE】

☐ トレース	
トレース・データの用途	トレース
トレース・データの選択	分岐命令とアクセス・アドレス/データ値
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・タイム・タグの分周率	1/1 (20ns/1.4min)
トレース・メモリ・サイズ[フレーム]	8K
トレースを補完する	はい

(a) [トレース・データの用途]

トレース機能／リアルタイム表示更新機能（RRM 機能）／カバレッジ機能は一部排他使用の機能です。

このため、このプロパティで、上記のどの機能を優先的に使用するかを指定します。ここでは、次のドロップダウン・リストより、[トレース] を選択してください。

トレース	<p>トレース機能を優先的に使用します。</p> <ul style="list-style-type: none"> - リアルタイム表示更新機能 (RRM 機能) 使用不可となります。 - カバレッジ機能 使用不可となります。
RRM	<p>リアルタイム表示更新機能 (RRM 機能) を優先的に使用します (デフォルト)。</p> <ul style="list-style-type: none"> - トレース機能 内部的に分岐命令のみトレースします ([トレース・データの選択] プロパティにおいて [分岐命令] が指定されている状態と同等の動作)。 また、トレース関連のイベントが無効状態となります。 - カバレッジ機能 使用不可となります。
カバレッジ	<p>カバレッジ機能を優先的に使用します。</p> <ul style="list-style-type: none"> - トレース機能 内部的に命令のみトレースします ([トレース・データの選択] プロパティにおいて [全命令] が指定されている状態と同等の動作)。 また、トレース関連のイベントが無効状態となります。 - リアルタイム表示更新機能 (RRM 機能) 使用不可となります。

(b) [トレース・データの選択]

このプロパティは、[トレース・データの用途] プロパティで [トレース] を選択している場合のみ表示されます。

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

分岐命令
全命令
アクセス・アドレス/データ値
分岐命令とアクセス・アドレス/データ値 (デフォルト)
全命令とアクセス・アドレス/データ値
アクセス命令とアクセス・アドレス/データ値
分岐/アクセス命令とアクセス・アドレス/データ値

備考 1. 収集するトレース・データの種類の意味は次のとおりです。

分岐命令	分岐元と分岐先の命令の PC 値
全命令	全命令の PC 値
アクセス・アドレス/データ値	アクセス・アドレスとアクセス・データ
アクセス命令	アクセスを発生させた命令の PC 値

2. DMA の開始点、および終了点はトレース条件に関わらず必ずトレースされます。

(c) [トレースの優先度]

このプロパティは、[トレース・データの用途] プロパティで [トレース] を選択している場合のみ表示されます。


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・リストにより指定します。

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。

(d) [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) するかどうかをドロップダウン・リストにより指定します。

クリアする場合は [はい] を指定してください (デフォルト)。

備考 **トレースパネル【IECUBE】【IECUBE2】【シミュレータ】** のツールバーの  ボタンをクリックすることにより、トレース・メモリを強制的にクリアすることができます。

(e) [トレース・メモリを使い切った後の動作]

収集したトレース・データでトレース・メモリがいっぱいになった際の動作を、次のドロップダウン・リストにより指定します。

トレース・メモリを上書きし実行を続ける	トレース・メモリがいっぱいになると、古いトレース・データに上書きを続けます (デフォルト)。 [実行前にトレース・メモリをクリアする] プロパティで [はい] を指定している場合は、再実行時、トレース・データをクリアしたのちトレース・データの書き込みを行います。
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)。 [実行前にトレース・メモリをクリアする] プロパティで [いいえ] を指定している場合は、再び実行してもトレース・データの書き込みは行いません。
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。 [実行前にトレース・メモリをクリアする] プロパティで [いいえ] を指定している場合は、再び実行してもプログラムは実行せずに停止します。

(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【IECUBE】【IECUBE2】【シミュレータ】** 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・リストにより指定します。

ドロップダウン・リストには、次の分周率が表示されます (“()” 内は分解能、およびタイマ・カウンタが 32 ビットで外部クロックが 50MHz の場合の最大測定時間を示す)。

1/1(20ns/1.4min) (デフォルト), 1/2(40ns/2.8min), 1/4(80ns/5.7min),
1/8(160ns/11.4min), 1/16(320ns/22.8min), 1/32(640ns/45.6min),
1/64(1280ns/1.5h), 1/128(2560ns/3.0h), 1/256(5120ns/6.0h),

1/512(10240ns/12.2h), 1/1024(20480ns/24.4h), 1/2048(40960ns/48.8h),
 1/4096(81920ns/97.6h)

(g) [トレース・メモリ・サイズ [フレーム]]

トレース・メモリのサイズ (トレース・フレーム数) をドロップダウン・リストにより指定します。

なお、トレース・フレームはトレース・データの一単位を表し、フェッチ/ライト/リードなどで、それぞれ1つのトレース・フレームを使用します。

ドロップダウン・リストには、次のトレース・フレーム数が表示されます。

8K (デフォルト), 32K, 64K, 128K, 256K

(h) [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【IECUBE】【IECUBE2】【シミュレータ】** で表示する際に、補完表示するか否かを指定します。

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補完表示が可能となります。

補完表示を行う場合は [はい] を指定してください (デフォルト)。

なお、このプロパティの指定を変更すると、自動的にトレース・メモリがクリアされます。

(2) **【IECUBE2】** 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—217 [トレース] カテゴリ **【IECUBE2】**

目 トレース	
トレース・データの用途	トレース
トレース・データの選択	分岐命令とアクセス・アドレス/データ値
DMAトレースを行う	はい
DMAトレース・データの選択	アクセス種別/アクセス・アドレス/データ・サイズ/データ値
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・メモリ・サイズ [フレーム]	8K
トレースを補完する	はい

(a) [トレース・データの用途]

トレース機能/タイマ機能 (Run-Break タイマを除く) /カバレッジ機能は一部排他使用の機能です。このため、このプロパティで、上記のどの機能を優先的に使用するかを指定します。ここでは、次のドロップダウン・リストより、[トレース] を選択してください。

トレース	トレース機能を優先的に使用します (デフォルト)。 - タイマ機能 使用不可となります。 また、タイマ関連のイベント (Run-Break タイマを除く) が無効状態となります。 - カバレッジ機能 使用不可となります。
------	---

タイマ	<p>タイマ機能を優先的に使用します。</p> <ul style="list-style-type: none"> - トレース機能 使用不可となります。 また、トレース関連のイベントが無効状態となります。 - カバレッジ機能 使用不可となります。
カバレッジ	<p>カバレッジ機能を優先的に使用します。</p> <ul style="list-style-type: none"> - トレース機能 内部的に分岐命令のみトレースします（[トレース・データの選択] プロパティにおいて [分岐命令] が指定されている状態と同等の動作）。 また、トレース関連のイベントが無効状態となります。 - タイマ機能 使用不可となります。 また、タイマ関連のイベント（Run-Break タイマを除く）が無効状態となります。

(b) [トレース・データの選択]

このプロパティは、[トレース・データの用途] プロパティで [トレース] を選択している場合のみ表示されます。

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

分岐命令
アクセス・アドレス/データ値
分岐命令とアクセス・アドレス/データ値（デフォルト）
アクセス命令とアクセス・アドレス/データ値
分岐/アクセス命令とアクセス・アドレス/データ値

備考 収集するトレース・データの種類の意味は次のとおりです。

分岐命令	分岐元と分岐先の命令の PC 値
アクセス・アドレス/データ値	アクセス・アドレスとアクセス・データ
アクセス命令	アクセスを発生させた命令の PC 値

(c) [DMA トレースを行う]

DMA（Direct Memory Access）によるアクセス・データをトレースするか否かをドロップダウン・リストにより指定します。

DMA トレースを行う場合は [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) [DMA トレース・データの選択]

このプロパティは、[DMA トレースを行う] プロパティで [はい] を選択している場合のみ表示されます。

収集する DMA トレース・データの種類の種類を次のドロップダウン・リストにより指定します。

アクセス種別／アクセス・アドレス／データ・サイズ／データ値
アクセス種別／アクセス・アドレス／データ・サイズ
アクセス種別／チャンネル番号
アクセス種別／チャンネル番号／転送回数
アクセス種別／データ値／チャンネル番号／転送回数

備考 収集する DMA トレース・データの種類の意味は次のとおりです。

アクセス種別	メモリ・アクセスの種類 (リード, ライト, リード/ライト)
アクセス・アドレス	メモリ・アクセスの対象アドレス
データ・サイズ	アクセス・サイズ
データ値	アクセス・データ
チャンネル番号	DMA チャンネル番号
転送回数	DMA 転送回数

(e) [トレースの優先度]

このプロパティは、[トレース・データの用途] プロパティで [トレース] を選択している場合のみ表示されます。


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・リストにより指定します。

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。

(f) [実行前にトレース・メモリをクリアする]

トレース機能を開始する前に、トレース・メモリを一度クリア (初期化) するか否かをドロップダウン・リストにより指定します。

クリアする場合は [はい] を指定してください (デフォルト)。

備考 **トレース パネル** [IECUBE] [IECUBE2] [シミュレータ] のツールバーの  ボタンをクリックすることにより、トレース・メモリを強制的にクリアすることができます。

(g) [トレース・メモリを使い切った後の動作]

収集したトレース・データでトレース・メモリがいっぱいになった際の動作を、次のドロップダウン・リストにより指定します。

トレース・メモリを上書きし実行を続ける	トレース・メモリがいっぱいになると、古いトレース・データを上書きを続けます (デフォルト)。 [実行前にトレース・メモリをクリアする] プロパティで [はい] を指定している場合は、再実行時、トレース・データをクリアしたのちトレース・データの書き込みを行います。
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)。 [実行前にトレース・メモリをクリアする] プロパティで [いいえ] を指定している場合は、再び実行してもトレース・データの書き込みは行いません。
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。 [実行前にトレース・メモリをクリアする] プロパティの設定に関わらず、プログラム実行前にトレース・メモリをクリアします。

(h) [トレース・メモリ・サイズ [フレーム]]

トレース・メモリのサイズ (トレース・フレーム数) をドロップダウン・リストにより指定します。

なお、トレース・フレームはトレース・データの一単位を表し、フェッチ/ライト/リードなどで、それぞれ1つのトレース・フレームを使用します。

ドロップダウン・リストには、次のトレース・フレーム数が表示されます。

8K (デフォルト), 32K, 64K, 128K, 256K

(i) [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** [IECUBE] [IECUBE2] [シミュレータ] で表示する際に、補完表示するか否かを指定します。

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補完表示が可能となります。

補完表示を行う場合は [はい] を指定してください (デフォルト)。

なお、このプロパティの指定を変更すると、自動的にトレース・メモリがクリアされます。

(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—218 【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ドロップダウン・リストにより指定します。

トレース機能を使用する場合は【はい】を指定してください（デフォルトでは【いいえ】が指定されます）。

(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）するか否かをドロップダウン・リストにより指定します。

クリアする場合は【はい】を指定してください（デフォルト）。

備考 [トレースパネル【IECUBE】【IECUBE2】【シミュレータ】](#)のツールバーの  ボタンをクリックすることにより、トレース・メモリを強制的にクリアすることができます。

(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・リストにより指定します。

トレース・メモリを上書きし実行を続ける	トレース・メモリがいっぱいになると、古いトレース・データを上書きを続けます（デフォルト）。 【実行前にトレース・メモリをクリアする】 プロパティで【はい】を指定している場合は、再実行時、トレース・データをクリアしたのちトレース・データの書き込みを行います。
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）。 【実行前にトレース・メモリをクリアする】 プロパティで【いいえ】を指定している場合は、再び実行してもトレース・データの書き込みは行いません。
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。 【実行前にトレース・メモリをクリアする】 プロパティで【いいえ】を指定している場合は、再び実行してもプログラムは実行せずに停止します。

(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ドロップダウン・リストにより指定します。

トレースの時間表示を積算表示にする場合は [はい] を、差分表示にする場合は [いいえ] を指定してください (デフォルト)。

(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・リストにより指定します。

なお、トレース・フレームはトレース・データの一単位を表し、フェッチ/ライト/リードなどで、それぞれ1つのトレース・フレームを使用します。

ドロップダウン・リストには、次のトレース・フレーム数が表示されます。

4K (デフォルト), 8K, 12K, 16K, 20K, 24K, 28K, 32K, 36K, 40K, 44K, 48K, 52K, 56K,
60K, 64K, 128K 192K, 256K, 320K, 384K, 448K, 512K, 576K, 640K, 704K, 768K, 832K,
896K, 960K, 1M, 2M, 3M

(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 [IECUBE] [IECUBE2] [シミュレータ] 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・リストにより指定します。

ドロップダウン・リストには、次の分周率が表示されます。

1/1 (デフォルト), 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, 1/1K, 1/4K, 1/8K,
1/16K, 1/64K, 1/256K, 1/1M

2.11.2 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、「[2.11.6 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 ([2.11.3 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。

2.11.3 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）、[「2.15.6 イベント設定に関する留意事項」](#)も参照してください。

2. トレース開始／終了イベントとして設定可能なイベントの種別（実行系／アクセス系）は、デバッグ・ツールにより異なります。

3. 【IECUBE】【IECUBE2】

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・データの用途」プロパティの指定を、「トレース」以外に変更した場合、現在設定されているトレース関連のイベントはすべて無効状態となります。

4. 【IECUBE】【IECUBE2】

トレース動作中は、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。ただし、[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」プロパティを「はい」に指定することにより、トレーサを一瞬停止してこの操作を行うことができます。

5. 【シミュレータ】

トレース動作中は、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・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。

(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／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カー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。

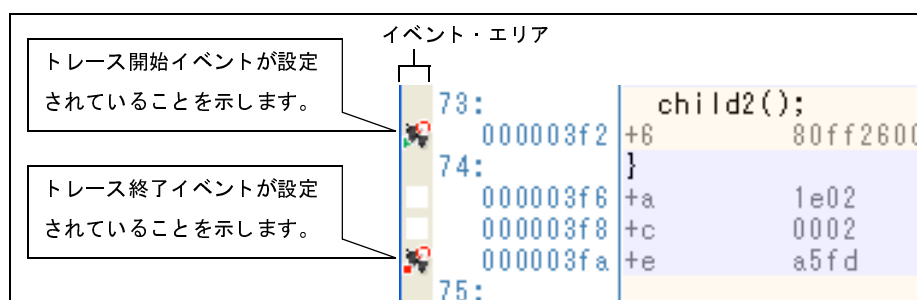
トレース開始イベント／トレース終了イベントは、キャレット位置の行に対応する先頭アドレスの命令に設定されます。

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—17 トレース開始イベント／トレース終了イベント・マーク

種別	イベント・マーク
トレース開始	
トレース終了	

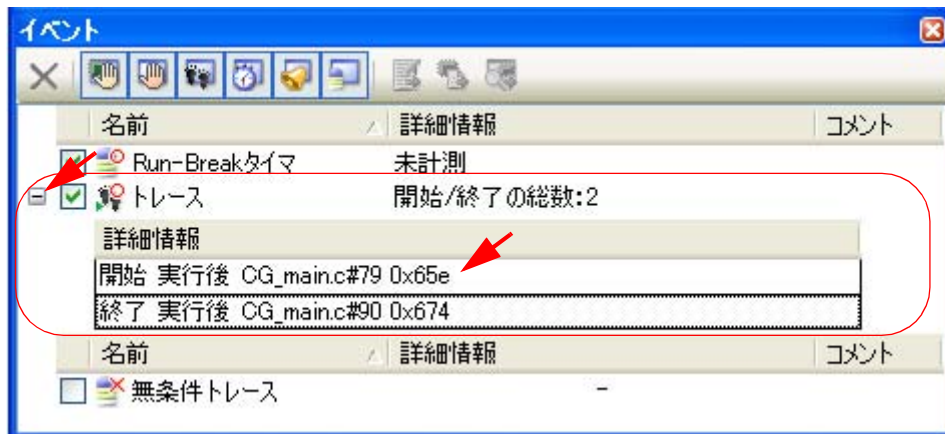
図 2—219 トレース開始イベント／トレース終了イベントの設定例（逆アセンブルパネルの場合）

**(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（「[2.15 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—220 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



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スは動作しません）。

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。

3. イベントの設定状態によりイベント・マークは異なります（「2.15.1 設定状態（有効/無効）を変更する」参照）。

また、すでにイベントが設定されている箇所で、新たにイベントを設定した場合は、複数のイベントが設定されていることを示すイベント・マーク（)が表示されます。

4. 【シミュレータ】

トレース開始イベント/トレース終了イベントのいずれかが有効状態で設定されると、プロパティパネルの【デバッグ・ツール設定】タブ上の【トレース】カテゴリ内【トレース機能を使用する】プロパティの指定を自動的に【はい】に変更し、トレース機能が有効化されます。

(2) プログラムを実行する


プログラムを実行します（「2.7 プログラムの実行」参照）。

トレース開始イベント/トレース終了イベントに設定されている条件が成立した時点で、トレース・データの収集を開始/終了します。

なお、収集したトレース・データの確認方法についての詳細は、「2.11.6 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（「2.15.4 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／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）。

2.11.4 条件を満たしたときのみの実行履歴を収集する

ある条件を満たした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

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）、[「2.15.6 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。

ただし、[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」](#) プロパティを [「はい」](#) に指定することにより、トレースを一瞬停止してこの操作を行うことができます（[「シミュレータ」](#)を除く）。

備考 1. [【IECUBE2】](#)

DMA によるアクセスも対象となります。

2. [【シミュレータ】](#)

ポイント・トレース・イベントのいずれかが [有効状態](#) で設定されると、[プロパティパネルの「デバッグ・ツール設定」タブ](#)上の [「トレース」](#) カテゴリ内 [「トレース機能を使用する」](#) プロパティの指定を自動的に [「はい」](#) に変更し、トレース機能が有効化されます。

(a) [エディタパネル／逆アセンブルパネル](#)上の変数 I/O レジスタへのアクセスの場合

操作は、[ソース・テキスト／逆アセンブル・テキスト](#)を表示している [エディタパネル／逆アセンブルパネル](#)上で行います。

対象となる変数、または I/O レジスタを選択したのち、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（「[2.9.6 ウォッチ式を表示/変更する](#)」参照）。

ただし、対象となるウォッチ式は、グローバル変数/関数内スタティック変数/ファイル内スタティック変数/I/Oレジスタのみとなります。

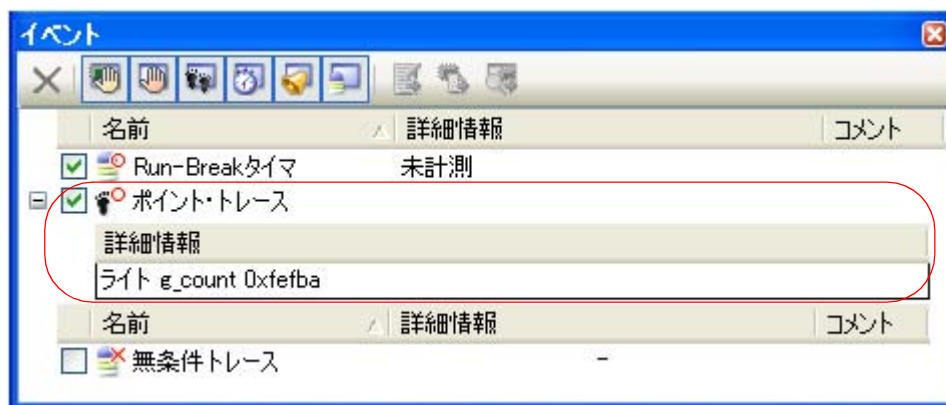
アクセス種別	操作方法
リード	[トレース出力] → [値をトレースに記録 (読み込み時)] を選択します。
ライト	[トレース出力] → [値をトレースに記録 (書き込み時)] を選択します。
リード/ライト	[トレース出力] → [値をトレースに記録 (読み書き時)] を選択します。

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きみなされ、[イベントパネル](#)で管理されます（「[2.15 イベントの管理](#)」参照）。

図 2-221 イベントパネルのポイント・トレース・イベントの設定例

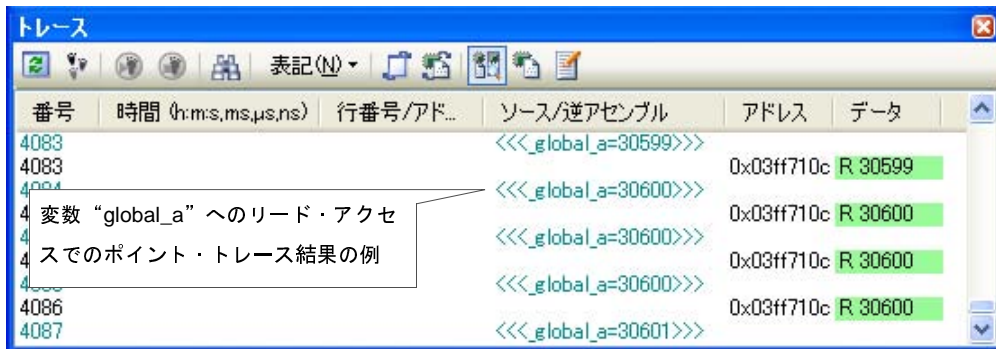


(2) プログラムを実行する


プログラムを実行します（「2.7 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、「2.11.6 実行履歴を表示する」を参照してください。

図 2—222 ポイント・トレース・イベントの結果表示例（シミュレータを使用した場合）



(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  ボタンをクリックします（「2.15 イベントの管理」参照）。


2.11.5 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。

注意 【IECUBE】 【IECUBE2】


この機能は、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・データの用途] プロパティに [トレース] が指定されている場合のみ有効となります。

(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【IECUBE】【IECUBE2】【シミュレータ】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【IECUBE】【IECUBE2】【シミュレータ】のツールバーの  ボ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。

2.11.6 実行履歴を表示する

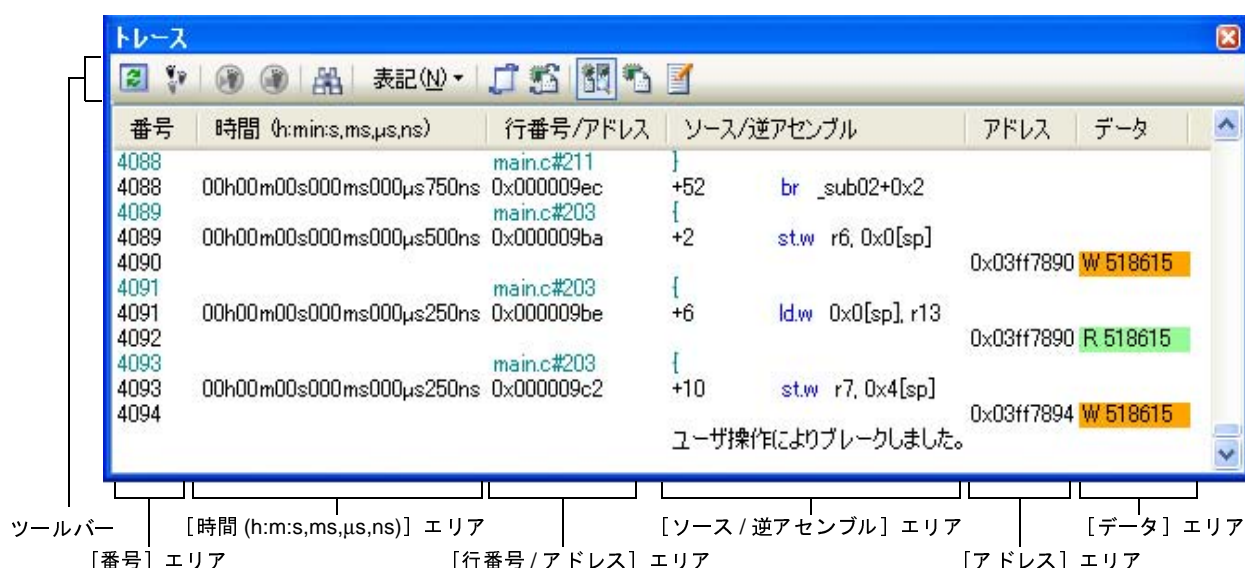
収集したトレース・データの表示は、次のトレースパネル【IECUBE】【IECUBE2】【シミュレータ】で行います。

[表示]メニュー→[トレース]を選択してください。

トレース・データは、デフォルトで逆アセンブル・テキストとソース・テキストを混合して表示しますが、表示モードを選択することにより、そのどちらか一方のみを表示させることもできます。

なお、各エリアの見方、および機能についての詳細は、トレースパネル【IECUBE】【IECUBE2】【シミュレータ】の項を参照してください。

図 2—223 トレース・データの表示（トレースパネル）



ここでは、次の操作方法について説明します。

- (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。

表 2—18 トレース パネルの表示モード

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



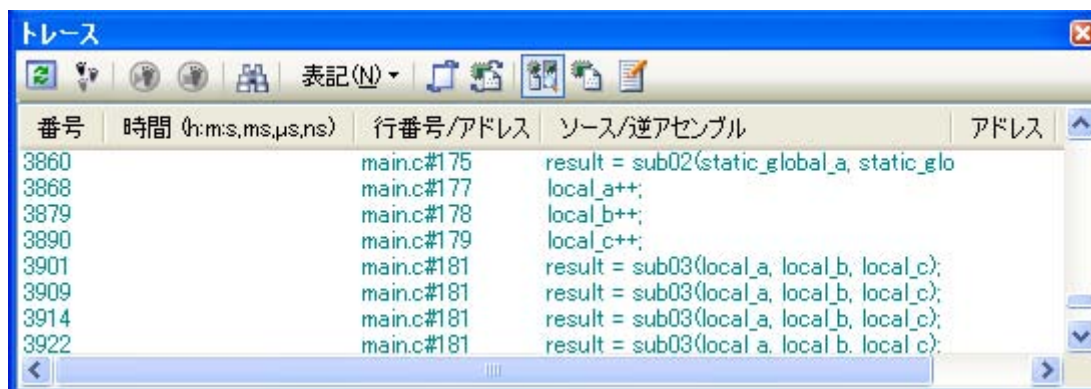
ボタン	表示モード	表示内容
	逆アセンブル 表示モード	命令（逆アセンブル）／ラベル名／ポイント・トレース結果／ブレーク要因 を表示します。
	ソース表示 モード	ソース・テキスト（対応するソース行）／ブレーク要因を表示します。 ただし、デバッグ情報が存在しない箇所を実行した場合は、“デバッグ情報の ない区間の実行”と表示します。





図 2—224 ソース表示モードの例（トレース パネル）



(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する次のボタンを表示します。
	このパネル上の値を 16 進数で表示します（デフォルト）。
	このパネル上の値を 10 進数で表示します。
	このパネル上の値を 8 進数で表示します。
	このパネル上の値を 2 進数で表示します。

(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 [ソースヘジャンプ] / [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）。

2.11.7 トレース・メモリをクリアする

収集したトレース・データの内容をクリアするには、ツールバーの  ボタンをクリックします。
ただし、トレー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] を指定している場合は、プログラムの実行ごとにトレース・メモリがクリアされます。

2.11.8 トレース・データを検索する

収集したトレース・データの検索は、ツールバーの  ボタンをクリックすることによりオープンする **トレース検索 ダイアログ** [IECUBE] [IECUBE2] [シミュレータ] により行います（プログラム実行中は無効）。

このダイアログにおいて、次の操作を行ってください。

なお、タブ選択エリア上のタブを選択することにより、命令レベル、またはソース・レベルでトレース・データを検索することができます。

ただし、命令レベルでトレース・データの検索を行う場合は、**トレースパネル** [IECUBE] [IECUBE2] [シミュレータ] を **混合表示モード**、または **逆アセンブル表示モード** で表示している必要があります。

また、ソース・レベルで検索を行う場合は、**混合表示モード**、または **ソース表示モード** で表示している必要があります。

図 2—225 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。

図 2—226 命令レベルでのトレース・データの検索



(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【IECUBE】【IECUBE2】【シミュレータ】](#) 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【IECUBE】](#)[【IECUBE2】](#)[【シミュレータ】](#)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【IECUBE】](#)[【IECUBE2】](#)[【シミュレータ】](#)上で選択状態にします。

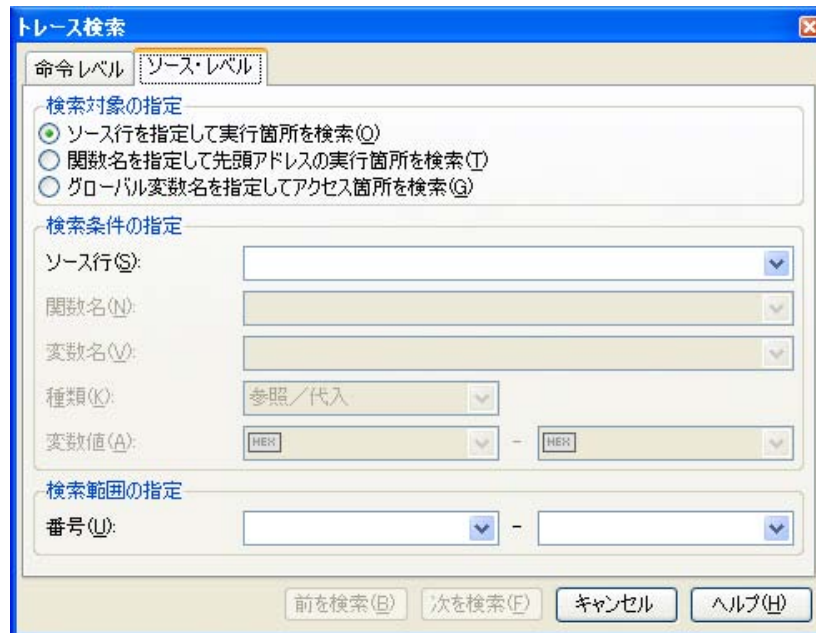
[次を検索] ボタンをクリックすると、番号の大きい方向に検索を行い、検索結果箇所を[トレースパネル【IECUBE】](#)[【IECUBE2】](#)[【シミュレータ】](#)上で選択状態にします。

(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。

図 2—227 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【IECUBE】【IECUBE2】【シミュレータ】** 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【IECUBE】【IECUBE2】【シミュレータ】** 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【IECUBE】【IECUBE2】【シミュレータ】](#) 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【IECUBE】【IECUBE2】【シミュレータ】](#) 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【IECUBE】【IECUBE2】【シミュレータ】](#) 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【IECUBE】【IECUBE2】【シミュレータ】](#) 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【IECUBE】【IECUBE2】【シミュレータ】](#) 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

検索するトレース・データの範囲を、[トレースパネル【IECUBE】【IECUBE2】【シミュレータ】](#)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【IECUBE】【IECUBE2】【シミュレータ】](#)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【IECUBE】【IECUBE2】【シミュレータ】](#)上で選択状態にします。

2.11.9 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—228 実行履歴の保存（データ保存 ダイアログ）



(1) [ファイル名] の指定

保存するファイル名を指定します。

テキスト・ボックスに直接入力するか（最大指定文字数：259 文字）、またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。

また、[...] ボタンをクリックすることでオープンするデータ保存ファイルを選択ダイアログにより、ファイルを選択することもできます。

(2) [ファイルの種類] の指定

保存するファイルの形式を次のドロップダウン・リストにより選択します。

選択できるファイルの形式は次のとおりです。

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

注 各データを“,”で区切り保存します。

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

(3) [保存範囲 番号] の指定

ファイルに保存する範囲を“開始トレース番号”と“終了トレース番号”で指定します。

それぞれのテキスト・ボックスに10進数の数値を直接入力するか、またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10個）。

なお、すべてのトレース・データを保存する場合は、左側のドロップダウン・リストにおいて、[すべてのトレース・データ]を選択してください（右側のテキスト・ボックスは無効）。

パネル上で範囲選択している場合は、デフォルトでその選択範囲がテキスト・ボックスに指定されます。範囲選択していない場合は、現在のパネルの表示範囲が指定されます。

(4) [保存] ボタンのクリック

指定したファイルに、指定した形式でトレース・データを保存します。

図 2—229 トレース・データ保存の際の出カイメージ

番号	時間	行番号 / アドレス	ソース / 逆アセンブル	アドレス	データ
番号	時間	行番号 / アドレス	ソース / 逆アセンブル	アドレス	データ
:	:	:	:	:	:

2.12 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。

2.12.1 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. 【MINICUBE2】

ステップ実行中の Run-Break 時間は計測できません。

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。Run-Break タイマ・イベントは常に有効状態に設定されています（変更不可）。

(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・ウィ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）。

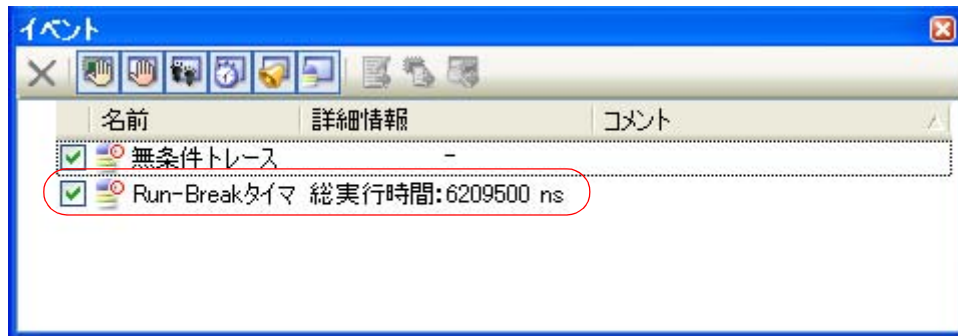
図 2—230 Run-Break タイマ・イベントの測定結果例（ステータスバー）



(2) イベントパネルでの確認

プログラムの実行停止後、【表示】メニュー→【イベント】の選択によりオープンするイベントパネル上において、Run-Break タイマ・イベントとして計測結果を表示します。

図 2—231 Run-Break タイマ・イベントの測定結果例（イベントパネル）



2. 12. 2 任意区間の実行時間を計測する

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. 【V850E1】【V850ES】：【MINICUBE2】【E1】【E20】【EZ Emulator】

タイマ計測イベントはサポートしていません。

2. 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. 15. 6 イベント設定に関する留意事項」](#)も参照してください。

3. 【IECUBE2】

トレース機能／タイマ機能（Run-Break タイマを除く）／カバレッジ機能は一部排他使用の機能です。このため、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「トレース」カテゴリ内「トレース・データの用途」プロパティ](#)において、[「タイマ」](#)が指定されている必要があります。

4. 【シミュレータ】

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。

(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caret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]
タイマ終了	[タイマ設定] → [実行時にタイマ終了]

注意 【シミュレータ】

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、caret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—19 タイマ開始イベント/タイマ終了のイベント・マーク



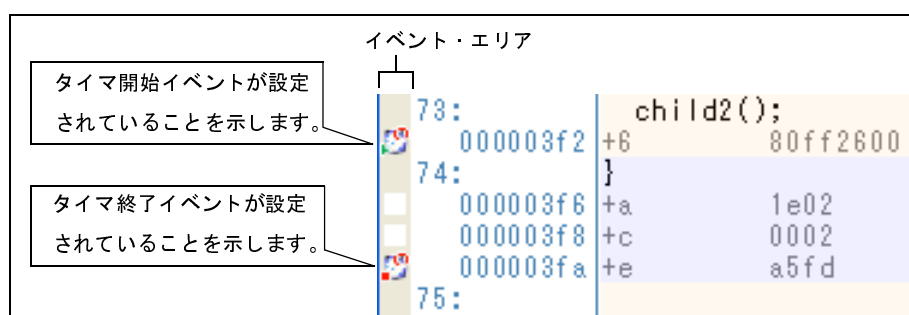
種別	イベント・マーク
タイマ開始	
タイマ終了	

図 2—232 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）

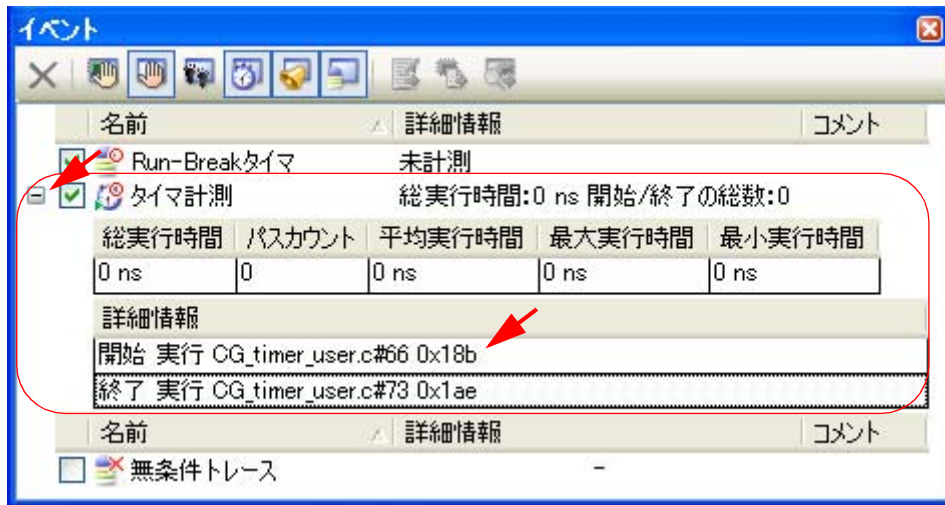


(b) アクセス系イベントの場合


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（[2.15 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—233 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



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（「2.15.1 設定状態（有効／無効）を変更する」参照）。

また、すでにイベントが設定されている箇所で、新たにイベントを設定した場合は、複数のイベントが設定されていることを示すイベント・マーク（)が表示されます。

(2) プログラムを実行する

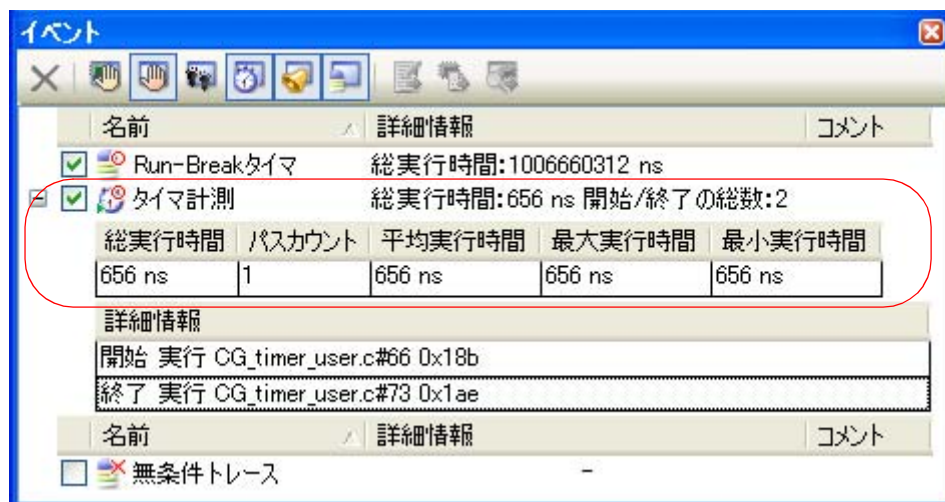
プログラムを実行します（「2.7 プログラムの実行」参照）。

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条件が成立した時点で、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、イベントパネルにおいて、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—234 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



(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（[2.15.4 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/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）。

2.12.3 測定可能時間の範囲

Run-Break タイマ・イベント（[2.12.1 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（[2.12.2 任意区間の実行時間を計測する](#)参照）によるタイマ計測の測定可能時間の範囲は次のとおりです。なお、測定可能な最大時間を越えた場合は、タイマ・オーバ・ブレイクを発生し、プログラムの実行を停止します。

表 2—20 測定可能時間の範囲【V850E1】【V850ES】

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	
	最小	最大	最小	最大
IECUBE	最小	20 ナノ秒	最小	20 ナノ秒
	最大	約 195 時間 12 分 (4K 分周時) オーバフロー検出あり	最大	約 195 時間 12 分 (4K 分周時) 最大パスカウント：65535 回 オーバフロー検出あり
MINICUBE E1/E20(JTAG)	最小	200 ナノ秒	—	
	最大	約 7 分 (TCK/DCK = 10 MHz 時) オーバフロー検出あり		
MINICUBE2 E1/E20(Serial) EZ Emulator	最小	100 マイクロ秒	—	
	最大	約 119 時間 18 分 オーバフロー検出あり		
シミュレータ	タイマ/トレース用クロック周波数に依存		タイマ/トレース用クロック周波数に依存	

表 2—21 測定可能時間の範囲【V850E2】

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	
	最小	最大	最小	最大
IECUBE2	最小	200 ナノ秒	最小	5 ナノ秒
	最大	約 7 分 (TCK/DCK = 10 MHz 時) オーバフロー検出あり	最大	約 24 時間 (4K 分周時) パスカウント・カウンタ：16 ビット オーバフロー検出あり

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
MINICUBE E1/E20(JTAG)	最小	200 ナノ秒	タイマ/トレース用クロック周波数に依存 パスカウント・カウンタ：32 ビット オーバーフロー検出あり
	最大	約 7 分 (TCK/DCK = 10 MHz 時) オーバーフロー検出あり	
E1/E20(LPD)	最小	100 ナノ秒 (CPU クロック = 80 MHz) 125 ナノ秒 (CPU クロック = 64 MHz) 166 ナノ秒 (CPU クロック = 48 MHz)	タイマ/トレース用クロック周波数に依存 パスカウント・カウンタ：32 ビット オーバーフロー検出あり
	最大	約 3.5 分 (CPU クロック = 80 MHz) 約 4.5 分 (CPU クロック = 64 MHz) 約 6 分 (CPU クロック = 48 MHz) オーバーフロー検出あり	
MINICUBE2 E1/E20(Serial)	最小	100 マイクロ秒	タイマ/トレース用クロック周波数に依存 パスカウント・カウンタ：32 ビット オーバーフロー検出あり
	最大	約 119 時間 18 分 オーバーフロー検出あり	
シミュレータ	タイマ/トレース用クロック周波数に依存		タイマ/トレース用クロック周波数に依存

2.13 カバレッジの測定【IECUBE】【IECUBE2】【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【シミュレータ】を行います。

カバレッジ測定の対象となる領域は次のとおりです。

表 2—22 カバレッジ測定の対象領域

デバッグ・ツール		対象領域
V850E1 V850ES	IECUBE シミュレータ	次の両領域 - アドレス 0x000000 ~ 0x0FFFFFF の 1M バイト空間（固定の測定領域） - 内蔵 ROM 領域以外の任意の 1M バイト空間（「2.13.1 カバレッジ測定の設定をする」参照）
V850E2 (シングルコアのみ)	IECUBE2 シミュレータ	内蔵 ROM の 4M バイト 次の両領域 - アドレス 0x000000 ~ 0x0FFFFFF の 1M バイト空間（固定の測定領域） - 内蔵 ROM 領域以外の任意の 1M バイト空間（「2.13.1 カバレッジ測定の設定をする」参照）

注意 1. 【IECUBE】

カバレッジ測定は、IECUBE がカバレッジ・ボード搭載品の場合のみサポートされる機能です。

また、トレース機能／リアルタイム表示更新機能（RRM 機能）／カバレッジ機能は一部排他使用の機能です。

2. 【IECUBE2】

トレース機能／タイマ機能（Run-Break タイマを除く）／カバレッジ機能は一部排他使用の機能です。

3. 【MINICUBE】【MINICUBE2】【E1】【E20】【EZ Emulator】

カバレッジ機能はサポートしていません。

備考 C0 カバレッジ：命令網羅率（ステートメント・カバレッジ）

たとえば、コード内のすべての命令（ステートメント）を少なくとも 1 回は実行した場合、C0 = 100 % となります。

2.13.1 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。

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) 【IECUBE】の場合
- (2) 【IECUBE2】の場合
- (3) 【シミュレータ】の場合

(1) 【IECUBE】の場合

注意 カバレッジ測定は、IECUBE がカバレッジ・ボード搭載品の場合のみサポートされる機能です。

したがって、カバレッジ・ボード非搭載品の場合、デバッグ・ツールと接続後、以降の設定は無効となります。

トレース機能／リアルタイム表示更新機能（RRM 機能）／カバレッジ機能は一部排他使用の機能です。このため、まず、これらの機能のうち、カバレッジ機能を優先して使用する設定を行います。

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】カテゴリ内【トレース・データの用途】プロパティにおいて、【カバレッジ】を指定してください。

なお、このプロパティにおいて、【カバレッジ】以外を指定している場合は、以降の設定は無効となります。

図 2—235 【トレース】カテゴリ（カバレッジ機能優先の指定）

トレース	
トレース・データの用途	カバレッジ
実行前にトレース・メモリをクリアする	はい
トレース・メモリを保持する	トレース・メモリを上書きし実行を続ける

次に、コード・カバレッジ測定に関する設定を、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—236 【カバレッジ】カテゴリ【IECUBE】

カバレッジ	
カバレッジ結果を再利用する	いいえ
カバレッジ測定領域(1Mバイト単位)	HEX 3F00000

(a) 【カバレッジ結果を再利用する】

デバッグ・ツールと切断時に、現在取得しているコード・カバレッジ測定結果を自動保存し、次回デバッグ・ツールと接続した際に、保存した測定結果の内容を再現するか否かをドロップダウン・リストにより指定します。

前回取得したコード・カバレッジ測定結果の内容を再現する場合は、【はい】を選択してください（デフォルトでは【いいえ】が指定されます）。

なお、測定結果を保存するファイル（raw.csv）は、現在ダウンロードしているロード・モジュール・ファイルが存在するフォルダに作成されます。

(b) 【カバレッジ測定領域(1Mバイト単位)】

コード・カバレッジ測定の対象領域を指定します（「表 2—22 カバレッジ測定の対象領域」参照）。

アドレス 0x100000 ~ 0x3FFFFFFF 内の任意の 1 M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（デフォルトでは、内蔵 RAM の開始アドレスが指定されます）。

(2) 【IECUBE2】の場合

トレース機能／タイマ機能（Run-Break タイマを除く）／カバレッジ機能は一部排他使用の機能です。このため、まず、これらの機能のうち、カバレッジ機能を優先して使用する設定を行います。

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】カテゴリ内【トレース・データの用途】プロパティにおいて、【カバレッジ】を指定してください。

なお、このプロパティにおいて、【カバレッジ】以外を指定している場合は、以降の設定は無効となります。

図 2—237 【トレース】カテゴリ（カバレッジ機能優先の指定）

トレース	
トレース・データの用途	カバレッジ
実行前にトレース・メモリをクリアする	はい
トレース・メモリを保持する	トレース・メモリを上書きし実行を続ける

次に、コード・カバレッジ測定に関する設定を、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—238 【カバレッジ】カテゴリ【IECUBE2】

カバレッジ	
カバレッジ結果を再利用する	いいえ

(a) 【カバレッジ結果を再利用する】

デバッグ・ツールと切断時に、現在取得しているコード・カバレッジ測定結果を自動保存し、次回デバッグ・ツールと接続した際に、保存した測定結果の内容を再現するか否かをドロップダウン・リストにより指定します。

前回取得したコード・カバレッジ測定結果の内容を再現する場合は、【はい】を選択してください（デフォルトでは【いいえ】が指定されます）。

なお、測定結果を保存するファイル（raw.csr.cv）は、現在ダウンロードしているロード・モジュール・ファイルが存在するフォルダに作成されます。

(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【カバレッジ】カテゴリ内で行います。

図 2—239 【カバレッジ】カテゴリ【シミュレータ】

カバレッジ	
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	いいえ
カバレッジ測定領域(1 Mバイト単位)	HEX 3F00000

(a) [カバレッジ機能を使用する]

カバレッジ機能を使用するか否かをドロップダウン・リストにより指定します。

カバレッジ機能を使用する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

(b) [カバレッジ結果を再利用する]

このプロパティは、[カバレッジ機能を使用する] プロパティにおいて [はい] を指定した場合のみ表示されます。

デバッグ・ツールと切断時に、現在取得しているコード・カバレッジ測定結果を自動保存し、次回デバッグ・ツールと接続した際に、保存した測定結果の内容を再現するか否かをドロップダウン・リストにより指定します。

前回取得したコード・カバレッジ測定結果の内容を再現する場合は、[はい] を選択してください（デフォルトでは [いいえ] が指定されます）。

(c) [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] を指定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（「表 2—22 カバレッジ測定の対象領域」参照）。

カバレッジ測定を行う内蔵 ROM 領域以外の任意の 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（デフォルトでは、内蔵 RAM の開始アドレスが指定されます）。

2.13.2 カバレッジ測定結果を表示する

プログラムの実行が開始すると自動的にカバレッジ測定が開始し、実行停止とともにカバレッジ測定も終了します。

(1) コード・カバレッジ率

(a) ソース行／逆アセンブル行に対するコード・カバレッジ率の表示

対象となるプログラムを表示している **エディタ パネル／逆アセンブル パネル**で行われます。

各パネルでは、表 2—23 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—24 のように色分け表示されます。

ただし、デバッグ・ツールと切断時、またはプログラム実行中は、結果の表示を行いません。

なお、取得したコード・カバレッジ測定結果は、**エディタ パネル／逆アセンブル パネル**上のコンテキスト・メニューの [カバレッジ情報のクリア] を選択することにより、すべてリセットすることができます（各パネル上の色分け表示もクリアされます）。

表 2—23 ソース行／逆アセンブル行に対するコード・カバレッジ率の計算方法

パネル	計算方法
エディタ パネル	“ソース行と対応するアドレス範囲内で実行されたバイト数” ÷ “ソース行と対応するアドレス範囲内の総バイト数”
逆アセンブル パネル	“逆アセンブル結果行と対応するアドレス範囲内で実行されたバイト数” ÷ “逆アセンブル結果行と対応するアドレス範囲内の総バイト数”

表 2—24 コード・カバレッジ測定結果の表示色（デフォルト）

コード・カバレッジ率	背景色
100 %	ソース・テキスト／逆アセンブル結果
1 ~ 99 %	ソース・テキスト／逆アセンブル結果
0 % (未実行)	ソース・テキスト／逆アセンブル結果

- 備考 1. 各パネルにおけるコード・カバレッジ測定結果の表示更新は、プログラム停止ごとに自動的に行われます。
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（「表 2—22 カバレッジ測定の対象領域」参照）の行に対しては行われません。
4. ダウンロードしているロード・モジュールの更新日時より、現在オープンしているソース・ファイルの更新日時が新しい場合、エディタ パネルではコード・カバレッジ測定結果の表示は行われません。

図 2—240 コード・カバレッジ測定結果の表示例（エディタ パネル）

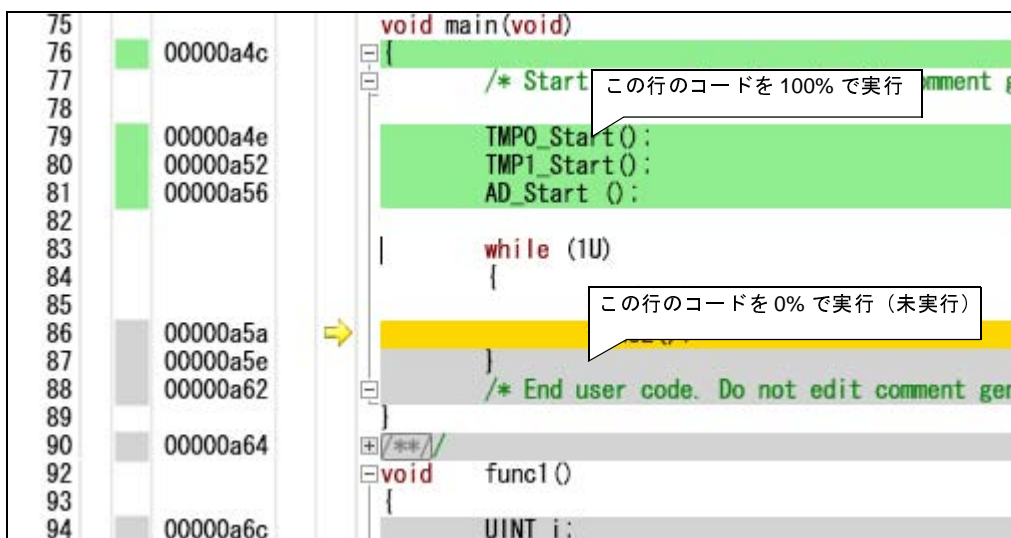


図 2—241 コード・カバレッジ測定結果の表示例（逆アセンブルパネル）

00000a4c	main:	d50d	br	main+0x1a
00000a4e		bfff32fd		_Start, lp
00000a52		bfff02fe	jarl	_imp1_Start, lp
00000a56		bfff46fc	jarl	_AD_Start, lp
00000a5a		80ff1200		lp
00000a5e		80ff3a00		lp
00000a62		c5fd	br	main+0xe
00000a64		1e02	callt	0x1e
00000a66		0002	callt	0x0
00000a68		b5f5	br	_main+0x2
00000a6a		0000	nop	
00000a6c	_func1:	a50d	br	_func1+0x14
00000a6e		00e8	mov	r0, r29
00000a70		80ff1400	jarl	func1a, lp

(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。

(2) データ・カバレッジ率【シミュレータ】

各変数に対するデータ・カバレッジ率は、解析ツールの変数パネル内 [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。

2.14 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。

2.14.1 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、ソフトウェア処理により 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. 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）、[「2.15.6 イベント設定に関する留意事項」](#)も参照してください。

2. ステップ実行中（ /  / ）、またはブレーク関連のイベントを無視した実行中（）にアクション・イベントは発生しません。

3. 【シミュレータ】

プロパティパネルの [「デバッグ・ツール設定」](#) タブの [ブレーク] カテゴリ内 [停止時にブレーク位置の命令を実行] プロパティを [はい] に指定している場合、設定したアクション・イベントはすべてブレーク・イベントとして動作します（Printf イベントは発生しません）。

(1) Printf イベントを設定する

エディタパネル／逆アセンブルパネル上で、printf コマンドを実行させたい箇所に Printf イベントを設定します。

エディタパネル／逆アセンブルパネルにおいて、Printf イベントを設定したい行／アドレス^注にキャレットを移動したのち、コンテキスト・メニューの [アクション・イベントの登録...] を選択すると、次の [アクション・イベント ダイアログ](#) がオープンします。

このダイアログにおいて、次の操作を行ってください。

注 アクション・イベントは、アドレス表示がない行に設定することはできません。

図 2—242 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）

**(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—22 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（[2.15 イベントの管理](#) 参照）。

(2) プログラムを実行する

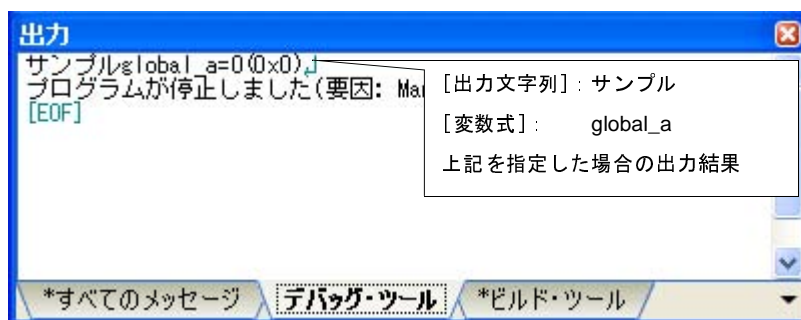
プログラムを実行します（[2.7 プログラムの実行](#) 参照）。

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出力されます（[図 A—61 Printf イベントの出力結果フォーマット](#) 参照）。

図 2—243 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。

2.15 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。

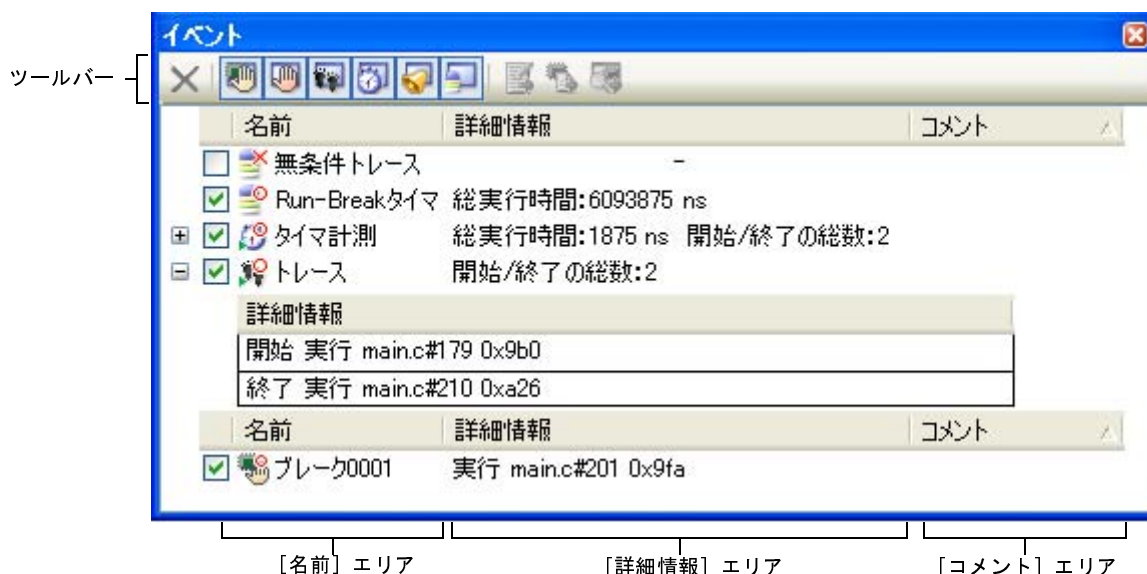
イベントは、一括して次の**イベントパネル**で管理されます。

[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。

図 2—244 設定したイベントの表示（イベントパネル）



備考 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。

- ブレークポイントを設定する
- ブレーク・イベント（実行系）を設定する
- ブレーク・イベント（アクセス系）を設定する
- トレース・イベントを設定する
- ポイント・トレース・イベントを設定する
- タイマ計測イベントを設定する

2.15.1 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—245 イベント名のチェック・ボックス

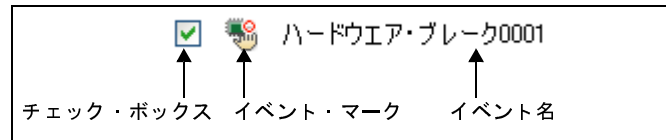


表 2—25 イベントの設定状態

<input checked="" type="checkbox"/>	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。
<input type="checkbox"/>	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。
<input type="checkbox"/>	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。
- 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。
 - 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。
 - 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。

2.15.2 特定のイベント種別のみ表示する

ツールバーの次のボタンをクリックすることで、特定のイベント種別のみを表示することができます。

	ハードウェア・ブレイク関連のイベントを表示します。
 【シミュレータ】以外	ソフトウェア・ブレイク関連のイベントを表示します。
 【IECUBE】【IECUBE2】 【シミュレータ】	トレース関連のイベントを表示します。

	タイマ関連のイベントを表示します ^注 。
	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（無条件トレース /Run-Break タイマ）を表示します。




注 【V850E1】【V850ES】：【MINICUBE2】【E1】【E20】【EZ Emulator】

このボタンは無効となります。


2.15.3 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caret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caret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caretを移動した状態で、 メモリパネル がオープンします。

2.15.4 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。

2. 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、 ボタンをクリックします（ビルトイン・イベントを除く）。

2.15.5 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。

2.15.6 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。

なお、表内の数値の意味は次のとおりです。

$x+y$	“実行後イベント： x 個” + “アクセス・イベント： y 個”
$x+y+z$	“実行後イベント： x 個” + “アクセス・イベント： y 個” + “DMA イベント： l 個”

表 2—26 有効イベント数の制限【V850E1】【V850ES】

イベント種別	デバッグ・ツール					シミュレータ
	IECUBE	MINICUBE E1/E20(JTAG)		MINICUBE2 E1/E20(Serial) EZ Emulator		
		TEU あり	TEU なし	コアイベ ントあり	コアイベ ントなし	
ハードウェア・ブレイク (実行系：実行前)	2注1	2	2注2	2注2	—	64注4
ハードウェア・ブレイク (実行系：実行後)	8	8	0	0	—	
ハードウェア・ブレイク (アクセス系)	6	4	2注2	2注2	—	
ソフトウェア・ブレイク	2000	2000 (フラッシュ用：0～8注3)			—	—
トレース (トレース開始 ／トレース終了)	4+3注5	—	—	—	—	32注5
ポイント・トレース	8+6注6	—	—	—	—	64注6
タイマ計測 (タイマ開始 ／タイマ終了)	4+3注5	—	—	—	—	1
アクション (Printf イベ ント)	100注7					64注8

表 2—27 有効イベント数の制限【V850E2】

イベント種別	デバッグ・ツール					シミュレータ
	IECUBE2	MINICUBE E1/E20(LPD)/(JTAG)		MINICUBE2 E1/E20(Serial)		
		TEU あり	TEU なし	コアイベント あり	コアイベント なし	
ハードウェア・ブレーク (実行系：実行前)	4	4	4 注2	4 注2	0	64 注4
ハードウェア・ブレーク (実行系：実行後)	8	8	0	0	0	
ハードウェア・ブレーク (アクセス系)	6	6	4 注2	4 注2	0	
ソフトウェア・ブレーク	8 (TRAM 非搭載時) 40 ~ 2000 (TRAM 搭載時)					—
トレース (トレース開始 /トレース終了)	4+3+3 注5, 9	—	—	—	—	32 注5
ポイント・トレース	8+6+6 注6, 9	—	—	—	—	64 注6
タイマ計測 (タイマ開始 /タイマ終了)	4+3+3 注5	2 注10				1
アクション (Printf イベント)	100 注7					64 注8

- 注1. フラッシュ・セルフ・エミュレーション機能を有効に指定している場合（プロパティパネル上の [フラッシュ・セルフ・エミュレーション設定] タブ【IECUBE】上の [フラッシュ・セルフ・エミュレーション] カテゴリ内 [フラッシュ・セルフ・プログラミング・エミュレーションを行う] プロパティを [はい] に指定) は1個
2. ハードウェア・ブレーク (実行系：実行前) とハードウェア・ブレーク (アクセス系) で兼用
3. フラッシュ・メモリ領域に設定可能なブレーク数は対象マイクロコントローラの持つROMコレクションに依存
4. 実行前ブレーク/実行後ブレークはプロパティパネルにおいて指定可
5. 今版では1組のみ設定可 (ただし, トレース開始イベント/トレース終了イベントは複数設定可)
6. 今版では1つのみ設定可 (ただし, イベント条件は複数設定可)
7. ソフトウェア・ブレークと兼用 (ただし, 有効/無効状態に関わらず 100 個まで)
8. ハードウェア・ブレーク (実行系：実行前) と兼用 (ただし, 有効/無効状態に関わらず 64 個まで)
9. トレース開始イベント/トレース終了イベントとポイント・トレース・イベントを同時に設定した場合, トレース開始イベント/トレース終了イベントで設定した区間内でポイント・トレース・イベントの条件に一致したデータのみをトレース・データとして取得可
10. PMU 搭載品の場合, ハードウェア・ブレーク (実行系：実行前) を 2 個使用

(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を示します。

なお、表内のマークの意味は次のとおりです。

○	可能
△	プログラムの実行を一瞬停止することで可能 (プロパティパネル→[デバッグ・ツール設定]タブ→[実行中のイベント設定]カテゴリ→[実行を一瞬停止してイベントを設定する]プロパティ→[はい])
□	トレーサ／タイマの実行を一瞬停止することで可能 (プロパティパネル→[デバッグ・ツール設定]タブ→[実行中のイベント設定]カテゴリ→[実行を一瞬停止してイベントを設定する]プロパティ→[はい])
▲	トレーサ／タイマ動作中は不可
—	不可、または非サポート

表 2—28 実行中に設定／削除可能なイベント種別【V850E1】【V850ES】

イベント種別	デバッグ・ツール					シミュレータ
	IECUBE	MINICUBE E1/E20(JTAG)		MINICUBE2 E1/E20(Serial) EZ Emulator		
		TEU あり	TEU なし	コアイベントあり	コアイベントなし	
ハードウェア・ブ레이크 (実行系：実行前)	△	△	△	△	—	▲
ハードウェア・ブ레이크 (実行系：実行後)	○	○	—	—	—	▲
ハードウェア・ブ레이크 (アクセス系)	○	○	△	△	—	▲
ソフトウェア・ブ레이크	△	△	△	△	△	—
トレース (トレース開始 ／トレース終了)	□注1	—	—	—	—	▲
ポイント・トレース	□注1	—	—	—	—	▲
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—29 実行中に設定／削除可能なイベント種別【V850E2】

イベント種別	デバッグ・ツール					シミュレータ
	IECUBE2	MINICUBE E1/E20(LPD)/(JTAG)		MINICUBE2 E1/E20(Serial)		
		TEU あり	TEU なし	コアイベン トあり	コアイベ ントなし	
ハードウェア・ブレーク (実行系：実行前)	△	△	△	△	—	▲
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 (アクセス系)	○	○	—	—	—	▲
ソフトウェア・ブレーク	△	△	△	△	△	—
トレース（トレース開始 ／トレース終了）	<input type="checkbox"/> 注2	—	—	—	—	▲
ポイント・トレース	<input type="checkbox"/> 注2	—	—	—	—	▲
タイマ計測（タイマ開始 ／タイマ終了）	<input type="checkbox"/>	—	—	—	—	▲
アクション（Printf イベ ント）	△	△	△	△	△	—

注 1. 【V850E1】【V850ES】


タイマ機能，またはカバレッジ機能が動作中の場合は削除できません。

2. 【V850E2】

タイマ機能，またはカバレッジ機能が動作中の場合は設定／削除できません。

(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。

- 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
- 【V850E1】【V850ES】：【シミュレータ以外】
ソフトウェア・ブレーク・イベントと次のハードウェア・ブレーク・イベントが競合した場合、PCレジスタ値を不正に補正する場合があります。
ソフトウェア・ブレーク・イベントではなく、ハードウェア・ブレーク・イベントを使用してください。
 - トレース・フル・ブレーク
 - タイム・アウト・ブレーク
 - ノンマップ・ブレーク
 - ライト・プロテクト・ブレーク
 - I/O イリーガル・アクセス・ブレーク
 - 強制ブレーク（ ボタンの押下）
 - ハードウェア・ブレーク・イベント
- 【V850E1】【V850ES】
ROM 化の対象がコードの場合、そのコードに対してソフトウェア・ブレーク・イベントを設定しても、RAM へのコピー時にブレーク用の命令が削除されるためブレークしません。
 - 【シミュレータ】以外
ハードウェア・ブレーク・イベントを使用してください。
 - 【シミュレータ】
ハードウェア・ブレーク・イベントを使用してもブレークしませんが、トレース機能、またはタイマ機能を有効化（プロパティパネルの【デバッグ・ツール設定】タブにおける【トレース】／【タイマ】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】プロパティを【はい】に設定）することによりブレークするようになります。
- 【IECUBE】【IECUBE2】以外
ブート・スワップ領域にソフトウェア・ブレーク・イベント（シミュレータの場合はハードウェア・ブレーク・イベント）を設定した場合、フラッシュ ROM にブレーク用の命令が書き込まれるため、ブート・スワップ後もブレーク用の命令が残ってしまいます。
 - 【E1】【E20】【EZ Emulator】
ブレークを設定する場合は、ハードウェア・ブレーク・イベントを使用してください。
 - 【シミュレータ】
ブートスワップ領域にブレーク・イベントを設定しないでください。

- 【V850E2】

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、イベントはすべてのコア（PE n ）で有効となるように設定されます。

2.16 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。

図 2—246 [フック処理] カテゴリ

フック処理設定	
ダウンロード前	ダウンロード前[0]
ダウンロード後	ダウンロード後[0]
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]
実行開始前	実行開始前[0]
ブレーク後	ブレーク後[0]

表 2—30 [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—247 テキスト編集ダイアログのオープン

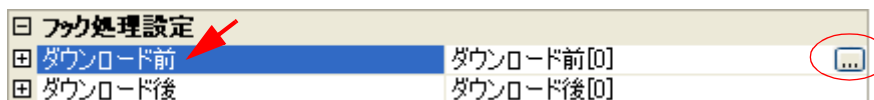
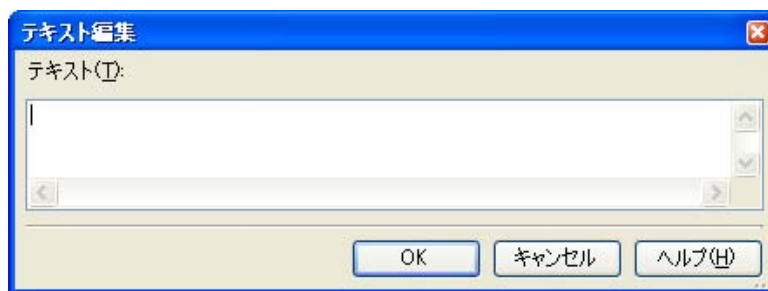


図 2—248 フック処理を設定する（テキスト編集 ダイアログ）



このダイアログにおいて、目的の処理を直接入力により指定します。処理の指定形式は次のいずれかです。

【指定形式：1】

I/O レジスタの内容を、*数値*に自動的に書き換えます。

I/O レジスタ名 数値

【指定形式：2】

CPU レジスタの内容を、*数値*に自動的に書き換えます。

CPU レジスタ名 数値

【指定形式：3】

Python スクリプト・パス（絶対パス／プロジェクト・フォルダを基点とした相対パス）で指定したスクリプト・ファイルを実行します。

Source Python スクリプト・パス

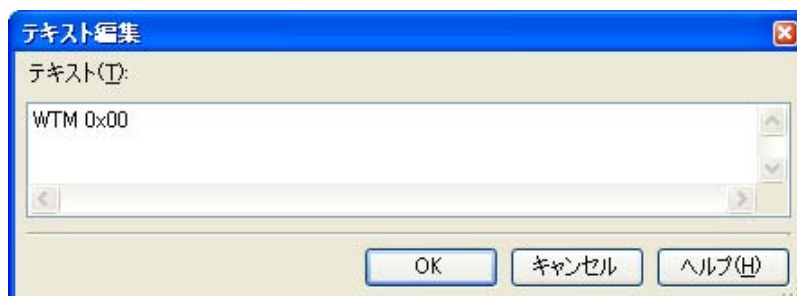
備考 1. 処理の指定の際、行頭に“#”を付与することにより、その行はコメント扱いとなります。

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—249 フック処理設定の例



2.17 シミュレータ GUI の使用【シミュレータ】

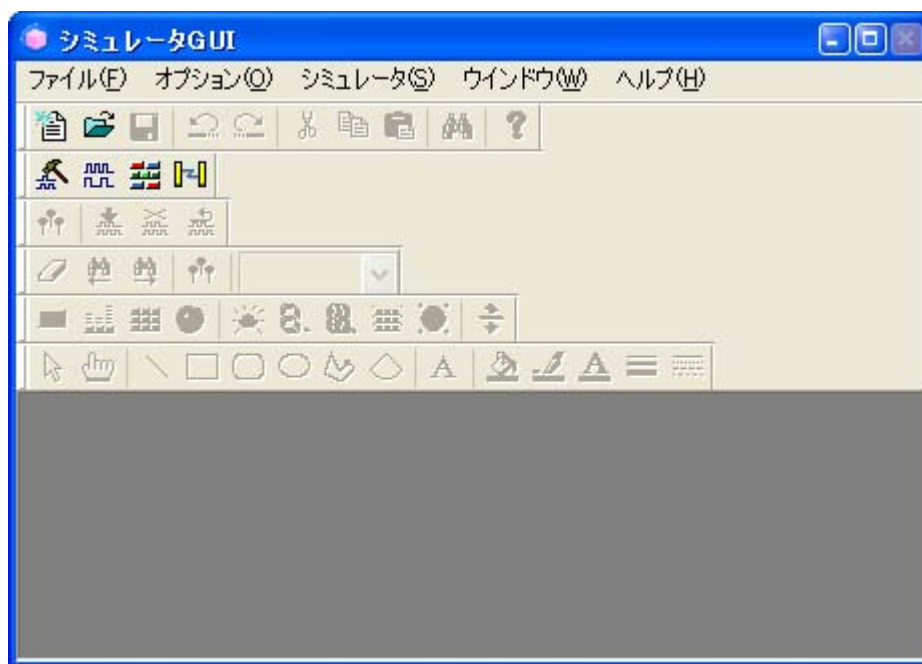
この節では、シミュレータ GUI の使用方法について説明します。

なお、この節で説明するシミュレータ GUI は、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る場合のみ提供する機能です。

シミュレータ GUI のすべての操作は、次のシミュレータ GUI ウィンドウより行います。このウィンドウは、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器シミュレーションをサポートしている場合で、かつ使用するデバッグ・ツールに“シミュレータ”を選択している場合、デバッグ・ツールと接続すると自動的にオープンします（デフォルト）。

備考 シミュレータ GUI ウィンドウ、およびこのウィンドウよりオープンする各種ウィンドウは、CubeSuite+ のメイン・ウィンドウとドッキング表示することはできません。

図 2—250 シミュレータ GUI を使用する（シミュレータ GUI ウィンドウ）



なお、シミュレータ GUI ウィンドウの表示に関する設定は、次のプロパティパネルの [デバッグ・ツール設定] タブ上の [シミュレータ GUI] カテゴリ内で行うことができます。

必要に応じて、次の設定を行ってください。

注意 デバッグ・ツールと接続後、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。

図 2—251 [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	はい

(1) [シミュレータ GUI を表示する]

シミュレータ GUI ウィンドウ を表示するか否かをドロップダウン・リストにより指定します。

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 ウィンドウ がクローズします。


(2) [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] を指定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 ウィンドウ を最前面に表示するか否かをドロップダウン・リストにより指定します。

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)。

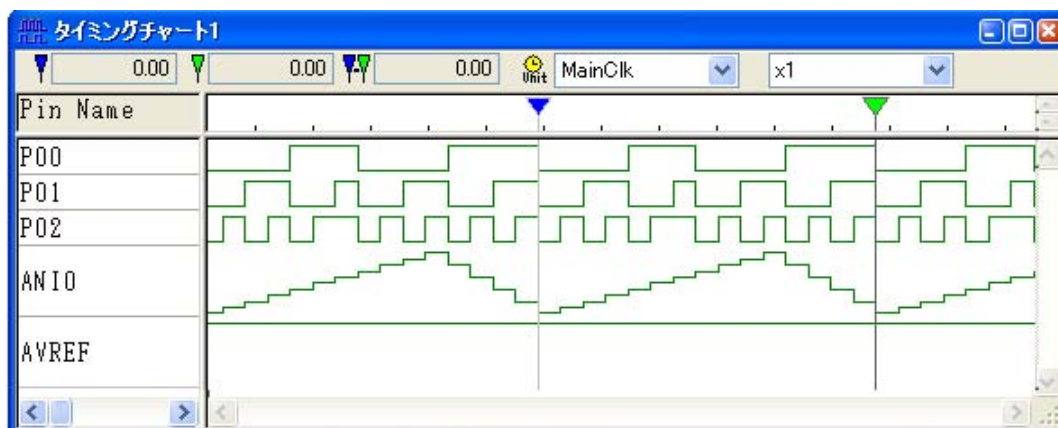
2.17.1 マイコンの入出力波形を確認する

マイコンの入出力波形の確認は、シミュレータ GUI ウィンドウ のツールバーの  ボタンをクリックすることによりオープンする、次のタイミングチャート ウィンドウで行うことができます。


このウィンドウでは、マイコンの端子に対する入力信号と出力信号をタイミング・チャートで表示します。

操作方法についての詳細は、タイミングチャート ウィンドウ の項を参照してください。

図 2—252 マイコンの入出力波形の確認 (タイミングチャート ウィンドウ)



2.17.2 端子へ信号を入力する

端子への入力信号の設定は、シミュレータ GUI ウィンドウのツールバーの  ボタンをクリックすることによりオープンする、次の信号データエディタ ウィンドウで行うことができます。


このウィンドウでは、入力端子に対して任意のタイミングの入力信号データを数値で設定することができます。操作方法についての詳細は、信号データエディタ ウィンドウの項を参照してください。

図 2—253 端子への入力信号の設定（信号データエディタ ウィンドウ）



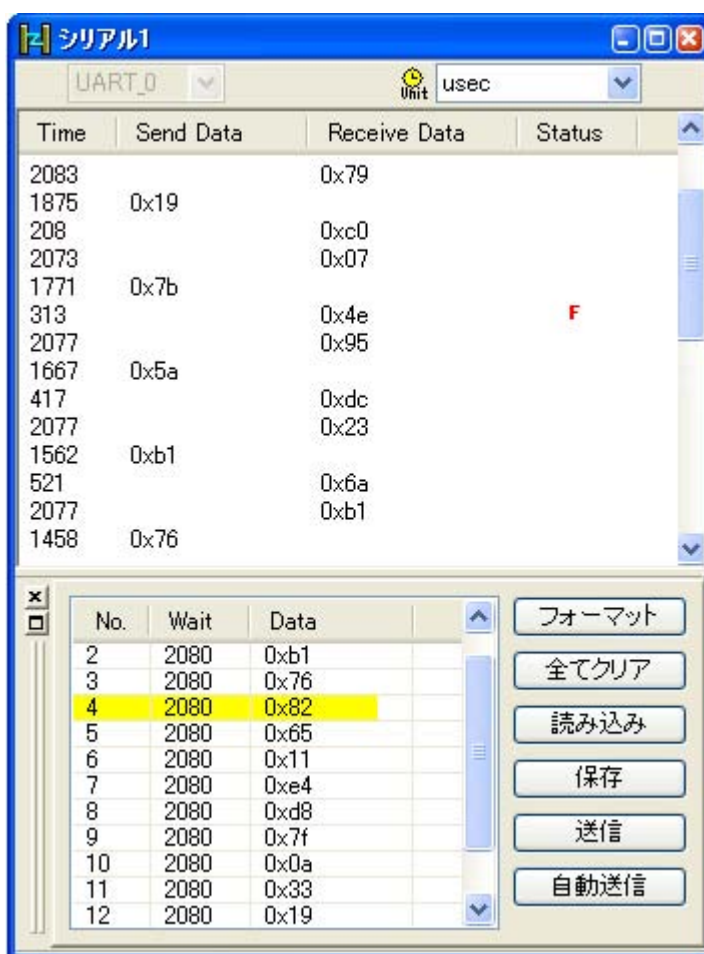
Unit	Mark	Wait	P00	P01	P02	ANIO	AVREF
1	↩	100	0	0	0	0	5000
2		100	0	0	1	500	5000
3		100	0	1	0	1000	5000
4		100	0	1	1	1500	5000
5		100	1	0	0	2000	5000
6		100	1	0	1	2500	5000
7		100	1	1	1	3000	5000
8		100	0	0	0	3500	5000
9		100	0	0	1	4000	5000
10		100	0	1	0	4500	5000

2.17.3 シリアル通信を行う

シリアル通信の設定は、シミュレータ GUI ウィンドウのツールバーの  ボタンをクリックすることによりオープンする、次のシリアルウィンドウで行うことができます。


このウィンドウでは、CPU に搭載されているシリアル・インタフェースと通信を行うためのシリアル入出力機能として、マイクロコントローラのシリアル受信端子へのデータ入力と送信端子からの出力データの取得が可能です。操作方法についての詳細は、シリアルウィンドウの項を参照してください。

図 2—254 シリアル通信を行う（シリアル ウィンドウ）



2.17.4 ボタン/LED/ レベル・ゲージなどの部品を使用する

シミュレータ GUI では、周辺 I/O との入出力部を GUI 化した標準的な接続部品（ボタン/LED/ レベル・ゲージなど）を提供することで、入力操作、および出力表示のシミュレーションを可能にしています。

各種接続部品の設定は、シミュレータ GUI ウィンドウのツールバーの  ボタンをクリックすることによりオープンする、次の入出力パネル ウィンドウで行うことができます。

このウィンドウにおいて各種接続部品の設定を行うことにより、疑似的なターゲット・システムを構築することができます。

操作方法についての詳細は、[入出力パネル ウィンドウ](#)の項を参照してください。

図 2—255 各種接続部品の設定（入出力パネル ウィンドウ）



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—31 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、全角のカタカナ、漢字（中国語を含む）、および半角のカタカナ

(2) 数 値

数値を入力する際に許可している進数は次のとおりです。

表 2—32 進数の一覧

進数表記	概要
2 進数	0b で始まり、0 ~ 1 の数字が続く数値
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	0x で始まり、0 ~ 9 の数字、および a ~ f の英字が続く数値 （英字の大文字／小文字については、不問） ただし、 HEX マークが表示されている入力エリアでは、0x の接頭辞は必要ありません。

(3) 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。

シンボルとして、I/O レジスタ名、ラベル名、関数名、変数名が記述された場合は、そのアドレスをシンボルの値として演算します。

式の基本入力形式は次のとおりです。

表 2—33 式の基本入力形式

式	説明
C 言語変数名	C 言語の変数の値
式[インデクス]	配列の要素値
式. メンバ名	構造体／共用体のメンバ値
式-> メンバ名	ポインタの指し示す構造体／共用体のメンバ値
* 式	ポインタの変数の値
CPU レジスタ名	CPU レジスタの値
I/O レジスタ名	I/O レジスタの値
ラベル名, EQU シンボル名, 即値アドレス	ラベルの値, EQU シンボルの値, 即値アドレスの値
ビット・シンボル	ビット・シンボルの値

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。

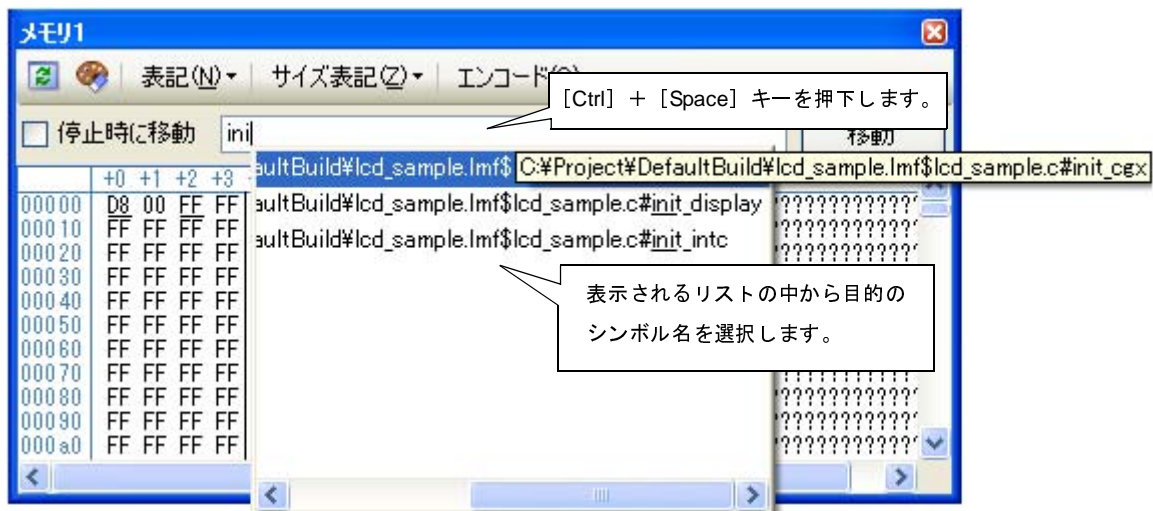
シンボル名のリストは、この機能に対応するテキスト・ボックスにおいて、目的のシンボル名の一部が入力されている状態で [Ctrl] + [Space] キーを押下することにより表示されます。リスト内において、目的のシンボル名をダブルクリックすることで（または、[↑] / [↓] キーによりシンボル名を選択したのち [Space] / [Enter] キーを押下）、入力中のシンボル名が補完されます。

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 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。


備考 シンボル名の入力時にこの機能を使用できるか否かは、該当するパネル／ダイアログの入力エリアの説明を参照してください。

図 2—256 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 ウィンドウ・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ィンドウ／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ィンドウ／パネル／ダイアログの一覧を示します。

表 A—1 ウィンドウ／パネル／ダイアログ一覧

ウィンドウ／パネル／ダイアログ名	機能概要
メイン・ウィンドウ	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【V850E2】	デバッグ対象とするコア（PE n ）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
I/O レジスタ 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【IECUBE】【IECUBE2】【シミュレータ】	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・マッピングの設定
ダウンロード・ファイル ダイアログ	ダウンロードする際のファイルの選択、およびダウンロード条件の設定
外部フラッシュ・メモリ・ダウンロード ダイアログ	外部フラッシュ・メモリにダウンロードする際の外部フラッシュ・メモリ情報ファイルの選択、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【V850E2】	フラッシュ・メモリのオプション設定

ウィンドウ／パネル／ダイアログ名	機能概要
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 ウィンドウ	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【IECUBE】 【IECUBE2】【シミュレータ】	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 パネルにおけるスクロール範囲の設定
指定行へのジャンプ ダイアログ	指定した行にカーレットを移動
指定位置へ移動 ダイアログ	指定した位置にカーレットを移動
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	ウォッチ式をインポートするためのファイルの選択
外部フラッシュ・メモリ情報ファイルを選択 ダイアログ	外部フラッシュ・メモリ情報ファイルの選択
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
シミュレータ・コンフィギュレーション・ファ イルを選択 ダイアログ【シミュレータ】	シミュレータ・コンフィギュレーション・ファイルの選択

表 A—2 ウィンドウ／ダイアログ一覧（シミュレータ GUI 部専用）

ウィンドウ／パネル／ダイアログ名	機能概要
シミュレータ GUI ウィンドウ	各種シミュレータ GUI ウィンドウのオープン、および操作
書式設定 ダイアログ	ウィンドウ色、フォント等の設定

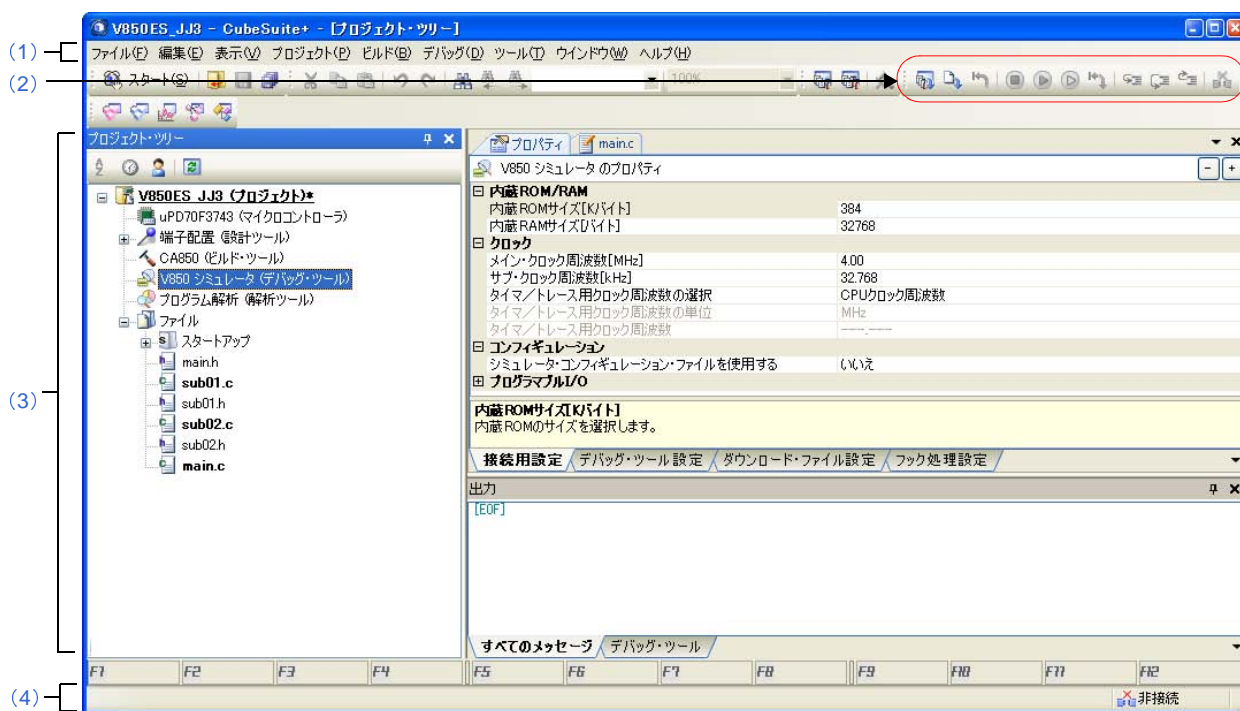
ウィンドウ／パネル／ダイアログ名	機能概要
信号データエディタ ウィンドウ	信号データの入力設定
ループ設定 ダイアログ	信号データエディタ ウィンドウのループ情報設定
端子選択 ダイアログ	信号データエディタ ウィンドウ、およびタイミングチャート ウィンドウの表示端子の選択
タイミングチャート ウィンドウ	入力信号と出力信号のタイミング・チャート表示
データ検索 ダイアログ	タイミングチャート ウィンドウの詳細検索
入出力パネル ウィンドウ	疑似的なターゲット・システムの構築
Parts Button Properties ダイアログ	ボタンの端子接続情報を設定
Analog Button Properties ダイアログ	アナログ・ボタンの端子接続情報を設定
Parts Key Properties ダイアログ	キーマトリクス LED の端子接続情報を設定
Parts Level Gauge Properties ダイアログ	レベルゲージの端子接続情報を設定
Parts Led Properties ダイアログ	LED の端子接続情報を設定
Parts Segment LED Properties ダイアログ	7 / 14 セグメント LED の端子接続情報を設定
Parts Matrix Led Properties ダイアログ	マトリクス LED の端子接続情報を設定
Parts Buzzer Properties ダイアログ	ブザーの端子接続情報を設定
プルアップ／プルダウン設定 ダイアログ	プルアップ／プルダウン抵抗の端子接続情報を設定
Object Properties ダイアログ	図形／文字／ビットマップの端子への接続情報の設定
部品一覧 ダイアログ	入出力パネル ウィンドウ上のオブジェクトの端子接続状況の一覧表示
シリアル ウィンドウ	シリアル・インタフェースの通信ウィンドウ
フォーマット (UART) ダイアログ	シリアル・フォーマットの設定 (UART)
フォーマット (CSI) ダイアログ	シリアル・フォーマットの設定 (CSI)

メイン・ウィンドウ

CubeSuite+ を起動した際、最初にオープンするウィンドウです。

デバッグを行う際は、このウィ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・ウィンドウ



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）。

デバッグ・マネージャ 【V850E2】	デバッグ・マネージャ パネル【V850E2】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V850E2 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。
ウォッチ	ウォッチ パネルをオープンするために、次のカスケード・メニューを表示します。 ただし、デバッグ・ツールと切断時は無効となります。
ウォッチ 1	ウォッチ パネル（ウォッチ 1）をオープンします。
ウォッチ 2	ウォッチ パネル（ウォッチ 2）をオープンします。
ウォッチ 3	ウォッチ パネル（ウォッチ 3）をオープンします。
ウォッチ 4	ウォッチ パネル（ウォッチ 4）をオープンします。
ローカル変数	ローカル変数 パネルをオープンします。 ただし、デバッグ・ツールと切断時は無効となります。
コール・スタック	コール・スタック パネルをオープンします。 ただし、デバッグ・ツールと切断時は無効となります。
メモリ	メモリ パネルをオープンするために、次のカスケード・メニューを表示します。 ただし、デバッグ・ツールと切断時は無効となります。
メモリ 1	メモリ パネル（メモリ 1）をオープンします。
メモリ 2	メモリ パネル（メモリ 2）をオープンします。
メモリ 3	メモリ パネル（メモリ 3）をオープンします。
メモリ 4	メモリ パネル（メモリ 4）をオープンします。
IOR	IOR パネルをオープンします。 ただし、デバッグ・ツールと切断時は無効となります。
CPU レジスタ	CPU レジスタ パネルをオープンします。 ただし、デバッグ・ツールと切断時は無効となります。
トレース 【IECUBE】【IECUBE2】 【シミュレータ】	トレース パネル【IECUBE】【IECUBE2】【シミュレータ】をオープンします。 ただし、デバッグ・ツールと切断時は無効となります。

逆アセンブル	逆アセンブル パネルをオープンするために、次のカスケード・メニューを表示します。 ただし、デバッグ・ツールと切断時は無効となります。
逆アセンブル 1	逆アセンブル パネル (逆アセンブル 1) をオープンします。
逆アセンブル 2	逆アセンブル パネル (逆アセンブル 2) をオープンします。
逆アセンブル 3	逆アセンブル パネル (逆アセンブル 3) をオープンします。
逆アセンブル 4	逆アセンブル パネル (逆アセンブル 4) をオープンします。
イベント	イベント パネルをオープンします。 ただし、デバッグ・ツールと切断時は無効となります。
現在の PC 位置を開く	カレント PC 位置 (PC レジスタ値) をエディタ パネルで表示します。 ただし、デバッグ・ツールと切断時は無効となります。
ジャンプ前の位置へ戻る	定義箇所へジャンプ (「(7) 関数へジャンプする」 / 「(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。
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・ツールヘダウンロード	アクティブ・プロジェクトで現在選択しているデバッグ・ツールに、指定されたファイルをダウンロードします。 デバッグ・ツールと切断時の場合は、自動的にデバッグ・ツールに接続し、ダウンロードを実行します。 ただし、プログラム実行中、または [ビルド & デバッグ・ツールヘダウンロード] 実行中は無効となります。
ビルド & デバッグ・ツールヘダウンロード	プロジェクトのビルドを行い、ビルド後にアクティブ・プロジェクトで現在選択しているデバッグ・ツールにダウンロードを実行します。 デバッグ・ツールと切断時の場合は、自動的にデバッグ・ツールに接続し、ダウンロードを実行します。 ただし、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。
ホット・プラグイン 【V850E2】 【E1/E20(LPD)】 【E1/E20(JTAG)】	現在実行中のターゲット・システムをデバッグするために、アクティブ・プロジェクトで現在選択しているデバッグ・ツールとホット・プラグイン接続します (「2.4.3 デバッグ・ツールをホット・プラグイン接続する【E1/E20(LPD)/ (JTAG) 【V850E2】】」参照)。 ただし、デバッグ・ツールと接続時は無効となります。

デバッグ・ツールからアップロード ...	メモリ内容をファイルに保存するためのデータ保存ダイアログをオープンします。 ただし、プログラム実行中、[ビルド & デバッグ・ツールヘダダウンロード] 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・ツールヘダダウンロード] 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・ツールヘダダウンロード] 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・ツールヘダダウンロード] 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・ツールヘダダウンロード] 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・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・ツールヘダダウンロード] 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・ツールヘダダウンロード] 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・ツールヘダダウンロード] 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・ツールヘダダウンロード] 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、「2.7.3 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。
2. ツールバー上を右クリックすることで表示されるコンテキスト・メニューにより、ツールバー上に表示／非表示するグループを選択することができます。

	<p>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。</p> <p>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。</p> <p>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。</p> <p>[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。</p>
	<p>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。</p> <p>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。</p> <p>ただし、プログラム実行中、または [ビルド & デバッグ・ツールヘダウンロード] 実行中は無効となります。</p> <p>[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。</p>
	<p>CPU をリセットします（プログラムは実行しません）。</p> <p>ただし、[ビルド & デバッグ・ツールヘダウンロード] 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。</p> <p>[デバッグ] メニュー → [CPU リセット] の選択と同等です。</p>
	<p>現在実行中のプログラムを強制的に停止します。</p> <p>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。</p> <p>[デバッグ] メニュー → [停止] の選択と同等です。</p>
	<p>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。</p> <p>ただし、プログラム実行中、[ビルド & デバッグ・ツールヘダウンロード] 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。</p> <p>[デバッグ] メニュー → [実行] の選択と同等です。</p>
	<p>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。</p> <p>ただし、プログラム実行中、[ビルド & デバッグ・ツールヘダウンロード] 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。</p> <p>[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。</p>
	<p>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。</p> <p>ただし、[ビルド & デバッグ・ツールヘダウンロード] 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。</p> <p>[デバッグ] メニュー → [リスタート] の選択と同等です。</p>

	<p>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・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。</p> <p>ただし、プログラム実行中、[ビルド&デバッグ・ツールヘダダウンロード] 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。</p> <p>[デバッグ] メニュー→ [ステップ・イン] の選択と同等です。</p>
	<p>カレント PC 位置からステップ実行し^注、各パネルの内容を更新します（ステップ・オーバー実行）。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）。</p> <p>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。</p> <p>ただし、プログラム実行中、[ビルド&デバッグ・ツールヘダダウンロード] 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。</p> <p>[デバッグ] メニュー→ [ステップ・オーバー] の選択と同等です。</p>
	<p>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）。</p> <p>ただし、プログラム実行中、[ビルド&デバッグ・ツールヘダダウンロード] 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。</p> <p>[デバッグ] メニュー→ [リターン・アウト] の選択と同等です。</p>
	<p>現在接続中のデバッグ・ツールとの通信を切断します。</p> <p>ただし、[ビルド&デバッグ・ツールヘダダウンロード] 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。</p> <p>[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。</p>

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、「[2.7.3 プログラムをステップ実行する](#)」を参照してください。

(3) パネル表示エリア

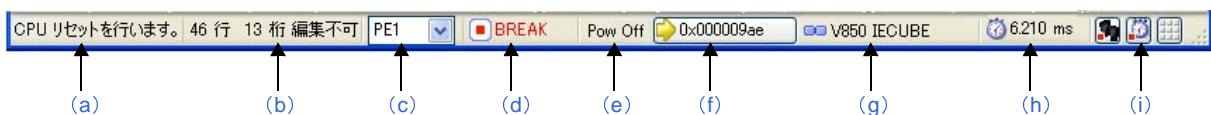
各種パネルを表示するエリアです。

表示内容についての詳細は、各パネルの項を参照してください。

(4) ステータスバー

ステータスバーは、次の情報を表示します。

図 A-2 ステータスバー



(a) ステータス・メッセージ

次のメッセージを表示します。

- 選択しているメニュー項目の簡易説明
- パネル/ダイアログにおいて入力値が不正な場合のメッセージ

- 検索・置換 ダイアログにより検索した際に、指定文字列が見つからなかった場合のメッセージ
- ブレークした際のブレーク要因（「2.8 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【V850E2】

デバッグ対象とするコア（PE *n*）を指定します。


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・ツールと切断時の場合は非表示となります。

備考 デバッグ対象コアの指定は、[デバッグ・マネージャ パネル【V850E2】](#)においても行うことができます。

(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
IECUBE	Halt	HALT モード中
IECUBE2	StopIdle	ハードウェア STOP/ ソフトウェア STOP/IDLE モード中
MINICUBE	Hold	バス・ホールド中
E1/E20(LPD)	Wait	ウエイト状態
E1/E20(JTAG)	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
MINICUBE2	Reset	リセット状態
E1/E20(Serial)	PowOff	ターゲットに電源が供給されていない状態
EZ Emulator		

デバッグ・ツール	表示内容	CPU 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

備考 CPU が上記以外の状態の場合は何も表示しません。

(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、次の情報をポップアップ表示します。

- カレント PC : 0x カレント PC 値 (ソース名# 行数^注)


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。

(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（[「2.12.1 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
オーバーフローした場合	OVERFLOW

(i) デバッグ・ツールの状態【IECUBE】【IECUBE2】【シミュレータ】

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。

機能が停止中の場合、対象アイコンをクリックすることにより、使用する／使用しない^注の状態を変更することができます。

ただし、デバッグ・ツールと切断時の場合は非表示となります。

機能	動作中	停止中（使用する）	使用しない
トレース			
タイマ			
カバレッジ			

注 【IECUBE】【IECUBE2】

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 [\[トレース・データの用途\]](#) プロパティの指定に反映します。

【シミュレータ】

プロパティパネルの [\[デバッグ・ツール設定\]](#) タブ上の [\[トレース\]](#) / [\[タイマ\]](#) / [\[カバレッジ\]](#) 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します。

デバッグ・マネージャ パネル【V850E2】

選択しているマイクロコントローラが V850E2 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 : Processer Element）の選択、およびその状態の表示を行います。

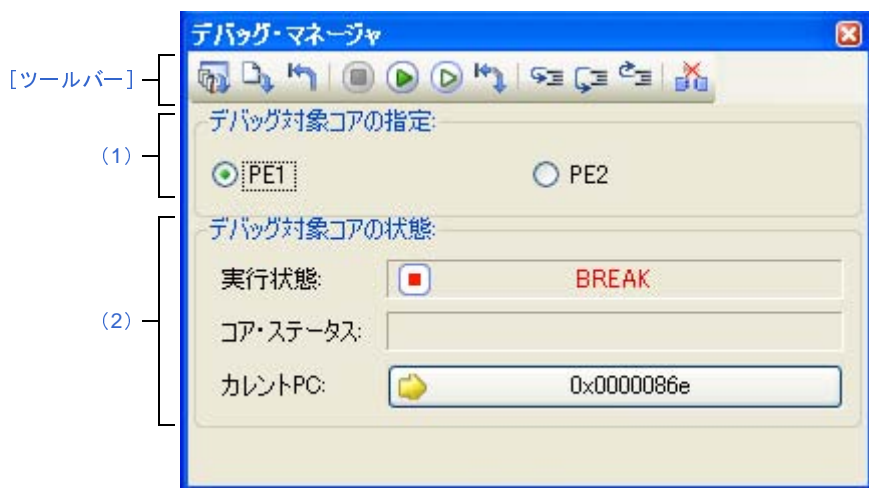
なお、このパネルは、デバッグ・ツールと接続時のみオープンすることができます。

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。

備考 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（イベントは全 PE で有効となるように自動設定されます）。

ただし、ステップ実行は、現在選択している PE n でのみ行います。

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コア（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・ウィンドウ](#)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・ウィンドウ](#)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
MINICUBE E1/E20(LPD) E1/E20(JTAG)	Halt	HALT モード中
	StopIdle	ハードウェア STOP/ ソフトウェア 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

(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・ウィンドウ](#)上の[デバッグ・ツールバー](#)の機能と同等です。

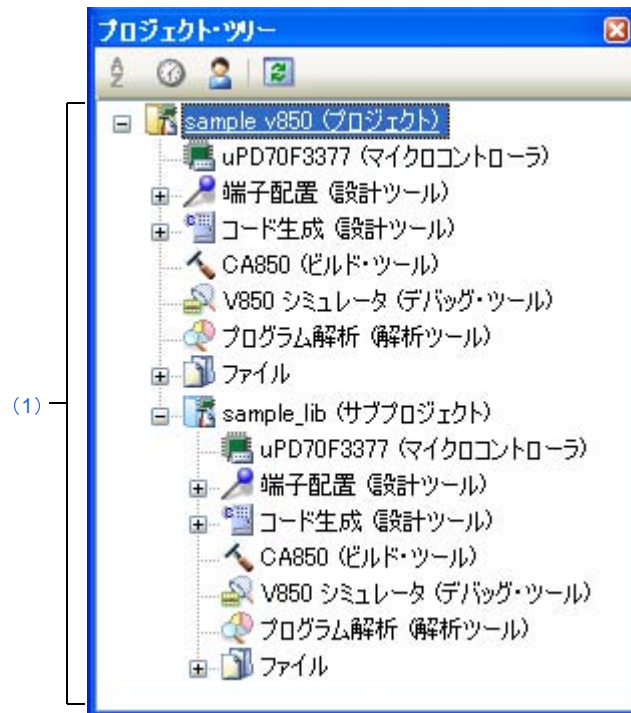
各ボタンの機能についての詳細は、「(2) [デバッグ・ツールバー](#)」を参照してください。

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 (デバッグ・ツール)	<ul style="list-style-type: none"> - 品種名 選択しているマイクロコントローラの品種名 (V850, V850E2) を表示します。 - デバッグ・ツール名 現在、プロジェクトで使用するデバッグ・ツール名 (IECUBE, IECUBE2, MINICUBE, MINICUBE2(Serial), E1(Serial), E1(LPD), E1(JTAG), E20(Serial), E20(LPD), E20(JTAG), EZ Emulator, シミュレータ) を表示します。 新規プロジェクト作成時は、シミュレータが設定されます^注。

注 選択しているマイクロコントローラがシミュレータを未サポートの場合、MINICUBE が設定されます。

ノードを選択すると、その詳細情報（プロパティ）が**プロパティパネル**に表示され、設定の変更を行うことができます（プロパティパネルがオープンしていない場合は、ノードをダブルクリックすることでオープンします）。

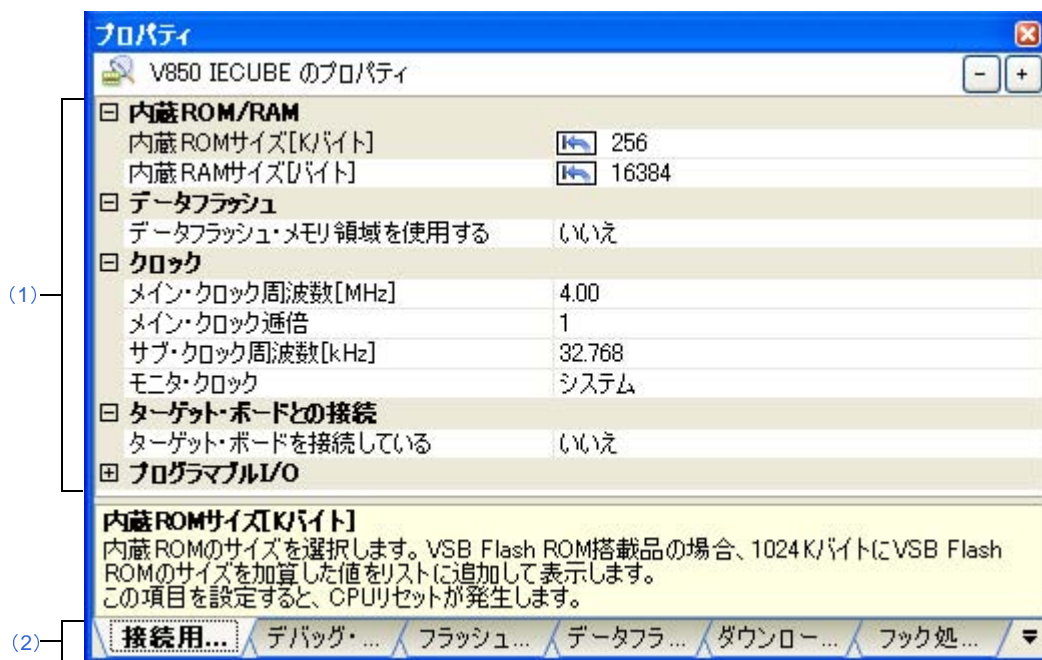
[コンテキスト・メニュー]

使用するデバッグ・ツール	使用するデバッグ・ツールを選択するためのカスケード・メニューを表示します。 なお、プロジェクトで選択しているマイクロコントローラの種類により、表示されるデバッグ・ツールは異なります（「表 2-1 マイクロコントローラの種類と接続可能なデバッグ・ツールの関係」参照）。
品種名 IECUBE	IECUBE を使用します。
品種名 IECUBE2	IECUBE2 を使用します。
品種名 MINICUBE	MINICUBE を使用します。
品種名 MINICUBE2(Serial)	MINICUBE2 を使用します。
品種名 E1(Serial)	E1 をシリアル通信方式で使用します。
品種名 E1(LPD)	E1 を LPD 通信方式で使用します。
品種名 E1(JTAG)	E1 を JTAG 通信方式で使用します。
品種名 E20(Serial)	E20 をシリアル通信方式で使用します。
品種名 E20(LPD)	E20 を LPD 通信方式で使用します。
品種名 E20(JTAG)	E20 を JTAG 通信方式で使用します。
品種名 EZ Emulator	評価キットなどを接続して使用します。
品種名 シミュレータ	シミュレータを使用します。
プロパティ	選択しているデバッグ・ツールのプロパティを プロパティパネル に表示します。

プロパティ パネル

プロジェクト・ツリー パネルで選択しているデバッグ・ツールについて、カテゴリ別に詳細情報の表示、および設定の変更を行います。

図 A—5 プロパティ パネル (IECUBE を選択した場合の例)



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [[編集] メニュー (プロパティ パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- プロジェクト・ツリー パネルにおいて、使用する [品種名 デバッグ・ツール名 (デバッグ・ツール)] ノードを選択したのち、[表示] メニュー、またはコンテキスト・メニューより [プロパティ] を選択
- プロジェクト・ツリー パネルにおいて、使用する [品種名 デバッグ・ツール名 (デバッグ・ツール)] ノードをダブルクリック

備考 このパネルがすでにオープンしている場合、プロジェクト・ツリー パネル上において、使用する [品種名 デバッグ・ツール名 (デバッグ・ツール)] ノードを選択することにより、選択したデバッグ・ツールの詳細情報を表示します。

[各エリアの説明]

(1) 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
- [フラッシュ・セルフ・エミュレーション設定] タブ【IECUBE】
- [データフラッシュ・エミュレーション設定] タブ【IECUBE】
- [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【V850E2】
- [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。

[コンテキスト・メニュー]

【文字列編集以外の場合】

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) [内蔵 ROM/RAM]
- (2) [データフラッシュ] 【V850E1】 【V850ES】 (【シミュレータ】以外)
- (3) [クロック]
- (4) [ターゲット・ボードとの接続] (【MINICUBE】 【E20(LPD)/(JTAG)】 【シミュレータ】以外)
- (5) [フラッシュ] (【IECUBE】 【シミュレータ】以外)
- (6) [コンフィギュレーション] 【シミュレータ】
- (7) [プログラマブル I/O] 【V850E1】 【V850ES】

図 A-6 プロパティ パネル: [接続用設定] タブ 【IECUBE】

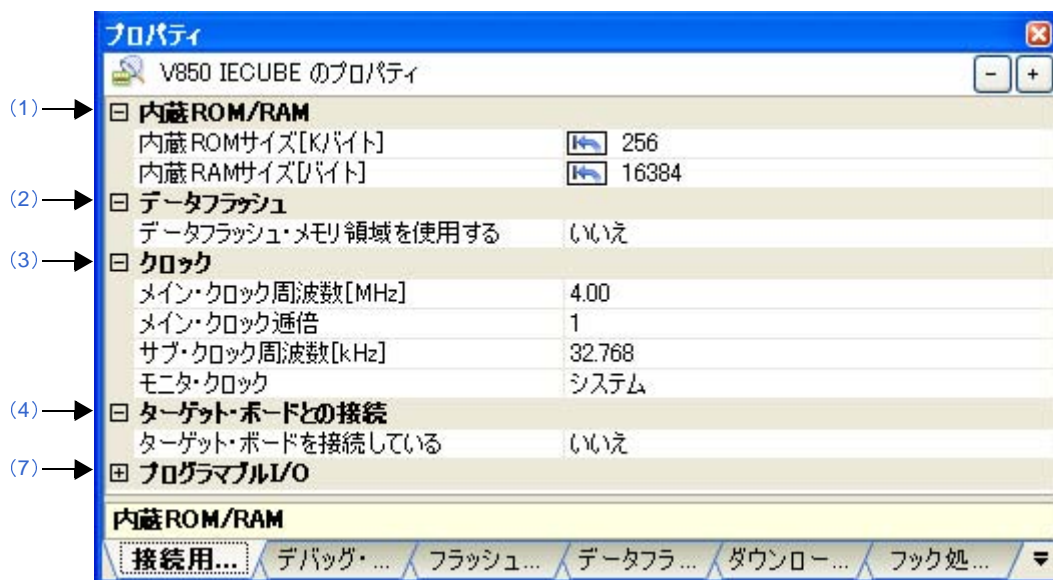


図 A—7 プロパティ パネル：[接続用設定] タブ【IECUBE2】

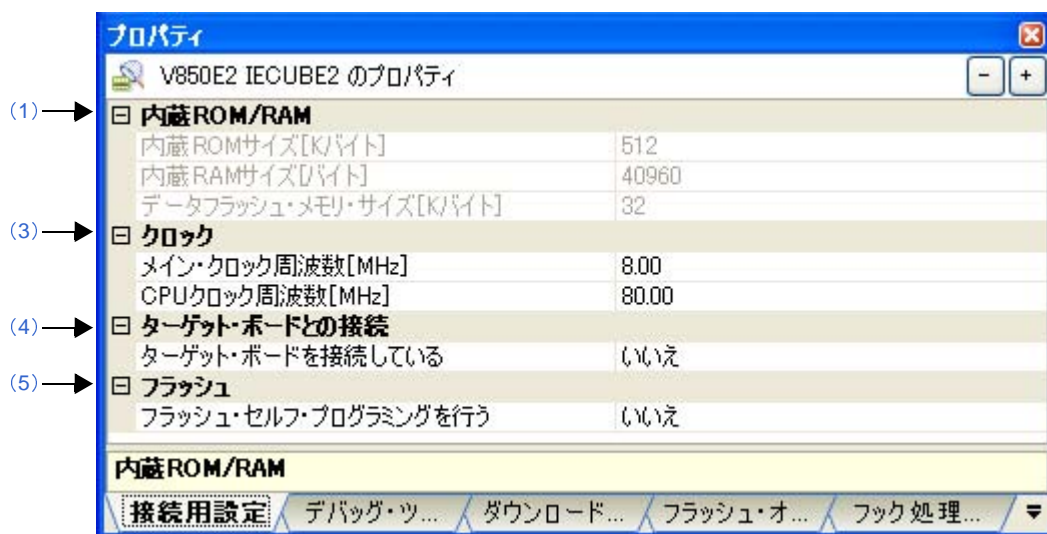


図 A—8 プロパティ パネル：[接続用設定] タブ【MINICUBE】【E1/E20(JTAG)】(【V850E1】【V850ES】)



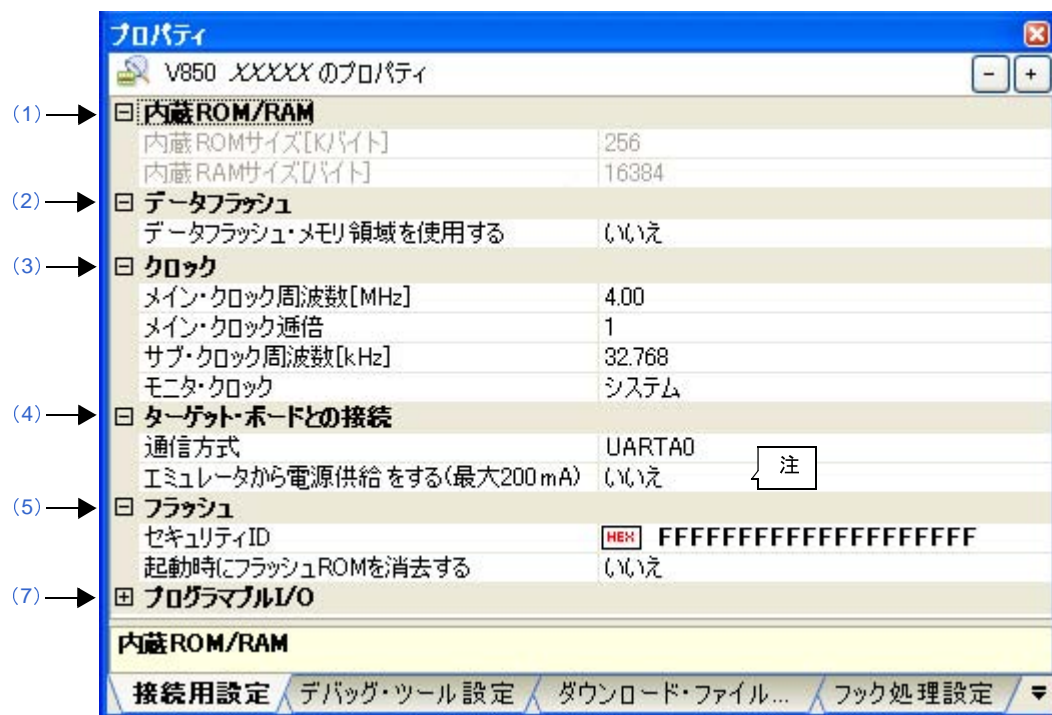
注 [ターゲット・ボードとの接続] カテゴリは、E1(JTAG) の場合のみ表示されます。

図 A—9 プロパティ パネル：[接続用設定] タブ【MINICUBE】【E1/E20(LPD)/(JTAG)】（【V850E2】）



注 [ターゲット・ボードとの接続] カテゴリは、E1(LPD)/(JTAG) の場合のみ表示されます。

図 A—10 プロパティ パネル：[接続用設定] タブ【MINICUBE2】【E1/E20(Serial)】【EZ Emulator】（【V850E1】 【V850ES】）



注 [エミュレータから電源供給する (最大 200mA)] プロパティは、E1(Serial) の場合のみ表示されます。

図 A—11 プロパティ パネル : [接続用設定] タブ【MINICUBE2】【E1/E20(Serial)】(【V850E2】)



注 [エミュレータから電源供給する (最大 200mA)] プロパティは、E1(Serial) の場合のみ表示されます。

図 A—12 プロパティ パネル : [接続用設定] タブ【シミュレータ】(【V850E1】【V850ES】)



図 A—13 プロパティ パネル：[接続用設定] タブ【シミュレータ】（V850E2）



[各カテゴリの説明]

(1) [内蔵 ROM/RAM]

内蔵 ROM/RAM に関する詳細情報の表示、および設定の変更を行います。

注意 1. 設定を変更する際は、他のメモリ・マッピング領域と重複しないよう注意が必要です。

2. [IECUBE]

このカテゴリ内のプロパティ設定を変更すると、自動的に CPU リセットが発生します。

内蔵 ROM サイズ [K バイト]	選択しているマイクロコントローラの内蔵 ROM サイズの表示と変更を行います。
	デフォルト - ROM 内蔵品の場合 選択しているマイクロコントローラの内蔵 ROM サイズ ^{注 1} - ROM レス品の場合 0
	変更方法 【IECUBE】【シミュレータ【V850E1】【V850ES】】 ドロップダウン・リストによる選択 ^{注 2} 【IECUBE2】【MINICUBE】【MINICUBE2】【E1】【E20】【EZ Emulator】 【シミュレータ【V850E2】】 変更不可
指定可能値	【IECUBE】 0, 8, 32, 64, 128, 256, 512, 1024, 1024 + VSB Flash ROM サイズ (VSB Flash ROM 搭載品の場合のみ) (単位: K バイト) 【シミュレータ【V850E1】【V850ES】】 0, 8, 32, 64, 128, 256, 512, 1024 (単位: K バイト)

内蔵 RAM サイズ [バイト]	選択しているマイクロコントローラの内蔵 RAM サイズの表示と変更を行います。	
	デフォルト	選択しているマイクロコントローラの内蔵 RAM サイズ ^{注 3}
	変更方法	【IECUBE】【シミュレータ 【V850E1】 【V850ES】】 ドロップダウン・リストによる選択 【IECUBE2】【MINICUBE】【MINICUBE2】【E1】【E20】【EZ Emulator】 【シミュレータ 【V850E2】】 変更不可
	指定可能値	【IECUBE】 4096, 12288, 28672, 61440, 61440 + VSB RAM サイズ (VSB RAM 搭載品の場合のみ) (単位: バイト) 【シミュレータ 【V850E1】 【V850ES】】 4096, 12288, 28672, 61440 (単位: バイト)
データフラッシュ・メモリ・サイズ [K バイト] 【V850E2】 (【シミュレータ】以外)	選択しているマイクロコントローラのデータフラッシュ・メモリ領域のサイズを表示します。	
	デフォルト	選択しているマイクロコントローラのデータフラッシュ・メモリ・サイズ
	変更方法	変更不可

注 1. 【V850E1】

VSB Flash ROM 搭載品の場合、VSB Flash ROM サイズを加算した値を表示します。

2. ROM レス品の場合、値を変更することはできません。

3. 【V850E1】

VSB RAM 搭載品の場合、VSB RAM サイズを加算した値を表示します。

【V850E2】

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、“合計 RAM サイズ”(コア 1 つ分の RAM サイズ x コアの数) を表示します。

(2) [データフラッシュ] 【V850E1】 【V850ES】 (【シミュレータ】以外)

データフラッシュ・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。

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データフラッシュ・メモリ内蔵品の場合のみ表示されます。

データフラッシュ・メモリ領域を使用する	データフラッシュ・メモリ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
	指定可能値	はい
	いいえ	データフラッシュ・メモリを使用しません。

データフラッシュ・メモリ・サイズ [K バイト]	<p>選択しているマイクロコントローラのデータフラッシュ・メモリ領域のサイズを表示します。</p> <p>なお、このプロパティは、[データフラッシュ・メモリ領域を使用する] 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。</p>	
	デフォルト	選択しているマイクロコントローラのデータフラッシュ・メモリ・サイズ
	変更方法	変更不可
チップ・セレクト	<p>データフラッシュ・メモリをマッピングする際のチップ・セレクトの表示と変更を行います。</p> <p>なお、このプロパティは、[データフラッシュ・メモリ領域を使用する] 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。</p>	
	デフォルト	選択しているマイクロコントローラのチップ・セレクト
	変更方法	<p>【IECUBE】</p> <ul style="list-style-type: none"> - 【V850ES】 <p>ドロップダウン・リストによる選択</p> <ul style="list-style-type: none"> - 【V850E1】 <p>変更不可</p> <p>【IECUBE】 i 以外</p> <p>変更不可</p>
	指定可能値	選択しているマイクロコントローラのチップ・セレクト選択肢 ^注

注 指定できるチップ・セレクトは、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。

(3) [クロック]

クロックに関する詳細情報の表示、および設定の変更を行います。

メイン・クロック周波数 [MHz]	メイン・クロック周波数（逡倍前）を MHz 単位で指定します。 【MINICUBE】【MINICUBE2】【E1】【E20】 このプロパティは、[CPU クロック周波数 [MHz]] プロパティを表示している場合は表示されません。	
	デフォルト	- 【V850E1】【V850ES】 4.00 - 【V850E2】 8.00
	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
	指定可能値	- ドロップダウン・リストによる次のいずれか 【V850E1】【V850ES】 【シミュレータ】以外 2.50, 4.00, 5.00, 6.00, 8.00, 13.50, 18.00（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, 8.00, 8.38, 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（単位：MHz） 【V850E2】 【シミュレータ】以外 4.00, 5.00, 7.20, 8.00, 9.60, 10.00, 13.50, 16.00（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（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] 【V850E2】 （【シミュレータ】以外）	【IECUBE2】 トレース用クロック周波数を MHz 単位で指定します。 【MINICUBE】【MINICUBE2】【E1】【E20】 CPU クロック周波数（逡倍後）を MHz 単位で指定します。 なお、このプロパティは、選択しているマイクロコントローラが V850E2 の一部製品でのみ表示されます。	
	デフォルト	80.00
	変更方法	【IECUBE2】 キーボードからの直接入力 【MINICUBE】【MINICUBE2】【E1】【E20】 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
	指定可能値	- ドロップダウン・リストによる次のいずれか 80.00, 100.00（単位：MHz） - テキスト入力による次の範囲 0.001 ~ 999.999（単位：MHz）

メイン・クロック通倍 【V850E1】【V850ES】 (【シミュレータ】以外)	メイン・クロック周波数の通倍を指定します。	
	デフォルト	1
	変更方法	ドロップダウン・リストによる選択、またはキーボードからの直接入力
	指定可能値	- ドロップダウン・リストによる次のいずれか 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 - テキスト入力による次の範囲 1 ~ 99 までの整数値
サブ・クロック周波数 [kHz] 【V850E1】【V850ES】	サブ・クロックの周波数を kHz 単位で指定します。 なお、このプロパティは、選択しているマイクロコントローラがサブ・クロックをサポートしている場合のみ表示されます。	
	デフォルト	【シミュレータ】以外 <i>選択しているマイクロコントローラのサブ・クロック周波数選択肢の最小値</i> 【シミュレータ】 32.768
	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
	指定可能値	- ドロップダウン・リストによる次のいずれか 【シミュレータ】以外 <i>選択しているマイクロコントローラのサブ・クロック周波数選択肢^注</i> 【シミュレータ】 32.768 - テキスト入力による次の範囲 0.001 ~ 999.999 (単位: kHz)
モニタ・クロック 【V850E1】【V850ES】 (【シミュレータ】以外)	プログラム停止中にモニタ・プログラムが動作するクロックを選択します。 なお、このプロパティは、選択しているマイクロコントローラがサブ・クロックをサポートしている場合のみ表示されます。	
	デフォルト	システム
	変更方法	ドロップダウン・リストによる選択
	指定可能値	システム ユーザ メイン・クロックで動作します。 プログラムで設定されているクロックで動作しません。
TCK/DCK クロック [MHz] 【V850E1】【V850ES】 【MINICUBE】 【E1/E20(JTAG)】	デバッグ・コントロール・ユニット (DCU) へ供給するクロックを選択します。 ただし、通常は、[10MHz] の指定を変更しないでください。[20MHz] を指定した場合、デバッグ・ツールと接続できない可能性があります。	
	デフォルト	10MHz
	変更方法	ドロップダウン・リストによる選択 ただし、デバッグ・ツールと切断中の場合のみ変更可
	指定可能値	10MHz, 20MHz

タイマ用クロック周波数 [MHz] 【V850E2】 (【IECUBE2】【シミュ レータ】以外)	タイマ用のクロック周波数を指定します。				
	デフォルト	80.00			
	変更方法	キーボードからの直接入力 ただし、[CPUクロック周波数 [MHz]] プロパティを表示している場合は 編集不可 ([CPUクロック周波数 [MHz]] と同値に固定)			
	指定可能値	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/トレース機能を使用する際のクロック周波数を選択します。				
	デフォルト	CPUクロック周波数			
	変更方法	ドロップダウン・リストによる選択			
	指定可能値	<table border="1"> <tr> <td>CPUクロック周波 数</td> <td>CPUクロック周波数を使用します。</td> </tr> <tr> <td>周波数の指定</td> <td>任意の周波数を使用します (下段のプロパティ項 目が有効となります)。</td> </tr> </table>	CPUクロック周波 数	CPUクロック周波数を使用します。	周波数の指定
CPUクロック周波 数	CPUクロック周波数を使用します。				
周波数の指定	任意の周波数を使用します (下段のプロパティ項 目が有効となります)。				
タイマ/トレース用ク ロック周波数の単位 【シミュレータ】	タイマ/トレース用クロックの周波数の単位を選択します。 なお、このプロパティは、[タイマ/トレース用クロック周波数の選択] プロパティに おいて [周波数の指定] を指定した場合のみ有効となります。				
	デフォルト	MHz			
	変更方法	ドロップダウン・リストによる選択			
	指定可能値	<table border="1"> <tr> <td>MHz</td> <td>周波数の単位を MHz とします。</td> </tr> <tr> <td>KHz</td> <td>周波数の単位を kHz とします。</td> </tr> </table>	MHz	周波数の単位を MHz とします。	KHz
MHz	周波数の単位を MHz とします。				
KHz	周波数の単位を kHz とします。				
タイマ/トレース用ク ロック周波数 【シミュレータ】	このプロパティは、[タイマ/トレース用クロック周波数の選択] プロパティでの指定 により、動作が異なります。 - [周波数の指定] の場合 タイマ/トレース用クロックの周波数を指定します。 - [CPUクロック周波数] の場合 次を表示します (変更不可)。 デバッグ・ツールと切断中: [---_---] デバッグ・ツールと接続中: [CPUクロック周波数]				
	デフォルト	- 【V850E1】【V850ES】 4.00 - 【V850E2】 8.00			
	変更方法	キーボードからの直接入力			
	指定可能値	1 kHz ~ 999.999 MHz ただし、単位は [タイマ/トレース用クロック周波数の単位] プロパ ティの指定に依存			

注 指定できるサブ・クロック周波数は、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。
 なお、指定できる周波数が固定の場合はドロップダウン・リストは表示されません。

(4) [ターゲット・ボードとの接続] ([MINICUBE] [E20(LPD)/(JTAG)] [シミュレータ] 以外)

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。

ターゲット・ボードを接続している 【IECUBE】 【IECUBE2】	IECUBE/IECUBE2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。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択 ただし、デバッグ・ツールと切断中の場合のみ変更可	
	指定可能値	はい	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。	
通信方式 【MINICUBE2】 【E1/E20(Serial)】 【EZ Emulator】	エミュレータが、ターゲット・ボード上のマイクロコントローラとシリアル通信を行う際に接続する通信方式を選択します。		
	デフォルト	選択しているマイクロコントローラに依存	
	変更方法	ドロップダウン・リストによる選択 ただし、デバッグ・ツールと切断中の場合のみ変更可	
	指定可能値	選択しているマイクロコントローラに依存 ^注	
エミュレータから電源供給をする (最大 200mA) 【E1】	E1からターゲット・ボードに電源を供給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 ただし、デバッグ・ツールと切断中の場合のみ変更可	
	指定可能値	はい	電源を供給します。
いいえ		電源を供給しません。	
供給電圧 【E1】	E1からターゲット・ボードに供給する電圧を選択します。 なお、このプロパティは、[エミュレータから電源供給をする (最大 200mA)] プロパティが表示されている際に [はい] を指定した場合のみ表示されます。		
	デフォルト	3.3V	
	変更方法	ドロップダウン・リストによる選択 ただし、デバッグ・ツールと切断中の場合のみ変更可	
	指定可能値	3.3V, 5.0V	

注 ドロップダウン・リスト内に表示されるポートの種類は、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。

(5) [フラッシュ] (【IECUBE】【シミュレータ】以外)

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。

セキュリティ ID 【MINICUBE】 【MINICUBE2】 【E1】【E20】 【EZ Emulator】	内蔵 ROM、または内蔵フラッシュ・メモリ上のコードを読み出す際のセキュリティ ID を指定します。 ^注 。 なお、このプロパティは、選択しているマイクロコントローラが、フラッシュ・メモリの ROM セキュリティ機能（オンチップ・デバッグ・セキュリティ ID）をサポートしている場合のみ表示されます。				
	デフォルト	- 【V850E1】【V850ES】 FFFFFFFFFFFFFFFFFF - 【V850E2】 FFFFFFFFFFFFFFFFFF			
	変更方法	キーボードからの直接入力 ただし、デバッグ・ツールと切断中の場合のみ変更可			
	指定可能値	- 【V850E1】【V850ES】 20 桁の 16 進数（10 バイト） - 【V850E2】 24 桁の 16 進数（12 バイト）			
起動時にフラッシュ ROM を消去する 【MINICUBE2】 【E1/E20(Serial)】 【EZ Emulator】	デバッグ・ツールと接続する際に、フラッシュ ROM を消去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 ただし、デバッグ・ツールと切断中の場合のみ変更可			
	指定可能値	<table border="1"> <tr> <td>はい</td> <td> デバッグ・ツールへ接続する際に、フラッシュ ROM の消去を行います。 なお、デバッグ・ツールと接続後は、自動的に [いいえ] の指定となります。 </td> </tr> <tr> <td>いいえ</td> <td> デバッグ・ツールと接続する際に、フラッシュ ROM の消去は行いません。 </td> </tr> </table>	はい	デバッグ・ツールへ接続する際に、フラッシュ ROM の消去を行います。 なお、デバッグ・ツールと接続後は、自動的に [いいえ] の指定となります。	いいえ
はい	デバッグ・ツールへ接続する際に、フラッシュ ROM の消去を行います。 なお、デバッグ・ツールと接続後は、自動的に [いいえ] の指定となります。				
いいえ	デバッグ・ツールと接続する際に、フラッシュ ROM の消去は行いません。				
フラッシュ・セルフ・プログラミングを行う 【V850E2】 【IECUBE2】 【MINICUBE】 【E1/E20(LPD)】 【E1/E20(JTAG)】	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、内蔵フラッシュ ROM を書き換えるか否かを選択します。 ただし、ホット・プラグイン接続時は、この設定は無効となります。				
	デフォルト	いいえ			
	変更方法	ドロップダウン・リストによる選択 ただし、デバッグ・ツールと切断中の場合のみ変更可			
	指定可能値	<table border="1"> <tr> <td>はい</td> <td> 内蔵フラッシュ ROM の書き換えを行います。この場合、内蔵フラッシュ ROM のキャッシュは行われません。 </td> </tr> <tr> <td>いいえ</td> <td> 内蔵フラッシュ ROM の書き換えを行いません。 </td> </tr> </table>	はい	内蔵フラッシュ ROM の書き換えを行います。この場合、内蔵フラッシュ ROM のキャッシュは行われません。	いいえ
はい	内蔵フラッシュ ROM の書き換えを行います。この場合、内蔵フラッシュ ROM のキャッシュは行われません。				
いいえ	内蔵フラッシュ ROM の書き換えを行いません。				

注 プロパティの設定を変更すると、ビルド・ツールのセキュリティ ID を変更するか否かを確認するメッセージを表示します（選択しているマイクロコントローラが V850E2 コアの場合を除く）。
 なお、オンチップ・デバッグ・セキュリティ ID についての詳細は、エミュレータのユーザーズ・マニュアルを参照してください。

(6) [コンフィギュレーション]【シミュレータ】

シミュレータをカスタマイズする際の詳細情報の表示、および設定の変更を行います。

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	シミュレータに対して、ユーザ・カスタマイズ（ユーザ・モデルの追加）を行うためのシミュレータ・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 ただし、デバッグ・ツールと切断中の場合のみ変更可
	指定可能値	はい
いいえ		シミュレータ・コンフィギュレーション・ファイルを使用しません。
シミュレータ・コンフィ ギュレーション・ファイル	使用するシミュレータ・コンフィギュレーション・ファイルを指定します。 なお、このプロパティは、[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] プロパティにおいて [はい] を指定した場合のみ表示されます。	
	デフォルト	空欄
	変更方法	キーボードからの直接入力、または [...] ボタンのクリックによるオープンするシミュレータ・コンフィギュレーション・ファイルを選択 ダイアログ【シミュレータ】 によるファイルの選択 ただし、デバッグ・ツールと切断中の場合のみ変更可

(7) [プログラマブル I/O]【V850E1】【V850ES】

プログラマブル I/O 領域に関する詳細情報の表示、および設定の変更を行います。

なお、このカテゴリでの設定は、デバッグを行う際に一時的に値を変更する場合にのみ行ってください。プロジェクトとして共通の値を設定する場合は、使用するビルド・ツールのプロパティ パネルの [共通オプション] タブ上の [デバイス] カテゴリ内 [プログラマブル I/O 領域開始アドレス] プロパティで設定を行ってください。

プログラマブル I/O 領域 を使用する	プログラマブル I/O 領域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 ただし、選択しているマイクロコントローラがプログラマブル I/O 領域をサポートしている場合のみ変更可
	指定可能値	はい
いいえ		プログラマブル I/O 領域を使用しません。
プログラマブル I/O 領域 開始アドレス	プログラマブル I/O 領域の開始アドレスを指定します。アドレスは、16 K バイトでアラインされます。 なお、このプロパティは、[プログラマブル I/O 領域を使用する] 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	0
	変更方法	キーボードからの直接入力
	指定可能値	プログラマブル I/O 領域のアドレス範囲内

[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] (【シミュレータ】以外)
- (4) [ブレーク]
- (5) [フェイルセーフ・ブレーク] 【IECUBE】
- (6) [トレース] 【IECUBE】 【IECUBE2】 【シミュレータ】
- (7) [タイマ] 【IECUBE】 【IECUBE2】 【シミュレータ】
- (8) [カバレッジ] 【IECUBE】 【IECUBE2】 【シミュレータ】
- (9) [入力信号のマスキ] (【シミュレータ】以外)
- (10) [外部フラッシュ・メモリ・ダウンロード] 【IECUBE】 【IECUBE2】 【MINICUBE】 【E1/E20(LPD)/(JTAG)】
- (11) [シミュレータ GUI] 【シミュレータ】

図 A—14 プロパティ パネル: [デバッグ・ツール設定] タブ 【IECUBE】



図 A—15 プロパティ パネル: [デバッグ・ツール設定] タブ [IECUBE2]



図 A—16 プロパティ パネル: [デバッグ・ツール設定] タブ [MINICUBE] [E1/E20(JTAG)] ([V850E1] [V850ES])



図 A—17 プロパティ パネル：[デバッグ・ツール設定] タブ【MINICUBE】[E1/E20(LPD)/(JTAG)] (〔V850E2〕)



図 A—18 プロパティ パネル：[デバッグ・ツール設定] タブ【MINICUBE2】[E1/E20(Serial)] [EZ Emulator]

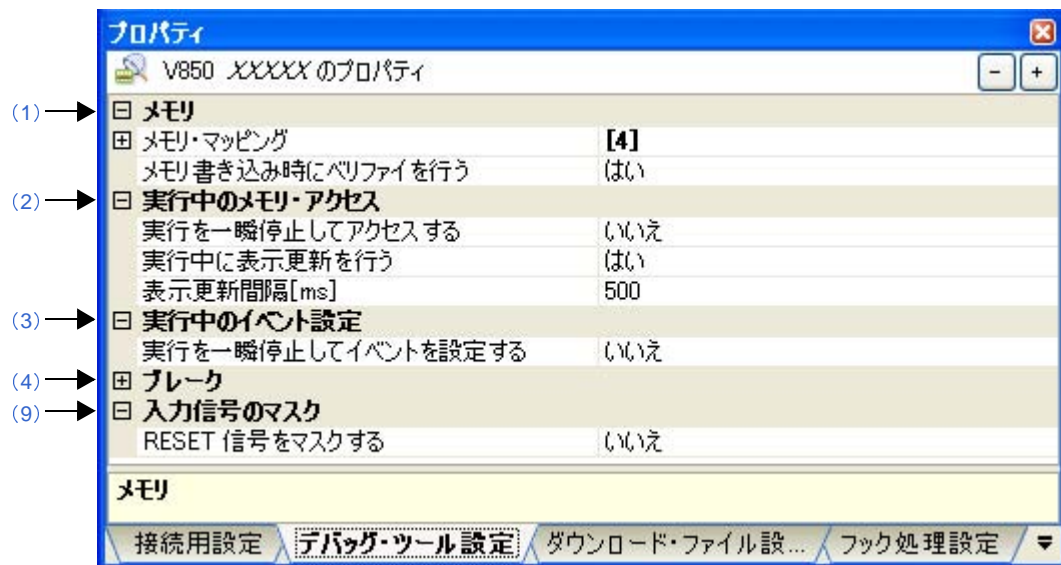


図 A-19 プロパティ パネル : [デバッグ・ツール設定] タブ【シミュレータ】 (【V850E1】 【V850ES】)



図 A-20 プロパティ パネル : [デバッグ・ツール設定] タブ【シミュレータ】 (【V850E2】)



[各カテゴリの説明]

(1) [メモリ]

メモリに関する詳細情報の表示、および設定の変更を行います。

なお、表示されるメモリ種別についての詳細は、[メモリ・マッピング ダイアログ](#)の項を参照してください。

メモリ・マッピング	現在のメモリ・マッピングの状況をメモリ領域の種別 ^注 ごとに表示します。	
	デフォルト	[マイクロコントローラ固有のメモリ・マッピング領域種別の合計]
	変更方法	メモリ・マッピング ダイアログによる指定 メモリ・マッピング ダイアログは、このプロパティを選択すると右端に表示される [...] ボタンをクリックすることでオープンします（このパネル上でメモリ・マッピングを変更することはできません）。
	表示内容	メモリ・マッピングの状況をメモリ領域の種別ごとに表示します。 なお、各メモリ種別の“+”マークをクリックすると、次の詳細情報を表示します。 - メモリ種別 - 開始アドレス - 終了アドレス - アクセス幅 [ビット]
メモリ書き込み時にベリファイを行う (【シミュレータ】以外)	メモリ値の初期化を行う際に、ベリファイを行うか否かを選択します。	
	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
	指定可能値	はい ベリファイを行います。 いいえ ベリファイを行いません。

注 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 【V850E2】 【IECIBE2】 【MINICUBE】 【E1/E20(LPD)】 【E1/E20(JTAG)】	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。 [フラッシュ] カテゴリ内 [フラッシュ・セルフ・プログラミングを行う] プロパティにおいて [いいえ] を選択している場合は、内蔵 ROM 領域も対象となります。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい 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。

実行を一瞬停止してアクセスする (【シミュレータ】以外)	プログラム実行中にはアクセスできないメモリ領域（ターゲット・メモリ / I/O レジスタ）に対して、実行を一瞬停止し、アクセスを許可するか否かを選択します。 ただし、内蔵 RAM 領域に対しては、この指定にかかわらず、一瞬停止することなくアクセスします（【フラッシュ】カテゴリ内【フラッシュ・セルフ・プログラミングを行う】プロパティにおいて【いいえ】を選択している場合は、内蔵 ROM 領域も同様です）。 【V850E2】 このプロパティは、【実行中にアクセスする】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。 ただし、選択しているマイクロコントローラがマルチコア版の場合、このプロパティは表示されません。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい 実行を一瞬停止し、アクセスします。 いいえ 実行中にメモリにアクセスしません。
HALT を解除してアクセスする 【IECUBE】 【IECUBE2】 【MINICUBE】 【E1/E20(LPD)】 【E1/E20(JTAG)】	HALT モードを解除して、内蔵 RAM 領域以外のメモリ領域（CPU レジスタ / I/O レジスタ）にアクセスするか否かを選択します。 【V850E2】 このプロパティは、【実行を一瞬停止してアクセスする】プロパティにおいて【はい】を指定した場合にのみ表示されます。	
	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 内蔵 RAM 領域以外に対し、HALT モードを解除してアクセスします。 いいえ 内蔵 RAM 領域以外は、HALT モード中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 の表示内容を自動的に更新するか否かを選択します。 【IECUBE】 このプロパティは、【トレース】カテゴリ内【トレース・データの用途】プロパティにおいて【トレース】を指定した場合、無効となります。	
	デフォルト	はい
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい 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 の表示内容を自動的に更新する間隔を 100 ms 単位で指定します。 なお、このプロパティは、【実行中に表示更新を行う】プロパティにおいて【はい】を指定した場合のみ表示されます。	
	デフォルト	500
	変更方法	キーボードからの直接入力
	指定可能値	100 ～ 65500 の整数（単位：100 ms 未満の端数切り上げ）

リアルタイム表示更新を自動設定する 【IECUBE】	ウォッチパネル／メモリパネルで表示している領域を、可能なかぎり自動的にリアルタイム表示更新機能の対象領域に設定し、プログラム実行中に表示内容を更新します。 なお、このプロパティは、[実行中に表示更新を行う] プロパティにおいて [はい] を指定した場合のみ表示されます。	
	デフォルト	はい
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい 自動的にリアルタイム表示更新の設定を行います。 いいえ リアルタイム表示更新設定はしません。

(3) [実行中のイベント設定] (【シミュレータ】以外)

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 (「(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。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい プログラムの実行、またはトレーサ／タイマの動作を一瞬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]

ブレーク機能に関する詳細情報の表示、および設定の変更を行います。

優先的に使用するブレークポイントの種類 (【シミュレータ】以外)	エディタパネル／逆アセンブルパネルにおいて、ソース行、または実行アドレスに対してマウスのワンクリック操作でブレークポイントを設定する際に、優先的に使用するブレークポイントの種別を選択します。 なお、このプロパティは、選択しているマイクロコントローラがブレークポイントの種類を1つのみしかサポートしていない場合は表示されません。	
	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
	指定可能値	ソフトウェア・ブレーク ソフトウェア・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 ハードウェア・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ションを停止する 【IECUBE】 【IECUBE2】 【MINICUBE】 【E1/E20(LPD)】 【E1/E20(JTAG)】	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		
	指定可能値	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。	
オープン・ブ레이크機能を使用する 【IECUBE】 【IECUBE2】 【MINICUBE】 【E1/E20(LPD)】 【E1/E20(JTAG)】	オープン・ブ레이크機能を使用するか否かを選択します。 なお、このプロパティは、選択しているマイクロコントローラがオープン・ブ레이크機能をサポートしている場合のみ表示されます。			
	デフォルト	いいえ (Output signal)		
	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 (Hi-Z)	CPU 停止時、オープン・ブ레이크の対象端子は Hi-Z になります。	
		いいえ (Output signal)	CPU 停止時、オープン・ブ레이크の対象端子は通常の状態のままで動作します。	
停止時にブ레이크位置の命令を実行 【シミュレータ】	ブ레이크ポイントによるプログラム実行停止のタイミングを、ブ레이크ポイントが設定されている位置の命令実行後とするか、または命令実行前とするかを選択します。			
	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		
	指定可能値	はい	命令実行後にブ레이크します ^注 。	
いいえ		命令実行前にブ레이크します。		

注 [はい] を指定した場合、現在設定されているアクション・イベントは、すべてブハードウェア・ブ레이크・イベントとして動作します（「2.14 プログラム内へのアクションの設定」参照）。

(5) [フェイルセーフ・ブ레이크] 【IECUBE】

フェイルセーフ・ブ레이크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。

内蔵 ROM への書き込み直後に停止する	内蔵 ROM への書き込み直後に、実行を停止するか否かを選択します ^注 。 なお、このプロパティは、選択しているマイクロコントローラが ROM レス品の場合は無効となります。		
	デフォルト	はい	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	書き込み直後に実行を停止します。
いいえ		書き込み後も実行を停止しません。	

内蔵 ROM の非メモリ・マッピング領域へのアクセス直後に停止する	内蔵 ROM の非メモリ・マッピング領域へのアクセス直後に、実行を停止するか否かを選択します ^注 。 なお、このプロパティは、選択しているマイクロコントローラが ROM レス品の場合は無効となります。	
	デフォルト	はい
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい アクセス直後に実行を停止します。 いいえ アクセス後も実行を停止しません。
内蔵 RAM の非メモリ・マッピング領域へのアクセス直後に停止する	内蔵 RAM の非メモリ・マッピング領域へのアクセス直後に、実行を停止するか否かを選択します ^注 。	
	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 アクセス直後に実行を停止します。 いいえ アクセス後も実行を停止しません。
内蔵 RAM の書き込み禁止領域への書き込みを検出する	プログラム停止時に、内蔵 RAM の書き込み禁止領域への書き込みを、ペリファイ・チェックすることにより検出するか否かを選択します ^注 。	
	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 書き込みを検出するとメッセージを表示します。 いいえ 書き込みのチェックを行いません。
I/O レジスタ領域内の読み込み禁止領域からの読み込み直後に停止する	I/O レジスタ領域内の読み込み禁止領域からの読み込み直後に、実行を停止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
	指定可能値	はい 読み込み直後に実行を停止します。 いいえ 読み込み後も実行を停止しません。
I/O レジスタ領域内の書き込み禁止領域への書き込み直後に停止する	I/O レジスタ領域内の書き込み禁止領域への書き込み直後に、実行を停止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
	指定可能値	はい 書き込み直後に実行を停止します。 いいえ 書き込み後も実行を停止しません。
I/O レジスタ領域内の非マッピング領域にアクセス直後に停止する	I/O レジスタ領域内の非マッピング領域にアクセス直後に、実行を停止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
	指定可能値	はい アクセス直後に実行を停止します。 いいえ アクセス後も実行を停止しません。

書き込み禁止外部メモリへの書き込み直後に停止する	書き込み禁止外部メモリへの書き込み直後に、実行を停止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	
	指定可能値	はい	書き込み直後に実行を停止します。
いいえ		書き込み後も実行を停止しません。	
外部メモリの非メモリ・マッピング領域へのアクセス直後に停止する	外部メモリの非メモリ・マッピング領域へのアクセス直後に、実行を停止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	
	指定可能値	はい	アクセス直後に実行を停止します。
いいえ		アクセス後も実行を停止しません。	

注 プロパティを設定すると、自動的に CPU リセットが発生します。

(6) [トレース] 【IECUBE】 【IECUBE2】 【シミュレータ】

トレース機能に関する詳細情報の表示、および設定の変更を行います。

注意 1. 【IECUBE】

トレース機能／リアルタイム表示更新機能（RRM 機能）／カバレッジ機能は一部排他使用の機能です。

2. 【IECUBE2】

トレース機能／タイマ機能（Run-Break タイマを除く）／カバレッジ機能は一部排他使用の機能です。

トレース・データの用途 【IECUBE】 【IECUBE2】	【IECUBE】 トレース機能／リアルタイム表示更新機能（RRM 機能）／カバレッジ機能は一部排他使用の機能であるため、どの機能を優先的に使用するかを選択します。 【IECUBE2】 トレース機能／タイマ機能（Run-Break タイマを除く）／カバレッジ機能は一部排他使用の機能であるため、どの機能を優先的に使用するかを選択します。 なお、このプロパティについての詳細は、「2. 11. 1 トレース動作の設定をする」を参照してください。		
	デフォルト	【IECUBE】 RRM 【IECUBE2】 トレース	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	トレース	トレース機能を優先的に使用します。
		【IECUBE】 RRM	リアルタイム表示更新機能（RRM 機能）を優先的に使用します。
		カバレッジ	カバレッジ機能を優先的に使用します。
	指定可能値	【IECUBE2】 トレース	トレース機能を優先的に使用します。
タイマ		タイマ機能を優先的に使用します。	
カバレッジ		カバレッジ機能を優先的に使用します。	

トレース・データの選択 【IECUBE】 【IECUBE2】	収集するトレース・データの種類を選択します（「2. 11. 1 トレース動作の設定をする」参照）。 なお、このプロパティは、[トレース・データの用途] プロパティにおいて [トレース] を選択した場合のみ表示されます。		
	デフォルト	分岐命令とアクセス・アドレス／データ値	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	分岐命令	
		全命令 【IECUBE】	
		アクセス・アドレス／データ値	
		分岐命令とアクセス・アドレス／データ値	
全命令とアクセス・アドレス／データ値 【IECUBE】			
アクセス命令とアクセス・アドレス／データ値			
	分岐／アクセス命令とアクセス・アドレス／データ値		

DMA トレースを行う 【IECUBE2】	DMA によるアクセス・データをトレース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
	指定可能値	はい
いいえ		DMA トレースを行いません。

DMA トレース・データの 選択 【IECUBE2】	収集する DMA トレース・データの種類を選択します。 なお、このプロパティは、[DMA トレースを行う] 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。 なお、このプロパティについての詳細は、「2.11.1 トレース動作の設定をする」を参照してください。		
	デフォルト	アクセス種別／アクセス・アドレス／データ・サイズ／データ値	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	アクセス種別／アクセス・アドレス／データ・サイズ／データ値	
		アクセス種別／アクセス・アドレス／データ・サイズ	
		アクセス種別／チャンネル番号	
アクセス種別／チャンネル番号／転送回数			
トレースの優先度 【IECUBE】 【IECUBE2】	トレース・データを収集する際の優先度を選択します。 なお、このプロパティは、[トレース・データの用途] プロパティにおいて [トレース] を選択した場合のみ表示されます。		
	デフォルト	スピード優先	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	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。
トレース機能を使用する 【シミュレータ】	トレース機能を使用するか否かを選択します ^{注1} 。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	トレース機能を使用します。
		いいえ	トレース機能を使用しません。
実行前にトレース・メモリをクリアする	実行前にトレース・メモリをクリア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	
	指定可能値	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・データで満たされた際の動作を選択します。		
	デフォルト	トレース・メモリを上書きし実行を続ける	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	トレース・メモリを上書きし実行を続ける トレースを停止する 停止する	トレース・メモリを使い切ると、古いトレース・データを上書きを続けます。 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（実行は停止しません）。 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・タグを積算する 【シミュレータ】	トレースパネル【IECUBE】【IECUBE2】【シミュレータ】に表示するトレース時間の表示方法を選択します。		
	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	
	指定可能値	はい いいえ	トレースの時間表示を積算値で表示します。 トレースの時間表示を差分値で表示します。
トレースのタイム・タグの分周率 【IECUBE】 【シミュレータ】	トレースのタイム・タグ表示（トレースパネル【IECUBE】【IECUBE2】【シミュレータ】の「時間」表示）で使用するカウンタの分周率を選択します。 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・アップに必要なクロック数も変更されます。		
	デフォルト	【IECUBE】 1/1(20ns/1.4min) 【シミュレータ】 1/1	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	【IECUBE】 1/1(20ns/1.4min), 1/2(40ns/2.8min), 1/4(80ns/5.7min), 1/8(160ns/11.4min), 1/16(320ns/22.8min), 1/32(640ns/45.6min), 1/64(1280ns/1.5h), 1/128(2560ns/3.0h), 1/256(5120ns/6.0h), 1/512(10240ns/12.2h), 1/1024(20480ns/24.4h), 1/2048(40960ns/48.8h), 1/4096(81920ns/97.6h) (“/”内は、分解能、およびタイマ・カウンタが32ビットで外部クロックが50MHzの場合の最大測定時間を示す) 【シミュレータ】 1/1, 1/2, 1/4, 1/8, 1/16, 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・メモリ・サイズ [フレーム]	トレース・データを格納するメモリ・サイズ (トレース・フレーム ^{注2} 数) を選択します。	
	デフォルト	【IECUBE】 【IECUBE2】 8K 【シミュレータ】 4K
	変更方法	ドロップダウン・リストによる選択
	指定可能値	【IECUBE】 8K, 32K, 64K, 128K, 256K 【IECUBE2】 8K, 32K, 64K, 128K, 256K, 512K 【シミュレータ】 4K, 8K, 12K, 16K, 20K, 24K, 28K, 32K, 36K, 40K, 44K, 48K, 52K, 56K, 60K, 64K, 128K, 192K, 256K, 320K, 384K, 448K, 512K, 576K, 640K, 704K, 768K, 832K, 896K, 960K, 1M, 2M, 3M
トレースを補完する 【IECUBE】 【IECUBE2】	収集したトレース・データを表示する際に、補完表示するか否かを選択します ^{注3} 。	
	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 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。

- 注1. エディタ パネル / 逆アセンブル パネルにおいて、コンテキスト・メニュー → [トレース開始の設定] / [トレース終了の設定] を選択した場合、このプロパティは自動的に [はい] に変更されます。
2. トレース・フレームはトレース・データの一単位を表します。
フェッチ / ライト / リードなどで、それぞれ1つのトレース・フレームを使用します。
3. 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補完表示が可能となります。
プロパティの設定を変更すると、自動的にトレース・メモリがクリアされます。

(7) [タイマ] 【IECUBE】 【IECUBE2】 【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。

注意 【IECUBE2】

[トレース] カテゴリ内 [トレース・データの用途] プロパティにおいて、[タイマ] 以外を指定している場合、このカテゴリ内のプロパティは無効となります。

タイマの分周率 【IECUBE】 【IECUBE2】	【IECUBE】 タイマ計測に使用するタイマ・カウンタ（32ビット/50MHz）の分周率を選択します。	
	【IECUBE2】 タイマ計測に使用するタイマ・カウンタ（33ビット/200MHz）の分周率を選択します。	
	デフォルト	【IECUBE】 1/1(20ns/1.4min) (“()”内は分解能/最大測定時間を示す) 【IECUBE2】 1/1(5ns/42.9s) (“()”内は分解能/最大測定時間を示す)
	変更方法	ドロップダウン・リストによる選択
指定可能値	【IECUBE】 1/1(20ns/1.4min), 1/2(40ns/2.8min), 1/4(80ns/5.7min), 1/8(160ns/11.4min), 1/16(320ns/22.8min), 1/32(640ns/45.6min), 1/64(1280ns/1.5h), 1/128(2560ns/3.0h), 1/256(5120ns/6.0h), 1/512(10240ns/12.2h), 1/1024(20480ns/24.4h), 1/2048(40960ns/48.8h), 1/4096(81920ns/97.6h) 【IECUBE2】 1/1(5ns/42.9s), 1/2(10ns/1.4min), 1/4(20ns/2.8min), 1/8(40ns/5.71min), 1/16(80ns/11.42min), 1/32(160ns/22.8min), 1/64(320ns/45.6min), 1/128(640ns/1.5h), 1/256(1280ns/3.0h), 1/512(2560ns/6.0h), 1/1024(5120ns/12.2h), 1/2048(10240ns/24.4h), 1/4096(20480ns/48.8h), 1/8192(40960ns/97.6h), 1/16384(81920ns/195.2h)	
タイマ機能を使用する 【シミュレータ】	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
	指定可能値	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。

(8) [カバレッジ] 【IECUBE】 【IECUBE2】 【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。

注意 1. 【IECUBE】

使用する IECUBE にカバレッジ・ボードが搭載されていない場合、または [トレース] カテゴリ内 [トレース・データの用途] プロパティにおいて、[カバレッジ] 以外を指定している場合、このカテゴリ内のプロパティは無効となります。

2. 【IECUBE2】

[トレース] カテゴリ内 [トレース・データの用途] プロパティにおいて、[カバレッジ] 以外を指定している場合、このカテゴリ内のプロパティは無効となります。

カバレッジ機能を使用する 【シミュレータ】	カバレッジ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
	指定可能値	はい いいえ
		カバレッジ機能を使用します。 カバレッジ機能を使用しません。
カバレッジ結果を再利用する	デバッグ・ツールと接続時/切断時に、カバレッジ測定結果のロード/セーブを行うか否かを選択します。 【IECUBE】【IECUBE2】 このプロパティは、[トレース・データの用途] プロパティにおいて [カバレッジ] を指定した場合のみ表示されます。 【シミュレータ】 このプロパティは、[カバレッジ機能を使用する] プロパティにおいて [はい] を指定した場合のみ表示されます。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい いいえ
		カバレッジ測定結果のロード/セーブを行います。 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 バイト) 【IECUBE】 【シミュレータ】	内蔵 ROM 領域以外の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任意の 1M バイト空間の開始アドレスを指定します。 【IECUBE】 このプロパティは、[トレース・データの用途] プロパティにおいて [カバレッジ] を指定した場合のみ表示されます。 【シミュレータ】 このプロパティは、[カバレッジ機能を使用する] プロパティにおいて [はい] を指定した場合のみ表示されます。	
	デフォルト	内蔵 RAM の開始アドレス
	変更方法	キーボードからの直接入力
	指定可能値	内蔵 ROM 領域以外のアドレス (シンボル指定不可)

(9) [入力信号のマスク] (【シミュレータ】以外)

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。

NMIn 信号をマスクする 【IECUBE】 【MINICUBE】 【E1/E20(LPD)】 【E1/E20(JTAG)】	NMIn 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	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい いいえ
		NMIn 信号をマスクします。 NMIn 信号をマスクしません。

HLDQR 信号をマスクする 【IECUBE】 【IECUBE2】 【MINICUBE】 【E1/E20(LPD)】 【E1/E20(JTAG)】	HLDQR 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	いいえ ^{注1}	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	HLDQR 信号をマスクします。
いいえ		HLDQR 信号をマスクしません。	
RESET 信号をマスクする (【IECUBE2】 以外)	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。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	RESET 信号をマスクします。
いいえ		RESET 信号をマスクしません。	
STOP 信号をマスクする 【IECUBE】 【IECUBE2】 【MINICUBE】 【E1/E20(LPD)】 【E1/E20(JTAG)】	STOP 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	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	STOP 信号をマスクします。
いいえ		STOP 信号をマスクしません。	
WAIT 信号をマスクする 【IECUBE】 【IECUBE2】 【MINICUBE】 【E1/E20(LPD)】 【E1/E20(JTAG)】	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	いいえ ^{注1}	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	WAIT 信号をマスクします。
いいえ		WAIT 信号をマスクしません。	
DBINT 信号をマスクする 【850E1】 【850ES】 【MINICUBE】 【E1/E20(JTAG)】	DBINT 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	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	DBINT 信号をマスクします。
いいえ		DBINT 信号をマスクしません。	
RESET 信号のマスク 【IECUBE2】	マスクする RESET 信号を選択します。		
	デフォルト	TARGET RESET 信号をマスクする	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	マスクしない ^{注2}	RESET 信号をマスクしません。
TARGET RESET 信号をマスクする		TARGET RESET 信号をマスクします。	
TARGET RESET 信号と INTERNAL RESET 信号をマスクする		TARGET RESET 信号と INTERNAL RESET 信号をマスクします。	

注 1. 【IECUBE】 【IECUBE2】

【接続用設定】 タブ上の 【ターゲット・ボードとの接続】 (【MINICUBE】 【E20(LPD)/(JTAG)】 【シミュレータ】 以外) 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)。

2. 【IECUBE2】

【接続用設定】タブ上の【ターゲット・ボードとの接続】(【MINICUBE】【E20(LPD)/(JTAG)】【シミュレータ】以外) 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、この項目は表示されません。

(10) 【外部フラッシュ・メモリ・ダウンロード】【IECUBE】【IECUBE2】【MINICUBE】【E1/E20(LPD)/(JTAG)】

外部フラッシュ・メモリへの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。

外部フラッシュ・メモリ・ダウンロード設定	外部フラッシュ・メモリにダウンロードする際の外部フラッシュ・メモリ情報ファイル (*.fdb) の選択、およびダウンロード条件を指定します。 サブプロパティとして、登録された外部フラッシュ・メモリ情報ファイルに対応するダウンロード条件を下段に展開表示します。	
	デフォルト	[0]
	変更方法	外部フラッシュ・メモリ・ダウンロードダイアログによる選択 外部フラッシュ・メモリ・ダウンロードダイアログは、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 (このパネル上で外部フラッシュ・メモリ情報ファイルを指定することはできません)。

(11) 【シミュレータ GUI】【シミュレータ】

シミュレータ GUI 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・ツールと接続後、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 を表示する	シミュレータ GUI を使用するため、シミュレータ GUI ウィンドウを表示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 ただし、プログラム実行中は変更不可
	指定可能値	はい シミュレータ GUI を使用します。 いいえ シミュレータ GUI を使用しません。

実行開始時に最前面表示する	プログラムの実行開始時に、シミュレータ GUI ウィンドウを最前面に表示するか否かを選択します。 なお、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] を指定した場合のみ表示されます。	
	デフォルト	はい
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい
	いいえ	最前面に表示しません。

[フラッシュ・セルフ・エミュレーション設定] タブ【IECUBE】

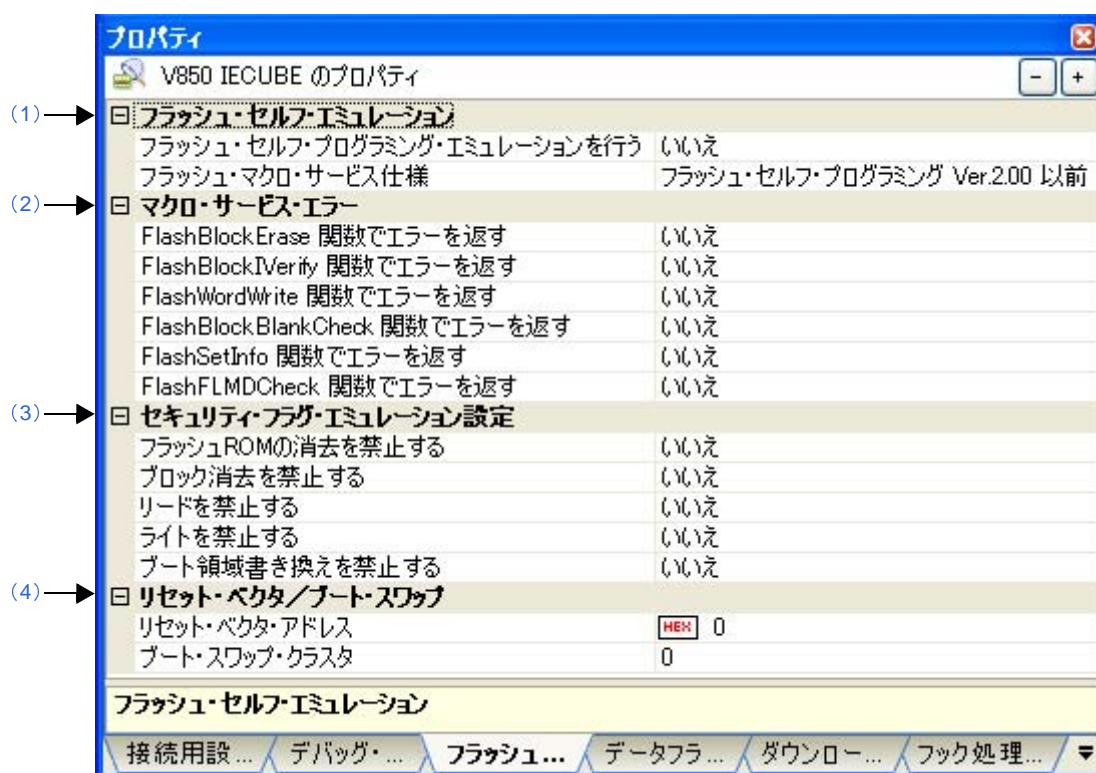
[フラッシュ・セルフ・エミュレーション設定] タブでは、次に示すカテゴリごとに詳細情報の表示、および設定の変更を行います。

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・メモリ内蔵品の場合のみ表示されます。

なお、フラッシュ関数のエミュレーション可否一覧、および注意事項に関しては、[\[フラッシュ・セルフ・プログラミング・エミュレーション特記事項\]](#) を参照してください。

- (1) [フラッシュ・セルフ・エミュレーション]
- (2) [マクロ・サービス・エラー]
- (3) [セキュリティ・フラグ・エミュレーション設定]
- (4) [リセット・ベクタ/ブート・スワップ]

図 A—21 プロパティ パネル：[フラッシュ・セルフ・エミュレーション設定] タブ



[各カテゴリの説明]

- (1) [フラッシュ・セルフ・エミュレーション]

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・セルフ・プログラミング・エミュレーションを行う	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	
	指定可能値	はい	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を使用します。
いいえ		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を使用しません。	
フラッシュ・マクロ・サービス仕様	フラッシュ・マクロ・サービスの仕様を選択します。 なお、このプロパティは、使用するフラッシュ・メモリのプロセスが、Type01 の場合のみ表示されます ^注 。		
	デフォルト	フラッシュ・セルフ・プログラミング Ver.2.00 以前	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	フラッシュ・セルフ・プログラミング Ver.2.00 以前	フラッシュ・マクロ・サービスの仕様を Ver.2.00 以前のものとしします。
フラッシュ・セルフ・プログラミング Ver.3.00 以降		フラッシュ・マクロ・サービスの仕様を Ver.3.00 以降のものとしします。	

注 デバッグ・ツールと接続時に、使用するフラッシュ・メモリのプロセスの判定を行います。
したがって、デバッグ・ツールと切断時は非表示とはなりません。

(2) [マクロ・サービス・エラー]

フラッシュ・マクロ・サービスに関する詳細情報の表示、および設定の変更を行います。

なお、フラッシュ・メモリがダメージを受けた場合に返るエラーは、通常のエミュレーションでは発生しません。

FlashBlockErase 関数でエラーを返す	エミュレートを行う FlashBlockErase 関数で発生するエラー値を強制的に返す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	
	指定可能値	はい	エラー値を強制的に返します。
いいえ		エラー値を返しません。	
FlashBlockIVerify 関数でエラーを返す	エミュレートを行う FlashBlockIVerify 関数で発生するエラー値を強制的に返す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	
	指定可能値	はい	エラー値を強制的に返します。
いいえ		エラー値を返しません。	

FlashWordWrite 関数でエラーを返す	エミュレートを行う FlashWordWrite 関数で発生するエラー値を強制的に返す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	
	指定可能値	はい	エラー値を強制的に返します。
		いいえ	エラー値を返しません。
FlashBlockBlankCheck 関数でエラーを返す	エミュレートを行う FlashBlockBlankCheck 関数で発生するエラー値を強制的に返す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	
	指定可能値	はい	エラー値を強制的に返します。
		いいえ	エラー値を返しません。
FlashSetInfo 関数でエラーを返す	エミュレートを行う FlashSetInfo 関数で発生するエラー値を強制的に返す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	
	指定可能値	はい	エラー値を強制的に返します。
		いいえ	エラー値を返しません。
FlashFLMDCheck 関数でエラーを返す	エミュレートを行う FlashFLMDCheck 関数で発生するエラー値を強制的に返す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	
	指定可能値	はい	エラー値を強制的に返します。
		いいえ	エラー値を返しません。

(3) [セキュリティ・フラグ・エミュレーション設定]

セキュリティ・フラグ・エミュレーション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。

フラッシュ ROM の消去を禁止する	フラッシュ ROM の消去禁止のエミュレーションを行うか否かを選択します。 なお、このプロパティは、使用するフラッシュ ROM がこの機能をサポートしている場合のみ表示されます。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	フラッシュ ROM 消去禁止のエミュレーションを行います。
		いいえ	フラッシュ ROM 消去禁止のエミュレーションを行いません。
ブロック消去を禁止する	ブロック消去禁止のエミュレーションを行う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	
	指定可能値	はい	ブロック消去禁止のエミュレーションを行います。
		いいえ	ブロック消去禁止のエミュレーションを行いません。

リードを禁止する	リード禁止のエミュレーションを行うか否かを選択します。 なお、このプロパティは、使用するフラッシュ・メモリのプロセスが Type01 ^注 で、フラッシュ・セルフ・プログラミング Ver2.00 以前の場合は表示されません。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい リード禁止のエミュレーションを行います。 いいえ リード禁止のエミュレーションを行いません。
ライトを禁止する	ライト禁止のエミュレーションを行う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
	指定可能値	はい ライト禁止のエミュレーションを行います。 いいえ ライト禁止のエミュレーションを行いません。
ブート領域書き換えを禁止する	ブート領域書き換え禁止のエミュレーションを行うか否かを選択します。 なお、このプロパティは、使用するフラッシュ・メモリのプロセスが Type02, Type03, または Type04 ^注 の場合のみ表示されます。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい ブート領域書き換え禁止のエミュレーションを行います。 いいえ ブート領域書き換え禁止のエミュレーションを行いません。

注 デバッグ・ツールと接続時に、使用するフラッシュ・メモリのプロセスの判定を行います。

(4) [リセット・ベクタ/ブート・スワップ]

リセット・ベクタ、およびブート・スワップ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。

リセット・ベクタ・アドレス	リセット・ベクタ・アドレスを指定します。 なお、このプロパティは、使用するフラッシュ・メモリのプロセスが Type01 でフラッシュ・セルフ・プログラミング Ver3.00 以降の場合のみ表示されます ^{注1} 。	
	デフォルト	0
	変更方法	キーボードからの直接入力
	指定可能値	0x0 ~ 0xFFFFFFFF の 16 進数
ブート・スワップ・クラスタ	ブート・スワップ・クラスタの対象領域を指定します。 ここでの設定値により、ブート・ブロック・クラスタの対象領域も自動的に変更されま ^{注2} 。 なお、このプロパティは、使用するフラッシュ・メモリのプロセスが Type04 の場合のみ表示されます ^{注1} 。	
	デフォルト	0
	変更方法	キーボードからの直接入力
	指定可能値	0 ~ 127 の 10 進数

- 注1. デバッグ・ツールと接続時に、使用するフラッシュ・メモリのプロセスの判定を行います。
したがって、デバッグ・ツールと切断時は非表示とはなりません。
2. ブート・ブロック・クラスタの対象領域は、マイクロコントローラのセキュリティ機能により、ブート領域書き換え禁止設定が可能な領域となります。
- なお、ここでの設定値とブート・ブロック・クラスタの対象領域の関係については、「[表 2-2 ブート・スワップ・クラスタとブート・ブロック・クラスタの関係](#)」を参照してください。

[フラッシュ・セルフ・プログラミング・エミュレーション特記事項]

IECUBE でフラッシュ・セルフ・プログラミングのエミュレーションを行う場合、次に示すフラッシュ関数のエミュレーション可否、および注意事項を確認してください。

なお、表内のマークの意味は次のとおりです。

○	エミュレーション可能
△	制限つきでエミュレーション可能
×	エミュレーション不可

表 A-3 フラッシュ・セルフ・プログラミング Type01 使用時のフラッシュ関数エミュレーション可否

フラッシュ関数名	機能概要／制限	エミュレーション可否
FlashEnv	フラッシュ環境初期化／終了関数	○
FlashBlockErase	1 ブロック消去関数	○
FlashWordWrite	1 ワード分の書き込み関数 制限：第三引数にガード領域が指定されていた場合、意図しないアドレスでフェイル・セーフ・ブレイクが発生します。	△
FlashBlockVerify	1 ブロックの内部ベリファイ処理関数	○
FlashBlockBlankCheck	1 ブロックのブランク・チェック関数	○
FlashGetInfo	フラッシュ情報取得関数	
	Option = 2 : CPU 番号、および CPU で持っているブロック総数 制限：CPU 番号はコンフィギュレーションで設定したデバイス名称 (4 桁番号) が返却されます。	△
	Option = 3 : セキュリティ情報	○
	Option = 4 : ブート領域入れ替え情報取得 制限：ブート領域入れ替え情報が反映されません。	△
	Option = 5 + ブロック番号 : ブロックの最終アドレス取得	○
FlashSetInfo	フラッシュ情報設定関数 制限：ブート領域入れ替え設定が無視されます。	△
FlashStatusCheck	直前に実行したフラッシュ関数の動作状況確認関数 制限：FlashBlockErase、および FlashBlockBlankCheck で FE_BUSY から FE_OK になるタイミングが実デバイスと異なります。	△
FlashBootSwap	ブート領域のブロック入れ替え関数	×

フラッシュ関数名	機能概要／制限	エミュレーション可否
FlashSetUserHandler	ユーザ割り込みハンドラ登録関数	○
FlashFLMDCheck	FLMD0 端子の状態チェック関数	○
FlashSetInfoEx	フラッシュ情報設定関数 制限：ブート領域入れ替え設定が無視されます。	△
FlashNWordRead	Nワード分の読み出し関数 制限：第三引数にガード領域が指定されていた場合、意図しないアドレスでフェイル・セーフ・ブレイクが発生します。	△

表 A—4 フラッシュ・セルフ・プログラミング Type02c 使用時のフラッシュ関数エミュレーション可否

フラッシュ関数名	機能概要／制限	エミュレーション可否
FlashEnv	フラッシュ環境初期化／終了関数	○
FlashBlockErase	1 ブロック消去関数	○
FlashWordWrite	1 ワード分の書き込み関数 制限：第三引数にガード領域が指定されていた場合、意図しないアドレスでフェイル・セーフ・ブレイクが発生します。	△
FlashBlockIVerify	1 ブロックの内部ベリファイ処理関数	○
FlashBlockBlankCheck	1 ブロックのブランク・チェック関数	○
FlashGetInfo	フラッシュ情報取得関数	
	Option = 2 : CPU 番号、および CPU で持っているブロック総数 制限：CPU 番号は DF の名称 (4 桁番号) が返却されます。	△
	Option = 3 : セキュリティ情報	○
	Option = 4 : ブート領域入れ替え情報取得 制限：ブート領域入れ替え情報が反映されません。	△
	Option = 5 + ブロック番号 : ブロックの最終アドレス取得	○
FlashSetInfo	フラッシュ情報設定関数 制限：ブート領域入れ替え設定が無視されます。	△
FlashBootSwap	ブート領域のブロック入れ替え関数	×
FlashFLMDCheck	FLMD0 端子の状態チェック関数	○
FlashWordRead	データ読み出し関数 制限：第三引数にガード領域が指定されていた場合、意図しないアドレスでフェイル・セーフ・ブレイクが発生します。	△

表 A—5 フラッシュ・セルフ・プログラミング Type03 使用時のフラッシュ関数エミュレーション可否

フラッシュ関数名	機能概要／制限	エミュレーション可否
FlashEnv	フラッシュ環境初期化／終了関数	○
FlashBlockErase	1 ブロック消去関数	○

フラッシュ関数名	機能概要/制限	エミュレーション可否
FlashWordWrite	1ワード分の書き込み関数 制限：第三引数にガード領域が指定されていた場合、意図しないアドレスでフェイル・セーフ・ブレイクが発生します。	△
FlashBlockIVerify	1ブロックの内部ベリファイ処理関数	○
FlashBlockBlankCheck	1ブロックのブランク・チェック関数	○
FlashGetInfo	フラッシュ情報取得関数	
	Option = 2 : CPU 番号, および CPU で持っているブロック総数 制限：CPU 番号は DF の名称 (4 桁番号) が返却されます。	△
	Option = 3 : セキュリティ情報	○
	Option = 4 : ブート領域入れ替え情報取得 制限：ブート領域入れ替え情報が反映されません。	△
	Option = 5 + ブロック番号 : ブロックの最終アドレス取得	○
FlashSetInfo	フラッシュ情報設定関数 制限：ブート領域入れ替え設定が無視されます。	△
FlashBootSwap	ブート領域のブロック入れ替え関数	×
FlashFLMDCheck	FLMD0 端子の状態チェック関数	○
FlashWordRead	データ読み出し関数 制限：第三引数にガード領域が指定されていた場合、意図しないアドレスでフェイル・セーフ・ブレイクが発生します。	△
FlashIVerify	内部ベリファイ関数 (EEPROM 専用)	×
FlashBlankCheck	ブランク・チェック関数 (EEPROM 専用)	×
EEPROM_Init	EEPROM 領域初期化関数 (EEPROM 専用)	×
EEPROM_Write	EEPROM 書き込み関数 (EEPROM 専用)	×
EEPROM_Read	EEPROM 読み出し関数 (EEPROM 専用)	×
EEPROM_Copy	EEPROM コピー関数 (EEPROM 専用)	×
EEPROM_VChk	EEPROM 有効領域チェック関数 (EEPROM 専用)	×
EEPROM_Erase	EEPROM 消去関数 (EEPROM 専用)	×

表 A-6 フラッシュ・セルフ・プログラミング Type04 使用時のフラッシュ関数エミュレーション可否

フラッシュ関数名	機能概要/制限	エミュレーション可否
FlashInit	セルフ・ライブラリの初期化関数	○
FlashEnv	フラッシュ環境初期化/終了関数	○
FlashBlockErase	1ブロック消去関数	○
FlashWordWrite	1ワード分の書き込み関数	○
FlashBlockIVerify	1ブロックの内部ベリファイ処理関数	○
FlashBlockBlankCheck	1ブロックのブランク・チェック関数	○

フラッシュ関数名	機能概要/制限	エミュレーション可否
FlashGetInfo	フラッシュ情報取得関数	
	Option = 2 : デバイス情報 (ブロック総数, デバイス番号)	○
	Option = 3 : セキュリティ・フラグ, ブート・ブロックの末尾ブロック番号	○
	Option = 4 : デバイス情報	○
	Option = 5 : リセット・ベクタ・アドレス	○
	Option = 6 + ブロック番号 <i>n</i> : ブロック番号 <i>n</i> の末尾アドレス	○
FlashSetInfo	フラッシュ情報設定関数 制限 : ブート領域入れ替え設定が無視されます。	△
FlashStatusCheck	直前に実行したフラッシュ関数の動作状況確認 制限 : SELFLIB_BUSY は返却されません。	△
FlashBootSwap	ブート領域のブロック入れ替え関数 制限 : 関数の呼び出しはできますが, ブート・スワップは実行されません。	×
FlashFLMDCheck	FLMD0 端子の状態チェック関数	○

表 A—7 注意事項一覧

番号	注意事項
(1)	<p>次の場合, フラッシュ・メモリ・セルフ・プログラミング・エミュレーションを有効にできません。</p> <ul style="list-style-type: none"> - 内蔵 ROM サイズをデフォルト・サイズ以外に設定した場合 回避策として, [接続用設定] タブ上の [内蔵 ROM/RAM] カテゴリ内 [内蔵 ROM サイズ [Kバイト]] プロパティをデフォルト値に戻してください。 - 実行前ブレークを 2 個使用している場合 回避策として, 実行前ブレーク 1 個を無効にするか, 削除してください。
(2)	<p>フラッシュ・メモリ・セルフ・プログラミング・エミュレーションを有効にすると, 次のデバッグ機能に制限がかかります。</p> <ul style="list-style-type: none"> - 内蔵 ROM と内蔵 RAM サイズの変更はできません - プログラム DMM/ 疑似 RRM の機能は使用できません。 - SP レジスタが適当な位置 (内蔵 RAM 等) に初期化される前にイベント等のブレークが発生した場合は, スタックエリアに対する不正ブレーク要因となってしまいます。この間にブレークが発生する可能性がある場合は, プログラム実行前に SP を適当な値にしてください。 - ご使用の IECUBE で, 下記制限事項が存在する場合, 不正ブレークが発生する可能性があります。 [デバッグ・ツール設定] タブ上の [フェイルセーフ・ブレーク] カテゴリ内 [内蔵 RAM の非メモリ・マッピング領域へのアクセス直後に停止する] プロパティを [いいえ] に設定してください。 - 「内蔵 RAM でプログラム実行時のイリーガル・ブレーク制限事項」

番号	注意事項
(3)	<p>フラッシュ・メモリ・セルフ・プログラミング・エミュレーションを有効に設定すると、0番地から4バイト分は予約領域になり、0番地にjr 0xffffd6の4バイト命令が書き込まれます。</p> <p>そのため、リセットベクタ0番地で使用の際には、スタート・アップルーチンは4番地から配置してください。</p> <p>また、フラッシュ・メモリ・セルフ・プログラミングを有効→無効に設定すると、0番地から4バイト分は0が書き込まれます。リセットベクタ0番地以外でご使用の際も0番地に分岐させるコードは記述しないでください。</p> <p>FlashNWordReadまたはFlashWordReadで予約領域を読み出したときは、0の値が読めます。</p> <p>実デバイスでも同一のプログラムを動作させるためには、次のコードを作成してください。</p> <pre># RESET handler --0番地の場合 .section "RESET", text jr __start --jr 0xffffd6 に書き換えられる jr __start</pre>
(4)	<p>リセット・ベクタ・ハンドリング指定アドレスに、0番地を指定するとリセットベクタは4番地になります。</p> <p>0番地以外を指定すると+4されずに指定アドレスがリセットベクタになります。</p>
(5)	<p>FlashBlockErase()とFlashBlockBlankCheck()後のFlashStatusCheck()の動作について、エミュレーション時は、FlashStatusCheck()の返却値がFE_BUSYからFE_OKになるまでのタイミングが実デバイスと異なりますので注意してください。</p>
(6)	<p>FlashWordWrite, FlashWordRead, FlashNWordReadの第3引数に指定したアドレスがガード領域であった場合、不正なメモリアクセスとなり、意図しないアドレスでフェイル・セーフ・ブレイクが発生します。</p> <p>FlashWordWrite, FlashWordRead, FlashNWordReadで指定するアドレスを適切なアドレスに修正してください。</p>
(7)	<p>このタブ上の各設定を有効にするためには、設定後必ずCPUリセットして、再実行するようにしてください。</p> <p>CPUリセットを行わず再実行した場合、設定が反映されない場合があります。</p>
(8)	<p>デバッグのワーク分として、最低84(54H)バイトのスタックを確保してください。ブレイク時やフラッシュ書き換えエミュレーション処理時に最低84(54H)バイトのスタックを消費します。</p> <p>割り込みを許可する場合は、さらに、デバッグのワーク分として、84(54H)バイトのスタックが必要です。また、多重割り込みの場合は、1段ごとに84(54H)バイトのスタックを確保する必要があります。</p>
(9)	<p>CPUリセット時に内蔵RAMの内容が破壊されます。通常、実デバイスにおいて、リセット後の内蔵RAMデータは保証していませんが、動作が異なる場合がありますので注意してください。</p>
(10)	<p>フラッシュ関数を仕様範囲外の方法で使用したり、サポートしていないフラッシュ関数を呼び出した場合、戻り値として“1”が返ります。</p>
(11)	<p>Type04のエミュレーション時は、次の制限があります。</p> <ul style="list-style-type: none"> - 内蔵RAMの最終アドレスから48バイト分は予約領域としてデバッグが使用します。 - 内蔵フラッシュ・メモリが1Mバイトのデバイスの場合、0xFF300以降の内蔵フラッシュ領域は、デバッグ・ツールが使用します。 - フラッシュ関数を命令レベル単位でステップ実行した場合、デバッグ・ツールのエミュレーション用コードが実行されるため、実際のデバイスが実行するコードと異なります。このため、デバッグ時はソース・レベル単位でステップ実行してください。

[データフラッシュ・エミュレーション設定] タブ【IECUBE】

[データフラッシュ・エミュレーション設定] 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。

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データフラッシュ・メモリ内蔵品の場合のみ表示されます。

- (1) [データフラッシュ・エミュレーション]
- (2) [EEPROM ライブラリ・エラー]
- (3) [書き込み／消去時間設定]

図 A—22 プロパティ パネル：[データフラッシュ・エミュレーション設定] タブ



[各カテゴリの説明]

- (1) [データフラッシュ・エミュレーション]

データフラッシュ・エミュレーション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。

データフラッシュ・エミュレーションを行う	データフラッシュ・エミュレーション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		
	指定可能値	はい	データフラッシュ・エミュレーション機能を使用します。	
		いいえ	データフラッシュ・エミュレーション機能を使用しません。	

(2) [EEPROM ライブラリ・エラー]

EEPROM エミュレーション・ライブラ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。

このカテゴリ内の各プロパティを設定することにより、通常のエミュレーションでは返ることのないエラー値を強制的に返すことができます。

WordWrite 関数でエラーを返す	エミュレートを行う WordWrite 関数で発生するエラー値を強制的に返す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
	指定可能値	はい いいえ
WordWrite 関数エラーの発生アドレス	WordWrite 関数エラーを発生させるデータフラッシュ・メモリ領域内のアドレスを指定します。 なお、このプロパティは、[WordWrite 関数でエラーを返す] プロパティにおいて [はい] を指定した場合のみ表示されます。	
	デフォルト	0
	変更方法	キーボードからの直接入力
	指定可能値	0x0 ~ 0xFFFFFFFF の 16 進数
BlockErase 関数でエラーを返す	エミュレートを行う BlockErase 関数で発生するエラー値を強制的に返す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
	指定可能値	はい いいえ
BlockErase 関数エラーの発生アドレス	BlockErase 関数エラーを発生させるデータフラッシュ・メモリ領域内のアドレスを指定します。 なお、このプロパティは、[BlockErase 関数でエラーを返す] プロパティにおいて [はい] を指定した場合のみ表示されます。	
	デフォルト	0
	変更方法	キーボードからの直接入力
	指定可能値	0x0 ~ 0xFFFFFFFF の 16 進数
BlockVerify 関数でエラーを返す	エミュレートを行う BlockVerify 関数で発生するエラー値を強制的に返す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
	指定可能値	はい いいえ

BlockIVerify 関数エラーの発生アドレス	BlockIVerify 関数エラーを発生させるデータフラッシュ・メモリ領域内のアドレスを指定します。 なお、このプロパティは、[BlockIVerify 関数でエラーを返す] プロパティにおいて [はい] を指定した場合のみ表示されます。	
	デフォルト	0
	変更方法	キーボードからの直接入力
	指定可能値	0x0 ~ 0xFFFFFFFF の 16 進数
BlockBlankCheck 関数でエラーを返す	エミュレートを行う BlockBlankCheck 関数で発生するエラー値を強制的に返す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
	指定可能値	はい エラー値を強制的に返します。 いいえ エラー値を返しません。
BlockBlankCheck 関数エラーの発生アドレス	BlockBlankCheck 関数エラーを発生させるデータフラッシュ・メモリ領域内のアドレスを指定します。 なお、このプロパティは、[BlockBlankCheck 関数でエラーを返す] プロパティにおいて [はい] を指定した場合のみ表示されます。	
	デフォルト	0
	変更方法	キーボードからの直接入力
	指定可能値	0x0 ~ 0xFFFFFFFF の 16 進数

(3) [書き込み/消去時間設定]

データフラッシュ・メモリへの書き込み、または消去時間に関する詳細情報の表示、および設定の変更を行います。

データフラッシュ・メモリへの書き込み時間	データフラッシュ・メモリへ書き込む際の遅延時間のシミュレート値を選択します。		
	デフォルト	フラッシュ・マクロ・スペック上の想定回数	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	リトライなし	リトライ回数を 0 とし、遅延時間=0 (書き込み時間最速) とします。
		フラッシュ・マクロ・スペック上の想定回数	使用するフラッシュ・マクロのスペック上の想定回数を遅延時間とします。
フラッシュ・マクロ・スペック上の想定最大回数		使用するフラッシュ・マクロのスペック上の想定最大回数を遅延時間とします。	
リトライ回数最大		リトライ回数を最大とし、遅延時間を最大 (書き込み時間最長) とします。	

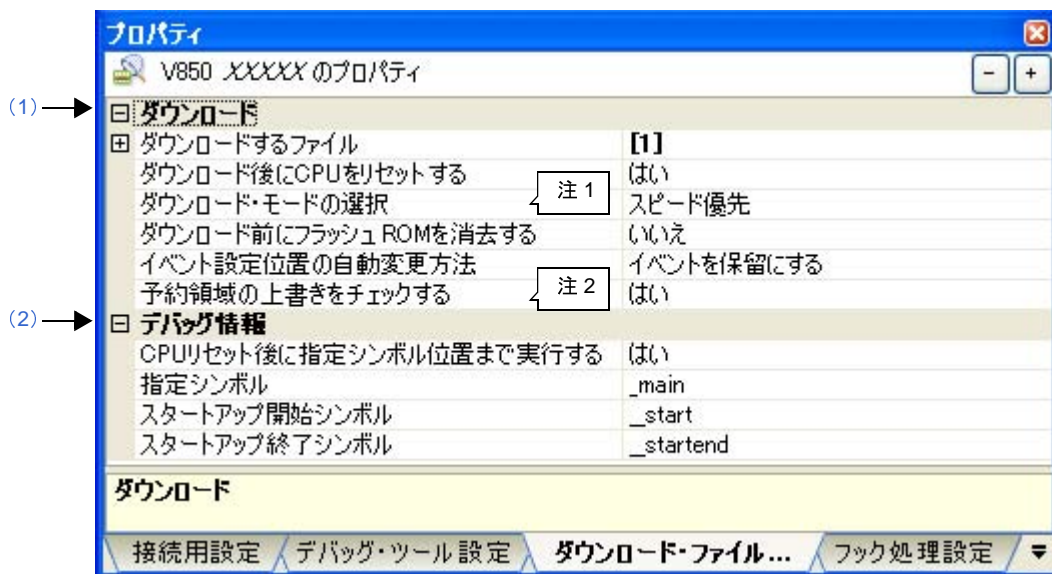
データフラッシュ・メモリの消去時間	データフラッシュ・メモリの消去を行う際の遅延時間のシミュレート値を選択します。		
	デフォルト	フラッシュ・マクロ・スペック上の想定回数	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	リトライなし	リトライ回数を0とし、遅延時間=0（消去時間最速）とします。
		フラッシュ・マクロ・スペック上の想定回数	使用するフラッシュ・マクロのスペック上の想定回数を遅延時間とします。
フラッシュ・マクロ・スペック上の想定最大回数		使用するフラッシュ・マクロのスペック上の想定最大回数を遅延時間とします。	
	リトライ回数最大	リトライ回数を最大とし、遅延時間を最大（消去時間最長）とします。	

[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—23 プロパティ パネル : [ダウンロード・ファイル設定] タブ



- 注1. [ダウンロード・モードの選択プロパティは、シミュレータ以外の場合のみ表示されます。
- 2. [予約領域の上書きをチェックする] プロパティは、MINICUBE2/E1(Serial)/E20(Serial)/EZ Emulator の場合のみ表示されます。

[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端に展開表示します。	
	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 をリセットする	ダウンロード後に CPU をリセットするか否かを選択します。		
	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	
	指定可能値	はい	ダウンロード後に CPU をリセットします。
いいえ		ダウンロード後に CPU をリセットしません。	
ダウンロード・モードの選択 (【シミュレータ】以外)	フラッシュ ROM へダウンロードを行う際のダウンロード・モードを選択します。		
	デフォルト	スピード優先	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	スピード優先	先頭データと最終データ間の空き領域は FFH で補完されま す。書き込みデータの量を減らすため、ダウンロード速度は 速くなります。
データ優先		空き領域はダウンロード前の値が保持されます。 空き領域のデータを一度読み込むため、ダウンロード速度は 非常に遅くなります。	
ダウンロード前にフラッシュ ROM を消去する	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。		
	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、[ダウンロード・モードの選択] プロパティにおいて [データ優先] を指定した場合は変更不可	
	指定可能値	はい	ダウンロード前にフラッシュ ROM を消去します。
いいえ		ダウンロード前にフラッシュ ROM を消去しません。	
イベント設定位置の自動変更方法	再ダウンロードすることにより、現在設定されているイベントの設定位置（アドレス）が命令の途中になる場合の再設定方法を選択します ^{注2} 。		
	デフォルト	イベントを保留にする	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	命令の先頭に移動する	命令の先頭アドレスに対象イベントを再設定します。
イベントを保留にする		対象イベントを保留状態にします。	
予約領域の上書きをチェックする 【MINICUBE2】 【E1/E20(Serial)】 【EZ Emulator】	ダウンロードの際に、エミュレータが使用する予約領域に対して上書きした場合に、メッセージを出力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	
	指定可能値	はい	予約領域に上書きした場合、メッセージを出力します。
いいえ		予約領域に上書きしても、メッセージを出力しません。	

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—3 ダウンロード可能なファイル形式」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ンボル位置まで実行する	CPU リセット後に、プログラムを指定シンボル位置まで実行するか否かを選択します。	
	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
	指定可能値	はい プログラムを指定シンボル位置まで実行します。 いいえ CPU リセット後にプログラムを実行しません。
指定シンボル	CPU リセット後に、プログラムを実行して停止する位置を指定します。 なお、このプロパティは、[CPU リセット後に指定シンボル位置まで実行する] プロパティにおいて [はい] を選択した場合のみ表示されます。	
	デフォルト	_main
	変更方法	キーボードからの直接入力
	指定可能値	0 ~ “アドレス空間の終了アドレス” のアドレス式
スタートアップ開始シンボル	スタートアップ・ルーチンのテキスト領域の開始シンボルを指定します。	
	デフォルト	__start
	変更方法	キーボードからの直接入力
	指定可能値	0 ~ “アドレス空間の終了アドレス” のアドレス式
スタートアップ終了シンボル	スタートアップ・ルーチンのテキスト領域の終了シンボルを指定します。	
	デフォルト	__startend
	変更方法	キーボードからの直接入力
	指定可能値	0 ~ “アドレス空間の終了アドレス” のアドレス式

[フラッシュ・オプション設定] タブ【V850E2】

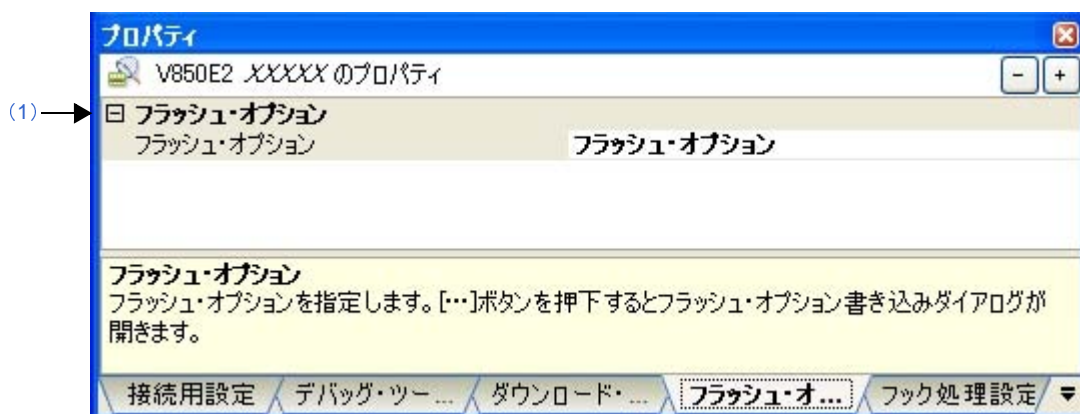
[フラッシュ・オプション設定] タブでは、V850E2 コアのマイクロコントローラに搭載されているフラッシュ・メモリのためのオプション設定を行います。

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【V850E2】](#) において行います。

- 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。

図 A—24 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。

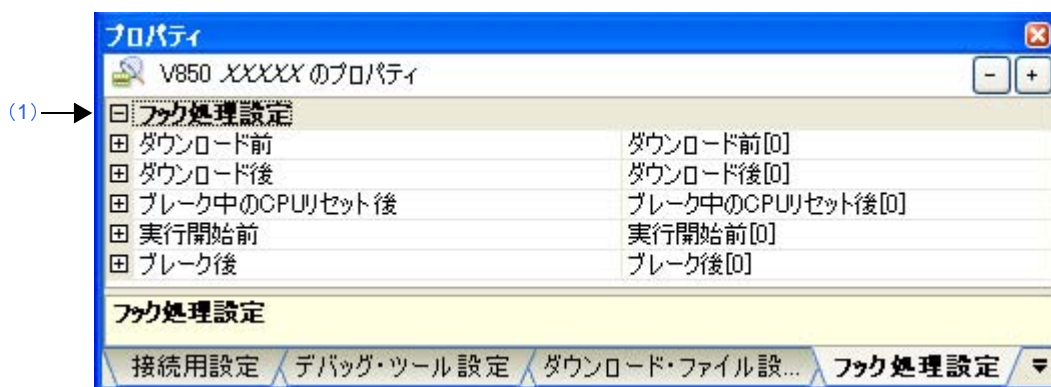
フラッシュ・オプション	このプロパティを選択すると欄内右端に表示される [...] ボタンをクリックすることにより、フラッシュ・オプションに関する詳細情報を設定するための フラッシュ・オプションの設定 ダイアログ【V850E2】 をオープンします。
-------------	--

[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、「2.16 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—25 プロパティ パネル : [フック処理設定] タブ



[各カテゴリの説明]

(1) [フック処理設定]

フック処理に関する詳細情報の表示、および設定の変更を行います。

なお、このタブ上のプロパティの設定は、各プロパティを選択すると右端に表示される [...] ボタンをクリックすることでオープンする **テキスト編集 ダイアログ** で行います（このパネル上で処理を指定することはできません）。

注意 1 処理につき 64 文字まで入力可能で、各プロパティごとに 128 個までの処理を指定することができます（**テキスト編集 ダイアログ** 上の [テキスト] エリア内の 1 行が 1 処理に相当）。

ダウンロード前	ロード・モジュール・ファイルをダウンロードする直前に行う処理を指定します。
	デフォルト ダウンロード前 [0]（“[]” 内は現在の指定処理数を示す）
	変更方法 テキスト編集 ダイアログ による指定
	指定形式 次のいずれか - I/O レジスタ名 + 半角スペース + 数値 【処理】 I/O レジスタの内容を数値に自動的に書き換えます。 - CPU レジスタ名 + 半角スペース + 数値 【処理】 CPU レジスタの内容を数値に自動的に書き換えます。 - Source+ 半角スペース + Python スクリプト・パス 【処理】 Python スクリプト・パスで指定したスクリプト・ファイルを実行します。

ダウンロード後	ロード・モジュール・ファイルをダウンロードした直後に行う処理を指定します。	
	デフォルト	ダウンロード後 [0] (“[]”内は現在の指定処理数を示す)
	変更方法	テキスト編集 ダイアログによる指定
	指定形式	次のいずれか - I/O レジスタ名 + 半角スペース + 数値 【処理】 I/O レジスタの内容を数値に自動的に書き換えます。 - CPU レジスタ名 + 半角スペース + 数値 【処理】 CPU レジスタの内容を数値に自動的に書き換えます。 - Source+ 半角スペース + Python スクリプト・パス 【処理】 Python スクリプト・パスで指定したスクリプト・ファイルを実行します。
ブレーク中の CPU リセット後	ブレーク中の CPU リセット直後に行う処理を指定します。	
	デフォルト	ブレーク中の CPU リセット後 [0] (“[]”内は現在の指定処理数を示す)
	変更方法	テキスト編集 ダイアログによる指定
	指定形式	次のいずれか - I/O レジスタ名 + 半角スペース + 数値 【処理】 I/O レジスタの内容を数値に自動的に書き換えます。 - CPU レジスタ名 + 半角スペース + 数値 【処理】 CPU レジスタの内容を数値に自動的に書き換えます。 - Source+ 半角スペース + Python スクリプト・パス 【処理】 Python スクリプト・パスで指定したスクリプト・ファイルを実行します。
実行開始前	プログラムの実行開始直前に行う処理を指定します。	
	デフォルト	実行開始前 [0] (“[]”内は現在の指定処理数を示す)
	変更方法	テキスト編集 ダイアログによる指定
	指定形式	次のいずれか - I/O レジスタ名 + 半角スペース + 数値 【処理】 I/O レジスタの内容を数値に自動的に書き換えます。 - CPU レジスタ名 + 半角スペース + 数値 【処理】 CPU レジスタの内容を数値に自動的に書き換えます。 - Source+ 半角スペース + Python スクリプト・パス 【処理】 Python スクリプト・パスで指定したスクリプト・ファイルを実行します。
ブレーク後	プログラムの実行がブレークした直後に行う処理を指定します。	
	デフォルト	ブレーク後 [0] (“[]”内は現在の指定処理数を示す)
	変更方法	テキスト編集 ダイアログによる指定
	指定形式	次のいずれか - I/O レジスタ名 + 半角スペース + 数値 【処理】 I/O レジスタの内容を数値に自動的に書き換えます。 - CPU レジスタ名 + 半角スペース + 数値 【処理】 CPU レジスタの内容を数値に自動的に書き換えます。 - Source+ 半角スペース + Python スクリプト・パス 【処理】 Python スクリプト・パスで指定したスクリプト・ファイルを実行します。

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・ファイルを表示している場合は、ソース・レベル・デバッグ/命令レベル・デバッグ（「2.7.3 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【IECUBE】【IECUBE2】【シミュレータ】（「2.13 カバレッジの測定【IECUBE】【IECUBE2】【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・コード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）。

- 備考 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。
3. ソース・ファイルをオープンする際、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オープンするソース・ファイルの更新日時が新しい場合、メッセージを表示します。参照されているソースコードとデバッグ情報が一致していないことが原因です。

図 A—26 エディタ パネル

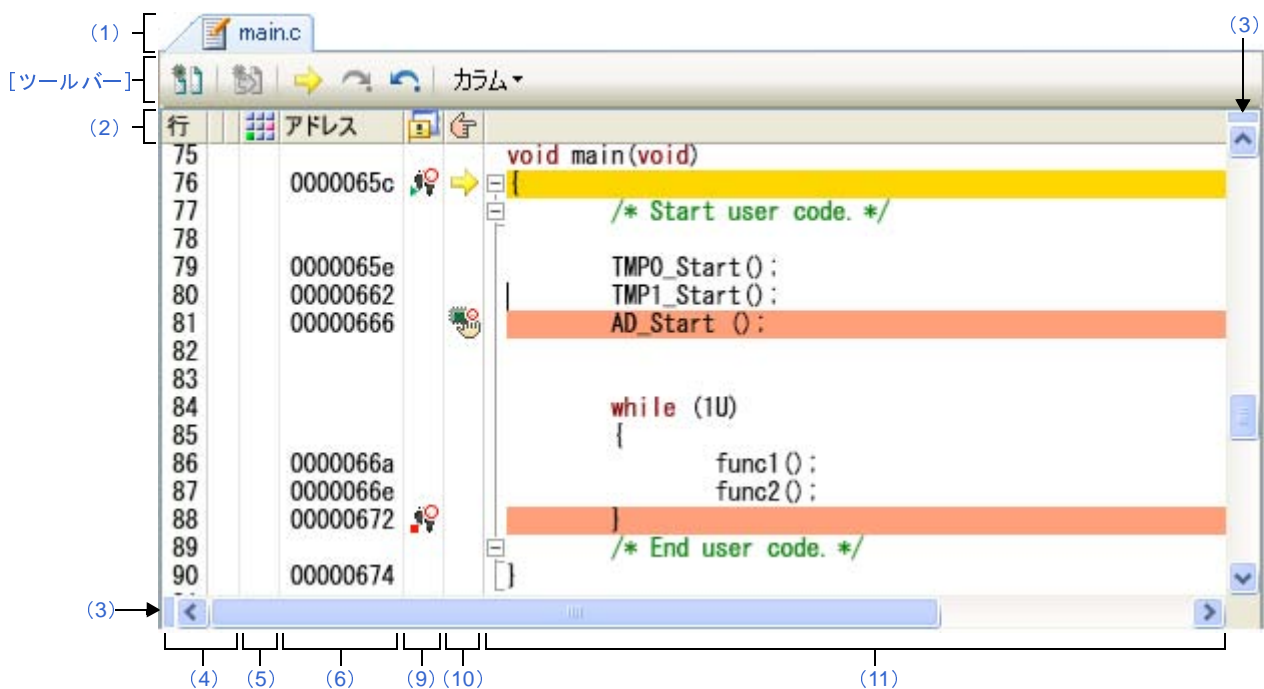


図 A—27 エディタ パネル (コード・カバレッジ測定結果を表示した場合)

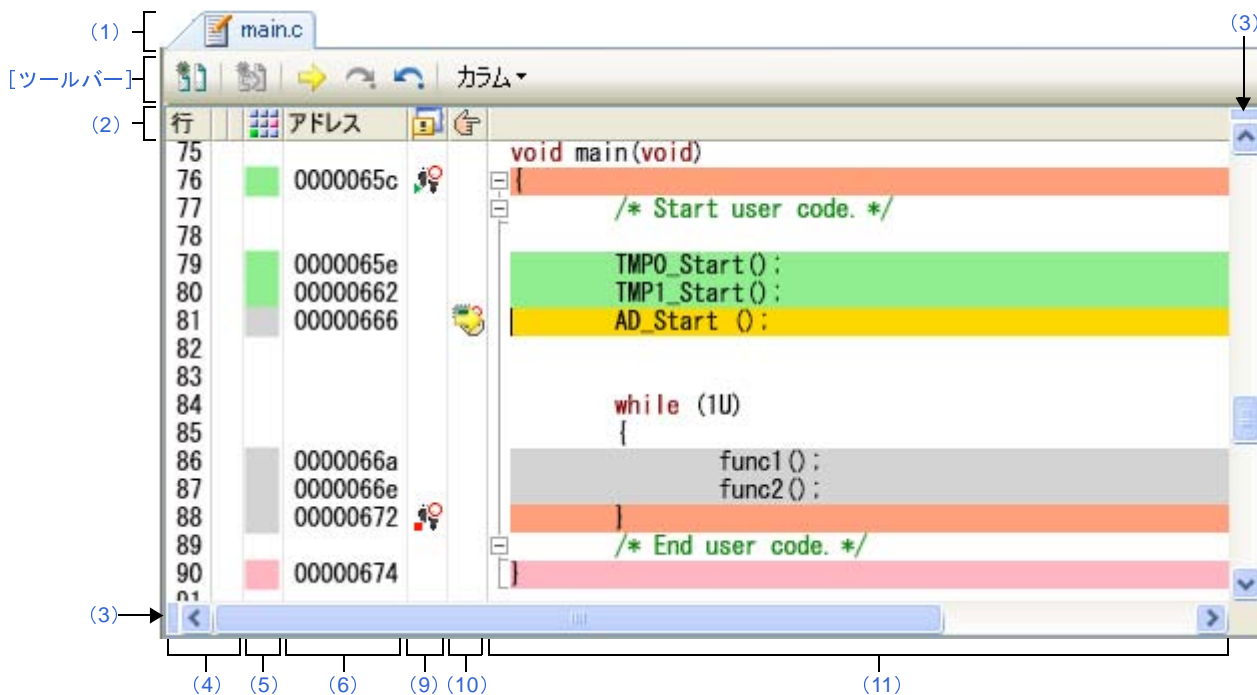
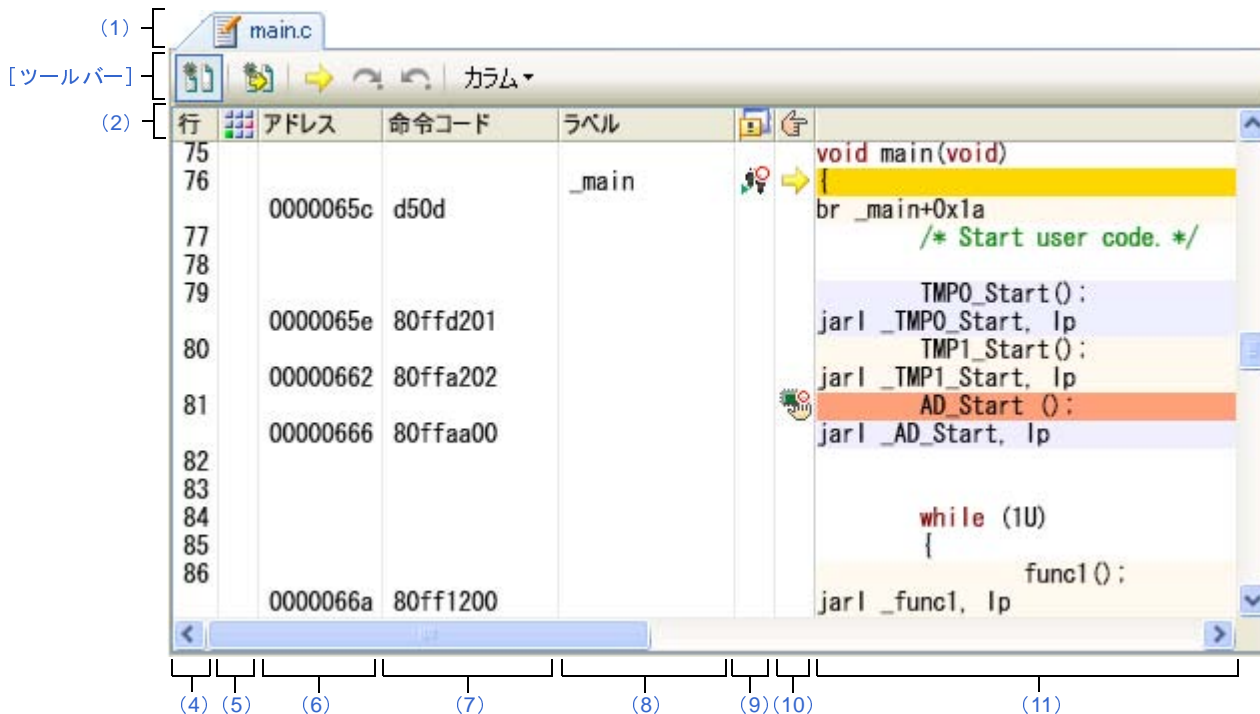


図 A—28 エディタ パネル (混合表示モードの場合)



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、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【IECUBE】【IECUBE2】【シミュレータ】/ イベントパネルにおいて、コンテキスト・メニューの [ソースヘジャンプ] 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

オープンしているテキスト・ファイル/ソース・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
*	テキスト・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テキスト・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)。

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 (「(4) 行番号エリア」参照)。 ただし、混合表示モード時は表示されません。

表示	タイトル名	説明
(表示なし)	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	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新しい場合に色表示を行います（「(4) 行番号エリア」参照）。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（「(5) 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（「(6) 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（「(7) 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（「(8) 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（「(9) 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	ブレークポイントを設定します（「(10) メイン・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。

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。

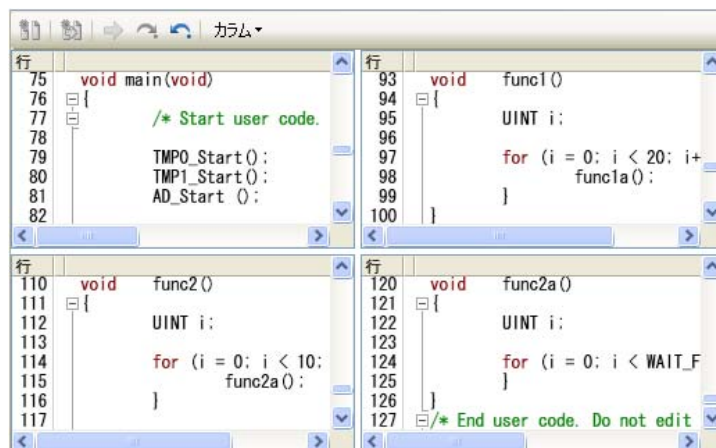
(3) 分割バー

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—29 エディタ パネル（縦横2分割表示した場合）



(4) 行番号エリア

表示しているテキスト・ファイル/ソース・ファイルの行番号を表示します。

また、各行に付いた色が、その行の編集状態を示します（混合表示モード時を除く）。

```

(1) 88 void main(void)
      89 {
(2) 90     /* Start user code.
      91
      92     TMPO_Start():
      93     TMP1_Start():
      94     AD_Start():
      95
      96     while (1U)
      97     {
  
```

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・ファイルを表示している場合は、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、次のマークを表示します（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）。

```

(3) 88
      89
      90
      91
      92
  
```

```

void main(void)
{
    /* Start user c
    TMPO_Start():
  
```

(1)		新規または変更したが保存されていない。
(2)		新規または変更後、保存済み。 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。
(3)		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が古い。 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。

(5) 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・ファイルを表示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] カテゴリの設定に依存します）。

カバレッジ測定についての詳細は、「2.13 カバレッジの測定【IECUBE】【IECUBE2】【シミュレータ】」を参照してください。

このエリア上にマウス・カーソルを重ねることにより、エリアのタイトル“カバレッジ”をポップアップ表示します。

このエリアは、次の機能を備えています。

(a) メニューによるカバレッジ情報のクリア

このエリアを右クリックすることにより、次のメニューを表示します。

カバレッジ情報をクリア	<p>現在デバッグ・ツールが保持しているコード・カバレッジ測定結果をすべてクリアします。</p> <p>ただし、使用するデバッグ・ツールがカバレッジ機能をサポートしていない場合、この項目は表示されません。</p>
-------------	--

(6) 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・ファイルを表示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。

(7) 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表示している場合のみ有効となるエリアです。

ソース・コードに対応した命令コードを表示します。

このエリア（命令コードを除く）上にマウス・カーソルを重ねることにより、エリアのタイトル“命令コード”をポップアップ表示します。

(8) 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表示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。

このエリア（ラベル名を除く）上にマウス・カーソルを重ねることにより、エリアのタイトル“ラベル”をポップアップ表示します。

(9) 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・ファイルを表示している場合のみ有効となるエリアです。

アドレス表示のある行は、タイマ/トレースなどのイベントを設定することができます。

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（イベント・マークを除く）上にマウス・カーソルを重ねることにより、エリアのタイトル“イベント”をポップアップ表示します。

このエリアは、次の機能を備えています。




(a) メニューによるイベントの各種設定

このエリアを右クリックすることにより、次のメニューを表示します。

実行時にタイム開始	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イム開始イベントを設定します（「2.12.2 任意区間の実行時間を計測する」参照）。
実行時にタイム終了	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イム終了イベントを設定します（「2.12.2 任意区間の実行時間を計測する」参照）。
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（「2.11.3 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	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（「2.11.3 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...	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、 アクション・イベントダイアログ をオープンします（「2.14.1 printfを挿入する」参照）。

(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、次のメニューを表示し、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	設定されているすべてのイベントを 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ブレークポイントを有効状態にします。
無効化	設定されているすべてのイベントを 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	設定されているすべてのイベントを削除します。 なお、複数のイベントが設定されていることを示すイベント・マーク（ )を選択している場合は、設定されているすべてのイベントを削除します。
イベントパネルで詳細を表示	選択しているイベントの詳細情報を表示する イベントパネル をオープンします。

(c) ポップアップ表示

[イベント・マーク](#)にマウス・カーソルを重ねることにより、そのイベントのイベント名／詳細情報／イベントに付加されたコメントをポップアップ表示します。

なお、該当箇所に複数のイベントが設定されている場合、最大3つまで、各イベントの情報を列挙して表示します。

備考 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。

(10) メイン・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・ファイルを表示している場合のみ有効となるエリアです。

アドレス表示のある行は、ブレークポイントを設定することができます。

また、現在設定しているブレークポイントがある場合、そのブレークポイ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。

また、このエリアでは、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。ただし、カレント PC マークは、カレント PC 値を変更した際、またはデバッグ・ツールが実行状態から停止状態に移行した際に、カレント PC 値がソース・テキスト行と対応する場合にのみ表示されます。

このエリア (イベント・マークを除く) 上にマウス・カーソルを重ねることにより、エリアのタイトル “メイン” をポップアップ表示します。

このエリアは、次の機能を備えています。

(a) ブレークポイントの設定／削除

ブレークポイントを設定したい箇所をマウスでクリックすることにより、容易にブレークポイントを設定することができます。

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

ブレークポイントを設定すると、設定した行に **イベント・マーク** が表示されます。また、設定したブレークポイントの詳細情報が **イベントパネル** に反映されます。

なお、すでにいずれかのイベント・マークが表示されている箇所において、この操作を行った場合は、そのイベントを削除し、ブレークポイントの設定は行いません。

ブレークポイントの設定方法についての詳細は、「[2.8.2 任意の場所で停止する \(ブレークポイント\)](#)」を参照してください。

(b) メニューによるブレークポイントの各種設定




このエリアを右クリックすることにより、次のメニューを表示します。

ブレークの設定	<p>キャレットのある行にブレークポイントを設定します (「2.8.2 任意の場所で停止する (ブレークポイント)」参照)。</p> <p>【シミュレータ】以外</p> <p>デフォルトは、リソースが使用可能であれば、デバッグ・ツールはハードウェア・ブレークポイントを設定します。[ハードウェア・ブレーク優先]、または [ソフトウェア・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。</p>
ハードウェア・ブレークポイントの設定 (【シミュレータ】以外)	キャレットのある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。
ソフトウェア・ブレークポイントの設定 (【シミュレータ】以外)	キャレットのある行にブレークポイント (ソフトウェア・ブレーク・イベント) を設定します。
ハードウェア・ブレーク優先 (【シミュレータ】以外)	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)。

ソフトウェア・ブレーク優先 （【シミュレータ】以外）	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ェア・ブレークポイントとします（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）。
-------------------------------	--

(c) 各種ブレークポイントの状態変更

各種イベント・マークを右クリックすることにより、次のメニューを表示し、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	選択しているブレークポイントを有効状態にします。 指定されている条件の成立で、プログラムを停止します。 なお、複数のブレークポイ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ブレークポイ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。 指定されている条件が成立しても、プログラムを停止しません。 なお、複数のブレークポイ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ブレークポイントを無効状態にします。
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。 なお、複数のブレークポイ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ブレークポイントを削除します。
イベントパネルで 詳細を表示	選択しているブレークポイ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。

(d) 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。

備考 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。

(11) 文字列エリア

テキスト・ファイル／ソース・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) 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、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/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 ファイルのサイズが1MBを越える場合は無効です。

(b) 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。

(c) 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、[表示]メニュー→[タグ・ジャンプ]、またはコンテキスト・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／該当桁へ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、「表 2—6 タグ・ジャンプの動作」を参照してください。

(d) カレント PC 行の表示

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]カテゴリの設定に依存します）。

ただし、この機能は、デバッグ・ツールと接続中で、ソース・ファイルを表示している場合のみ有効となります。

(e) ブレークポイント設定行の表示

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]カテゴリの設定に依存します）。

ただし、この機能は、デバッグ・ツールと接続中で、ソース・ファイルを表示している場合のみ有効となります。

(f) コード・カバレッジ測定結果の表示【IEUCBE】【IEUCBE2】【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]カテゴリの設定に依存します）。

カバレッジ測定についての詳細は、「2.13 カバレッジの測定【IEUCBE】【IEUCBE2】【シミュレータ】」を参照してください。

ただし、この機能は、デバッグ・ツールと接続中で、ソース・ファイルを表示している場合のみ有効となります。

(g) 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（“<変数名> = <変数値>”）。

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—17 ウォッチ式の表示形式（デフォルト）」と同等）。

ただし、この機能は、デバッグ・ツールと接続中で、ソース・ファイルを表示している場合のみ有効となります。

(h) 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] / [トレースの設定] / [タイマの設定] を選択することにより、現在キャレットのあるアドレス／行に各種イベントを設定することができます。

イベントを設定することにより、対応する **イベント・マーク** が **メイン・エリア** / **イベント・エリア** に表示されます。また、設定したイベントの詳細情報が **イベントパネル** に反映されます。

ただし、この機能は、デバッグ・ツールと接続中で、ソース・ファイルを表示している場合のみ有効となります。

イベントの設定方法についての詳細は次を参照してください。

- 「2. 8. 4 変数 /IO レジスタへのアクセスで停止する」
- 「2. 11. 3 任意区間の実行履歴を収集する」
- 「2. 11. 4 条件を満たしたときのみの実行履歴を収集する」
- 「2. 12. 2 任意区間の実行時間を計測する」

備考 ブレークポイントの設定／削除は、**メイン・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）。

(i) ウォッチ式の登録

表示されている C 言語変数 /CPU レジスタ /IO レジスタ /アセンブラ・シンボルをウォッチ式として **ウォッチパネル** に登録することができます。

操作方法についての詳細は、「(1) **ウォッチ式を登録する**」を参照してください。

ただし、この機能は、デバッグ・ツールと接続中で、ソース・ファイルを表示している場合のみ有効となります。

(j) ファイルの監視機能

ソース・ファイルを管理するために、次の監視機能を備えています。

- ソース・ファイルをオープンする際、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するソース・ファイルの更新日時が新しい場合、メッセージを表示します。
- 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。
- ソース・ファイルを保存する際（[ファイル] メニュー→ [ファイル名を保存] の選択）、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されていた場合、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。

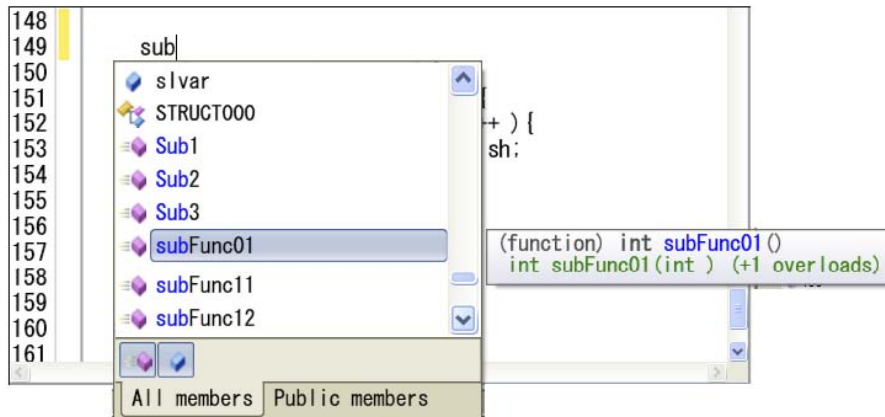
(k) スマート・エディット機能を使用する【V850E2】

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—30 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. スマート・エディット機能は、ビルド・ツールとして CX を使用している場合のみサポートされる機能です。

2. エディタパネルにおいて、混合表示モード（「(1) 表示モードを変更する」参照）を選択している場合、スマート・エディット機能を使用することはできません。

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、次のいずれかの操作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

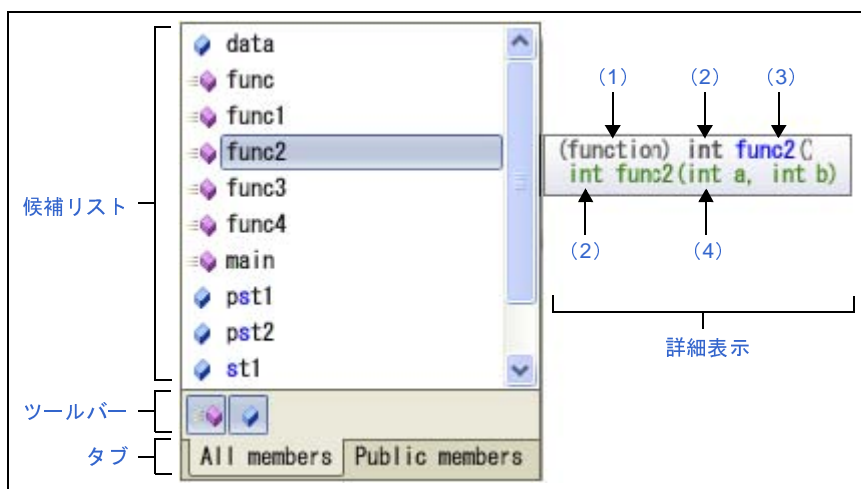
- キーボードより, [Ctrl] + [Space] キーを入力した場合 (すべての候補を表示)
 ただし, 候補が1つのみの場合は候補表示を行わず, 該当文字列がこの時点で挿入されます。

- 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—31 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。
	候補が関数であることを示します。
	候補が変数であることを示します。
	候補が構造体, または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
	選択することにより, 関数の候補を表示します。
	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。 ただし、今版では意味を持ちません。

- 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

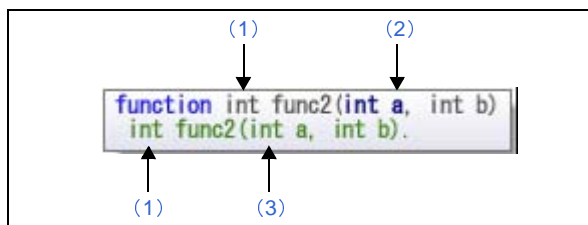
- 表示方法

引数の候補は、次のいずれかの操作により表示されます。

- 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

図 A—32 引数の候補表示



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。

項目	説明
(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

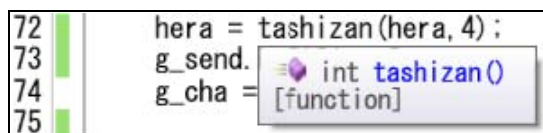
- 次の項目は候補表示に対応していません。
 - マクロ定義
 - ローカル変数
 - Typedef
- 関数中に構造体宣言／共用体宣言を行った場合、宣言以降関数内で候補表示は行いません。
- 変数のサイズに影響するコンパイル・オプションを設定した場合、表示される変数の型が実際の宣言と異なる場合があります。

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。
- 共用体の宣言内における共用体タグについてはポップアップ表示を行いません。
- ヘッダ・ファイル中で宣言されている構造体／共用体は表示されません。
- const, static, および volatile 属性は表示されません。

図 A—33 スマート・エディット機能によるポップアップ表示



(I) ブロックの選択

次の操作のいずれかより、複数行にわたるブロックを選択することができます。

- [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

```

void main(void)
{
    /* Start user code.
    TMPO_Start();
    TMP1_Start();
    AD_Start ();
    while (1U)
    {
    
```

備考 選択したブロックは、[編集] メニューの [切り取り] / [コピー] / [貼り付け] / [削除] による編集が可能です。

(m) 表示の拡大／縮小





[Ctrl] キーとマウス・ホイールを組み合わせることにより、エディタ パネルの表示を拡大／縮小することができます。


- [Ctrl] キーを押下しながらマウス・ホイールを前方に動かすと、エディタ パネルの表示を拡大し、表示が大きくなり見やすくなります（最大 300%）。
- [Ctrl] キーを押下しながらマウス・ホイールを後方に動かすと、エディタ パネルの表示を縮小し、表示が小さくなります（最小 50%）。

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

[ツールバー]

	<p>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）。</p> <p>クリックすると、混合表示モードに設定します。</p> <p>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・ファイルを表示している場合のみ有効となります。</p>
	<p>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（「2.7.3 プログラムをステップ実行する」参照）。</p> <p>クリックすると、命令レベル単位のステップ実行を行います（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）。</p> <p>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。</p>
	<p>現在の PC 位置を表示します。</p> <p>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。</p>
	<p>[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。</p> <p>ただし、混合表示モードの場合は無効となります。</p>

	[コンテキスト・メニュー] → [関数へジャンプ] 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	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・ファイルがロードモジュールより新しいことを示す表示	行番号エリアにおいて、ダウンロードされたロード・モジュールの更新状態を示すマークを表示します。 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。
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。 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）。

ファイル名を閉じる	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。
ファイル名を保存	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、[名前を付けてファイル名を保存] 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。
名前を付けてファイル名を保存 ...	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。 ただし、混合表示モードの場合は無効となります。

ファイル名の保存設定 ...	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。
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 ウィンドウ 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。
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、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。
置換 ...	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。

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（「(a) 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替える	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。
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（C++ など）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（C++ など）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（C++ など）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（C++ など）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です。
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、何も行きません。
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。

[[ウィンドウ] 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 [ウィンドウ] 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、2 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]

【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。
ファイル名の保存	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。 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。

【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。 何も選択されていない場合は、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。
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。
ジャンプ前の位置へ戻る	[関数へジャンプ] を実行する前の位置へ戻ります。
関数へジャンプ	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、該当する関数へジャンプします（「 (7) 関数へジャンプする 」参照）。
タグ・ジャンプ	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（「 (c) タグ・ジャンプ 」参照）。

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 (C++ など) 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 (C++ など) 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 (C++ など) 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 (C++ など) 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です。
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。

【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、既にすべてのチャン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・イベントの登録...	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、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	[関数へジャンプ] 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} 。 なお、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、下方向に有効な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・ツールヘダウンロード] 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 ^{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・ツールヘダウンロード] 実行中は無効となります。
関数へジャンプ	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、該当する関数へジャンプします (「 (7) 関数へジャンプする 」参照)。
タグ・ジャンプ	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 (「 (c) タグ・ジャンプ 」参照)。
逆アセンブルへジャンプ	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。

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。
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 (C++ など) 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 (C++ など) 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 (C++ など) 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 (C++ など) 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组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です。
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。

ブレークの設定	ブレーク関連のイベントを設定するために、次のカスケード・メニューを表示します。 なお、イベントは、イベントの設定が可能な行のみ設定することができます（「(9) イベント・エリア」参照）。
ハード・ブレークの設定	キャレットのある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（「(1) ブレーク・イベント（実行系）を設定する」参照）注1。
ソフト・ブレークの設定 （【シミュレータ】以外）	キャレットのある行にブレークポイント（ソフトウェア・ブレーク・イベント）を設定します（「(1) ブレーク・イベント（実行系）を設定する」参照）注1。
組み合わせブレークを設定	今版ではサポートしていません。
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	キャレット位置、または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）I/Oレジスタに、リード／ライト・アクセスのブレーク・イベントを設定します（「(1) ブレーク・イベント（アクセス系）を設定する」参照）。
読み込み組み合わせブレークを設定	今版ではサポートしていません。
書き込み組み合わせブレークを設定	今版ではサポートしていません。
読み書き組み合わせブレークを設定	今版ではサポートしていません。
ブレーク動作の設定	プロパティパネルをオープンし、ブレーク機能の設定を行います。

トレース設定 【IECUBE】【IECUBE2】 【シミュレータ】	トレース関連のイベントを設定するために、次のカスケード・メニューを表示します。 なお、イベントは、イベントの設定が可能な行のみ設定することができます（「(9) イベント・エリア」参照）。
トレース開始の設定	キャレット位置の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（「(1) トレース・イベントを設定する」参照）注1。 【シミュレータ】 プロパティパネル上の【トレース】【IECUBE】【IECUBE2】【シミュレータ】カテゴリ内【トレース機能を使用する】プロパティの設定を自動的に【はい】にします。
トレース終了の設定	キャレット位置の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（「(1) トレース・イベントを設定する」参照）注1。 【シミュレータ】 プロパティパネル上の【トレース】【IECUBE】【IECUBE2】【シミュレータ】カテゴリ内【トレース機能を使用する】プロパティの設定を自動的に【はい】にします。
値をトレースに記録（読み込み時）	キャレット位置、または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）I/Oレジスタにリード・アクセスした際に、その値をトレース・メモリに記録するポイント・トレース・イベントを設定します（「(1) 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（書き込み時）	キャレット位置、または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）I/Oレジスタにライト・アクセスした際に、その値をトレース・メモリに記録するポイント・トレース・イベントを設定します（「(1) 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（読み書き時）	キャレット位置、または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）I/Oレジスタにリード／ライト・アクセスした際に、その値をトレース・メモリに記録するポイント・トレース・イベントを設定します（「(1) ポイント・トレース・イベントを設定する」参照）。
読み書き時にトレース開始の設定	今版ではサポートしていません。
読み書き時にトレース終了の設定	今版ではサポートしていません。
トレース結果の表示	トレースパネル【IECUBE】【IECUBE2】【シミュレータ】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。
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 ^{注2} 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（「2.12.2 任意区間の実行時間を計測する」参照）。 なお、イベントは、イベントの設定が可能な行のみ設定することができます（「(9) イベント・エリア」参照）。
実行時にタイマ開始	キャレット位置の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 ^{注1} 。 【シミュレータ】 プロパティパネル上の【タイマ】【IECUBE】【IECUBE2】【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。
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イマ終了イベントを設定します ^{注1} 。 【シミュレータ】 プロパティパネル上の【タイマ】【IECUBE】【IECUBE2】【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。
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	現在デバッグ・ツールが保持しているコード・カバレッジ測定結果をすべてクリアします。ただし、使用するデバッグ・ツールがカバレッジ機能をサポートしていない場合、この項目は表示されません。

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。

2. 【V850E1】【V850ES】：【MINICUBE2】【E1】【E20】【EZ Emulator】

タイマ計測イベントをサポートしていないため、この項目は無効となります。

メモリパネル

メモリの内容の表示、変更を行います（「2.9.1 メモリを表示／変更する」参照）。

また、選択しているマイクロコントローラがデータフラッシュ・メモリ内蔵品の場合では、データフラッシュ・メモリ（ID タグを含む）の内容も表示、変更が可能です。

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“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 【V850E2】

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 リセットが発生する場合があります。


- 備考 1. ツールバーの [表示] →  ボタンをクリックすることによりオープンするスクロール範囲設定ダイアログにより、このパネルの垂直スクロール・バーのスクロール範囲を設定することができます。
2. ツールバーの ，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—34 メモリパネル（データフラッシュ・メモリ非内蔵品）

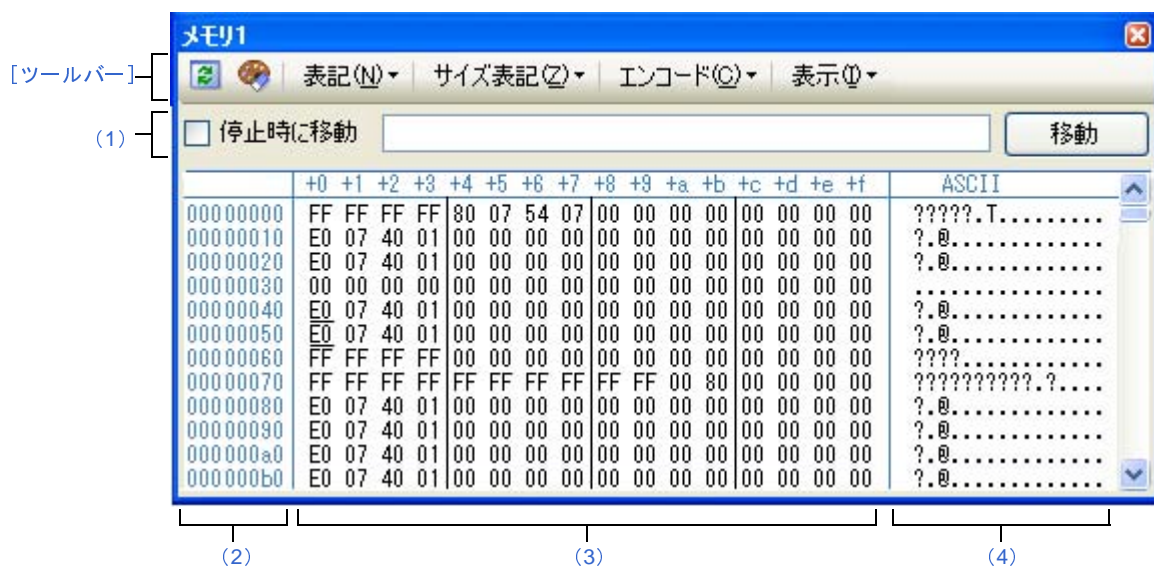


図 A—35 メモリパネル：[CPUメモリ] タブ (データフラッシュ・メモリ内蔵品)



図 A—36 メモリパネル：[IDタグ] タブ (データフラッシュ・メモリ内蔵品)



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ます、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]	<input checked="" type="checkbox"/>	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。
	<input type="checkbox"/>	プログラム停止後、アドレス式の評価を自動的に行いません。 この場合、[移動] ボタンをクリックすることにより、アドレス式の評価を行います。
[移動]		[停止時に移動] チェック・ボックスのチェックをしなかった場合、このボタンをクリックすることによりアドレス式の評価を行い、その計算結果のアドレスにキャレットが移動します。

(2) アドレス・エリア

メモリのアドレスを表示します（16 進数表記固定）。

デフォルトで、0x0 番地より表示を開始します。

ただし、コンテキスト・メニューの [表示アドレス・オフセット値を設定 ...] 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します)。

表示例 (デフォルト)			説明	
00	文字色	青	ユーザにより、値が変更されているメモリ値 ([Enter] キーによりターゲット・メモリに書き込まれます)	
	背景色	標準色		
00 (下線)	文字色	標準色	シンボルが定義されているアドレスのメモリ値 (ウォッチ式の登録を行うことができます)	
	背景色	標準色		
00	文字色	茶色	プログラムの実行により、値が変化したメモリ値 ^注 ツールバーの  ボタンをクリックすると、強調表示をリセットします。	
	背景色	クリーム		
00	文字色	ピンク	リアルタイム表示更新機能を行っているメモリ値	
	背景色	標準色		
00	文字色	標準色	リード／フェッチ	リアルタイム表示更新機能を行っている場合、現在のメモリ値のアクセス状態
	背景色	薄緑		
00	文字色	標準色	ライト	
	背景色	オレンジ		
00	文字色	標準色	リードとライト	
	背景色	薄青		
00	文字色	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)、またはメモリ値の取得に失敗した場合	
	背景色	標準色		
**	文字色	標準色	プログラム実行中に、リアルタイム表示更新領域以外の領域を表示指定した場合、またはメモリ値の取得に失敗した場合	
	背景色	標準色		

注 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。

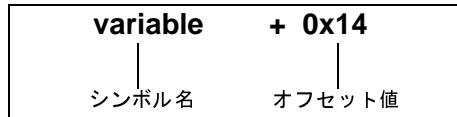
注意 このエリアの表示桁数は、コンテキスト・メニューの [サイズ表記] の指定に従って自動的に変更します。

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わり、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	サイズ	
Float	単精度浮動小数点数	32 ビット	
	数値		<符号><仮数部> e <符号><指数部>
	無限大		Inf, および -Inf
	非数		NaN
	例		+ 1.234567e+123
Double	倍精度浮動小数点数	64 ビット	
	数値		<符号><仮数部> e <符号><指数部>
	無限大		Inf, および -Inf
	非数		NaN
	例		+ 1.2345678901234e+123
Float Complex	単精度浮動小数点数の複素数	64 ビット	
	<単精度浮動小数点数><単精度浮動小数点数> * 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> * I	

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 caret 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わり、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 caret 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)。

(5) タブ選択エリア (【シミュレータ】以外)

このエリアは、選択しているマイクロコントローラがデータフラッシュ・メモリ内蔵品の場合のみ表示されます。

表示するメモリ種別を示すタブを選択します。

表示されるタブは次のとおりです。

タブ名	説明
CPU メモリ	マイクロコントローラにマッピングされているプログラム用のメモリ値を表示します。

タブ名	説明
ID タグ	データフラッシュ・メモリの ID タグの値を表示します。

(6) ID タグ表示エリア (【シミュレータ】以外)













このエリアは、選択しているマイクロコントローラがデータフラッシュ・メモリ内蔵品の場合のみ表示されます。













データフラッシュ・メモリの ID タグのメモリ値を表示します。

ID タグとは、データフラッシュ・メモリに割り当てられているパワー・フェイル検出用のビットで、“0”は、そのデータが正常に書き込めたことを意味しています。

このエリアは、“0”、または“1”の値でのみ変更が可能です。

[ツールバー]

	デバッグ・ツールから最新の情報を取得し、表示を更新します。
	プログラム実行により値が変化した箇所を示す強調表示をリセットします。 ただし、プログラム実行中は無効となります。
表記	メモリ値の表示形式を変更する次のボタンを表示します。 ただし、プログラム実行中は無効となります。
	メモリ値を 16 進数で表示します (デフォルト)。
	メモリ値を符号付き 10 進数で表示します。
	メモリ値を符号なし 10 進数で表示します。
	メモリ値を 8 進数で表示します。
	メモリ値を 2 進数で表示します。
サイズ表記	メモリ値のサイズの表示形式を変更する次のボタンを表示します。 ただし、プログラム実行中は無効となります。
	メモリ値を 4 ビット幅で表示します。
	メモリ値を 8 ビット幅で表示します (デフォルト)。
	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。
	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。
	メモリ値を 64 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。

エンコード	文字列のエンコードを変更する次のボタンを表示します。 ただし、プログラム実行中は無効となります。
	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。
	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。
	文字列を単精度浮動小数点数値で表示します。
	文字列を倍精度浮動小数点数値で表示します。
	文字列を単精度浮動小数点数の複素数で表示します。
	文字列を倍精度浮動小数点数の複素数で表示します。
	文字列を単精度浮動小数点数の虚数で表示します。
	文字列を倍精度浮動小数点数の虚数で表示します。
表示	表示形式を変更する次のボタンを表示します。
	スクロール範囲を設定するための 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・オフセット値を設定 ...	アドレス・エリア に表示するアドレスのオフセット値を設定するため、 アドレス・オフセット設定 ダイアログ をオープンします。

[[ファイル] メニュー (メモリ パネル専用部分)]

メモリ パネル専用の [ファイル] メニューは次のとおりです (その他の項目は共通)。
ただし、プログラム実行中はすべて無効となります。

メモリ・データを保存	メモリの内容を前回保存したテキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存します (「(g) メモリ値の保存 」参照)。 なお、起動後に初めてこの項目を選択した場合は、[名前を付けてメモリ・データを保存 ...] の選択と同等の動作となります。
名前を付けてメモリ・データを保存 ...	メモリの内容を指定したテキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存するために、 データ保存 ダイアログ をオープンします (「(g) メモリ値の保存 」参照)。

[[編集] メニュー (メモリ パネル専用部分)]

メモリ パネル専用の [編集] メニューは次のとおりです (その他の項目はすべて無効)。
ただし、プログラム実行中はすべて無効となります。

コピー	選択している範囲を文字列としてクリップ・ボードにコピーします。
貼り付け	クリップ・ボードにコピーされている文字列を caret 位置に貼り付けます。 - メモリ値エリアに貼り付ける場合 : 「(e) コピー/貼り付け 」参照 - 文字列エリアに貼り付ける場合 : 「(c) コピー/貼り付け 」参照

検索 ...	メモリ検索 ダイアログをオープンします。 検索対象となる箇所は、メモリ値エリアと文字列エリアのうち、キャレットのあるエリア内となります。
--------	---

【コンテキスト・メニュー】

ウォッチ 1 に登録	キャレット位置のシンボルをウォッチパネル（ウォッチ 1）に登録します。 ウォッチ式として登録される際は変数名として登録されるため、スコープにより表示されるシンボル名は変化します。 ただし、キャレット位置のメモリ値に対応するアドレスにシンボルが定義されていない場合は無効となります（「(f) ウォッチ式の登録」参照）。
検索 ...	メモリ検索 ダイアログをオープンします。 検索対象となる箇所は、メモリ値エリアと文字列エリア（浮動小数点数値表示を選択している場合を除く）のうち、キャレットのあるエリア内となります。 ただし、プログラム実行中は無効となります。
初期化 ...	メモリ初期化 ダイアログをオープンします。
最新の情報に更新	デバッグ・ツールから最新の情報を取得し、表示を更新します。
コピー	選択している範囲を文字列としてクリップ・ボードにコピーします。 ただし、プログラム実行中は無効となります。
貼り付け	クリップ・ボードにコピーされている文字列をキャレット位置に貼り付けます。 ただし、プログラム実行中は無効となります。 -メモリ値エリアに貼り付ける場合：「(e) コピー／貼り付け」参照 -文字列エリアに貼り付ける場合：「(c) コピー／貼り付け」参照
表記	メモリ値エリアの表示進数を指定するため、次のカスケード・メニューを表示します。 ただし、プログラム実行中は無効となります。
16 進数	メモリ値を 16 進数で表示します（デフォルト）。
符号付き 10 進数	メモリ値を符号付き 10 進数で表示します。
符号無し 10 進数	メモリ値を符号なし 10 進数で表示します。
8 進数	メモリ値を 8 進数で表示します。
2 進数	メモリ値を 2 進数で表示します。
サイズ表記	メモリ値エリアの表示幅を指定するため、次のカスケード・メニューを表示します。 ただし、プログラム実行中は無効となります。
4 ビット	メモリ値を 4 ビット幅で表示します。
1 バイト	メモリ値を 8 ビット幅で表示します（デフォルト）。
2 バイト	メモリ値を 16 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。
4 バイト	メモリ値を 32 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。
8 バイト	メモリ値を 64 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。

エンコード	文字列エリアの表示形式を指定するため、次のカスケード・メニューを表示します。 ただし、プログラム実行中は無効となります。
ASCII	文字列を ASCII コードで表示します (デフォルト)。
Shift_JIS	文字列を Shift_JIS コードで表示します。
EUC-JP	文字列を EUC-JP コードで表示します。
UTF-8	文字列を UTF-8 コードで表示します。
UTF-16	文字列を UTF-16 コードで表示します。
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。
表示	表示形式を変更するため、次のカスケード・メニューを表示します。
スクロール範囲を設定 ...	スクロール範囲を設定するため、 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 をオープンします。
表示アドレス・オフセット値を設定 ...	アドレス・エリア に表示するアドレスのオフセット値を設定するため、 アドレス・オフセット設定 ダイアログ をオープンします。
強調表示	チェックすることにより、プログラムの実行により値が変更されたメモリ値を強調表示します (デフォルト)。 ただし、プログラム実行中は無効となります。
リアルタイム表示更新設定	リアルタイム表示更新設定のため、次のカスケード・メニューを表示します (「(b) リアルタイム表示更新機能」参照)。
リアルタイム表示更新全体設定	リアルタイム表示更新機能の全般設定を行うため、 プロパティ パネル をオープンします。

逆アセンブルパネル

メモリ内容を逆アセンブルした結果（逆アセンブル・テキスト）の表示、ライン・アセンブル（「2.6.4 ライン・アセンブルを行う」参照）、命令レベル・デバッグ（「2.7.3 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【IECUBE】【IECUBE2】【シミュレータ】（「2.13 カバレッジの測定【IECUBE】【IECUBE2】【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。

混合表示モードにすることにより、コード・データに対応するソース・ファイル中のソース・テキストも表示することができます（デフォルト）。

なお、このパネルは、デバッグ・ツールと接続時のみオープンすることができます。

注意 このパネルにフォーカスがある状態でプログラムをステップ実行した場合、実行単位は命令レベル単位となります（「2.7.3 プログラムをステップ実行する」参照）。


- 備考 1.** ツールバーの [表示] →  ボタンをクリックすることでオープンするスクロール範囲設定ダイアログにより、このパネルの垂直スクロール・バーのスクロール範囲を設定することができます。
- 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。
- 3.** ツールバーの , 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—37 逆アセンブルパネル（混合表示モードの場合）

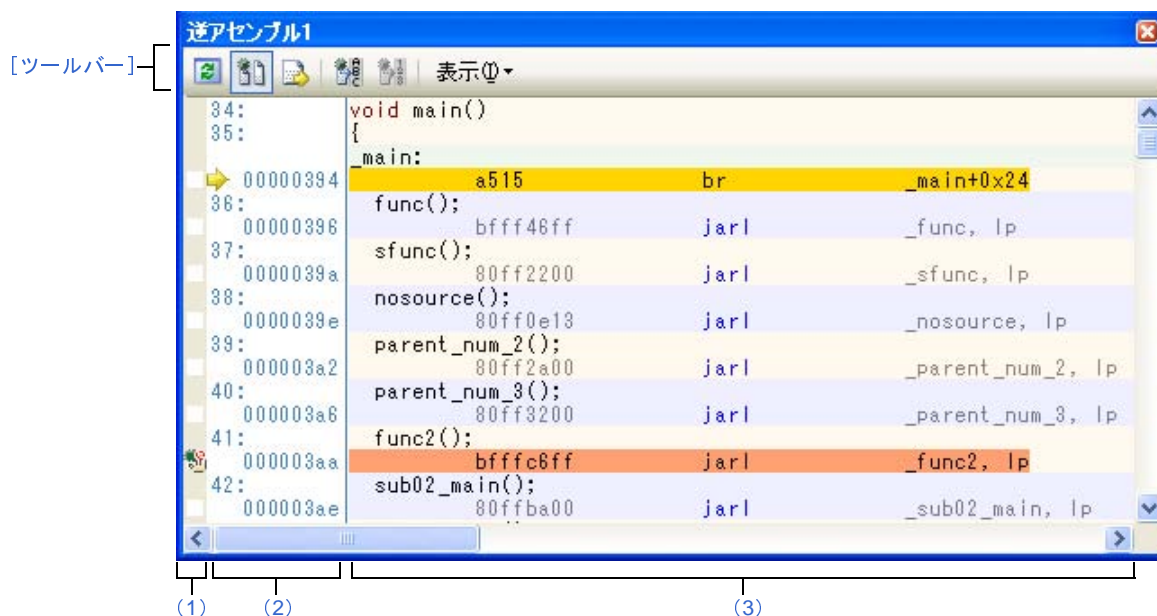


図 A—38 逆アセンブルパネル（混合表示モードを解除した場合）

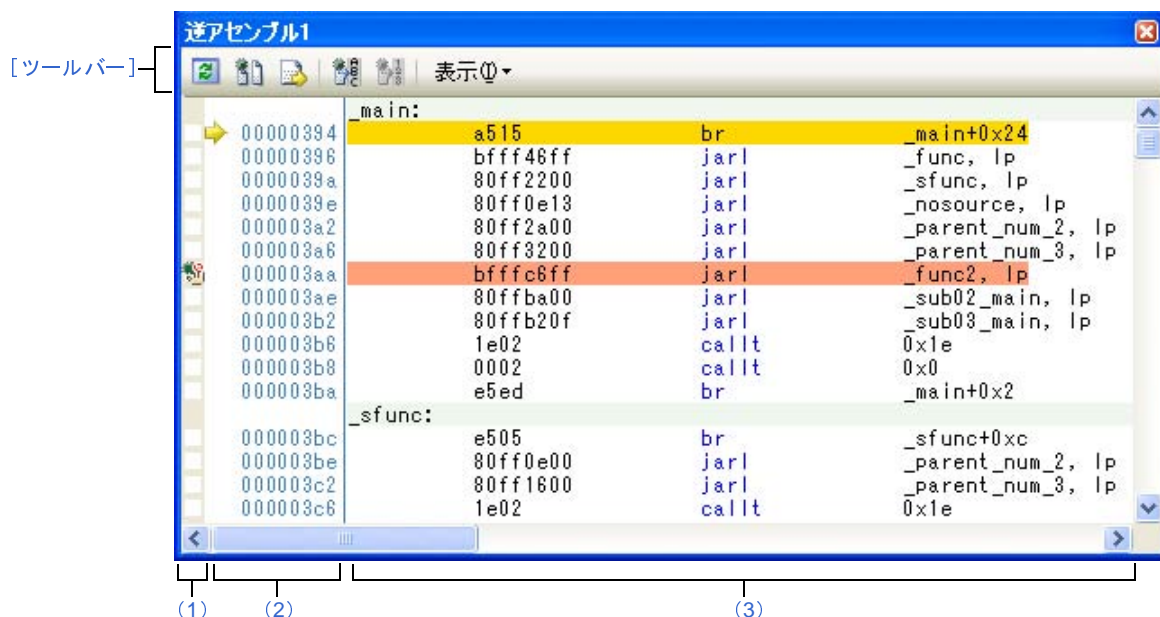
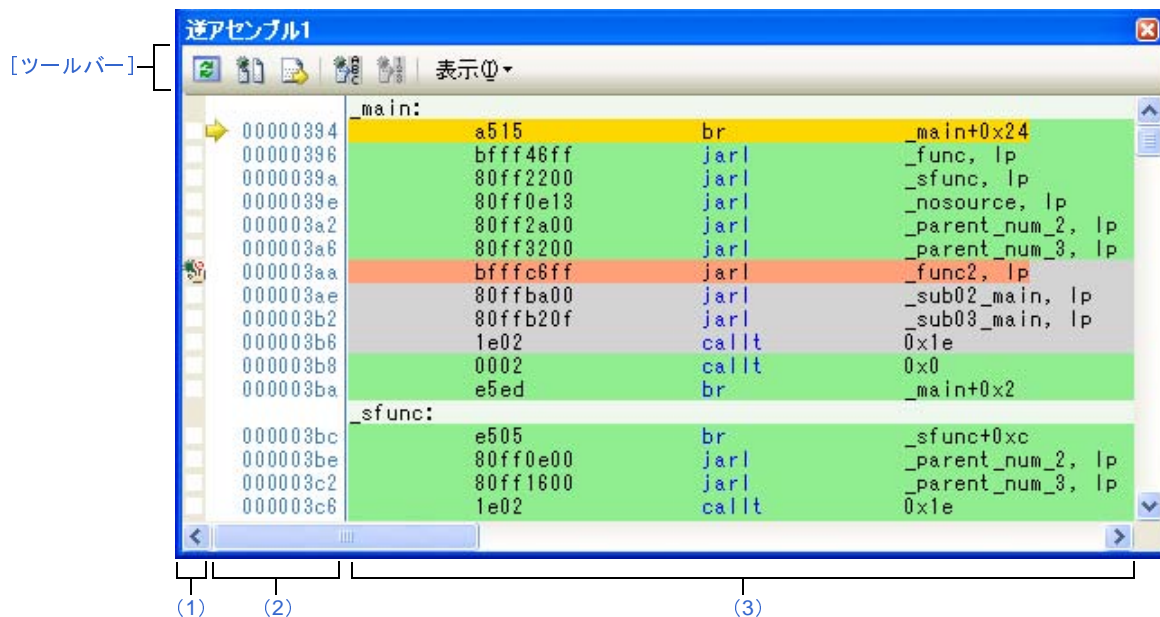


図 A—39 逆アセンブルパネル（コード・カバレージ測定結果を表示した場合）



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（逆アセンブルパネル専用部分）]
- [[編集] メニュー（逆アセンブルパネル専用部分）]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー → [逆アセンブル] → [逆アセンブル 1 ~ 4] を選択

[各エリアの説明]

(1) イベント・エリア

イベントの設定が可能な行は、背景色を白色で表示します（背景色がグレー表示の行は、イベントの設定が不可能であることを示すします）。

また、現在設定しているイベントがある場合、そのイベント設定行に、イベント種別を示す**イベント・マーク**を表示します。

このエリアは、次の機能を備えています。

(a) ブレークポイントの設定／削除

ブレークポイントを設定したい箇所をマウスでクリックすることにより、容易にブレークポイントを設定することができます。

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

ブレークポイントを設定すると、設定した行に**イベント・マーク**が表示されます。また、設定したブレークポイントの詳細情報が**イベントパネル**に反映されます。




なお、すでにいずれかのイベント・マークが表示されている箇所において、この操作を行った場合は、そのイベントを削除し、ブレークポイントの設定は行いません。

ただし、イベントの設定は、背景色が白色で表示されている行に対してのみ行うことができます。

ブレークポイントの設定方法についての詳細は、「[2.8.2 任意の場所で停止する（ブレークポイント）](#)」を参照してください。

(b) 各種イベントの状態変更

各種イベント・マークを右クリックすることにより、次のメニューが表示され、選択したイベントの状態の変更を行うことができます。

有効化	<p>選択しているイベントを有効状態にします。</p> <p>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。</p> <p>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。</p>
無効化	<p>選択しているイベントを無効状態にします。</p> <p>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。</p> <p>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。</p>
イベント削除	<p>選択しているイベントを削除します。</p> <p>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。</p>
詳細設定情報表示	<p>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。</p>

(c) ポップアップ表示

イベント・マークにマウス・カーソルを重ねることにより、そのイベントのイベント名／詳細情報／イベントに付加されたコメントをポップアップ表示します。

なお、該当箇所に複数のイベントが設定されている場合、最大3つまで、各イベントの情報を列挙して表示します。

(2) アドレス・エリア

行ごとの逆アセンブル開始アドレスを表示します（16進数表記固定）。

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

アドレス幅は、プロジェクトで指定しているマイクロコントローラのメモリ空間のアドレス幅となります。

なお、混合表示モード時におけるソース・テキスト行に対しては、開始アドレスに対応するソース・ファイル中の行番号（xxx:）を表示します。

このエリアは、次の機能を備えています。

(a) ポップアップ表示

アドレス／ソース行番号にマウス・カーソルを重ねることにより、次の情報をポップアップ表示します。

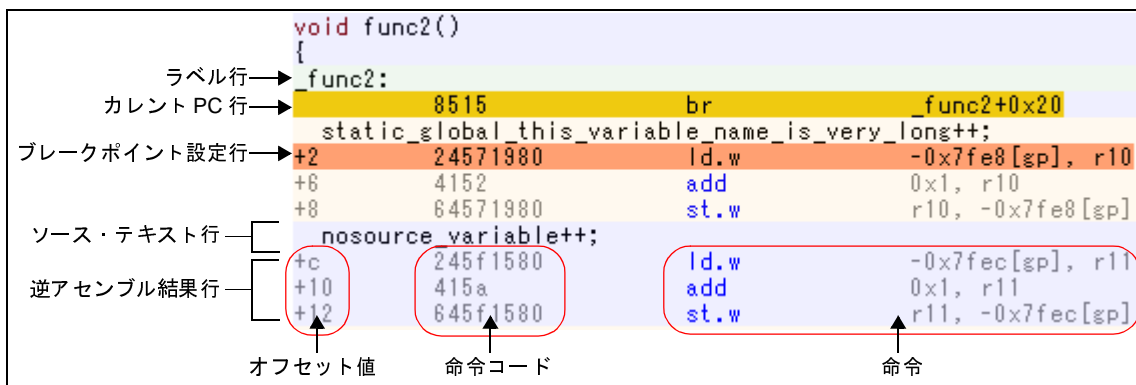
アドレス	形式： <ロード・モジュール名>注\$<ラベル名>+<オフセット値> 例 1： test1.out\$main + 0x10 例 2： subfunction + 0x20
ソース行番号	形式： <ロード・モジュール名>注\$<ファイル名>#<行番号> 例 1： test1.out\$main.c#40 例 2： main.c#100

注 ロード・モジュール名は、ロード・モジュール・ファイルが複数ダウンロードされている場合のみ表示します。

(3) 逆アセンブル・エリア



対象となるソース・テキスト行に続き、逆アセンブル結果行を次のように表示します。

図 A—40 逆アセンブル・エリアの表示内容（混合表示モードの場合）



ラベル行	アドレスにラベルが定義されている場合は、ラベル名を表示し、行全体を薄緑色で強調表示します。	
カレント PC 行	カレント PC 位置 (PC レジスタ値) のアドレスと対応する行を強調表示 ^{注1} します。	
ブレークポイント設定行	ブレークポイントが設定されている行を強調表示 ^{注1} します。	
ソース・テキスト行	コード・データに対応するソース・テキストを表示します ^{注2} 。	
逆アセンブル結果行	オフセット値	アドレスにラベルが定義されていない場合は、一番近いラベルからのオフセット値を表示します ^{注3} 。
	命令コード	逆アセンブルの対象となったコードを 16 進数で表示します。
	命令	逆アセンブル結果として命令を表示します。ニモニックは青色で強調表示します。

注 1. 強調色は、[オプション ダイアログ](#)における [\[全般 - フォントと色\]](#) カテゴリの設定に依存します。

2. ツールバーの  ボタン (トグル) のクリック、またはコンテキスト・メニューの [\[混合表示モード\]](#) のチェックを外すことにより、ソース・テキストを非表示にすることができます (デフォルトでチェックされています)。
3. オフセット値はデフォルトでは表示されません。表示する場合は、ツールバーの  ボタンのクリック、またはコンテキスト・メニューの [\[ラベルのオフセットを表示\]](#) を選択してください。

このエリアは、次の機能を備えています。

(a) ライン・アセンブル

表示されている命令/コードは、編集 (ライン・アセンブル) することができます。

操作方法についての詳細は、「[2.6.4 ライン・アセンブルを行う](#)」を参照してください。

(b) 命令レベルでのプログラム実行

このパネルにフォーカスがある状態でプログラムをステップ実行することにより、命令レベル単位で実行を制御することができます。

操作方法についての詳細は、「[2.7.3 プログラムをステップ実行する](#)」を参照してください。

(c) 各種イベントの設定

コンテキスト・メニューの [\[ブレークの設定\]](#) / [\[トレース設定\]](#) / [\[タイマ設定\]](#) を選択することにより、現在キャレットのあるアドレス/行に各種イベントを設定することができます。

イベントを設定することにより、対応する [イベント・マーク](#) が [イベント・エリア](#) に表示されます。また、設定したイベントの詳細情報が [イベント パネル](#) に反映されます。

ただし、イベントの設定は、[イベント・エリア](#)において、背景色が白色で表示されている行に対してのみ行うことができます。

イベントの設定方法についての詳細は次を参照してください。

- 「[2.8.4 変数 I/O レジスタへのアクセスで停止する](#)」
- 「[2.11.3 任意区間の実行履歴を収集する](#)」
- 「[2.11.4 条件を満たしたときのみの実行履歴を収集する](#)」
- 「[2.12.2 任意区間の実行時間を計測する](#)」


備考 ブレークポイントの設定／削除は、[イベント・エリア](#)においても簡単に行うことができます
(「[\(a\) ブレークポイントの設定／削除](#)」参照)。


(d) ウォッチ式の登録

表示されている C 言語変数 /CPU レジスタ /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カーレットを移動した状態で、ツールバーの  ボタンをクリック、またはコンテキスト・メニューの [シンボルへ移動] を選択することにより、カーレット位置のシンボルが定義されているアドレスにカー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、カーレット移動前のシンボルを参照している命令にカー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、現在のカーレット位置のアドレスに対応するソース行にカーレットを移動した状態で [エディタ パネル](#)がオープンします (すでにオープンしている場合は、エディタ パネルにジャンプ)。

また、同様に [メモリへジャンプ] を選択することにより、現在のカーレット位置のアドレスに対応するメモリ値にカー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 【IEUCBE】 【IEUCBE2】 【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、「[2.13 カバレッジの測定 【IEUCUBE】 【IEUCUBE2】 【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]

	デバッグ・ツールから最新の情報を取得し、表示を更新します。
	逆アセンブル結果とソース・テキストとの対応を表示する。混合表示モードに設定します（デフォルト）。
	キャレット位置をカレント PC 値に追従するように指定します。
	選択しているシンボルの定義位置へキャレットを移動します。
	 ボタンで移動する直前の位置（アドレス）へ移動します。
表示	逆アセンブル・エリアの表示形式を変更する次のボタンを表示します。
	ラベルのオフセット値を表示します。アドレスにラベルが定義されていない場合、一番近いラベルからのオフセット値を表示します。
	アドレス値を“シンボル+オフセット値”で表示します（デフォルト）。 ただし、アドレス値にシンボルが定義されている場合は、シンボルのみを表示します。
	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。
	スクロール範囲を設定するための スクロール範囲設定 ダイアログ がオープンします。

[[ファイル] メニュー（逆アセンブルパネル専用部分）]

逆アセンブルパネル専用の [ファイル] メニューは次のとおりです（その他の項目は共通）。

ただし、プログラム実行中はすべて無効となります。

逆アセンブル・データを保存	逆アセンブルの内容を前回保存したテキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存します（「(h) 逆アセンブル・データの保存 」参照）。 なお、起動後に初めてこの項目を選択した場合は、[名前を付けて逆アセンブル・データを保存 ...] の選択と同等の動作となります。
名前を付けて逆アセンブル・データを保存 ...	逆アセンブルの内容を指定したテキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存するために、 データ保存 ダイアログ をオープンします（「(h) 逆アセンブル・データの保存 」参照）。
印刷 ...	このパネルの内容を印刷するために、 印刷アドレス範囲設定 ダイアログ をオープンします。

[[編集] メニュー（逆アセンブルパネル専用部分）]

逆アセンブルパネル専用の [編集] メニューは次のとおりです（その他の項目はすべて無効）。

コピー	行を選択している場合、選択している行の内容を文字列としてクリップ・ボードにコピーします。 編集モードの場合、選択している文字列をクリップ・ボードにコピーします。
名前の変更	キャレット位置の命令／命令コードを編集するために、編集モードに移行します（ 2.6.4 ライン・アセンブルを行う 参照）。 ただし、プログラム実行中は無効となります。
検索 ...	検索・置換 ダイアログ を [一括検索] タブが選択状態でオープンします。

置換 ...	検索・置換 ダイアログを [一括置換] タブが選択状態でオープンします。
移動 ...	指定したアドレスへキャレットを移動するため、 指定位置へ移動 ダイアログ をオープンします。

[コンテキスト・メニュー]

【逆アセンブル・エリア/アドレス・エリア】

ウォッチ 1 に登録	選択している文字列、またはキャレット位置の単語をウォッチ式として ウォッチ パネル (ウォッチ 1) に登録します (単語の判断は現在のビルド・ツールに依存)。 ウォッチ式として登録する際は変数名として登録されるため、スコープにより表示されるシンボル名は変化します。
アクション・イベントの登録 ...	キャレット位置のアドレスにアクション・イベントを設定するため、 アクション・イベント ダイアログ をオープンします。
ここまで実行	カレント PC 値で示されるアドレスから、キャレット位置の行に対応するアドレスまでプログラムを実行します。 ただし、プログラム実行中、または [ビルド & デバッグ・ツールヘダダウンロード] 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・ツールヘダダウンロード] 実行中は無効となります。
移動 ...	指定したアドレスへキャレットを移動するため、 指定位置へ移動 ダイアログ をオープンします。
シンボルへ移動	選択しているシンボルの定義位置へキャレットを移動します。
アドレスへ戻る	[シンボルへ移動] で移動する直前の位置 (アドレス) へ移動します。 ただし、アドレスにシンボル名が表示されていない場合は無効となります。

ブレークの設定	ブレーク関連のイベントを設定するために、次のカスケード・メニューを表示します。 なお、イベントは、イベントの設定が可能な行のみ設定することができます（「(1) イベント・エリア」参照）。
ハード・ブレークの設定	キャレット位置のアドレスにブレークポイント（ハードウェア・ブレーク・イベント）を設定します（「(1) ブレーク・イベント（実行系）を設定する」参照）。
ソフト・ブレークの設定 （【シミュレータ】以外）	キャレット位置のアドレスにブレークポイント（ソフトウェア・ブレーク・イベント）を設定します（「(1) ブレーク・イベント（実行系）を設定する」参照）。
組み合わせブレークを設定	今版ではサポートしていません。
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	キャレット位置、または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）I/Oレジスタに、リード／ライト・アクセスのブレーク・イベントを設定します（「(1) ブレーク・イベント（アクセス系）を設定する」参照）。
読み込み組み合わせブレークを設定	今版ではサポートしていません。
書き込み組み合わせブレークを設定	今版ではサポートしていません。
読み書き組み合わせブレークを設定	今版ではサポートしていません。
ブレーク動作の設定	プロパティパネルをオープンし、ブレーク機能の設定を行います。

トレース設定 【IECUBE】【IECUBE2】 【シミュレータ】	トレース関連のイベントを設定するために、次のカスケード・メニューを表示します。 なお、イベントは、イベントの設定が可能な行のみ設定することができます（「(1) イベント・エリア」参照）。
トレース開始の設定	キャレット位置のアドレスの命令が実行された際に、プログラムの実行履歴を示すトレース・データの収集を開始するトレース開始イベントを設定します（「2.11.3 任意区間の実行履歴を収集する」参照）。 【シミュレータ】 プロパティパネル上の【トレース】【IECUBE】【IECUBE2】【シミュレータ】カテゴリ内【トレース機能を使用する】プロパティの設定を自動的に【はい】にします。
トレース終了の設定	キャレット位置のアドレスの命令が実行された際に、プログラムの実行履歴を示すトレース・データの収集を終了するトレース終了イベントを設定します（「2.11.3 任意区間の実行履歴を収集する」参照）。 【シミュレータ】 プロパティパネル上の【トレース】【IECUBE】【IECUBE2】【シミュレータ】カテゴリ内【トレース機能を使用する】プロパティの設定を自動的に【はい】にします。
値をトレースに記録（読み込み時）	キャレット位置、または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）I/Oレジスタにリード・アクセスした際に、その値をトレース・メモリに記録するポイント・トレース・イベントを設定します（「(1) 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（書き込み時）	キャレット位置、または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）I/Oレジスタにライト・アクセスした際に、その値をトレース・メモリに記録するポイント・トレース・イベントを設定します（「(1) 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（読み書き時）	キャレット位置、または選択している変数（グローバル変数／関数内スタティック変数／ファイル内スタティック変数）I/Oレジスタにリード／ライト・アクセスした際に、その値をトレース・メモリに記録するポイント・トレース・イベントを設定します（「(1) ポイント・トレース・イベントを設定する」参照）。
読み書き時にトレース開始の設定	今版ではサポートしていません。
読み書き時にトレース終了の設定	今版ではサポートしていません。
トレース結果の表示	トレースパネル【IECUBE】【IECUBE2】【シミュレータ】をオープンし、取得したトレース・データを表示します。
トレース動作の設定	プロパティパネルをオープンし、トレース機能の設定を行います。 ただし、トレーサ動作中は無効となります。

タイマ設定	タイマ関連のイベント ^注 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 (「2.12.2 任意区間の実行時間を計測する」参照)。 なお、イベントは、イベントの設定が可能な行のみ設定することができます (「(1) イベント・エリア」参照)。
実行時にタイマ開始	キャレット位置のアドレスの命令が実行された際に、プログラムの実行時間の計測を開始するタイマ開始イベントを設定します。 【シミュレータ】 プロパティパネル上の [タイマ] 【IECUBE】 【IECUBE2】 【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。
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。 【シミュレータ】 プロパティパネル上の [タイマ] 【IECUBE】 【IECUBE2】 【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。
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	現在デバッグ・ツールが保持しているコード・カバレッジ測定結果をすべてクリアします。 ただし、使用するデバッグ・ツールがカバレッジ機能をサポートしていない場合、この項目は表示されません。
命令の編集	キャレット位置の行の命令を編集するために、編集モードに移行します (「2.6.4 ライン・アセンブルを行う」参照)。 ただし、プログラム実行中は無効となります。
コードの編集	キャレット位置の行の命令コードを編集するために、編集モードに移行します (「2.6.4 ライン・アセンブルを行う」参照)。 ただし、プログラム実行中は無効となります。
表示	逆アセンブル・エリアの表示内容を設定するために、次のカスケード・メニューを表示します。
ラベルのオフセットを表示	ラベルのオフセット値を表示します。アドレスにラベルが定義されていない場合、一番近いラベルからのオフセット値を表示します。
アドレス値をシンボルで表示	アドレス値を“シンボル+オフセット値”で表示します (デフォルト)。 ただし、アドレス値にシンボルが定義されている場合は、シンボルのみを表示します。
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	逆アセンブル結果とソース・テキストとの対応を表示する、混合表示モードに設定します (デフォルト)。
ソースヘジャンプ	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 【V850E1】 【V850ES】 : 【MINICUBE2】 【E1】 【E20】 【EZ Emulator】

タイマ計測イベントをサポートしていないため、この項目は無効となります。

【イベント・エリア】（【シミュレータ】以外）

ハードウェア・ブ레이크を優先する	マウスのワンクリック操作で設定できるブ레이크の種類をハードウェア・ブ레이크ポイントとします（プロパティパネル上の【ブ레이크】カテゴリ内【優先的に使用するブ레이크ポイントの種類】プロパティの設定に反映されます）。
ソフトウェア・ブ레이크を優先する	マウスのワンクリック操作で設定できるブ레이크の種類をソフトウェア・ブ레이크ポイントとします（プロパティパネル上の【ブ레이크】カテゴリ内【優先的に使用するブ레이크ポイントの種類】プロパティの設定に反映されます）。

CPU レジスタ パネル

CPU レジスタ（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（「[2.9.2 CPU レジスタを表示／変更する](#)」参照）。

なお、このパネルは、デバッグ・ツールと接続時のみオープンすることができます。

- 備考 1. ツールバーの 、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。

図 A—41 CPU レジスタ パネル



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー（CPU レジスタ パネル専用部分）]
- [[編集] メニュー（CPU レジスタ パネル専用部分）]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー → [CPU レジスタ] を選択





[各エリアの説明]

(1) [レジスタ名] エリア

レジスタの種別をカテゴリ（フォルダ）として分類し、各レジスタ名を一覧表示します。

表示される各アイコンの意味は次のとおりです。

なお、カテゴリ名／レジスタ名を編集／削除することはできません。

	このカテゴリに属するレジスタ名を表示している状態を示します。アイコンをダブルクリック、または“-”マークをクリックすると、カテゴリを閉じてレジスタ名を非表示にします。
	このカテゴリに属するレジスタ名が非表示の状態を示します。アイコンをダブルクリック、または“+”マークをクリックすると、カテゴリを開いてレジスタ名を表示します。
	レジスタ名を表示します。アイコンをダブルクリック、または“+” / “-” マークをクリックすると、下階層のレジスタ名（レジスタの部分を表す名称）を表示／非表示します。
	レジスタ名（レジスタの部分を表す名称）を表示します。

表示されるカテゴリ名／レジスタ名は次のとおりです（各レジスタ名の先頭の“+”マークの数は、表示される際の階層の深さを示します）。

表 A—8 [汎用レジスタ] カテゴリとレジスタ名【V850E1】【V850ES】

レジスタ名（別名）	ビット幅	レジスタ名（別名）	ビット幅
+ r0	32	+ r5(TP)	32
+ r1	32	+ r6 ~ + r29	32
+ r2	32	+ r30(EP)	32
+ r3(SP)	32	+ r31(LP)	32
+ r4(GP)	32		

表 A—9 [汎用レジスタ] カテゴリとレジスタ名【V850E2】

レジスタ名（別名）	ビット幅	レジスタ名（別名）	ビット幅
+ 整数	-	+ 単精度浮動小数点数	-
++ r0	32	++ fp1 ~ + fp30	32
++ r1	32	++ fp31	32
++ r2	32	+ 倍精度浮動小数点数	-
++ r3(SP)	32	++ dp2	64
++ r4(GP)	32	++ dp2 ~ + dp28	64
++ r5(TP)	32	++ dp30	64
+ r6 ~ + r29	32		
++ r30(EP)	32		
++ r31(LP)	32		

表 A—10 [システム・レジスタ] カテゴリとレジスタ名【V850E1】【V850ES】

レジスタ名	ビット幅	レジスタ名	ビット幅
+ PC	32	+ CTPC	32
+ EIPC	32	+ CTPSW	32
+ EIPSW	32	++ NP	1
++ NP	1	++ EP	1
++ EP	1	++ ID	1
++ ID	1	++ SAT	1
++ SAT	1	++ CY	1
++ CY	1	++ OV	1
++ OV	1	++ S	1
++ S	1	++ Z	1
++ Z	1	+ DBPC	32
+ FEPC	32	+ DBPSW	32
+ FEPSW	32	++ NP	1
++ NP	1	++ EP	1
++ EP	1	++ ID	1
++ ID	1	++ SAT	1
++ SAT	1	++ CY	1
++ CY	1	++ OV	1
++ OV	1	++ S	1
++ S	1	++ Z	1
++ Z	1	+ CTBP	32
+ ECR	32		
+ PSW	32		
++ NP	1		
++ EP	1		
++ ID	1		
++ SAT	1		
++ CY	1		
++ OV	1		
++ S	1		
++ Z	1		

表 A—11 [システム・レジスタ] カテゴリとレジスタ名 [V850E2]

レジスタ名	ビット幅	レジスタ名	ビット幅
+ PC	32	++ ソフトウェア・ページング・バンク	-
+ CPU 機能グループ	-	+++ MPM	32
++ 基本バンク	-	+++ MPC	32
+++ EIPC	32	+++ TID	32
+++ EIPSW	32	+++ VMECR	32
+++ FEPC	32	+++ VMTID	32
+++ FEPSW	32	+++ VMADR	32
+++ ECR	32	+++ IPA0L	32
+++ PSW	32	+++ IPA0U	32
++++ PP	1	+++ IPA1L	32
++++ NPV	1	+++ IPA1U	32
++++ DMP	1	+++ IPA2L	32
++++ IMP	1	+++ IPA2U	32
++++ NP	1	+++ IPA3L	32
++++ EP	1	+++ IPA3U	32
++++ ID	1	+++ IPA4L	32
++++ SAT	1	+++ IPA4U	32
++++ CY	1	+++ DPA0L	32
++++ OV	1	+++ DPA0U	32
++++ S	1	+++ DPA1L	32
++++ Z	1	+++ DPA1U	32
+++ SCCFG	32	+++ DPA2L	32
+++ SCBP	32	+++ DPA2U	32
+++ EIIC	32	+++ DPA3L	32
+++ FEIC	32	+++ DPA3U	32
+++ CTPC	32	+++ DPA4L	32
+++ CTPSW	32	+++ DPA4U	32
+++ CTBP	32	+++ DPA5L	32
++ 例外ハンドラ切り替え機能バンク 0	-	+++ DPA5U	32
+++ SW_CTL	32	+ FPU 機能グループ	-
+++ SW_CFG	32	++ FPU ステータス・バンク	-
+++ SW_BASE	32	+++ FPSR	32
++ 例外ハンドラ切り替え機能バンク 1	-	+++ FPEPC	32
+++ EH_CFG	32	+++ FPST	32

レジスタ名	ビット幅	レジスタ名	ビット幅
+++ EH_RESET	32	+++ FPCC	32
+++ EH_BASE	32	+++ FPCFG	32
+ プロセッサ保護グループ	-	+++ FPEC	32
++ プロセッサ保護違反バンク	-	+ ユーザ・グループ	-
+++ VSECR	32	++ ユーザ0バンク	-
+++ VSTID	32	+++ PSW	32
+++ VSADR	32	++++ PP	1
+++ VMECR	32	++++ NPV	1
+++ VMTID	32	++++ DMP	1
+++ VMADR	32	++++ IMP	1
+++ MCA	32	++++ NP	1
+++ MCS	32	++++ EP	1
+++ MCC	32	++++ ID	1
+++ MCR	32	++++ SAT	1
++ プロセッサ保護設定バンク	-	++++ CY	1
+++ MPM	32	++++ OV	1
+++ MPC	32	++++ S	1
+++ TID	32	++++ Z	1
+++ IPA0L	32	+++ FPST	32
+++ IPA0U	32	+++ FPCC	32
+++ IPA1L	32	+++ FPCFG	32
+++ IPA1U	32	+++ CTPC	32
+++ IPA2L	32	+++ CTPSW	32
+++ IPA2U	32	++++ PP	1
+++ IPA3L	32	++++ NPV	1
+++ IPA3U	32	++++ DMP	1
+++ IPA4L	32	++++ IMP	1
+++ IPA4U	32	++++ NP	1
+++ DPA0L	32	++++ EP	1
+++ DPA0U	32	++++ ID	1
+++ DPA1L	32	++++ SAT	1
+++ DPA1U	32	++++ CY	1
+++ DPA2L	32	++++ OV	1
+++ DPA2U	32	++++ S	1
+++ DPA3L	32	++++ Z	1
+++ DPA3U	32	+++ CTBP	32
+++ DPA4L	32	+ 共通グループ	-

レジスタ名	ビット幅	レジスタ名	ビット幅
+++ DPA4U	32	++ EIWR	32
+++ DPA5L	32	++ FEWR	32
+++ DPA5U	32	++ BSEL	32
		+++ GRP	8
		+++ BNK	8

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（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します）。

表示例（デフォルト）			説明
0x0	文字色	青色	ユーザにより、値が変更されている CPU レジスタの値（[Enter] キーによりターゲット・メモリに書き込まれます）
	背景色	標準色	
0x0	文字色	茶色	プログラムの実行により、値が変化した CPU レジスタの値 プログラムを再実行させることにより、強調色をリセットします。
	背景色	クリーム	

このエリアは、次の機能を備えています。

(a) CPU レジスタ値の変更

CPU レジスタ値の変更は、対象 CPU レジスタ値を選択したのち、再度クリックし、キーボードからの直接入力により行います（[Esc] キーの押下で編集モードをキャンセルします）。












CPU レジスタ値を編集したのち、[Enter] キーの押下、または編集領域以外へのフォーカスの移動により、デバッグ・ツールのレジスタに書き込まれます。

(b) CPU レジスタ値の保存

[ファイル] メニュー→ [名前を付けて CPU レジスタ・データを保存...] を選択することにより、**名前を付けて保存 ダイアログ**をオープンし、このパネルのすべての内容をテキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存することができます。

CPU レジスタ値の保存方法についての詳細は、「(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]

	デバッグ・ツールから最新の情報を取得し、表示を更新します。 ただし、プログラム実行中は無効となります。
表記	値の表示形式を変更する次のボタンを表示します。
	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）。
	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。
	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。
	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。
	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。
	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。
	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。 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。
	選択している項目を Double で表示します。 ただし、8 バイト・データ以外の場合は、規定値で表示します。
	値表示の末尾に、その値の 16 進数表記を “()” で囲んで併記します。

[[ファイル] メニュー (CPU レジスタ パネル専用部分)]

CPU レジスタ パネル専用の [ファイル] メニューは次のとおりです（その他の項目は共通）。

ただし、プログラム実行中はすべて無効となります。

CPU レジスタ・データを保存	このパネルの内容を前回保存したテキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存します（「(b) CPU レジスタ値の保存」参照）。 なお、起動後に初めてこの項目を選択した場合は、[名前を付けて CPU レジスタ・データを保存...] の選択と同等の動作となります。
名前を付けて CPU レジスタ・データを保存 ...	このパネルの内容を指定したテキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存するために、 名前を付けて保存 ダイアログ をオープンします（「(b) CPU レジスタ値の保存」参照）。

[[編集] メニュー (CPU レジスタ パネル専用部分)]

CPU レジスタ パネル専用の [編集] メニューは次のとおりです (その他の項目はすべて無効)。

切り取り	選択範囲の文字列を切り取り、クリップ・ボードにコピーします。 ただし、文字列を編集中の場合のみ有効となります。
コピー	編集中の場合、選択している文字列をクリップ・ボードにコピーします。 行を選択している場合、レジスタ/カテゴリをクリップ・ボードにコピーします。 なお、コピーした項目は、 ウォッチ パネル に貼り付け可能です。
貼り付け	クリップ・ボードにコピーされている文字列をキャレット位置に貼り付けます。 ただし、文字列を編集中の場合のみ有効となります。
すべて選択	すべての項目を選択状態にします。
検索 ...	検索・置換 ダイアログを [一括検索] タブが選択状態でオープンします。
置換 ...	検索・置換 ダイアログを [一括置換] タブが選択状態でオープンします。

[コンテキスト・メニュー]

ウォッチ 1 に登録	選択しているレジスタ名/カテゴリを ウォッチ パネル (ウォッチ 1) に登録します。
コピー	編集中の場合、選択している文字列をクリップ・ボードにコピーします。 行選択している場合、レジスタ項目/カテゴリをクリップ・ボードにコピーします。 なお、コピーした項目は、 ウォッチ パネル に貼り付け可能です。
表記	表示形式を指定するため、次のカスケード・メニューを表示します。
自動	選択している項目 (下位項目を含む) の値を規定値で表示します (デフォルト)。
16 進数	選択している項目 (下位項目を含む) の値を 16 進数で表示します。
符号付き 10 進数	選択している項目 (下位項目を含む) の値を符号付き 10 進数で表示します。
符号無し 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（「2.9.3 I/O レジスタを表示／変更する」参照）。

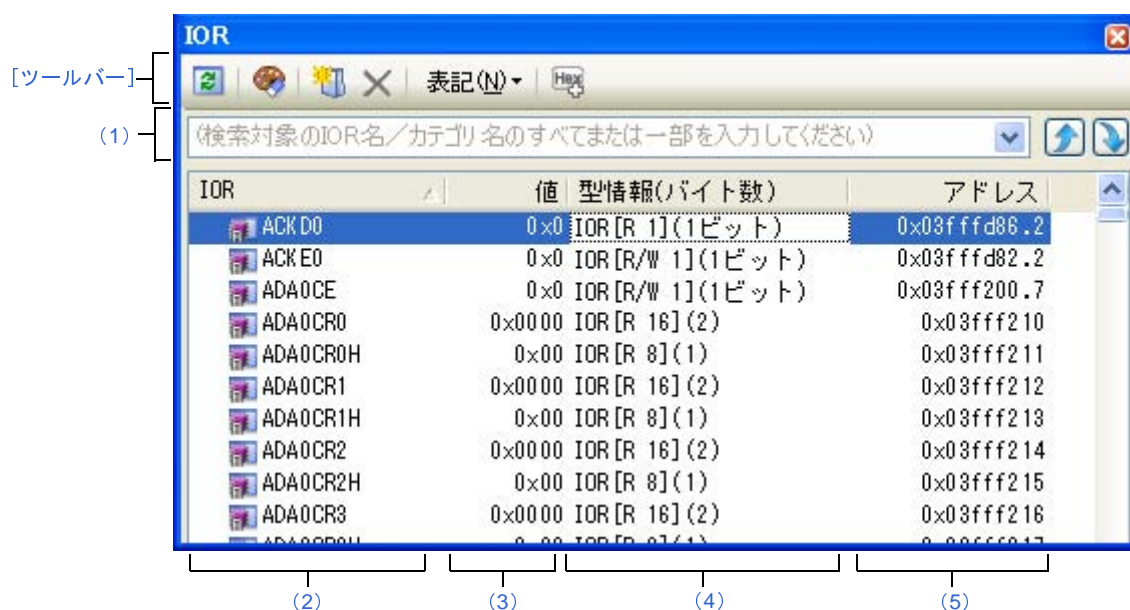
なお、このパネルは、デバッグ・ツールと接続時のみオープンすることができます。

注意 読み込み動作によってマイクロコントローラが動作してしまう I/O レジスタは、読み込み保護対象となるため、値の読み込みは行いません（[値] に“?” を表示）。

読み込み保護対象の I/O レジスタ の内容を取得したい場合は、コンテキスト・メニューの [値を強制読み込み] を選択してください。

- 備考 1. ツールバーの 、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。

図 A—42 IOR パネル



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー (IOR パネル専用部分)]
- [[編集] メニュー (IOR パネル専用部分)]
- [コンテキスト・メニュー]




[オープン方法]

- [表示] メニュー→ [IOR] を選択



[各エリアの説明]

(1) 検索エリア

I/O レジスタ名の検索を行います。

	検索対象の文字列を指定します（大文字／小文字不問）。 キーボードより文字列を直接入力するか（最大指定文字数：512 文字）、ドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。
	テキスト・ボックスで指定している文字列を含む I/O レジスタ名を上方向に検索し、検索結果を選択状態にします。
	テキスト・ボックスで指定している文字列を含む I/O レジスタ名を下方向に検索し、検索結果を選択状態にします。




備考 1. カテゴリ（フォルダ）により分類されて非表示の状態の I/O レジスタ名も検索します（展開して選択状態となります）。

2. 検索対象の文字列入力後、[Enter] キーを押下することにより、 ボタンのクリックと同等の動作を行い、[Shift] + [Enter] キーを押下することにより、 ボタンのクリックと同等の動作を行います。

(2) [IOR] エリア

I/O レジスタの種別をカテゴリ（フォルダ）として分類し、各 I/O レジスタ名を一覧表示します。

表示される各アイコンの意味は次のとおりです。


	このカテゴリに属する I/O レジスタ名を表示している状態を示します。アイコンをダブルクリック、または“-”マークをクリックすると、カテゴリを閉じ I/O レジスタ名を非表示にします。 なお、カテゴリはデフォルトでは存在しません。必要な場合は、カテゴリを新規作成したのち、 ツリーの編集 を行ってください。
	I/O レジスタ名が非表示の状態を示します。アイコンをダブルクリック、または“+”マークをクリックすると、カテゴリを開き I/O レジスタ名を表示します。 なお、カテゴリはデフォルトでは存在しません。必要な場合は、カテゴリを新規作成したのち、 ツリーの編集 を行ってください。
	I/O レジスタ名を示します。


備考 このエリアのヘッダ部をクリックすることにより、カテゴリ名を文字コード順でソートします（カテゴリ内 I/O レジスタ名も同様にソートします）。

このエリアは、次の機能を備えています。

(a) ツリーの編集

各 I/O レジスタを任意のカテゴリ（フォルダ）で分類し、ツリー形式を編集することができます。

カテゴリを新規に作成する場合は、作成したい I/O レジスタ名にキャレットを移動したのち、ツールバーの  ボタンのクリック、またはコンテキスト・メニューの [カテゴリを作成] を選択し、任意にカテゴリ名称を入力することにより行います（最大指定文字数：1024 文字）。

なお、カテゴリを削除する場合は、削除したいカテゴリを選択したのち、ツールバーの  ボタンのクリック、またはコンテキスト・メニューの [削除] を選択します。ただし、削除できるカテゴリは、空のカテゴリのみです。

また、カテゴリ名を編集する場合は、編集したいカテゴリ名を選択したのち、次のいずれかの操作により行います。

- 再度クリック後、キーボードよりカテゴリ名を直接編集
- [編集] メニュー → [名前の変更] を選択後、キーボードよりカテゴリ名を直接編集
- [F2] キーを押下後、キーボードよりカテゴリ名を直接編集

カテゴリを作成したのち、I/O レジスタ名をカテゴリ内に直接ドラッグ・アンド・ドロップすることにより、各 I/O レジスタをカテゴリで分類したツリー形式で表示します。

同様に、カテゴリと I/O レジスタ名の表示の順番（上下位置）も、ドラッグ・アンド・ドロップ操作により自由に変更することができます。

注意 1. カテゴリ内にカテゴリを作成することはできません。

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（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します）。

表示例（デフォルト）			説明
0x0	文字色	青色	ユーザにより、値が変更されている I/O レジスタの値（[Enter] キーによりターゲット・メモリに書き込まれます）
	背景色	標準色	

表示例（デフォルト）			説明
0x0	文字色	茶色	プログラムの実行により、値が変化した I/O レジスタの値 ツールバーの  ボタン、またはコンテキスト・メニューの「表示色をリセット」を選択することにより、強調表示をリセットします。
	背景色	クリーム	
?	文字色	グレー	読み込み保護対象の I/O レジスタ ^注 の値
	背景色	標準色	

注 読み込み動作によってマイクロコントローラが動作してしまう I/O レジスタを示します。

読み込み保護対象の I/O レジスタの内容を取得する場合は、コンテキスト・メニューの「値を強制読み込み」を選択することにより行ってください。

注意 1 バイト /2 バイト I/O レジスタと、1 バイト /2 バイト I/O レジスタに割り付けられている 1 ビット I/O レジスタでは、値を取得するタイミングが異なります。このため、同一の I/O レジスタの値を表示して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 I/O 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 > < すべてのアクセス可能サイズ >] (< サイズ >)

アクセス属性	アクセス属性として、次のいずれかをを表示します。	
	R	リードのみ可能
	W	ライトのみ可能
	R/W	リード/ライト可能
すべてのアクセス可能サイズ	アクセス可能なサイズをビット単位で小さい順に“,”で区切り列挙します。	
サイズ	I/Oレジスタのサイズを表示します。 バイト単位で表示可能な場合はバイト単位で、ビット単位でのみ表示可能な場合はビット単位で単位を付与して表示します。	

例 1. 「IOR [R/W 1.8] (1 バイト)」の場合

リード/ライト可能, 1 ビット・アクセス / 8 ビット・アクセス可能, サイズが 1 バイトの I/O レジスタ

2. 「IOR [R/W 1] (1 ビット)」の場合

リード/ライト可能, 1 ビット・アクセス可能, サイズが 1 ビットの I/O レジスタ

備考 このエリアのヘッダ部をクリックすることにより、型情報を文字コード順でソートします。

(5) [アドレス] エリア

各 I/O レジスタがマッピングされているアドレスを表示します (16 進数表記固定)。

ただし、ビット・レジスタの場合は、次の例のようにビット・オフセット値を付与して表示します。

例 1. 「0xFF40」の場合





アドレス “0xFF40” に割り当てられている








2. 「0xFF40.4」の場合

アドレス “0xFF40” のビット 4 に割り当てられている (ビット・レジスタ)

備考 このエリアのヘッダ部をクリックすることにより、アドレスを数値の昇順でソートします。

[ツールバー]

	デバッグ・ツールから最新の情報を取得し、表示を更新します。 読み込み保護対象の I/O レジスタの再読み込みは行いません。 ただし、プログラム実行中は無効となります。
	選択している I/O レジスタに対して、プログラム実行により値が変化したことを示す強調表示をリセットします。 ただし、プログラム実行中は無効となります。
	新規カテゴリ (フォルダ) を追加します。テキスト・ボックスに直接カテゴリ名を入力します。 なお、新規に作成できるカテゴリの数に制限はありますが、カテゴリ内にカテゴリを作成することはできません。 ただし、プログラム実行中は無効となります。
	選択している範囲の文字列を削除します。 空のカテゴリが選択状態の場合は、そのカテゴリを削除します (I/O レジスタの削除不可)。

表記	値の表示形式を変更する次のボタンを表示します。
	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)。
	選択している項目の値を符号付き 10 進数で表示します。
	選択している項目の値を符号なし 10 進数で表示します。
	選択している項目の値を 8 進数で表示します。
	選択している項目の値を 2 進数で表示します。
	選択している項目の値を ASCII コードで表示します。
	選択している項目の値表示の末尾に、その値の 16 進数表記を “()” で囲んで併記します。

[[ファイル] メニュー (IOR パネル専用部分)]

IOR パネル専用の [ファイル] メニューは次のとおりです (その他の項目は共通)。

ただし、プログラム実行中はすべて無効となります。

IOR データを保存	このパネルの内容を前回保存したテキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存します (「(b) I/O レジスタ値の保存」参照)。 なお、起動後に初めてこの項目を選択した場合は、[名前を付けて IOR データを保存 ...] の選択と同等の動作となります。
名前を付けて IOR データを保存 ...	このパネルの内容を指定したテキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存するために、名前を付けて保存 ダイアログをオープンします (「(b) I/O レジスタ値の保存」参照)。

[[編集] メニュー (IOR パネル専用部分)]

IOR パネル専用の [編集] メニューは次のとおりです (その他の項目はすべて無効)。

切り取り	選択している範囲の文字列を切り取ってクリップ・ボードに移動します (I/O レジスタ /カテゴリの切り取り不可)。
コピー	選択している範囲の文字列をクリップ・ボードにコピーします。 I/O レジスタ /カテゴリが選択状態の場合は、その項目をコピーします。 なお、コピーした項目は、ウォッチパネルに貼り付け可能です。
貼り付け	テキストが編集状態の場合、クリップ・ボードの内容をキャレット位置に挿入します (I/O レジスタ /カテゴリの貼り付け不可)。
削除	選択している範囲の文字列を削除します。 空のカテゴリが選択状態の場合は、その項目を削除します (I/O レジスタの削除不可)。
すべて選択	テキストが編集状態の場合、すべての文字列を選択します。 テキストが編集状態以外の場合、すべての I/O レジスタ /カテゴリを選択状態にします。
名前の変更	選択しているカテゴリの名称を編集します。
検索 ...	検索エリアのテキスト・ボックスにフォーカスを移動します。
移動 ...	指定した I/O レジスタへキャレットを移動するため、指定位置へ移動 ダイアログをオープンします。

[コンテキスト・メニュー]

ウォッチ 1 に登録	選択している I/O レジスタ／カテゴリを ウォッチ パネル (ウォッチ 1) に登録します。
最新の情報に更新	デバッグ・ツールから最新の情報を取得し、表示を更新します。 読み込み保護対象の I/O レジスタの再読み込みは行いません。 ただし、プログラム実行中は無効となります。
値を強制読み込み	読み込み保護対象の I/O レジスタの値を 1 回強制的に読み込みます。
移動 ...	指定位置へ移動 ダイアログ をオープンします。
カテゴリを作成	新規カテゴリ (フォルダ) を追加します。テキスト・ボックスに直接カテゴリ名を入力します。 なお、新規に作成できるカテゴリの数に制限はありませんが、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（「2.9.5 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。

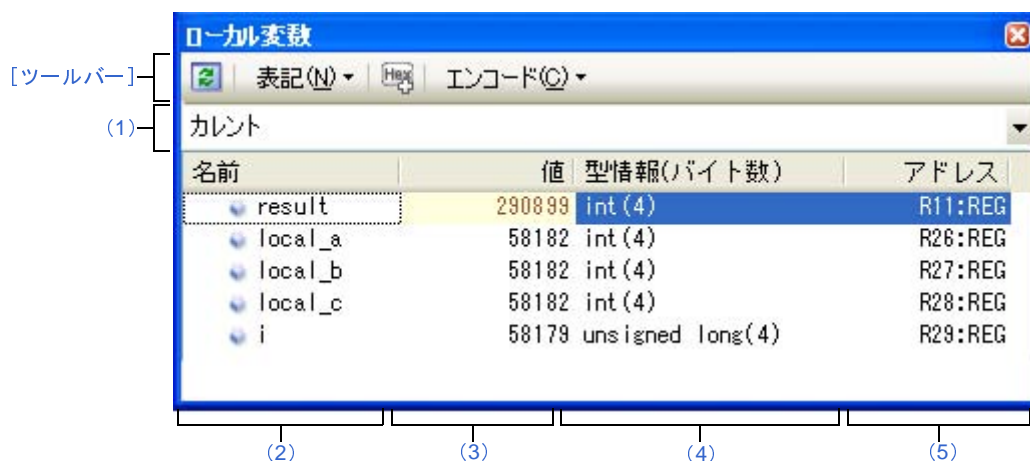
注意 1. プログラム実行中は、このパネルには何も表示されません。プログラムの実行が停止したタイミングで、各エリアの表示を行います。

2. コンパイラによる最適化のため、対象となる変数を使用していないブロックでは変数データがスタック／レジスタに存在しない場合があります。この場合は対象となる変数は表示されません。

備考 1. ツールバーの 、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。

図 A—43 ローカル変数 パネル



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー (ローカル変数 パネル専用部分)]
- [[編集] メニュー (ローカル変数 パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー → [ローカル変数] を選択

[各エリアの説明]

(1) スコープ・エリア

表示するローカル変数のスコープをドロップダウン・リストにより選択します。

選択できる項目は次のとおりです。

項目	動作
カレント	カレント PC 値のスコープでのローカル変数を表示します。
<深さ> <関数名 ([ファイル名 # 行番号]> ^注	呼び出し元の関数のスコープでのローカル変数を表示します。 プログラム実行後、選択したスコープが存在するかぎり、ここで選択したスコープを保ちます。

注 コール・スタック パネルで表示している関数呼び出し元を表示します。




(2) [名前] エリア

ローカル変数名、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。

	変数を表示します。 Auto 変数、内部スタティック変数、Register 変数の表示も行います ^注 。 配列、ポインタ型変数、構造体/共用体は、階層構造をツリー形式で表示します。 先頭に“+”マークがある場合は、これをクリックすることにより次を展開表示します（展開後“-”マークに変化）。	
	配列	配列中の全要素
	ポインタ型変数	ポインタが指し示す先の変数 なお、ポインタが指し示す先がポインタの場合は、さらに“+”マークを付与し、これをクリックすることにより参照先を表示します。 ただし、ポインタの指す値が不明な場合は、“?”を表示します。
	構造体/共用体	構造体/共用体の全メンバ
	引数を表示します。	
	関数を表示します。	

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 [メモリへジャンプ] を選択することにより、選択しているローカル変数が配置されているアドレスにcaret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 (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します)。

表示例 (デフォルト)			説明
0x0	文字色	青色	ユーザにより、値が変更されているローカル変数値 ([Enter] キーによりターゲット・メモリに書き込まれます)
	背景色	標準色	
0x0	文字色	茶色	プログラムの実行により、値が変化したローカル変数値 ^注 プログラムを再び実行することにより、強調色がリセットされます。
	背景色	クリーム	
?	文字色	グレー	ローカル変数の値を取得できない場合
	背景色	標準色	

注 プログラムの実行開始位置からブレークした位置で同じ変数名を表示していて、かつ、その変数値が変化している場合が対象となります。

このエリアは、次の機能を備えています。

(a) ローカル変数値/引数値の変更

ローカル変数値、および引数値の変更は、対象ローカル変数値を選択したのち、再度クリックし、キーボードからの直接入力により行います ([Esc] キーの押下で編集モードをキャンセルします)。

ローカル変数値、および引数値を編集したのち、[Enter] キーの押下、または編集領域以外へのフォーカスの移動により、デバッグ・ツールのターゲット・メモリに書き込まれます。

ローカル変数値/引数値の変更方法についての詳細は、「(2) 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られている場合は、レジスタ名を表示します。

このエリアを編集することはできません。

[ツールバー]

ツールバーの各ボタンは、プログラム実行中は無効となります。

	デバッグ・ツールから最新の情報を取得し、表示を更新します。
表記	値の表示形式を変更する次のボタンを表示します。
	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。
	このパネル上の値を 16 進数で表示します。
	このパネル上の値を 10 進数で表示します。
	このパネル上の値を 8 進数で表示します。
	このパネル上の値を 2 進数で表示します。
	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)。
	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。
	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。
	このパネル上の値を Double で表示します。 ただし、8 バイト・データ以外、または型情報を持つ場合は、規定値で表示します。
	値表示の末尾に、その値の 16 進数表記を “()” で囲んで併記します。
エンコード	文字列変数のエンコードを変更する次のボタンを表示します。
	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。
	文字列変数を EUC-JP コードで表示します。
	文字列変数を UTF-8 コードで表示します。
	文字列変数を UTF-16 コードで表示します。

[[ファイル] メニュー (ローカル変数 パネル専用部分)]

ローカル変数 パネル専用の [ファイル] メニューは次のとおりです (その他の項目は共通)。

ただし、プログラム実行中はすべて無効となります。

ローカル変数データを保存	このパネルの内容を前回保存したテキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存します (「(b) ローカル変数値の保存」参照)。 なお、起動後に初めてこの項目を選択した場合は、[名前を付けてローカル変数データを保存 ...] の選択と同等の動作となります。
名前を付けてローカル変数データを保存 ...	このパネルの内容を指定したテキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存するために、名前を付けて保存 ダイアログをオープンします (「(b) ローカル変数値の保存」参照)。

[[編集] メニュー (ローカル変数 パネル専用部分)]

ローカル変数 パネル専用の [編集] メニューは次のとおりです (その他の項目はすべて無効)。

コピー	選択している行の内容、または文字列をクリップ・ボードにコピーします。
すべて選択	項目をすべて選択状態にします。
名前の変更	選択しているローカル変数の値を変更するために、編集モードに移行します (「(2) ローカル変数の内容を変更する」参照)。 ただし、プログラム実行中は無効となります。
検索 ...	検索・置換 ダイアログを [一括検索] タブが選択状態でオープンします。
置換 ...	検索・置換 ダイアログを [一括置換] タブが選択状態でオープンします。

[コンテキスト・メニュー]

コンテキスト・メニューの各項目は、プログラム実行中は無効となります。

ウォッチ 1 に登録	選択しているローカル変数を ウォッチ パネル (ウォッチ 1) に登録します。
コピー	選択している行の内容、または文字列をクリップ・ボードにコピーします。
表記	表示形式を指定するために、次のカスケード・メニューを表示します。
自動	このパネル上の値の表記を変数ごとの規定値で表示します (デフォルト)。
16 進数	このパネル上の値を 16 進数で表示します。
10 進数	このパネル上の値を 10 進数で表示します。
8 進数	このパネル上の値を 8 進数で表示します。
2 進数	このパネル上の値を 2 進数で表示します。
配列のインデックスを 10 進表記	このパネル上の配列のインデックスを 10 進数で表示します (デフォルト)。
配列のインデックスを 16 進表記	このパネル上の配列のインデックスを 16 進数で表示します。
Float	このパネル上の値を Float で表示します。 ただし、4 バイト・データ以外、または型情報を持つ場合は、規定値で表示します。
Double	このパネル上の値を Double で表示します。 ただし、8 バイト・データ以外、または型情報を持つ場合は、規定値で表示します。
16 進数値を併記	値表示の末尾に、その値の 16 進数表記を “()” で囲んで併記します。
エンコード	文字コードを指定するため、次のカスケード・メニューを表示します。
ASCII	文字列変数を ASCII コードで表示します。
Shift_JIS	文字列変数を Shift_JIS コードで表示します (デフォルト)。
EUC-JP	文字列変数を EUC-JP コードで表示します。
UTF-8	文字列変数を UTF-8 コードで表示します。
UTF-16	文字列変数を UTF-16 コードで表示します。
メモリヘジャンプ	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（「2.9.6 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には保持されます（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。

- 備考 1. ツールバーの 、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。

図 A—44 ウォッチ パネル



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー (ウォッチ パネル専用部分)]
- [[編集] メニュー (ウォッチ パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー → [ウォッチ] → [ウォッチ 1～4] を選択

[各エリアの説明]









(1) [ウォッチ式] エリア

登録しているウォッチ式を一覧で表示します。

このエリアの表タイトル部をクリックすることにより、一覧内のウォッチ式をアルファベット順でソートすることができます。

また、カテゴリ（フォルダ）を自由に作成してウォッチ式を分類し、ツリー形式で表示することができます（「(a) ツリーの編集」参照）。


表示される各アイコンの意味は次のとおりです。


	このカテゴリに属するウォッチ式を表示している状態を示します。アイコンをダブルクリック、または“-”マークをクリックすると、カテゴリを閉じウォッチ式を非表示にします。
	このカテゴリに属するウォッチ式が非表示の状態を示します。アイコンをダブルクリック、または“+”マークをクリックすると、カテゴリを開きウォッチ式を表示します。
	ウォッチ式が変数であることを示します。 配列、ポインタ型変数、構造体／共用体を示すウォッチ式の先頭には、“+” / “-” マークを表示し、これをクリックすることにより展開／折りたたみ表示します。
	ウォッチ式が関数であることを示します。
	ウォッチ式が即値であることを示します。
	ウォッチ式が式であることを示します。
	ウォッチ式が I/O レジスタであることを示します。
	ウォッチ式が CPU レジスタであることを示します。 下階層のレジスタ（レジスタの部分）を持つウォッチ式の先頭には、“+” / “-” マークを表示し、これをクリックすることにより展開／折りたたみ表示します。

このエリアは、次の機能を備えています。

(a) ツリーの編集

ウォッチ式をカテゴリ（フォルダ）で分類し、ツリー形式で表示することができます。

カテゴリを新規に作成する場合は、作成したい位置にカーレットを移動したのち、ツールバーの  ボタンのクリック、またはコンテキスト・メニューの [カテゴリを作成] を選択し、任意にカテゴリ名称を入力することにより行います。

なお、カテゴリを削除する場合は、削除したいカテゴリを選択したのち、ツールバーの  ボタンのクリック、またはコンテキスト・メニューの [削除] を選択します。

また、作成したカテゴリ名を編集する場合は、編集したいカテゴリ名を選択したのち、次のいずれかの操作により行います。

- 再度クリック後、キーボードよりカテゴリ名を直接編集

- [編集] メニュー→ [名前の変更] を選択後、キーボードよりカテゴリ名を直接編集
- [F2] キーを押下後、キーボードよりカテゴリ名を直接編集

カテゴリを作成したのち、登録済みのウォッチ式をカテゴリ内に直接ドラッグ・アンド・ドロップすることにより、各ウォッチ式をカテゴリで分類したツリー形式で表示します。

同様に、カテゴリとウォッチ式の表示の順番（上下位置）も、ドラッグ・アンド・ドロップ操作により自由に変更することができます。

注意 1. カテゴリ内にカテゴリを作成することはできません。

2. 1つのウォッチパネルにおいて、カテゴリは64個まで作成することができます（上限値を越えて作成しようとした場合、メッセージを表示します）。

備考 ウォッチ式／カテゴリを他のウォッチパネル（ウォッチ1～ウォッチ4）にドラッグ・アンド・ドロップすると、ドロップ先のウォッチパネルにウォッチ式／カテゴリがコピーされます。

(b) 展開／折りたたみ表示

配列、ポインタ型変数、構造体／共用体、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。

- 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—12 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）
I/O 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  ボタンをクリック、またはコンテキスト・メニューの [新規ウォッチ式を追加] を選択することにより、このエリアの最下段に新規ウォッチ式用のエントリ・ボックスが表示されます。

エントリ・ボックスの [ウォッチ式] エリアにおいて、キーボードより直接ウォッチ式を入力したのち、[Enter] キーを押下します。

この際のウォッチ式の入力形式は次のとおりです。

表 A—13 ウォッチ式の入力形式

ウォッチ式	表示する値
C 言語変数名	C 言語の変数の値
ウォッチ式 [ウォッチ式]	配列の要素値
ウォッチ式.メンバ名	構造体 / 共用体のメンバ値
ウォッチ式->メンバ名	ポインタの指し示す構造体 / 共用体のメンバ値
* ウォッチ式	ポインタの変数の値
CPU レジスタ名	CPU レジスタの値
I/O レジスタ名	I/O レジスタの値
ラベル、EQU シンボル、即値アドレス	ラベルの値、EQU シンボルの値、即値アドレスの値
ビット・シンボル	ビット・シンボルの値

また、ウォッチ式は、スコープを指定して登録することができます。スコープ指定してウォッチ式を登録した場合の扱いは次のとおりです。

表 A—14 C 言語変数をスコープ指定してウォッチ登録した場合の扱い

スコープ指定	ロード・モジュール名	ソース・ファイル名	関数名	変数名
prog\$file#func#var	prog	file	func	var
prog\$file#var	prog	file	グローバル	var
prog\$var	prog	グローバル	グローバル	var
file#func#var	カレント	file	func	var
file#var	カレント	file	グローバル	var
var	カレント	カレント	カレント	var

表 A—15 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—16 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
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caret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。
4. 同名の 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

また、ウォッチ式の先頭に "\$" を指定した場合は、次の順にシンボルを解決し、値を表示します。

CPU レジスタ > I/O レジスタ > C 言語変数

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. 【V850E2】

次の指定により、同名のシステム・レジスタを区別することができます。

スコープ指定	システム・レジスタ・バンク	システム・レジスタ名
PSW:SRBff00	CPU 機能グループ (0xff)・ユーザ 0 バンク (0x00)	PSW
PSW:SRB0000	ユーザ・グループ (0x00)・基本バンク (0x00)	PSW

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 128 個まで登録することができます（上限値を越えて登録しようとした場合、メッセージを表示します）。

2. コンパイラによる最適化のため、対象となる変数を使用していないブロックでは変数データがスタック /レジスタに存在しない場合があります。この場合、対象となる変数をウォッチ式として登録しても値の表示は“?”のままとなります。

備考 1. 各ウォッチ パネル（ウォッチ 1～ウォッチ 4）上で登録したウォッチ式は、それぞれ個別に管理され、プロジェクトのユーザ情報として保存されます。


2. ウォッチ式は、同名を複数登録することができます。

(d) ウォッチ式の編集

登録済みのウォッチ式の編集は、対象ウォッチ式をダブルクリックし、対象ウォッチ式を編集モードにしたのち、キーボードから編集内容を直接入力して行います（[Esc] キーの押下で編集モードをキャンセルします）。

ウォッチ式を編集したのち、[Enter] キーを押下すると編集を完了します。

(e) ウォッチ式の削除

ツールバーの  ボタンのクリック、またはコンテキスト・メニューの [削除] を選択することにより、選択しているウォッチ式を削除します。

(f) 各種イベントの設定

コンテキスト・メニューの [アクセス・ブレークの設定] / [トレース出力] を選択することにより、選択しているウォッチ式に各種イベントを設定することができます。

アクセス系のブレーク・イベントが設定された場合、ウォッチ式のアイコンが変化します（ウォッチ式のアイコンの下にブレーク・イベントのイベント・マークを重ねて表示）。トレース・イベントの場合は、ウォッチ式のマークに変化はありません。

イベントを設定することにより、設定したイベントの詳細情報が **イベント パネル** に反映されます。

ただし、イベントの設定は、対象となるウォッチ式がグローバル変数 /関数内スタティック変数 /ファイル内スタティック変数 /I/O レジスタの場合にのみ行うことができます。

イベントの設定方法についての詳細は次を参照してください。

- 「2. 8. 4 変数 /I/O レジスタへのアクセスで停止する」
- 「2. 11. 4 条件を満たしたときのみの実行履歴を収集する」

(g) メモリ定義アドレスへのジャンプ

コンテキスト・メニューの [メモリへジャンプ] を選択することにより、選択しているウォッチ式が定義されているアドレスにカー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。

表 A—17 ウォッチ式の表示形式（デフォルト）

ウォッチ式の型	表示形式
char, signed char, unsigned char	ASCII 文字に続き “()” 内に 16 進数値を併記
short, signed short, short int, signed short int, int, signed, signed int, long, signed long, long int, signed long int	符号付き 10 進数値に続き “()” 内に 16 進数値を併記
unsigned short, unsigned short int, unsigned, unsigned int, unsigned long, unsigned long int	符号なし 10 進数値に続き “()” 内に 16 進数値を併記
float	Float（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

また、ウォッチ式の値として表示されるマークや色の意味は次のとおりです（表示の際の文字色／背景色はオプションダイアログにおける [全般 - フォントと色] カテゴリの設定に依存します）。

表示例（デフォルト）			説明
0x0	文字色	青色	ユーザにより、値が変更されているウォッチ式の値（[Enter] キーによりターゲット・メモリに書き込まれます）
	背景色	標準色	
0x0	文字色	ピンク	リアルタイム表示更新機能を行っているウォッチ式の値
	背景色	標準色	
0x0	文字色	茶色	プログラムの実行により、値が変化したウォッチ式の値 ツールバーの  ボタン、またはコンテキスト・メニューの [表示色をリセット] を選択することにより、強調表示をリセットします。
	背景色	クリーム	
?	文字色	グレー	存在しない変数をウォッチ式として登録した場合、またはウォッチ式の値を取得できなかった場合（変数がスコープを外れた場合など）
	背景色	標準色	

- 備考 1.** 読み込み動作によってマイクロコントローラが動作してしまう I/O レジスタは、読み込み保護対象となり、値の読み込みは行いません。読み込み保護対象の I/O レジスタ の内容を読み込みたい場合には、コンテキスト・メニューの [値を強制読み込み] を選択してください。
- 2.** 各ウォッチ式は、登録された順序で値の取得を行います。
このため、同一の I/O レジスタを複数登録した場合、値を取得するタイミングに差が生じるため、表示される値が異なる場合があります。
- 3.** 16 進数値を併記している場合は、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} >)	
単独の I/O レジスタ	< I/O レジスタの種類 > (< アクセス属性 > < アクセス・タイプ > < サイズ ^{注1} >)	
	アクセス属性	R : 読み出しのみ可能 W : 書き込みのみ可能 R/W : 読み出し/書き込み可能
	アクセス・タイプ	1 : 1 ビット・アクセス可能 8 : バイト・アクセス可能 16 : ハーフ・ワード・アクセス可能 32 : ワード・アクセス可能
判別不能	?	
上記以外	< C コンパイラの判定に従ったウォッチ式の型 ^{注2} > (< サイズ ^{注1} >)	

注 1. ウォッチ式のサイズをバイト単位で示します。

ただし、ビット IOR/C 言語ビット・フィールドについては、ビット単位で表示し、数値の末尾に“ビット”表記を付与します。

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]

	登録しているウォッチ式のすべての値を再取得し、表示を更新します。 ただし、読み込み保護対象の I/O レジスタの再読み込みは行いません。
	選択しているウォッチ式に対して、プログラムの実行により値が変化したことを示す強調表示をリセットします。 ただし、プログラム実行中は無効となります。
	新規ウォッチ式を登録します。テキスト・ボックスに直接ウォッチ式を入力します（「(c) 新規ウォッチ式の登録」参照）。 なお、1つのウォッチパネルに登録可能なウォッチ式数は、最大 128 個までです。
	新規カテゴリ（フォルダ）を追加します。テキスト・ボックスに直接カテゴリ名を入力します。 なお、1つのウォッチパネルに作成可能なカテゴリ数は、最大 64 個までです（カテゴリ内のカテゴリ作成は不可）。
	選択している範囲の文字列を削除します。 ウォッチ式／カテゴリが選択状態の場合は、その項目を削除します。 ただし、ウォッチ式の展開項目を選択している場合は無効となります。
表記	値の表示形式を変更する次のボタンを表示します。
	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—17 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）。
	選択している項目の値を 16 進数で表示します。
	選択している項目の値を符号付き 10 進数で表示します。
	選択している項目の値を符号なし 10 進数で表示します。
	選択している項目の値を 8 進数で表示します。
	選択している項目の値を 2 進数で表示します。
	選択している項目の値を ASCII コードで表示します。
	選択している項目の値を 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。
	選択している項目の値を Double で表示します。 ただし、選択しているウォッチ式が 8 バイト・データの場合のみ有効となります。
	選択している項目の値表示の末尾に、その値の 16 進数表記を “()” で囲んで併記します。 ただし、16 進数表記をしている場合は併記しません。

[[ファイル] メニュー (ウォッチ パネル専用部分)]

ウォッチ パネル専用の [ファイル] メニューは次のとおりです (その他の項目は共通)。

ただし、プログラム実行中はすべて無効となります。

ウォッチ・データを保存	このパネルの内容を前回保存したテキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存します (「(c) ウォッチ式の値の保存」参照)。 なお、起動後に初めてこの項目を選択した場合は、[名前を付けてウォッチ・データを保存 ...] の選択と同等の動作となります。
名前を付けてウォッチ・データを保存 ...	このパネルの内容を指定したテキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存するために、名前を付けて保存 ダイアログをオープンします (「(c) ウォッチ式の値の保存」参照)。

[[編集] メニュー (ウォッチ パネル専用部分)]

ウォッチ パネル専用の [編集] メニューは次のとおりです (その他の項目はすべて無効)。

切り取り	選択範囲の文字列を切り取り、クリップ・ボードにコピーします。 ウォッチ式/カテゴリが選択状態の場合は、その項目を切り取ります。 ただし、ウォッチ式の展開項目を選択している場合は無効となります。
コピー	選択している範囲を文字列としてクリップ・ボードにコピーします。 ウォッチ式/カテゴリが選択状態の場合は、その項目をコピーします。 ただし、ウォッチ式の展開項目を選択している場合は無効となります。
貼り付け	テキストが編集状態の場合、クリップ・ボードの内容をカー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、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カーレット位置に登録します。
削除	選択している範囲の文字列を削除します。 ウォッチ式/カテゴリが選択状態の場合は、その項目を削除します。 ただし、ウォッチ式の展開項目を選択している場合は無効となります。
すべて選択	テキストが編集状態の場合、すべての文字列を選択します。 テキストが編集状態以外の場合、すべてのウォッチ式/カテゴリを選択状態にします。
名前の変更	選択しているウォッチ式、またはカテゴリの名称を編集します。
検索 ...	検索・置換 ダイアログを [一括検索] タブが選択状態でオープンします。
置換 ...	検索・置換 ダイアログを [一括置換] タブが選択状態でオープンします。

[コンテキスト・メニュー]

アクセス・ブレークの設定	この項目は、選択しているウォッチ式がグローバル変数／関数内スタティック変数／ファイル内スタティック変数、および I/O レジスタの場合のみ有効です（複数選択不可）。 アクセス系のブレーク・イベントを設定するために、次のカスケード・メニューを表示します（「(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。
読み込み組み合わせブレークを設定	今版ではサポートしていません。
書き込み組み合わせブレークを設定	今版ではサポートしていません。
読み書き組み合わせブレークを設定	今版ではサポートしていません。
トレース出力 【IECUBE】【IECUBE2】 【シミュレータ】	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（読み込み時）	選択しているウォッチ式にリード・アクセスした際に、その値をトレース・メモリに記録するポイント・トレース・イベントを設定します。
値をトレースに記録（書き込み時）	選択しているウォッチ式にライト・アクセスした際に、その値をトレース・メモリに記録するポイント・トレース・イベントを設定します。
値をトレースに記録（読み書き時）	選択しているウォッチ式にリード／ライト・アクセスした際に、その値をトレース・メモリに記録するポイント・トレース・イベントを設定します。
読み書き時にトレース開始の設定	今版ではサポートしていません。
読み書き時にトレース終了の設定	今版ではサポートしていません。
トレース	トレースパネル【IECUBE】【IECUBE2】【シミュレータ】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 128 個までです。
カテゴリを作成	新規カテゴリ (フォルダ) を追加します。テキスト・ボックスに直接カテゴリ名を入力します。 なお、1つのウォッチ パネルに作成可能なカテゴリ数は、最大 64 個までです (カテゴリ内のカテゴリ作成は不可)。
削除	選択している範囲の文字列を削除します。 ウォッチ式/カテゴリが選択状態の場合は、その項目を削除します。 ただし、ウォッチ式の展開項目を選択している場合は無効となります。
切り取り	選択している範囲の文字列を切り取ってクリップ・ボードに移動します。 ウォッチ式/カテゴリが選択状態の場合は、その項目を切り取ります。 ただし、ウォッチ式の展開項目を選択している場合は無効となります。
コピー	選択している範囲の文字列をクリップ・ボードにコピーします。 ウォッチ式/カテゴリが選択状態の場合は、その項目をコピーします。
貼り付け	テキストが編集状態の場合、クリップ・ボードの内容をキャレット位置に挿入します。 テキストが編集状態以外の場合で、ウォッチ式がクリップ・ボードにコピーされている場合は、コピーされているウォッチ式をキャレット位置に登録します。 ただし、ウォッチ式の展開項目を選択している場合は無効となります。
名前の変更	選択しているウォッチ式、またはカテゴリの名称を編集します。
ウォッチ式をインポート ...	ウォッチ式をインポートするために、ウォッチ式データ・ファイルを開くダイアログをオープンします (「(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—17 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—17 ウォッチ式の表示形式 (デフォルト)」参照) で表示します。
Double	選択している項目を Double で表示します。 ただし、選択しているウォッチ式が 8 バイト・データ以外、または型情報を持つ場合は、規定値 (「表 A—17 ウォッチ式の表示形式 (デフォルト)」参照) で表示します。
配列のインデックスを 10 進表記	すべての配列のインデックスを 10 進数で表示します。
配列のインデックスを 16 進表記	すべての配列のインデックスを 16 進数で表示します。

エンコード	文字コードを指定するため、次のカスケード・メニューを表示します。
ASCII	選択している項目を ASCII コードで表示します。
Shift_JIS	選択している項目を Shift_JIS コードで表示します (デフォルト)。
EUC-JP	選択している項目を EUC-JP コードで表示します。
UTF-8	選択している項目を UTF-8 コードで表示します。
UTF-16	選択している項目を UTF-16 コードで表示します。
サイズ表記	サイズを指定するため、次のカスケード・メニューを表示します。
1 バイト	選択している項目を 8 ビット・データとして表示します。
2 バイト	選択している項目を 16 ビット・データとして表示します。
4 バイト	選択している項目を 32 ビット・データとして表示します。
8 バイト	選択している項目を 64 ビット・データとして表示します。
メモリへジャンプ	選択しているウォッチ式が定義されているアドレスへキャレットを移動した状態で メモリパネル (メモリ 1) をオープンします (「(g) メモリ定義アドレスへのジャンプ 」参照)。
表示色をリセット	選択しているウォッチ式に対して、プログラムの実行により値が変化したことを示す強調表示をリセットします。 ただし、プログラム実行中は無効となります。

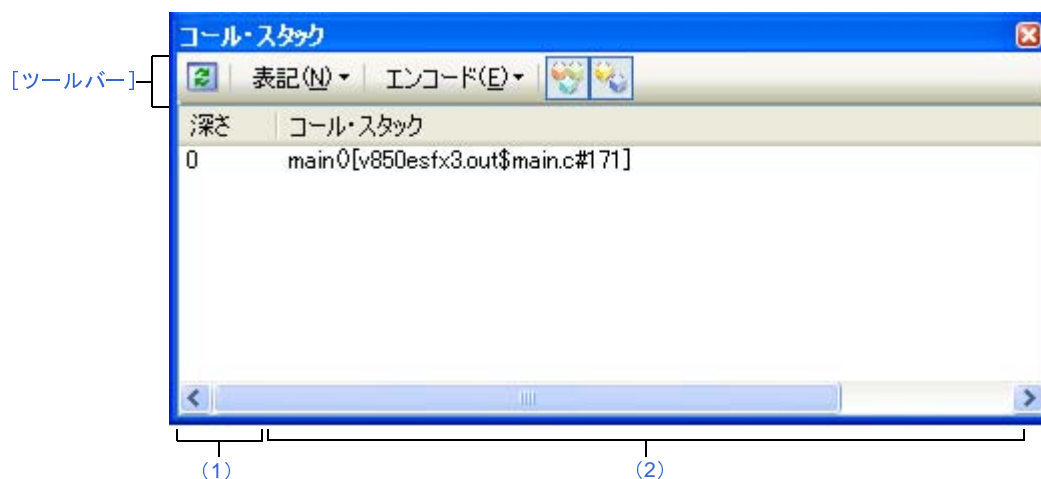
コール・スタック パネル

関数呼び出しのコール・スタック情報の表示を行います（「2.10.1 コール・スタック情報を表示する」参照）。
なお、このパネルは、デバッグ・ツールと接続時のみオープンすることができます。

注意 プログラム実行中は、このパネルには何も表示されません。
プログラムの実行が停止したタイミングで、各エリアの表示を行います。

備考 ツールバーの 、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—45 コール・スタック パネル



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー（コール・スタック パネル専用部分）]
- [[編集] メニュー（コール・スタック パネル専用部分）]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー → [コール・スタック] を選択

[各エリアの説明]



(1) [深さ] エリア

呼び出しの深さを表示します。

カレント PC 位置を表示している行を 0 とし、呼び出し元に 1 から順に番号を付与します。

(2) [コール・スタック] エリア

現在のソース位置とスタックに積まれているコール・スタック情報（関数呼び出し元位置／各関数の引数など）を表示します。

ツールバーの  /  ボタン、またはコンテキスト・メニューの [引数表示] / [モジュール・ファイル名表示] の選択による状態により、このエリアに表示する表示形式は次のように異なります。

状態	表示形式
- 引数を表示する - モジュール・ファイル名を表示する	<関数> (<引数> = <引数値 ^注 >, ...)[<モジュール・ファイル名> \$ <ファイル名> # <行番号>] (デフォルト)
- 引数を表示する - モジュール・ファイル名を表示しない	<関数> (<引数> = <引数値 ^注 >, ...)[<ファイル名> # <行番号>]
- 引数を表示しない - モジュール・ファイル名を表示する	<関数> ([<モジュール・ファイル名> \$ <ファイル名> # <行番号>])
- 引数を表示しない - モジュール・ファイル名を表示しない	<関数> ([<ファイル名> # <行番号>])

注 引数値が文字列の場合、最大 20 文字まで表示します。

備考 配列の引数は、配列としてではなくポインタとして渡されます（C 言語仕様）。そのため、引数が配列の場合、ポインタ扱いとして表示します。

このエリアは、次の機能を備えています。

(a) ソース行／逆アセンブル行へのジャンプ

コンテキスト・メニューの [ソースヘジャンプ] 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、現在のキャレット位置の関数呼び出し元のアドレスを示す逆アセンブル行にキャレットを移動した状態で **逆アセンブル パネル**（逆アセンブル 1）がオープンします（すでにオープンしている場合は、逆アセンブル パネル（逆アセンブル 1）にジャンプ）。

備考 行をダブルクリックすることでも、対象ソース行ヘ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]

ツールバーの各ボタンは、プログラム実行中は無効となります。

	デバッグ・ツールから最新の情報を取得し、表示を更新します。
表記	値の表示形式を変更する次のボタンを表示します。
	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)。
	このパネル上の値を 16 進数で表示します。
	このパネル上の値を 10 進数で表示します。
	このパネル上の値を 8 進数で表示します。
	このパネル上の値を 2 進数で表示します。
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。
	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。
	このパネル上の文字列変数を EUC-JP コードで表示します。
	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。
	モジュール・ファイル名を付加して表示します (デフォルト)。
	関数呼び出しのパラメータ (引数) を付加して表示します (デフォルト)。

[[ファイル] メニュー (コール・スタック パネル専用部分)]

コール・スタック パネル専用の [ファイル] メニューは次のとおりです (その他の項目は共通)。

ただし、プログラム実行中はすべて無効となります。

コール・スタック・データを保存	このパネルの内容を前回保存したテキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存します (「(b) コール・スタック情報の保存」参照)。 なお、起動後に初めてこの項目を選択した場合は、[名前を付けてコール・スタック・データを保存 ...] の選択と同等の動作となります。
名前を付けてコール・スタック・データを保存 ...	このパネルの内容を指定したテキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存するために、名前を付けて保存 ダイアログをオープンします (「(b) コール・スタック情報の保存」参照)。

[[編集] メニュー (コール・スタック パネル専用部分)]

コール・スタック パネル専用の [編集] メニューは次のとおりです (その他の項目はすべて無効)。

コピー	選択している行の内容を文字列としてクリップ・ボードにコピーします。
すべて選択	項目をすべて選択状態にします。
検索 ...	検索・置換 ダイアログを [一括検索] タブが選択状態でオープンします。
置換 ...	検索・置換 ダイアログを [一括置換] タブが選択状態でオープンします。

[コンテキスト・メニュー]

コンテキスト・メニューの各項目は、プログラム実行中は無効となります。

コピー	選択している行の内容を文字列としてクリップ・ボードにコピーします。
モジュール・ファイル名表示	モジュール・ファイル名を付加して表示します (デフォルト)。
引数表示	関数呼び出しのパラメータ (引数) を付加して表示します (デフォルト)。
表記	表示形式を指定するために、次のカスケード・メニューを表示します。
自動	このパネル上の値の表記を変数ごとの規定値で表示します (デフォルト)。
16 進数	このパネル上の値を 16 進数で表示します。
10 進数	このパネル上の値を 10 進数で表示します。
8 進数	このパネル上の値を 8 進数で表示します。
2 進数	このパネル上の値を 2 進数で表示します。
エンコード	文字コードを指定するため、次のカスケード・メニューを表示します。
ASCII	文字列変数を ASCII コードで表示します (デフォルト)。
Shift_JIS	文字列変数を Shift_JIS コードで表示します。
EUC-JP	文字列変数を EUC-JP コードで表示します。
UTF-8	文字列変数を UTF-8 コードで表示します。
UTF-16	文字列変数を UTF-16 コードで表示します。
逆アセンブルヘジャンプ	選択している行が示す関数呼び出し元のアドレスにカー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カー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【IECUBE】【IECUBE2】【シミュレータ】

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（「2.11 実行履歴の収集【IECUBE】【IECUBE2】【シミュレータ】」参照）。

トレース・データは、デフォルトで逆アセンブル・テキストとソース・テキストを混合して表示しますが、表示モードを選択することにより、そのどちらか一方のみを表示させることもできます。

プログラムの実行停止後、最新のトレース・データが表示されるよう表示位置を自動更新します。

なお、このパネルは、デバッグ・ツールと接続時のみオープンすることができます。

注意 1. 【IECUBE】

トレース機能／リアルタイム表示更新機能（RRM 機能）／カバレッジ機能は一部排他使用の機能です。

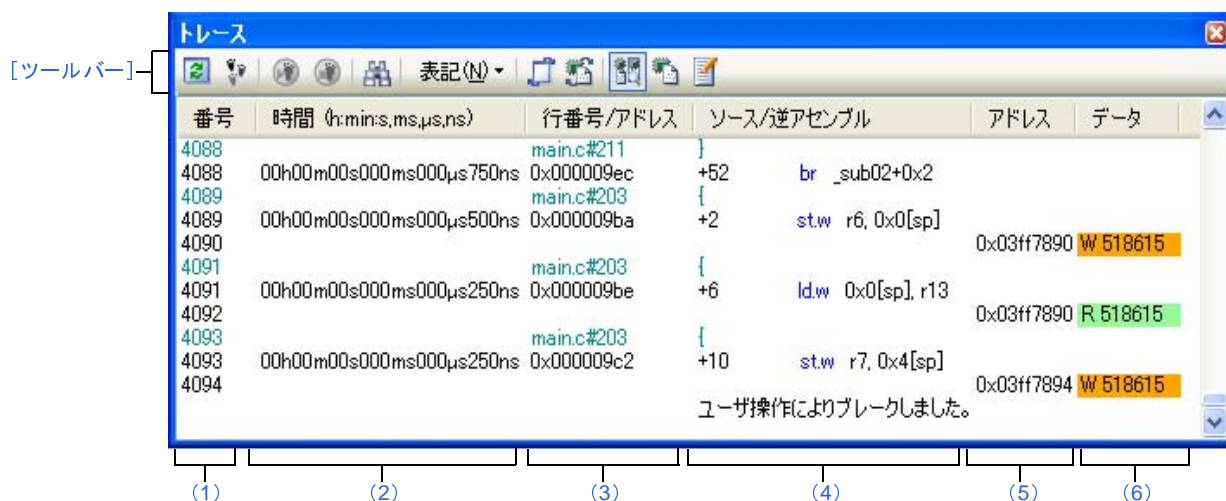
2. 【IECUBE2】

トレース機能／タイマ機能（Run-Break タイマを除く）／カバレッジ機能は一部排他使用の機能です。

備考 1. ツールバーの , 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。

図 A—46 トレース 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 (トレース パネル専用部分)]
- [[編集] メニュー (トレース パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー → [トレース] を選択
- [エディタ パネル](#) / [逆アセンブル パネル](#) において、コンテキスト・メニューの [トレース設定] → [トレース結果の表示] を選択

[各エリアの説明]

(1) [番号] エリア

トレース・フレームに対応するトレース番号を表示します。

(2) [時間 (h:min:s,ms,μs,ns)] エリア

プログラムの実行開始から、各フレームの命令実行、またはメモリ・アクセスの要因が発生するまでに要した時間を“時間、分、秒、ミリ秒、マイクロ秒、ナノ秒”の単位で表示します。

なお、オーバフローした場合、このエリアは無効色（グレー）で表示されます。

備考 1. 【IECUBE】【IECUBE2】

時間の精度は、[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・タグの分周率] プロパティの設定に依存します。

2. 【シミュレータ】

時間表示を積算値とするか差分値とするかは、[プロパティ パネル](#)の [\[デバッグ・ツール設定\]](#) タブ上の [トレース] カテゴリ内 [トレース・タイム・タグを積算] プロパティの設定に依存します。

(3) [行番号/アドレス] エリア

アセンブル命令のアドレス、またはソース・ファイルの行番号を表示します。

表示進数や文字列のエンコードは、[ツールバー](#)のボタン、またはコンテキスト・メニューより選択することができます。

表示形式は次のとおりです。

表示行の種類	表示形式
命令（逆アセンブル）	<アドレス>
ソース・テキスト	<ファイル名> # <行番号>
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。

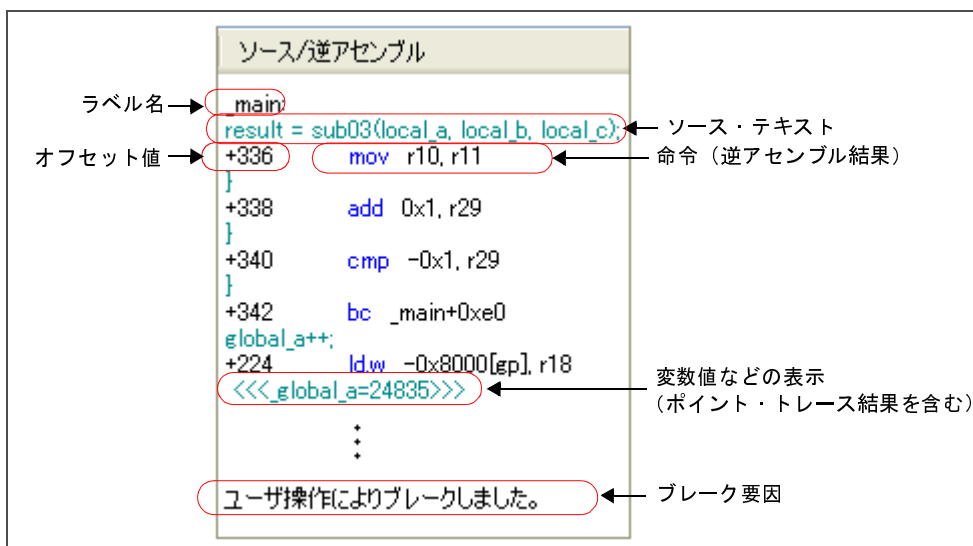
- CPU レジスタ・アクセス
- オペランド・アクセス
- 無効フェッチ

(4) [ソース/逆アセンブル] エリア

収集したトレース・データを次のように表示します。

なお、表示モードの選択により、このエリアに表示される項目は異なります（「(a) 表示モード」参照）。

図 A—47 [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 レジスタ名 = I/O レジスタ値 >>> 例：a=b; <<<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 【シミュレータ】	プログラムがブレークした要因を表示します。

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。

2. 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します）。

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／ブレイ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／ブレイ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）／ブレイ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 [逆アセンブルへジャンプ] を選択することにより、現在のキャレット位置の行のフェッチ・アドレスにキャレットを移動した状態で **逆アセンブル パネル**（逆アセンブル 1）がオープンします（すでにオープンしている場合は、逆アセンブル パネル（逆アセンブル 1）にジャンプ）。

(c) 他のパネルとの連動

ツールバーの  /  ボタン、またはコンテキスト・メニューの [ウィンドウ連動] → [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、「[2.11.9 実行履歴の表示内容を保存する](#)」を参照してください。

(5) [アドレス] エリア

メモリ・アクセスの対象アドレスを表示します。

ただし、I/O レジスタへのアクセスの場合は、アドレスの代わりに I/O レジスタ名を表示します（アクセスが複数ある場合は次の行に表示）。

表示進数は、ツールバーのボタン、またはコンテキスト・メニューより選択することができます。

備考 【IECUBE2】

DMA によるメモリ・アクセスの対象アドレスを表示する場合は、[プロパティパネルの \[デバッグ・ツール設定\] タブ](#)上の [トレース] カテゴリ内 [DMA トレースを行う] プロパティにおいて [はい] を選択してください。

(6) [データ] エリア

アクセスしたデータ値、およびその際のアクセス種別を表示します。

ただし、CPU レジスタ・アクセスは表示しません。

表示進数や文字列のエンコードは、ツールバーのボタン、またはコンテキスト・メニューより選択することができます。

データ値、およびアクセス種別の表示形式は次のとおりです（表示の際の文字色／背景色は[オプションダイアログ](#)における [\[全般 - フォントと色\] カテゴリ](#)の設定に依存します）。

表示例（デフォルト）			メモリ・アクセス種別
R データ値	文字色	標準色	リード・アクセス
	背景色	薄緑	
W データ値	文字色	標準色	ライト・アクセス
	背景色	オレンジ	
RW データ値	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	

備考 1. 【IECUBE】

DMA により、I/O レジスタ／メモリへのアクセスが発生した場合、アクセスが開始／終了したフレームのデータ値の直後に“DMA”を表示します。

2. 【IECUBE2】

DMA によりアクセスしたデータ値を表示する場合は、[プロパティパネルの \[デバッグ・ツール設](#)

定] タブ上の [トレース] カテゴリ内 [DMA トレースを行う] プロパティにおいて [はい] を選択してください。この場合、同カテゴリ内 [DMA トレース・データの選択] プロパティで選択した DMA トレース・データを次の形式で表示します。

<アクセス種別><データ値>< DMA Channel number= チャンネル番号>< Size= データ・サイズ>
< Transfer Count= 転送回数>

[ツールバー]

	デバッグ・ツールから最新の情報を取得し、表示を更新します。 ただし、トレーサ動作中は無効となります。
	トレース・メモリをクリア（初期化）し、このパネルの表示もクリアします。 ただし、トレーサ動作中は無効となります。
	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。
	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。
	トレース検索ダイアログ [IECUBE] [IECUBE2] [シミュレータ] 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。
	このパネル上の値を 16 進数で表示します（デフォルト）。
	このパネル上の値を 10 進数で表示します。
	このパネル上の値を 8 進数で表示します。
	このパネル上の値を 2 進数で表示します。
	選択している行に連動してエディタパネルをスクロールします。
	選択している行に連動して逆アセンブルパネルをスクロールします。
	表示モードを混合表示モードにします（デフォルト）。 ただし、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、トレーサ動作中は無効となります。
	表示モードをソース表示モードにします。 ただし、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・データを保存 ...	トレース・データの内容を指定したテキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存するために、 データ保存 ダイアログ をオープンします (「(e) トレース・データの保存」参照)。 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 (複数行選択不可)。 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 【IECUBE】 【IECUBE2】 【シミュレータ】 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。 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。 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 【IECUBE】 【IECUBE2】 【シミュレータ】 をオープンします。
コピー	選択している行の内容を文字列としてクリップ・ボードにコピーします (複数行選択不可)。 ただし、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。
ウィンドウ連動	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。
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。
ソースヘジャンプ	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。
メモリヘジャンプ	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。

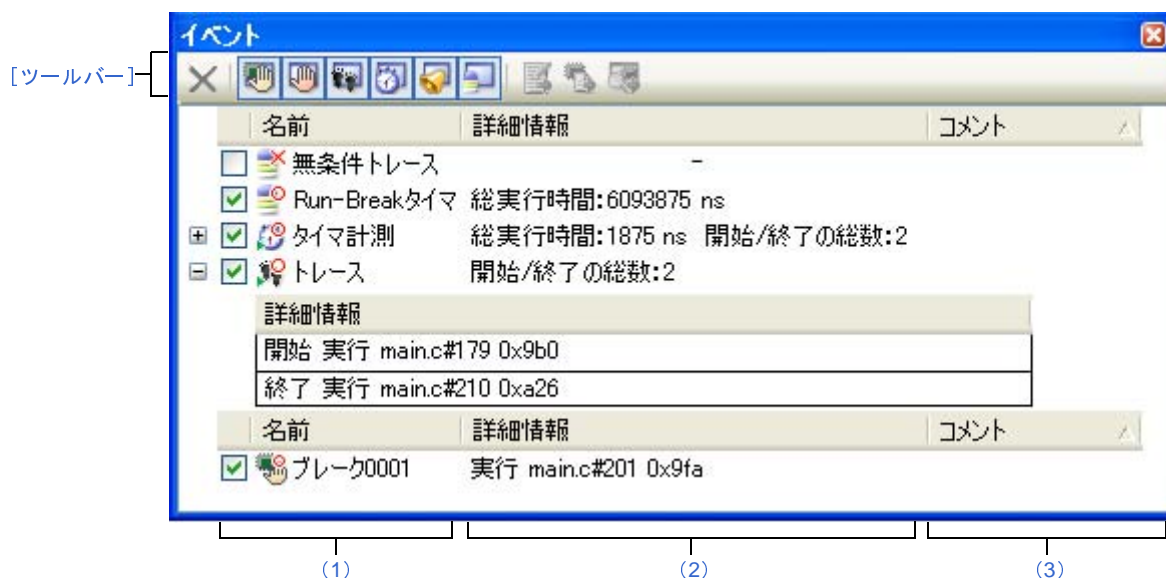
イベントパネル

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（「2.15 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. 選択しているマイクロコントローラ、および使用しているデバッグ・ツールの種類により、使用可能なイベント種別は異なります。詳細は、「表 A—19 イベント種別」内の各イベント種別ごとの参照部、または「(1) 有効イベント数の制限」を参照してください。
- 解析ツールの関数パネル／変数パネルで設定したイベントもこのパネルで管理されます。
 - ツールバーの , 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—48 イベントパネル



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]
- [[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]

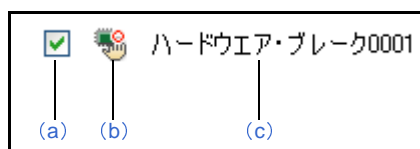
- [表示] メニュー → [イベント] を選択
- [IECUBE] [IECUBE2] [シミュレータ]

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。



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。

<input checked="" type="checkbox"/>	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。
<input type="checkbox"/>	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。
<input type="checkbox"/>	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。

表 A—18 イベント・マーク

イベント種別	有効状態	無効状態	保留状態	備考
ハードウェア・ブレーク				—
ソフトウェア・ブレーク				—
関数の先頭へのブレーク				解析ツールにより設定可能なブレーク・イベント
変数のアクセス・ブレーク				
無条件トレース			—	—
Run-Break タイマ		—	—	—
トレース				イベントパネルでのみ表示
トレース開始				エディタパネル／逆アセンブルパネルでのみ表示
トレース終了				ブルパネルでのみ表示
タイマ計測				イベントパネルでのみ表示
タイマ開始				エディタパネル／逆アセンブルパネルでのみ表示
タイマ終了				ブルパネルでのみ表示
ポイント・トレース				—
Printf イベント				—
上記イベントの複数設定	注1	注2	注3	エディタパネル／逆アセンブルパネルでのみ表示

注1. 複数のイベントの中で、1つでも有効状態のイベントがある場合。

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—19 イベント種別

イベント種別	説明
ハードウェア・ブレーク (ブレーク ^{注1})	デバッグ・ツールが、プログラム実行中にブレーク条件を逐次確認し、条件を満たした際にプログラムをブレークさせるイベントです。 →「2.8.2 任意の場所で停止する (ブレークポイント)」参照 →「2.8.3 任意の場所で停止する (ブレーク・イベント)」参照 →「2.8.4 変数 I/O レジスタへのアクセスで停止する」参照
ソフトウェア・ブレーク (ブレーク ^{注1})	ブレークさせるアドレスの命令コードをブレーク用の命令に書き換え、その命令を実行した際にプログラムをブレークさせるイベントです。 →「2.8.2 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ェア・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ェア・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)。 →「2.11.2 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)。 →「2.12.1 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)。 →「2.11.3 任意区間の実行履歴を収集する」参照
タイマ計測	タイマ開始イベント、およびタイマ終了イベントにより設定された条件を満たした際に、プログラムの実行時間の計測を開始/終了するイベントです (タイマ開始イベント/タイマ終了イベントのいずれかが設定されると表示されます)。 →「2.12.2 任意区間の実行時間を計測する」参照
ポイント・トレース	プログラムの実行により、指定した変数 I/O レジスタにアクセスした際に、その情報をトレース・メモリに記録するイベントです。 →「2.11.4 条件を満たしたときのみの実行履歴を収集する」参照
Printf イベント	プログラムの実行を任意の箇所で一瞬停止させたのち、ソフトウェア処理により 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)。 →「2.14.1 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（「(2) ブレークポイントを設定する」参照）は，“ブレーク”と表示します。
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。

表 A—20 イベント種別ごとの詳細情報

イベント種別	表示内容 ^{注1}	
ハードウェア・ブレーク (発生条件：実行系)	表示形式1	<発生条件> <ファイル名# 行番号> <アドレス>
	表示例	実行前 main.c#39 0x100
		実行後 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	<発生条件> <シンボル + オフセット> <アドレス>
	表示例	実行前 funcA + 0x10 0x100
		実行後 funcB + 0x20 0x200
実行前 — 0x300		
ハードウェア・ブレーク (発生条件：アクセス系)	表示形式1	<発生条件> <ファイル名# 変数名> <アドレス (範囲)> <比較条件> <比較値>
	表示例	リード main.c#variable1 0x100 - 0x101 == 0x5
		ライト sub.c#variable2 0x200 - 0x200 == 0x7
		リード/ライト sub2.c#variable3 0x300 - 0x303 == 0x8
	表示形式2	<発生条件> <ファイル名# 関数名# 変数名> <アドレス (範囲)> <比較条件> <比較値>
	表示例	リード main.c#func1#variable1 0x100 - 0x101 == 0x10
	表示形式3	<発生条件> <変数名> <アドレス (範囲)> <比較条件> <比較値>
	表示例	ライト variable1 0x100 - 0x101 == 0x10
ソフトウェア・ブレーク	表示形式1	<発生条件> <ファイル名# 行番号> <アドレス>
	表示例	実行前 main.c#40 0x102
		実行前 sub.c#101 0x204
	表示形式2	<発生条件> <シンボル + オフセット> <アドレス>
	表示例	実行前 funcA + 0x12 0x102
無条件トレース	表示形式	—
	表示例	—

イベント種別	表示内容 ^{注1}	
Run-Break タイマ	表示形式	総実行時間: <総実行時間>
	表示例	総実行時間: 1000ms
		総実行時間: OVERFLOW
トレース (発生条件: 実行系)	表示形式	開始/終了の総数: <トレース開始/トレース終了の総数> ^{注2} <開始/終了> <トレース開始/トレース終了の詳細情報>
	表示例	開始/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 main.c#200 0x100 - 終了 実行後 funcA + 0x10 0x100
タイマ計測 (発生条件: 実行系)	表示形式	総実行時間: <総実行時間> 開始/終了の総数: <タイマ開始/タイマ終了の総数> ^{注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> - <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 main.c#100 0x300 - 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 (発生条件: アクセス系)	表示形式 1	<発生条件> <変数名> <変数のアドレス>
	表示例	リード variable1 0x100
	表示形式 2	<発生条件> <ファイル名# 変数名> <変数のアドレス>
	表示例	ライト sub.c#variable2 0x200
	表示形式 3	<発生条件> <ファイル名# 関数名# 変数名> <変数のアドレス>
	表示例	リード/ライト sub.c#func1#variabl3 0x300
Printf イベント (アクション・イベント)	表示形式	<発生条件> <ファイル名# 行番号> <アドレス> <Print イベントの設定>
	表示例	実行前 main.c#39 0x100 aaa, bbb, ccc
		実行後 sub.c#100 0x200 aaaの結果の表示: aaa

注 1. 表示形式の詳細は次のとおりです。

<発生条件>	<p>次の条件のいずれか1つを表示します。</p> <p>【シミュレータ】以外</p> <p>実行系： 実行前, 実行後</p> <p>アクセス系： リード, ライト, リード/ライト</p> <p>【シミュレータ】</p> <p>実行系： 実行</p> <p>アクセス系： リード, ライト, リード/ライト</p>
<ファイル名 # 行番号>	<p>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。</p> <p>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は、<ロード・モジュール名 \$ ファイル名 # 行番号>を表示します。</p> <p>なお、逆アセンブルパネルで設定されたイベントでは、次の場合、行番号をシンボル+ オフセット形式で表示します。</p> <ul style="list-style-type: none"> - 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 <p>また、次の場合は、行番号を“-”で表示します。</p> <ul style="list-style-type: none"> - 行情報がなく、シンボル情報がない場合
<変数名>	<p>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。</p>
<比較条件>	<p>比較の条件 (==) を表示します。比較値が指定されなかった場合は表示しません。</p>
<比較値>	<p>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。</p>
<アドレス>	<p>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)。</p>
<開始/終了>	<p>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。</p>
<パスカウント>	<p>タイマのパスカウントを表示します。</p> <p>なお、タイマ・オーバーフロー発生時 (2.12.3 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。</p> <p>また、未計測の場合は、“未計測”を表示します。</p>
<総実行時間>	<p>タイマの総実行時間の測定結果を表示します。</p> <p>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)。</p> <p>なお、タイマ・オーバーフロー発生時 (2.12.3 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。</p> <p>また、未計測の場合は、“未計測”を表示します。</p>
<平均実行時間>	<p>タイマの平均実行時間の測定結果を表示します。</p> <p>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)。</p> <p>なお、タイマ・オーバーフロー発生時 (2.12.3 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。</p> <p>また、未計測の場合は、“未計測”を表示します。</p>

<最大実行時間>	<p>タイマの最大実行時間の測定結果を表示します。</p> <p>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）。</p> <p>なお、タイマ・オーバーフロー発生時（「2.12.3 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。</p> <p>また、未計測の場合は、“未計測”を表示します。</p>
<最小実行時間>	<p>タイマの最小実行時間の測定結果を表示します。</p> <p>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）。</p> <p>なお、タイマ・オーバーフロー発生時（「2.12.3 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。</p> <p>また、未計測の場合は、“未計測”を表示します。</p>
<Print イベントの設定>	<p>アクション・イベントダイアログ上で指定した、出力文字列：変数式を表示します。</p>

2. この行をクリックすることにより、下行の詳細情報を表示します。

(3) [コメント] エリア








設定されている各イベントに対して、ユーザが自由にコメントを入力できるエリアです。




コメントの入力は、コメントを入力したいイベントを選択後、このエリアをクリックするか、またはコンテキスト・メニューの「コメントの編集」を選択したのち、任意のテキストをキーボードから直接入力します（[Esc] キーの押下で編集モードをキャンセルします）。

コメントを編集したのち、[Enter] キーの押下、または編集領域以外へのフォーカスの移動により、編集を完了します。

なお、コメントは最大 256 文字まで入力することができ、使用中のユーザの設定として保存されます。

[ツールバー]

	<p>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。</p> <p>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。</p>
	ハードウェア・ブレイク関連のイベントを表示します（デフォルト）。
 （【シミュレータ】以外）	ソフトウェア・ブレイク関連のイベントを表示します（デフォルト）。
 【IECUBE】【IECUBE2】 【シミュレータ】	トレース関連のイベントを表示します（デフォルト）。
	タイマ関連のイベントを表示します（デフォルト） ^{注1} 。
	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）。
	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2} が設定されているアドレスに対応するソース行にcaretを移動した状態で、 エディタパネル がオープンします。
	選択しているイベント ^{注2} 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caret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2} が設定されているアドレスに対応するメモリ値にcaret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. 【V850E1】【V850ES】: 【MINICUBE2】【E1】【E20】【EZ Emulator】

このボタンは無効となります。

2. トレース・イベント/タイマ計測イベント/ビルトイン・イベント (無条件トレース・イベント/Run-Break タイマ・イベント) 以外のイベントが対象となります。

[[編集] メニュー (イベントパネル専用部分)]

イベントパネル専用の [編集] メニューは次のとおりです (その他の項目はすべて無効)。

削除	選択しているイベントを削除します。 ただし、ビルトイン・イベント (無条件トレース・イベント/Run-Break タイマ・イベント) を削除することはできません。
すべて選択	このパネルに表示されているすべてのイベントを選択状態にします。
検索 ...	検索・置換 ダイアログを [一括検索] タブが選択状態でオープンします。
置換 ...	検索・置換 ダイアログを [一括置換] タブが選択状態でオープンします。

[コンテキスト・メニュー]

有効化	選択しているイベントを有効状態にします。 ただし、選択しているイベントがすでに有効状態の場合は無効となります。
無効化	選択しているイベントを無効状態にします。 ただし、選択しているイベントがすでに無効状態の場合は無効となります。
削除	選択しているイベント、およびイベント条件を削除します。 ただし、ビルトイン・イベント (無条件トレース・イベント/Run-Break タイマ・イベント) を削除することはできません。
すべて選択	現在表示しているすべてのイベントを選択状態にします。

表示選択	表示するイベント種別を限定するために、次のカスケード・メニューを表示します。 デフォルトでは、すべての項目が選択されています。
ハードウェア・ブレイク	ハードウェア・ブレイク関連のイベントを表示します。
ソフトウェア・ブレイク	ソフトウェア・ブレイ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 【IECUBE】【IECUBE2】 【シミュレータ】	トレース関連のイベントを表示します。
アクション・イベント	アクション・イベント（Printf イベント）を表示します。
ビルトイン・イベント	ビルトイン・イベント（無条件トレース・イベント /Run-Break タイマ）を表示します。
タイマ設定	タイマ関連の設定をするために、次のカスケード・メニューを表示します。 ただし、タイマ関連のイベントを選択している場合のみ有効です。
タイマの初期化	選択しているイベント（Run-Break タイマ・イベントを除く）で使用するタイマを初期化します。
ナノ秒表示	選択しているイベントのタイマ結果をナノ秒（ns）単位で表示します。
マイクロ秒表示	選択しているイベントのタイマ結果をマイクロ秒（ μ s）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。
メモリジャンプ	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 ...	選択しているイベントを編集するために、次のダイアログをオープンします。 - アクション・イベント（Printf イベント）の場合 アクション・イベントダイアログ
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。

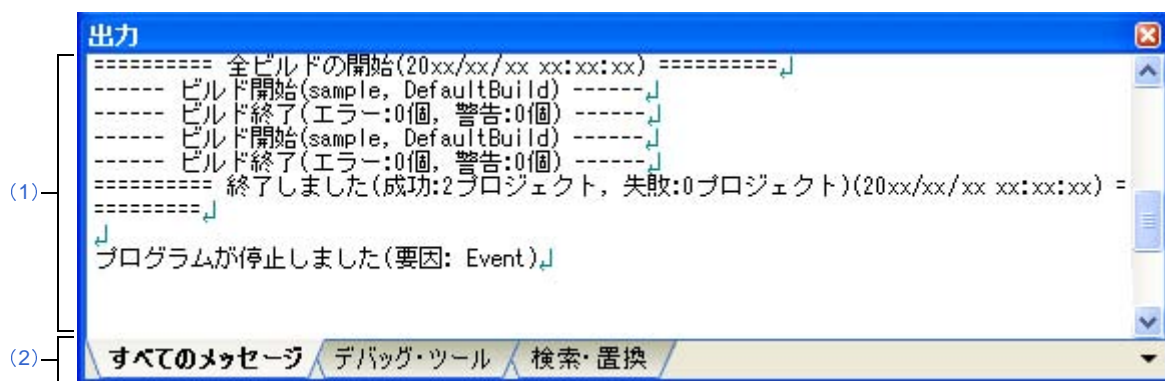
出力パネル

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／解析ツールなど）から出力されるメッセージの表示、または検索・置換 ダイアログによる一括検索を行った際の結果、および Printf イベント（「2.14.1 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 ，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—49 出力パネル



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [[ファイル] メニュー（出力パネル専用部分）]
- [[編集] メニュー（出力パネル専用部分）]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [出力] を選択

[各エリアの説明]

(1) メッセージ・エリア

各ツールから出力されたメッセージ、検索結果、および Printf イベントによる出力結果を表示します。

検索結果（一括検索）の表示では、検索を行うごとに、以前のメッセージをクリアしたのち新しいメッセージを表示します（[すべてのメッセージ] タブを除く）。

なお、メッセージの表示色は、出力メッセージの種別により、次のように異なります（表示の際の文字色／背景色はオプション ダイアログにおける [全般 - フォントと色] カテゴリの設定に依存します）。

メッセージ種別	表示例（デフォルト）		説明	
通常メッセージ	AaBbCc	文字色	黒	何らかの情報を通知する際に表示されます。
		背景色	白	
警告メッセージ	AaBbCc	文字色	青	操作に対して、何らかの警告を通知する際に表示されます。
		背景色	標準色	
エラー・メッセージ	AaBbCc	文字色	赤	致命的なエラー、または操作ミスにより実行が不可能な場合に表示されます。
		背景色	薄グレー	

このエリアは、次の機能を備えています。

(a) タグ・ジャンプ

出力されたメッセージをダブルクリック、またはメッセージにキャレットを移動したのち [Enter] キーを押下することにより、[エディタパネル](#)をオープンして該当ファイルの該当行番号を表示します。

これにより、ビルド時に出力されたエラー・メッセージなどから、ソース・ファイルの該当するエラー行へジャンプすることができます。

(b) ヘルプの表示

警告メッセージ、またはエラー・メッセージを表示している行にキャレットがある状態で、コンテキスト・メニューの [メッセージに関するヘルプ] を選択するか、または [F1] キーを押下することにより、その行のメッセージに関するヘルプを表示します。

(c) ログの保存

[ファイル] メニュー→ [名前を付けて出力 - タブ名を保存 ...] を選択することにより、[名前を付けて保存ダイアログ](#)をオープンし、現在選択しているタブ上に表示されている全内容をテキスト・ファイル (*.txt) に保存することができます（非選択状態のタブ上のメッセージは保存の対象となりません）。

(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/解析ツールなど）から出力されるメッセージを表示します（ラピッド・ビルドの実行によるメッセージを除く）。
デバッグ・ツール	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/解析ツールなど）から出力されるメッセージのうち、デバッグ・ツールが出力するメッセージを表示します。
検索・置換	検索・置換ダイアログによる一括検索結果を表示します。

注意 新たなメッセージが非選択状態のタブ上に出力されても、自動的なタブの表示切り替えは行いません。この場合、タブ名の先頭に“*”マークが付加し、新たなメッセージが出力されていることを示します。

[[ファイル] メニュー (出力パネル専用部分)]

出力パネル専用の [ファイル] メニューは次のとおりです (その他の項目は共通)。

ただし、プログラム実行中はすべて無効となります。

出力 - タブ名を保存	現在選択しているタブ上に表示されている内容を、前回保存したテキスト・ファイル (*.txt) に保存します (「(c) ログの保存」参照)。 なお、起動後に初めてこの項目を選択した場合は、[名前を付けてタブ名を保存 ...] の選択と同等の動作となります。 ただし、ビルド実行中は無効となります。
名前を付けて出力 - タブ名を保存 ...	現在選択しているタブ上に表示されている内容を、指定したテキスト・ファイル (*.txt) に保存するために、名前を付けて保存 ダイアログをオープンします (「(c) ログの保存」参照)。

[[編集] メニュー (出力パネル専用部分)]

出力パネル専用の [編集] メニューは次のとおりです (その他の項目はすべて無効)。

コピー	選択している文字列をクリップ・ボードにコピーします。
すべて選択	現在選択しているタブ上に表示されているすべてのメッセージを選択状態にします。
検索 ...	検索・置換 ダイアログを [クイック検索] タブが選択状態でオープンします。
置換 ...	検索・置換 ダイアログを [一括置換] タブが選択状態でオープンします。

[[コンテキスト・メニュー]]

コピー	選択している文字列をクリップ・ボードにコピーします。
すべて選択	現在選択しているタブ上に表示されているすべてのメッセージを選択状態にします。
クリア	現在選択しているタブ上に表示されているすべてのメッセージを消去します。
タグ・ジャンプ	エディタ パネルをオープンし、キャレット位置のメッセージに該当するファイルの該当行番号にジャンプします。
検索の中止	現在実行中の検索を中止します。 ただし、検索を実行していない場合は無効となります。
メッセージに関するヘルプ	現在のキャレット位置のメッセージに関するヘルプを表示します。 ただし、警告メッセージ/エラー・メッセージのみが対象となります。

メモリ・マッピング ダイアログ

メモリ・マッピングの設定をメモリ種別ごとに行います。

注意 デバッグ・ツールと切断中の場合では、ユーザにより追加されたメモリ・マッピング領域のみが表示対象となります。デバッグ・ツールと接続することにより（「2.4.1 デバッグ・ツールを接続する」）、各メモリ種別ごとの詳細表示を行います。

図 A—50 メモリ・マッピング ダイアログ (IECUBE [V850ES]) [シミュレータ] 以外)

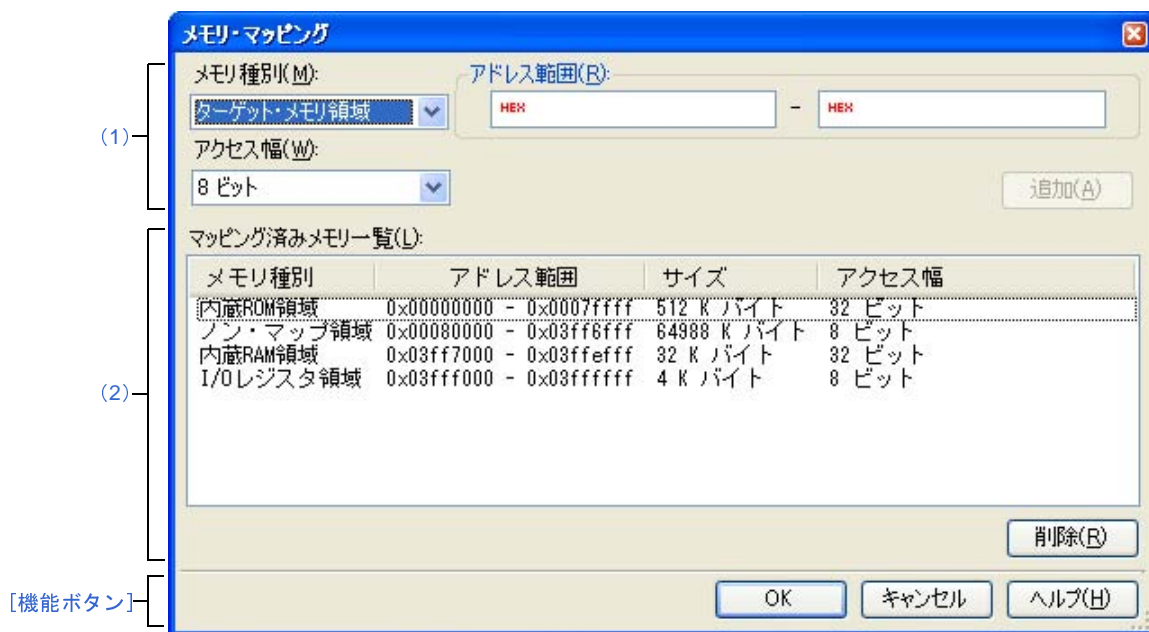


図 A—51 メモリ・マッピング ダイアログ (IECUBE [V850E1])

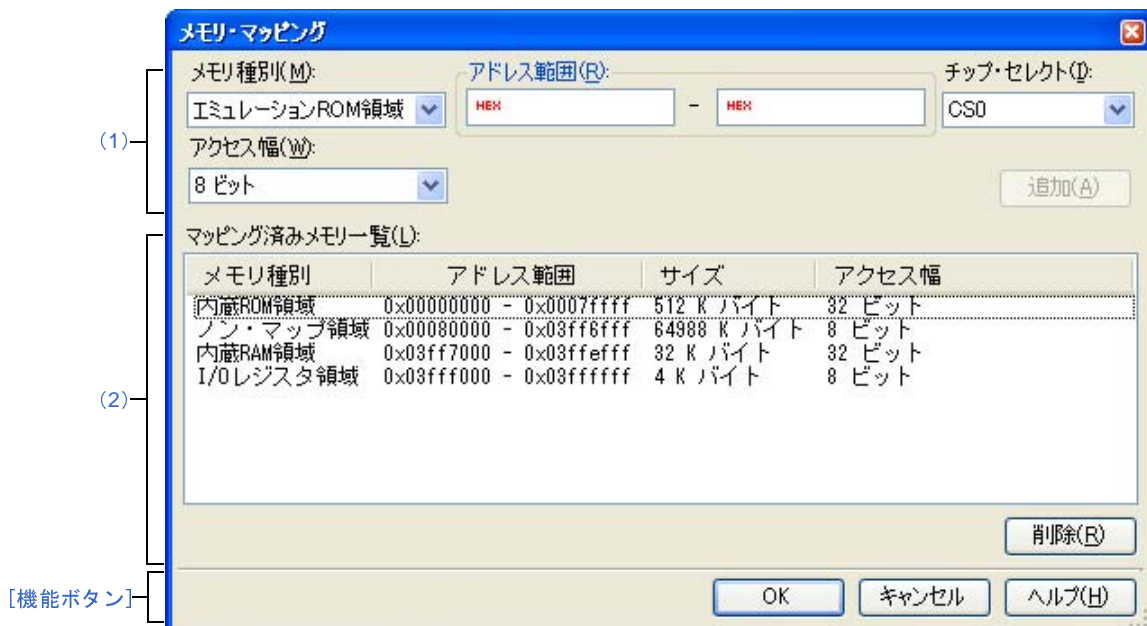
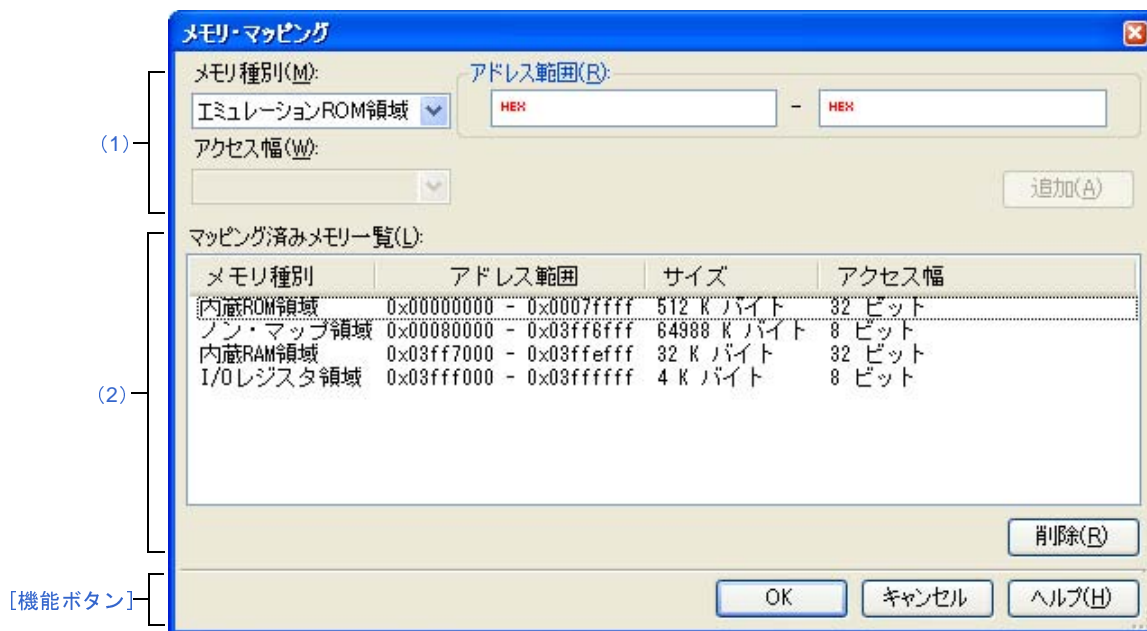


図 A—52 メモリ・マッピング ダイアログ【シミュレータ】



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- プロパティ パネルの [デバッグ・ツール設定] タブにおいて、[メモリ] カテゴリ内 [メモリ・マッピング] プロパティを選択することにより表示される [...] ボタンをクリック

注意 プログラム実行中は、このダイアログをオープンすることはできません。

[各エリアの説明]

(1) 追加メモリ・マッピング指定エリア

新たに追加するメモリ・マッピングの情報を指定します。

(a) [メモリ種別]

追加するメモリ・マッピングのメモリ種別を次のドロップダウン・リストより選択します（デフォルトで選択される項目は使用するデバッグ・ツールに依存します）。

エミュレーションROM 領域	エミュレーションROM 領域を追加します。
【IECUBE 【V850E1】 【V850ES】】	【IECUBE】: IECUBE 代替 ROM を使用します。
【シミュレータ】	【シミュレータ】: シミュレータ代替 ROM を使用します。

エミュレーションRAM領域 【IECUBE 【V850E1】 【V850ES】】 【シミュレータ】	エミュレーションRAM領域を追加します。 【IECUBE】: IECUBE 代替 RAM を使用します。 【シミュレータ】: シミュレータ代替 RAM を使用します。
ターゲット・メモリ領域	ターゲット・メモリ領域を追加します。
ターゲットROM領域 【IECUBE 【V850E1】 【V850ES】】	ターゲットROM領域を追加します。 ターゲットROMに指定される領域は、フェイルセーフ・ブレークのライト・プロテクトの対象となります。
I/O保護領域	I/O保護領域を追加します ^注 。

注 I/O保護領域は、デバッグ機能によるアクセスを保護する領域です。

I/O保護領域に設定されたアドレス範囲は、メモリパネルなどからアクセスできません（ロード・モジュールの実行によるアクセスのみ可能）。I/O保護領域は、ターゲット・メモリ領域として指定されている範囲内でのみ設定可能です。

備考 マイクロコントローラの外部メモリ領域と使用禁止領域（ガード領域）は、ノン・マップ領域として扱われます。このため、使用禁止領域に重なってマッピングを指定した場合、ノン・マップ領域と重なることを警告するメッセージを表示します。

なお、外部メモリ領域と使用禁止領域のマッピング情報については、使用するマイクロコントローラのマニュアルを参照してください。

なお、メモリ・マッピングが可能な属性／サイズは次のとおりです。

表 A—21 設定可能なメモリ・マッピング属性

属性	デバッグ・ツール				
	IECUBE	IECUBE2	MINICUBE E1(LPD) E1(JTAG) E20(LPD) E20(JTAG)	MINICUBE2 E1(Serial) E20(Serial) EZ Emulator	シミュレータ
エミュレーションROM領域	△注	—	—	—	○
エミュレーションRAM領域	△注	—	—	—	○
ターゲット・メモリ領域	○	○	○	○	○
ターゲットROM領域	○	—	—	—	—
I/O保護領域	○	○	○	○	○

○：可能（マッピング単位：1Mバイト（ただし、シミュレータは64KB））

△：オプション・ボード搭載時のみ可能（マッピング単位：1Mバイト）

—：不可

注 エミュレーションROM／RAM領域は16Mバイトで構成

(b) [アドレス範囲]

追加するメモリ・マッピングの開始アドレスと終了アドレスを指定します。それぞれのテキスト・ボックスに、16進数を直接入力します。

ただし、次の指定の場合、新たにメモリ・マッピングを追加することはできません（このエリアの[追加]ボタンをクリックした際に、メッセージを表示します）。

- メモリ種別として [ターゲット・メモリ領域] を選択した際に、指定したアドレス範囲が他のメモリ領域と重複している場合
- メモリ種別として [I/O 保護] を選択した際に、指定したアドレス範囲が1つのターゲット・メモリ領域内で収まらない場合

(c) [チップ・セレクト] [IECUBE [V850E1]]

選択しているマイクロコントローラが V850E1 コアの場合で、[メモリ種別] において [エミュレーションROM] / [エミュレーションRAM] が選択されている場合にのみ表示される項目です。

[アドレス範囲] で指定したアドレスに該当するチップ・セレクトをドロップダウン・リストにより選択します。

(d) [アクセス幅] ([シミュレータ] 以外)

追加するメモリ・マッピングのアクセス幅を次のドロップダウン・リストより選択します（直接入力不可）。

なお、メモリ種別として [I/O 保護領域] を選択した場合、アクセス幅は、対象となるターゲット・メモリ領域のアクセス幅と同値にする必要があります。

8 ビット	追加するメモリ・マッピングのアクセス幅を 8 ビットにします（デフォルト）。
16 ビット	追加するメモリ・マッピングのアクセス幅を 16 ビットにします ^注 。
32 ビット	追加するメモリ・マッピングのアクセス幅を 32 ビットにします ^注 。

注 メモリ・マッピングに追加される際に、終了アドレスに指定したアドレス値の端数を 16 ビットの場合は 2 の倍数に、32 ビットの場合は 4 の倍数に自動的に切り上げます。

(e) ボタン

ボタン	機能
追加	このエリアで指定した内容をメモリ・マッピングに追加します。 追加されたメモリ・マッピングは、[マッピング済みメモリー一覧] エリアに表示されます。 なお、[OK] ボタンを押下するまでは、変更内容の設定は行われません。

(2) [マッピング済みメモリー一覧] エリア

(a) 一覧の表示

追加メモリ・マッピング指定エリアで追加したメモリ・マッピングと、マイクロコントローラ内のメモリ・マッピングの情報を表示します。このエリアを編集することはできません。

メモリ種別	<p>次のメモリ種別を表示します。</p> <p>【V850E1】【V850ES】</p> <ul style="list-style-type: none"> - 内蔵 ROM 領域^{注1} - 内蔵 RAM 領域 - データフラッシュ領域^{注2} (【シミュレータ】以外) - I/O レジスタ領域 - ターゲット・メモリ領域 - ターゲット ROM 領域【IECUBE】 - エミュレーション ROM 領域【IECUBE】【シミュレータ】 - エミュレーション RAM 領域【IECUBE】【シミュレータ】 - ノン・マップ領域 - I/O 保護領域 <p>【V850E2】</p> <ul style="list-style-type: none"> - 内蔵 ROM 領域^{注1} - 内蔵 RAM 領域 - データフラッシュ領域^{注2} (【シミュレータ】以外) - Pバス周辺 I/O 領域 - CPU 周辺 I/O 領域 - 内蔵周辺 I/O 領域 - Hバス周辺 I/O 領域 - Hバス共有メモリ領域 - バックアップ RAM 領域 - ターゲット・メモリ領域 - エミュレーション ROM 領域【シミュレータ】 - エミュレーション RAM 領域【シミュレータ】 - ノン・マップ領域 - I/O 保護領域
アドレス範囲	<p>アドレス範囲を<開始アドレス> - <終了アドレス>で表示します。</p> <p>“0x”を付与した16進数表示固定です。</p>
サイズ	<p>サイズを10進数で表示します(単位: バイト / Kバイト^{注3})。</p>
アクセス幅	<p>アクセス幅を表示します(単位: ビット)。</p>

注1. 選択しているマイクロコントローラがROMレス品の場合は表示しません。

ただし、デバッグ・ツールにエミュレーション内蔵ROM領域が存在する場合で、[プロパティパネル](#)の[\[接続用設定\]](#)タブ上の[内蔵ROM/RAM]カテゴリ内[内蔵ROMサイズ[Kバイト]]プロパティに“0”より大きな値を設定した場合にかぎり、“内蔵ROM領域”を表示します。

2. 選択しているマイクロコントローラがデータフラッシュ・メモリ内蔵品の場合のみ表示します。

3. 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	<p>このエリアで選択しているメモリ・マッピングを削除します。</p> <p>削除できるメモリ領域は、ターゲット・メモリ領域 / I/O 保護領域 / エミュレーション ROM 領域【IECUBE】【シミュレータ】 / エミュレーション RAM 領域【IECUBE】【シミュレータ】のいずれかです（マイクロコントローラ内のメモリ・マッピングを削除することはできません）。</p> <p>ただし、I/O 保護領域が設定されているターゲット・メモリ領域を削除しようとした場合は、メッセージを表示し、[OK] ボタンがクリックされた場合のみ、選択されたターゲット・メモリ領域とその領域にマッピングされている I/O 保護領域をすべて削除します。</p>

[機能ボタン]

ボタン	機能
OK	現在設定されているメモリ・マッピングをデバッグ・ツールに設定し、このダイアログをクローズします。
キャンセル	メモリ・マッピングの変更を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

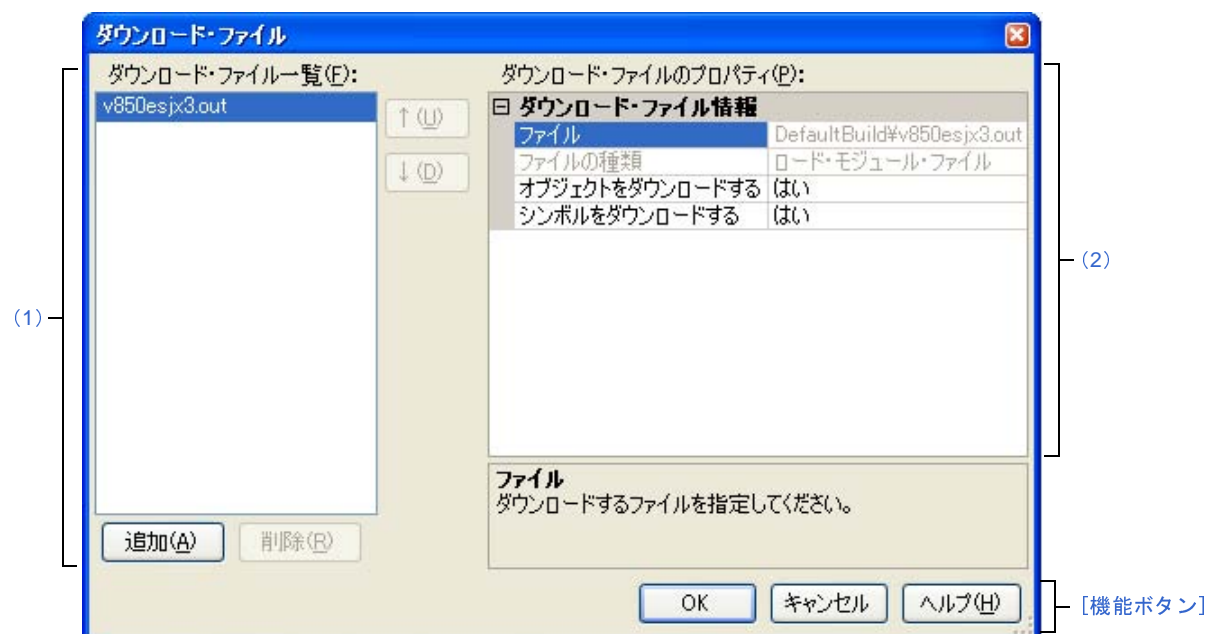
ダウンロード・ファイル ダイアログ

ダウンロードする際のファイルの選択、およびダウンロード条件の設定を行います（「2.5 ダウンロード／アップロード」参照）。

プロジェクト（メイン・プロジェクト／サブプロジェクト）でビルド対象に指定しているファイルは、自動的にダウンロードの対象ファイルとして登録されます（削除不可）。

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。

図 A—53 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
↑	選択しているファイルを1行上に移動します。 ただし、最上部のファイル、またはプロジェクトのビルド対象に指定しているファイルを選択している場合は無効となります。
↓	選択しているファイルを1行下に移動します。 ただし、最下部のファイル、またはプロジェクトのビルド対象に指定しているファイルを選択している場合は無効となります。
追加	一覧に空欄の項目（“-”）を1つ追加し、選択状態にします。 [ダウンロード・ファイルのプロパティ] エリアにおいて、追加するファイルのダウンロード条件を指定してください。 ただし、すでに20個以上のファイルが登録されている場合は無効となります。
削除	選択しているファイルを一覧から削除します。 ただし、プロジェクトのビルド対象に指定しているファイルは削除することはできません。

備考 1. ファイル名にマウス・カーソルを合わせることで、対象ファイルのパス情報をポップアップ表示します。

2. ファイル名をマウスでドラッグすることにより、一覧内の表示順序を変更することができます。ただし、プロジェクトでビルド対象に指定しているファイルの表示順序を変更することはできません。

(2) [ダウンロード・ファイルのプロパティ] エリア

(a) [ダウンロード・ファイル情報]

[\[ダウンロード・ファイル一覧\]](#) エリアで選択しているファイルに対して、ダウンロード条件の表示/設定変更を行います。

また、[追加] ボタンにより、新規にダウンロード・ファイルを追加する場合は、ここで追加ファイルのダウンロード条件を指定します。

ファイル	ダウンロードするファイルを指定します。	
	デフォルト	ファイル名（ただし、新規追加の場合は空欄）
	変更方法	キーボードからの直接入力。またはこの項目を選択すると欄内右端に表示される [...] ボタン ^{注1} のクリックによりオープンする ダウンロードするファイルを選択 ダイアログ による指定
	指定可能値	「 表 2-3 ダウンロード可能なファイル形式 」参照 最大指定文字数：259 文字
ファイルの種類	ダウンロードするファイルのファイル形式を指定します。	
	デフォルト	ロード・モジュール・ファイル
	変更方法	ドロップダウン・リストによる選択
	指定可能値	次のいずれか - ロード・モジュール・ファイル - ヘキサ・ファイル - ヘキサ・ファイル [ID タグ付] ^{注2} - ヘキサ・ファイル [ID タグ付 (プログラム形式)] ^{注2} - バイナリ・データ・ファイル
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] 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
	指定可能値	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
	指定可能値	0x0 ~ 0xFFFFFFFF の 16 進数
オブジェクトをダウンロードする	指定したファイルからオブジェクト情報をダウンロードするか否かを指定します。 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 いいえ

シンボルをダウンロードする	指定したファイルからシンボル情報をダウンロードするか否かを指定します ^{注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 いいえ

注1. [ダウンロード・ファイル一覧] エリアにおいて、プロジェクトのビルド対象のファイルを選択している場合、またはプログラム実行中は、[...] ボタンは表示されません。

2. 【シミュレータ】以外

選択しているマイクロコントローラがデータフラッシュ・メモリ内蔵品でIDタグを実装しており、かつデータフラッシュ・メモリ機能を有効にしている場合（プロパティパネルの[接続用設定] タブ上の[データフラッシュ] カテゴリ内 [データフラッシュ・メモリ領域を使用する] プロパティを [はい] に設定）のみ表示されます。

3. 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。

[機能ボタン]

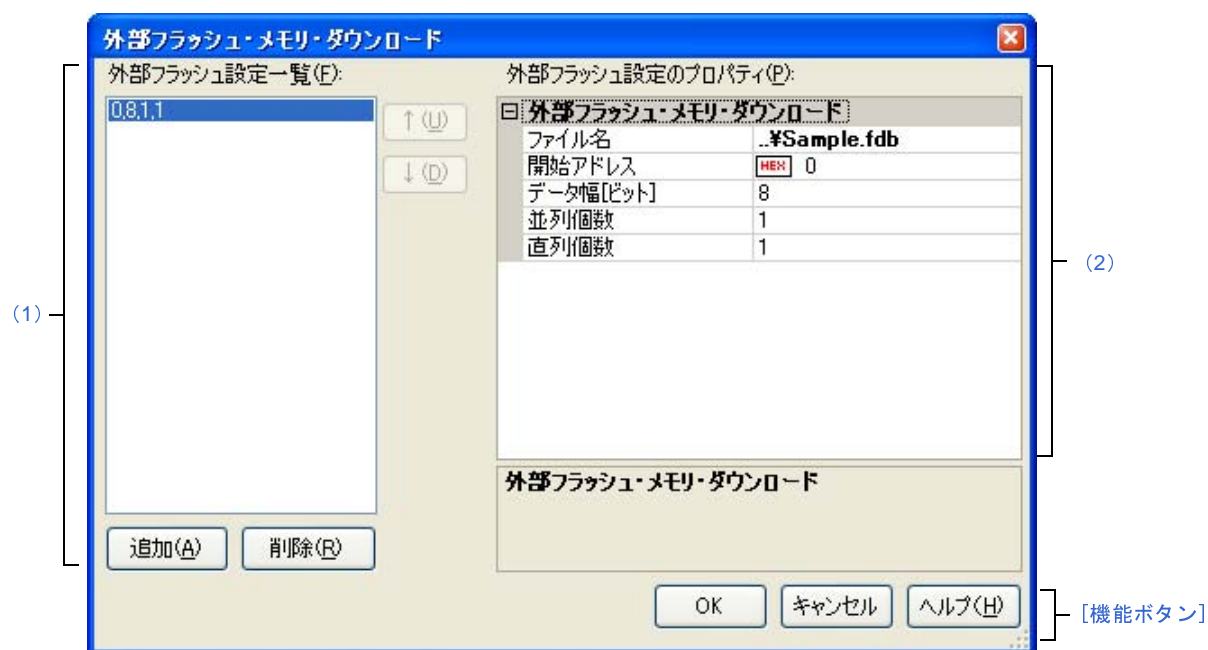
ボタン	機能
OK	ダウンロード・ファイルの設定を終了し、このダイアログをクローズします。
キャンセル	ダウンロード・ファイルの変更を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

外部フラッシュ・メモリ・ダウンロード ダイアログ

外部フラッシュ・メモリにダウンロードする際の外部フラッシュ・メモリ情報ファイル (*.fdb) の選択、およびダウンロード条件の設定を行います（「(5) 外部フラッシュ・メモリへダウンロードする【IECUBE】【IECUBE2】【MINICUBE】【E1/E20(LPD)/(JTAG)】」参照）。

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。

図 A—54 外部フラッシュ・メモリ・ダウンロード 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、[外部フラッシュ・メモリ・ダウンロード] カテゴリ内 [外部フラッシュ・メモリ・ダウンロード設定] 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) [外部フラッシュ設定一覧] エリア

(a) 一覧の表示

登録した外部フラッシュ・メモリ情報ファイルの一覧を、対応する“開始アドレス、データ幅、並列個数、直列個数”で表示します。

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。

新規に外部フラッシュ・メモリ情報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、[外部フラッシュ設定のプロパティ]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
↑	選択しているファイルを1行上に移動します。 ただし、最上部のファイルを選択している場合は無効となります。
↓	選択しているファイルを1行下に移動します。 ただし、最下部のファイル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。 [外部フラッシュ設定のプロパティ]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10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。

備考1. ファイルにマウス・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。

2. ファイル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。

(2) [外部フラッシュ設定のプロパティ] エリア

(a) [外部フラッシュ・メモリ・ダウンロード]

[外部フラッシュ設定一覧]エリアで選択しているファイルに設定したダウンロード条件（外部フラッシュ・メモリのための設定）を表示／編集します。

また、[追加]ボタンにより、新規に外部フラッシュ・メモリ情報ファイルを追加する場合は、ここで追加ファイルのためのダウンロード条件を指定します。

ファイル名	外部フラッシュ・メモリ情報ファイル (*.fbd) を指定します。	
	デフォルト	空欄
	変更方法	キーボードからの直接入力。またはこの項目を選択すると欄内右端に表示される [...] ボタン ^注 のクリックによりオープンする外部フラッシュ・メモリ情報ファイルを選択 ダイアログによる指定
開始アドレス	外部フラッシュ・メモリの開始アドレスを指定します。	
	デフォルト	0
	変更方法	キーボードからの直接入力
	指定可能値	0x0 ~ 0xFFFFFFFF の 16 進数
データ幅 [ビット]	外部フラッシュ・メモリのデータ幅 (アクセス・サイズ) を指定します。	
	デフォルト	8
	変更方法	ドロップダウン・リストによる選択
	指定可能値	8, 16, 32
並列個数	外部フラッシュ・メモリの並列個数を指定します。	
	デフォルト	1
	変更方法	ドロップダウン・リストによる選択。またはキーボードからの直接入力
	指定可能値	- ドロップダウン・リストによる次のいずれか 1, 2, 4 - テキスト入力による次の範囲 1 ~ 99 の整数値
直列個数	外部フラッシュ・メモリの直列個数を指定します。	
	デフォルト	1
	変更方法	ドロップダウン・リストによる選択。またはキーボードからの直接入力
	指定可能値	- ドロップダウン・リストによる次のいずれか 1, 2 - テキスト入力による次の範囲 1 ~ 99 の整数値

注 プログラム実行中は, [...] ボタンは表示されません。

[機能ボタン]

ボタン	機能
OK	外部フラッシュ・メモリへのダウンロード条件の設定を終了し、このダイアログをクローズします。
キャンセル	外部フラッシュ・メモリへのダウンロード条件の変更を無効とし、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【V850E2】

V850E2 コアの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、選択しているマイクロコントローラの種類により CPU リセットが発生します。

図 A—55 フラッシュ・オプションの設定 ダイアログ【IECUBE2】

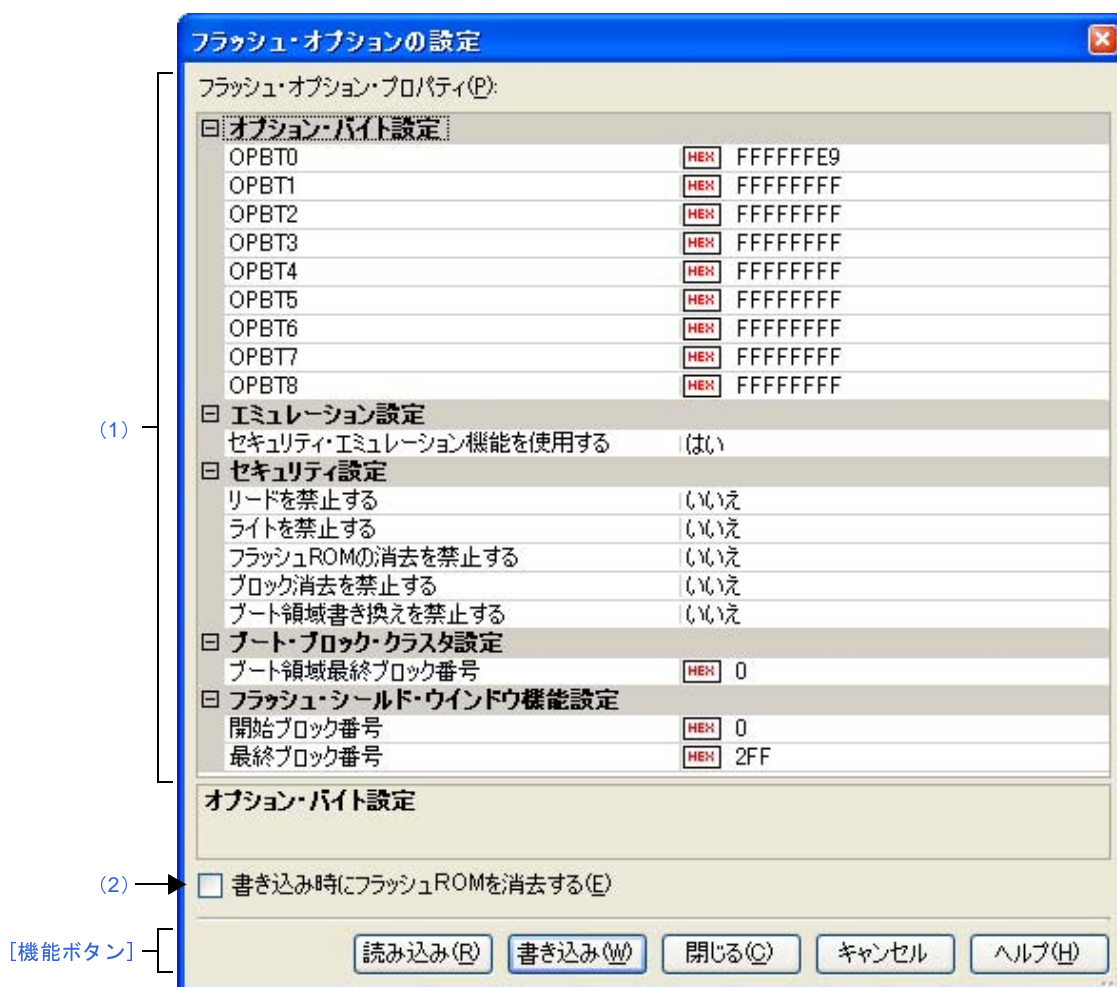


図 A—56 フラッシュ・オプションの設定 ダイアログ【MINICUBE】【MINICUBE2】【E1】【E20】



図 A—57 フラッシュ・オプションの設定 ダイアログ【シミュレータ】



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [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【V850E2】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

注意 選択しているマイクロコントローラの種類により、表示されるプロパティ項目は異なります。

(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。

OPBT0 ~ 8	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
	指定可能値	0x0 ~ 0xFFFFFFFF の範囲の 16 進数

注意 OPBT0 のビット 31/30 は、使用するデバッグ・ツールにより 0、または 1 固定となります。

デバッグ・ツール	ビット 31	ビット 30
IECUBE2	-	-
MINICUBE E1/E20(JTAG)	1 注	-
MINICUBE2 E1/E20(Serial)	0	1
E1/E20(LPD)	-	-

注 “0” を設定したい場合は、フラッシュ・プログラマにより書き込みを行ってください。

(b) [オンチップ・デバッグ・セキュリティ ID 設定] 【MINICUBE】 【MINICUBE2】 【E1】 【E20】

オンチップ・デバッグ・セキュリティ ID の設定を行います。

なお、オンチップ・デバッグ・セキュリティ ID 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。

セキュリティ ID	オンチップ・デバッグ・セキュリティ ID を指定します。	
	デフォルト	FFFFFFFFFFFFFFFFFFFFFFFF
	変更方法	キーボードからの直接入力
	指定可能値	24 桁の 16 進数 (12 バイト)

注意 1. ビット 95 は、“1” 固定となります。

“0” を設定したい場合は、フラッシュ・プログラマにより書き込みを行ってください。

2. この項目は、[読み込み] ボタンによる反映の対象外です。

(c) [エミュレーション設定] [IECUBE2]

セキュリティ・エミュレーション機能を使用するか否かを指定します。

セキュリティ・エミュレーション機能を使用する	セキュリティ・エミュレーション機能を使用するか否かを指定します。	
	デフォルト	はい
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい
いいえ		セキュリティ・エミュレーション機能を使用しません。

(d) [セキュリティ設定] [IECUBE2]

セキュリティ・エミュレーション機能に関する設定を行います。

ただし、このカテゴリは、[セキュリティ・エミュレーション機能を使用する] プロパティにおいて [はい] を指定した場合のみ表示されます。

注意 このカテゴリ内のプロパティ値を [はい] に変更したのち [書き込み] ボタンをクリックすると、プロパティ値を [いいえ] に戻すことができません。

[いいえ] に戻す場合は、[書き込み時にフラッシュ ROM を消去する] を選択したのち [書き込み] ボタンをクリックしてください。

リードを禁止する	リード禁止のエミュレーションを行うか否かを指定します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい
いいえ		リード禁止のエミュレーションを行いません。
ライトを禁止する	ライト禁止のエミュレーションを行うか否かを指定します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい
いいえ		ライト禁止のエミュレーションを行いません。
フラッシュ ROM の消去を禁止する	フラッシュ ROM 消去禁止のエミュレーションを行うか否かを指定します。 なお、このプロパティは、選択しているマイクロコントローラがこの機能をサポートしている場合のみ表示されます。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい
いいえ		フラッシュ ROM 消去禁止のエミュレーションを行いません。

ブロック消去を禁止する	ブロック消去禁止のエミュレーションを行うか否かを指定します。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	ブロック消去禁止のエミュレーションを行います。
いいえ		ブロック消去禁止のエミュレーションを行いません。	
ブート領域書き換えを禁止する	ブート領域書き換え禁止のエミュレーションを行うか否かを指定します。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	ブート領域書き換え禁止のエミュレーションを行います ^注 。
いいえ		ブート領域書き換え禁止のエミュレーションを行いません。	

注 [はい] に指定した場合、[ブート領域最終ブロック番号] プロパティを、[フラッシュ・シールド・ウィンドウ機能設定] カテゴリ内の [開始ブロック番号] / [最終ブロック番号] プロパティの範囲外に設定する必要があります。

(e) [ブート・ブロック・クラスタ設定] [IECUBE2]

ブート・スワップ機能に関する設定を行います。

なお、このカテゴリは [セキュリティ・エミュレーション機能を使用する] プロパティにおいて [はい] を指定した場合のみ表示されます。

ブート領域最終ブロック番号	ブート領域の最終ブロック番号を指定します。	
	デフォルト	0
	変更方法	キーボードからの直接入力 ただし、[ブート領域書き換えを禁止する] プロパティが [いいえ] の場合のみ
	指定可能値	16 進数

(f) [フラッシュ・シールド・ウィンドウ機能設定] ([シミュレータ] 以外)

フラッシュ・シールド・ウィンドウ機能に関する設定を行います。

開始ブロック番号	フラッシュ・シールド・ウィンドウの開始ブロック番号を指定します。	
	デフォルト	0
	変更方法	キーボードからの直接入力
	指定可能値	0x0 ~ 0x7FF の 16 進数
最終ブロック番号	フラッシュ・シールド・ウィンドウの最終ブロック番号を指定します。	
	デフォルト	選択しているマイクロコントローラの最終ブロック番号
	変更方法	キーボードからの直接入力
	指定可能値	0x0 ~ 0x7FF の 16 進数

(2) [書き込み時にフラッシュ ROM を消去する]

<input checked="" type="checkbox"/>	[書き込み] ボタンをクリックした際にフラッシュ ROM の消去を行います。
<input type="checkbox"/>	[書き込み] ボタンをクリックした際にフラッシュ ROM の消去を行いません (デフォルト)。

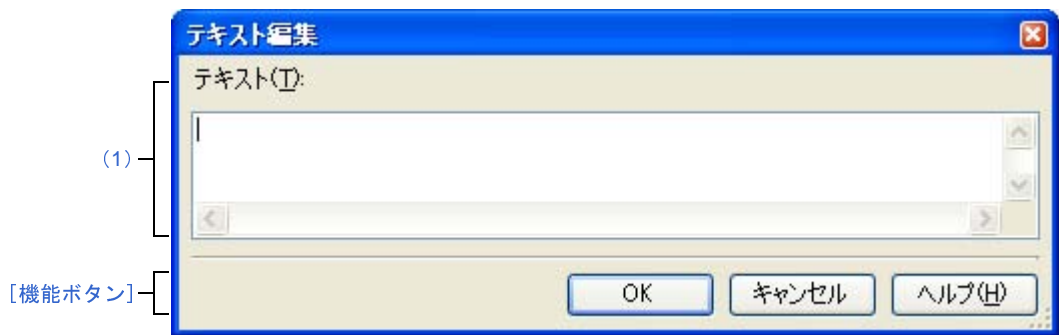
[機能ボタン]

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、 プロパティパネル 、および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。 なお、この際、選択しているマイクロコントローラによって、自動的にCPUリセットが発生する場合があります。
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。

テキスト編集 ダイアログ

複数行のテキストの入力, 編集を行います。

図 A—58 テキスト編集 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
OK	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（「2.14 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）、（「2.15.6 イベント設定に関する留意事項」も参照してください）。

図 A—59 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カー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カー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
OK	アクション・イベントの設定を終了し、指定したアクション・イベントを指定した位置に設定します。
キャンセル	アクション・イベントの設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

[Printf イベント] タブ

アクション・イベントとして、Printf イベントの設定を行います（「2.14 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、ソフトウェ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—60 アクション・イベント ダイアログ：[Printf イベント] タブ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- **エディタ パネル**において、Printf イベントを設定したい行に caret 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 caret 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は、選択している変数式がデフォルトで表示されます。

なお、変数式として指定できる基本入力形式と、その際にPrintf イベントとして出力される値は次のとおりです。

表 A—22 変数式と出力される値の関係（Printf イベント）

変数式	出力される値
C 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
変数式.メンバ名	構造体／共用体のメンバ値
変数式->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
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/EQU シンボル名/即値アドレス	ラベルの値/EQU シンボルの値/即値アドレスの値
ビット・シンボル	ビット・シンボルの値

注意 算術式（“+” / “-” など）を使用した変数式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caret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—61 Printf イベントの出力結果フォーマット

指定された文字列 変数式 1 = 値 1, 変数式 2 = 値 2, 変数式 3 = 値 3, ...	
指定された文字列	[出力文字列] で指定した文字列
変数式 1 ~ 10	[変数式] で指定した文字列
値 1 ~ 10	“変数式 1 ~ 10” に対する変数値 値は変数の型に応じた表示形式（「 表 A—17 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）。

[機能ボタン]

ボタン	機能
OK	Printf イベントの設定を終了し、ここで指定した Printf イベントを エディタパネル / 逆アセンブルパネル 上のキャレット位置の行/アドレスに設定します。
キャンセル	Printf イベントの設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

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—62 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ドロップダウン・リストにより選択します。

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。

[機能ボタン]

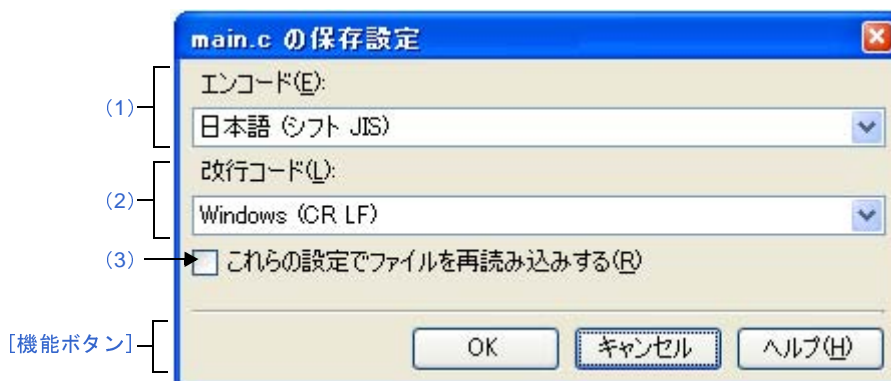
ボタン	機能
OK	指定したファイル・エンコードを使用し、 ファイルを開く ダイアログ で選択したファイルをオープンします。
キャンセル	ファイルを開く ダイアログ で選択したファイルをオープンせずに、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

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—63 ファイルの保存設定 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]

- 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]

<input checked="" type="checkbox"/>	[OK] ボタンをクリックした際に、指定したエンコード、および改行コードでファイルの再読み込みを行います。
<input type="checkbox"/>	[OK] ボタンをクリックした際に、ファイルの再読み込みを行いません (デフォルト)。

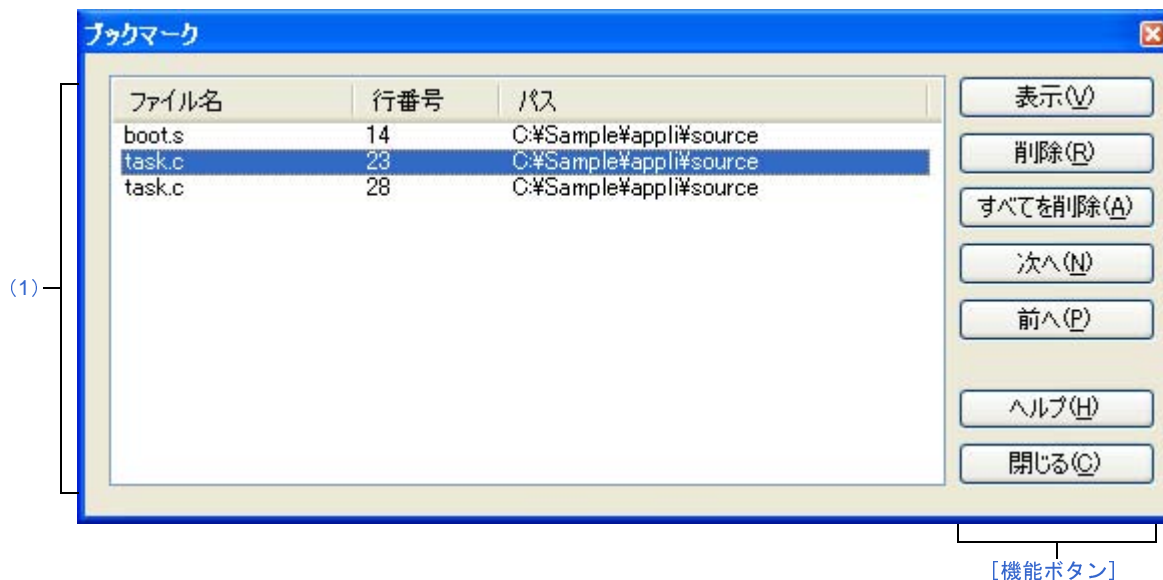
[機能ボタン]

ボタン	機能
OK	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 ダイアログ

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—64 ブックマーク ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- [編集] メニュー → [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。

注意 エディタ パネルをクローズした場合、クローズしたエディタ パネルに登録されていたブックマークは削除されます。

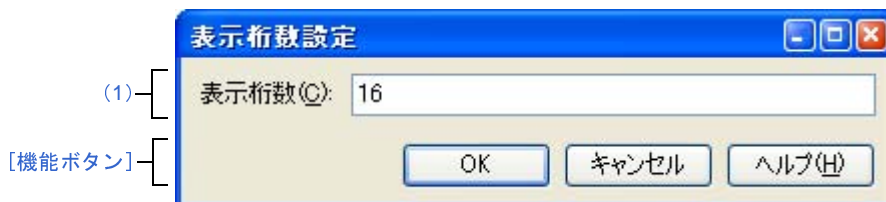
[機能ボタン]

ボタン	機能
表示	選択しているブックマーク位置に、caret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。
すべてを削除	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。
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、caretを移動します。 以下の場合、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
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、caretを移動します。 以下の場合、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
ヘルプ	このダイアログのヘルプを表示します。
閉じる	本ダイアログをクローズします。

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—65 表示桁数設定 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- メモリパネルにおいて、コンテキスト・メニューの [表示] → [表示桁数を設定 ...] を選択

[各エリアの説明]

(1) [表示桁数] エリア

表示する桁数を 10 進数で直接入力により指定します。

指定可能な値の範囲は、現在のメモリパネルにおける [サイズ表記] の設定により、次のように異なります。

サイズ表記	指定可能な範囲
4 ビット	2 ~ 512 ^注
1 バイト	1 ~ 256
2 バイト	1 ~ 128
4 バイト	1 ~ 64
8 バイト	1 ~ 32

注 偶数値でのみ指定できます (奇数値が入力された場合、1つ大きな値に変更されます)。

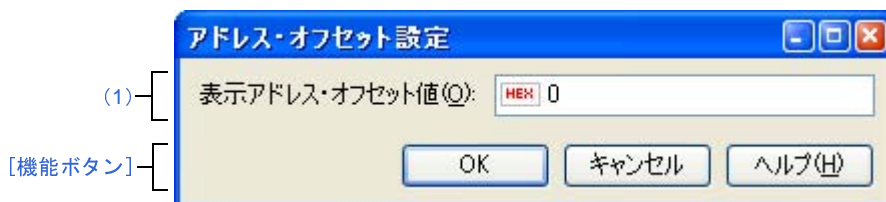
[機能ボタン]

ボタン	機能
OK	指定した桁数でメモリ値を表示します。
キャンセル	設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

アドレス・オフセット設定 ダイアログ

メモリパネルのアドレス・エリアにおいて、開始アドレスのオフセット値を設定します。

図 A—66 アドレス・オフセット設定 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- メモリパネルにおいて、コンテキスト・メニューの [表示] → [表示アドレス・オフセット値を設定 ...] を選択

[各エリアの説明]

(1) [表示アドレス・オフセット値] エリア

アドレス表示のオフセット値を 16 進数で直接入力により指定します。

指定可能な値の範囲は、現在のメモリパネルにおいて 1 行に表示されているメモリのバイト数により、次のように異なります。

指定可能な範囲： $0x0 \sim ([サイズ表記] の設定 \times 表示桁数) - 1$

例 [サイズ表記] : 1 バイト / 表示桁数 : 16 桁の場合

オフセット値	アドレス・エリアの表示内容
0x0 (デフォルト)	0000
	0010
	0020
0x1	0001
	0011
	0021
0x2	0002
	0012
	0022

[機能ボタン]

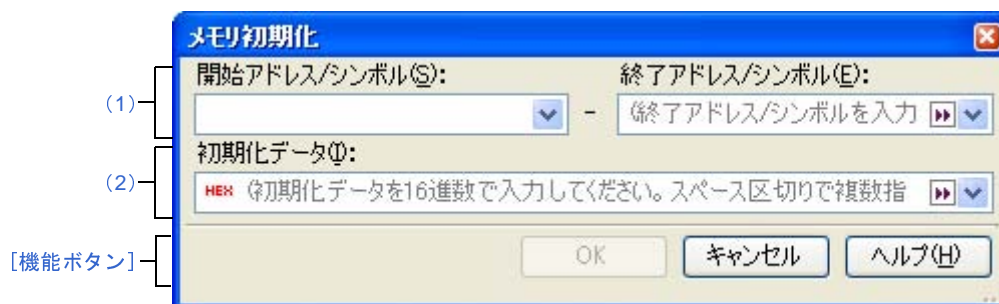
ボタン	機能
OK	指定したオフセット値でメモリのアドレス表示を行います。
キャンセル	設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

メモリ初期化 ダイアログ

メモリ値の初期化を行います（「(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。

注意 メモリの初期化は、データフラッシュ・メモリ用 ID タグの値に対して行うことはできません。

図 A—67 メモリ初期化 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- [メモリ パネル](#)において、コンテキスト・メニュー→ [初期化 ...] を選択

[各エリアの説明]

(1) 範囲指定エリア

メモリ値を初期化するアドレス範囲を [開始アドレス/シンボル] と [終了アドレス/シンボル] に指定します。それぞれのテキスト・ボックスにアドレス式を直接入力するか（最大指定文字数：1024 文字）、またはドロップダウン・リストにより入力履歴項目（最大履歴個数：10 個）を選択します。

入力したアドレス式の計算結果を、それぞれ開始アドレス/終了アドレスとして扱います。

なお、マイクロコントローラのアドレス空間よりも大きいアドレス値を指定することはできません。

注意 エンディアンの異なる領域をまたいだアドレス範囲を指定することはできません。

備考 このテキスト・ボックスで [Ctrl] + [Space] キーを押下することにより、現在のcaret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 (初期化データ)	書き込みイメージ (バイト単位)	
	リトル・エンディアン	ビッグ・エンディアン
1	01	01
0 12	00 12	00 12
00 012 345	00 12 00 45 03	00 00 12 03 45
000 12 000345	00 00 12 45 03 00	00 00 12 00 03 45

[機能ボタン]

ボタン	機能
OK	指定したアドレス範囲のメモリ領域に、指定した初期化データのパターンを繰り返し書き込みます（パターンの途中で終了アドレスに達した場合は書き込みを終了します）。
キャンセル	メモリ値の初期化の設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

メモリ検索 ダイアログ

メモリ値の検索を行います（「(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caret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、およびデータフラッシュ・メモリ用 ID タグの値などは検索対象となりません）。

図 A—68 メモリ検索 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- メモリパネルにおいて、コンテキスト・メニュー→ [検索 ...] を選択

[各エリアの説明]

(1) [検索するデータ] エリア

検索するデータを指定します。

テキスト・ボックスに直接入力するか（最大指定バイト数：256 バイト）、またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。

検索の対象がメモリパネル上のメモリ値エリアの場合、そのエリアと同じ表示形式（表示進数／サイズ）でデータを入力する必要があります。

また、検索の対象が文字列エリアの場合では、検索するデータとして、文字列を指定する必要があります。指定した文字列は、そのエリアで表示しているエンコード形式でデータに変換され検索されます。

なお、このダイアログをオープンする直前にメモリ値を選択していた場合は、デフォルトでその値が表示されます。

(2) [検索する範囲] エリア

検索する範囲を次のドロップダウン・リストより選択します。

アドレス範囲を指定する	[アドレス] エリアで指定したアドレス範囲内で検索を行います。
メモリ・マッピング	<p>選択したメモリ・マッピング範囲内で検索を行います。</p> <p>このリスト項目は、メモリ・マッピングダイアログで表示しているメモリ・マッピングを個々に表示します（ノン・マップ領域を除く）。</p> <p>表示形式：<メモリ種別> <アドレス範囲> <サイズ></p>

(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。

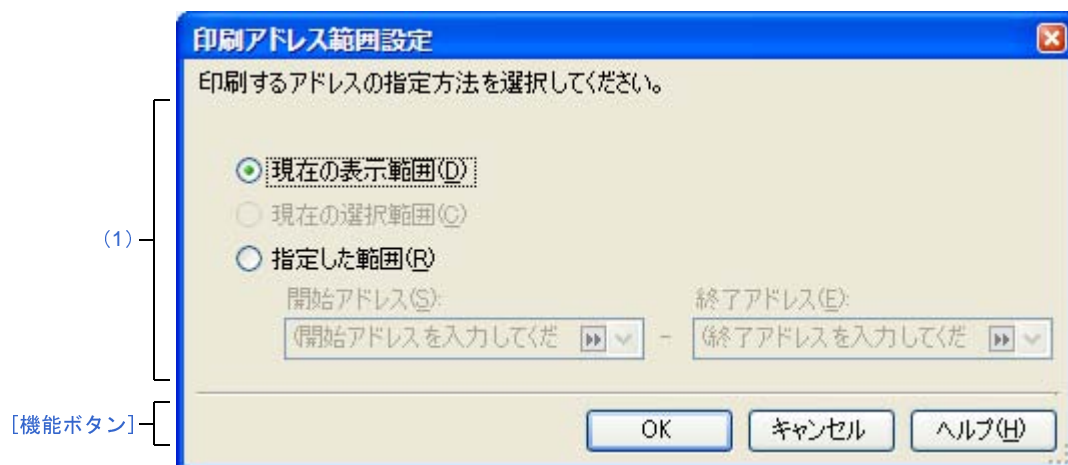
[機能ボタン]

ボタン	機能
前を検索	<p>[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。</p> <p>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。</p> <p>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。</p>
次を検索	<p>[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。</p> <p>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。</p> <p>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。</p>
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—69 印刷アドレス範囲設定 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- 逆アセンブルパネルにおいて、[ファイル]メニュー→[印刷...]を選択

[各エリアの説明]

(1) 範囲指定エリア

印刷する範囲を指定するために、次のオプション・ボタンのいずれか1つを選択します。

(a) [現在の表示範囲] (デフォルト)

逆アセンブルパネルで現在表示している範囲のみを印刷します。

(b) [現在の選択範囲]

逆アセンブルパネルで現在選択している範囲のみを印刷します。

ただし、逆アセンブルパネルにおいて、何も選択していない場合は無効となります。

(c) [指定した範囲]

印刷の対象となるアドレス範囲を [開始アドレス] と [終了アドレス] で指定します。

それぞれのテキスト・ボックスにアドレス式を直接入力するか (最大指定文字数: 1024 文字), またはドロップダウン・リストにより入力履歴項目 (最大履歴個数: 10 個) を選択します。

備考 このテキスト・ボックスで [Ctrl] + [Space] キーを押下することにより, 現在のキャレット位置のシンボル名を補完することができます (「[2.18.2 シンボル名の入力補完機能](#)」参照)。

[機能ボタン]

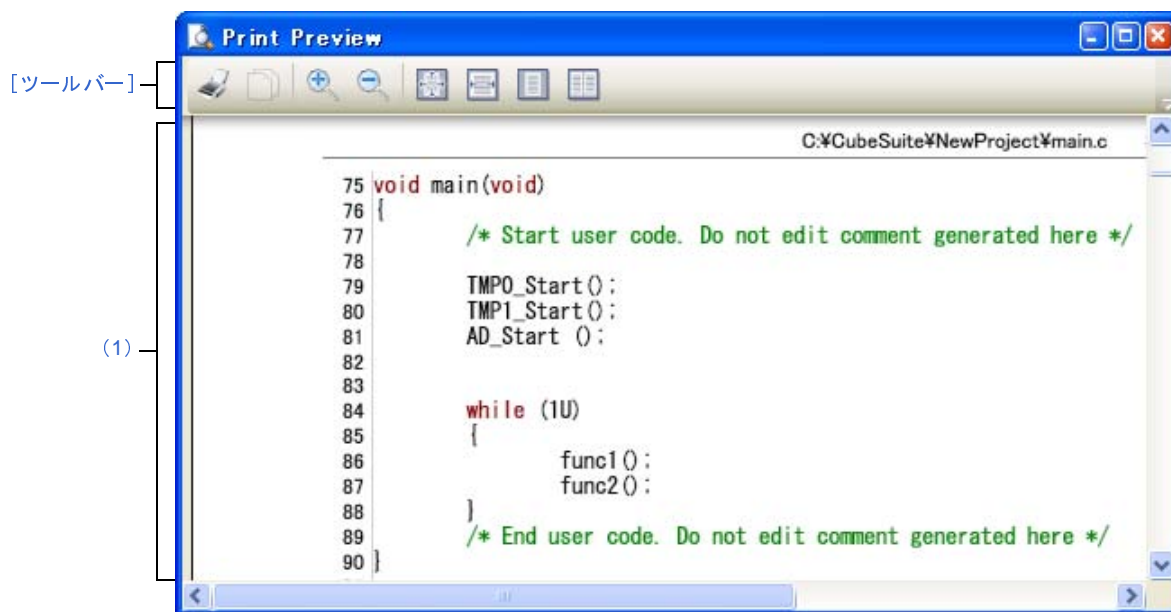
ボタン	機能
OK	指定した範囲で逆アセンブル パネルの内容を印刷するために, このダイアログをクローズして Windows の印刷用 ダイアログをオープンします。
キャンセル	範囲選択の設定を無視し, このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

Print Preview ウィンドウ

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、本ウィンドウの表示を拡大／縮小することができます。

図 A—70 Print Preview ウィンドウ



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／フッタ部に、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）。

なお、アウトライン設定をしている場合、折りたたんだ状態を表示するマーク（[「\(a\) コードのアウトライン表示」](#)参照）とともに、たたまれている行の内容も表示します。

[ツールバー]

	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。
	表示サイズを拡大します。
	表示サイズを縮小します。
	100% の倍率で表示します（デフォルト）。
	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【IECUBE】【IECUBE2】【シミュレータ】

トレース・データの検索を行います（「2.11.8 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-71 トレース検索 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- トレース パネル【IECUBE】【IECUBE2】【シミュレータ】において、ツールバーの  ボタンをクリック
- トレース パネル【IECUBE】【IECUBE2】【シミュレータ】において、コンテキスト・メニュー→ [検索 ...] を選択

[各エリアの説明]

(1) タブ選択エリア

タブを選択することにより、検索するレベルが切り替わります。

このダイアログには、次のタブが存在します。

- [命令レベル] タブ
- [ソース・レベル] タブ

(2) 検索条件設定エリア

検索する際の詳細条件を設定します。

表示内容／設定方法についての詳細は、該当するタブの項を参照してください。

[機能ボタン]

ボタン	機能
前を検索	<p>指定した範囲内で、番号の小さい方向に検索を行います。</p> <p>検索結果箇所をトレースパネル【IECUBE】【IECUBE2】【シミュレータ】上で選択状態にします。</p> <p>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。</p> <p>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。</p>
次を検索	<p>指定した範囲内で、番号の大きい方向に検索を行います。</p> <p>検索結果箇所をトレースパネル【IECUBE】【IECUBE2】【シミュレータ】上で選択状態にします。</p> <p>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。</p> <p>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。</p>
キャンセル	<p>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。</p>
ヘルプ	<p>このダイアログのヘルプを表示します。</p>

[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【IECUBE】【IECUBE2】【シミュレータ】をソース表示モードで表示している場合、このタブで命令レベルの検索を行っても対象を正しく検索することはできません。**
命令レベルの検索を行う際は、**混合表示モード**、または**逆アセンブル表示モード**で表示を行ってください。


図 A—72 トレース検索 ダイアログ：[命令レベル] タブ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- トレースパネル【IECUBE】【IECUBE2】【シミュレータ】において、ツールバーの  ボタンをクリック
- トレースパネル【IECUBE】【IECUBE2】【シミュレータ】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【IECUBE】【IECUBE2】【シミュレータ】の【ソース/逆アセンブル】](#) エリア内より検索します。

命令をテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。

なお、検索の際は、大文字/小文字は区別せず、部分一致も検索の対象とします。

(c) [アクセス・アドレス]

検索条件として必要な場合、アクセス・アドレスを指定します。

アドレス式をテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。

アクセス・アドレスの指定は範囲で指定することができます。この場合は、左右両方のテキスト・ボックスにアドレス式を指定することにより範囲を指定します。

右側のテキスト・ボックスが空欄、または“(範囲を指定する場合に入力)”の場合は、左側のテキスト・ボックスに指定された固定アドレスで検索を行います。

なお、マイクロコントローラのアドレス空間よりも大きいアドレス値が指定された場合は、上位のアドレス値をマスクして扱います。

また、32 ビットで表現できる値より大きいアドレス値を指定することはできません。

(d) [アクセスの種類]

この項目は [アクセス・アドレス] が指定された場合のみ有効となります。

アクセスの種類を次のドロップダウン・リストより選択します。

アクセスの種類を限定しない場合は、“(指定なし)” を選択してください。

(指定なし)
リード/ライト
リード
ライト
ベクタ・リード
DMA

(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** [IECUBE] [IECUBE2] [シミュレータ] の [番号] エリアに表示されている番号で指定します。

左右のテキスト・ボックスに、それぞれ開始番号と終了番号を指定します (デフォルトでは、“0” ~ “最終番号” が指定されます)。

10 進数で番号をテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目を選択します (最大履歴数 : 10 個)。

左側のテキスト・ボックスが空欄の場合は、“0” の指定として扱われます。

右側のテキスト・ボックスが空欄の場合は、最終番号の指定として扱われます。

[機能ボタン]

ボタン	機能
前を検索	<p>指定した範囲内で、番号の小さい方向に検索を行います。</p> <p>検索結果箇所を トレース パネル【IECUBE】【IECUBE2】【シミュレータ】 上で選択状態にします。</p> <p>ただし、不正な値を指定している場合はメッセージを表示し、トレース・データの検索は行いません。また、トレース パネルが非表示の場合、または他のパネルにフォーカスがある状態からこのダイアログへフォーカスを移動した場合、このボタンは無効となります。</p>
次を検索	<p>指定した範囲内で、番号の大きい方向に検索を行います。</p> <p>検索結果箇所を トレース パネル【IECUBE】【IECUBE2】【シミュレータ】 上で選択状態にします。</p> <p>ただし、不正な値を指定している場合はメッセージを表示し、トレース・データの検索は行いません。また、トレース パネルが非表示の場合、または他のパネルにフォーカスがある状態からこのダイアログへフォーカスを移動した場合、このボタンは無効となります。</p>
キャンセル	トレース・データの検索の設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

[ソース・レベル] タブ

取得したトレース・データをソース・レベルで検索します。

注意 **トレースパネル【IECUBE】【IECUBE2】【シミュレータ】**を**逆アセンブル表示モード**で表示している場合、このタブでソース・レベルの検索を行っても対象を正しく検索することはできません。
ソース・レベルの検索を行う際は、**混合表示モード**、または**ソース表示モード**で表示を行ってください。


図 A—73 トレース検索 ダイアログ : [ソース・レベル] タブ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- トレースパネル【IECUBE】【IECUBE2】【シミュレータ】において、ツールバーの  ボタンをクリック
- トレースパネル【IECUBE】【IECUBE2】【シミュレータ】において、コンテキスト・メニュー→ [検索 ...] を選択

[各エリアの説明]

(1) [検索対象の指定] エリア

検索する対象を次のオプション・ボタンの中から選択します。

ソース行を指定して実行箇所を検索	指定したソースの実行箇所を検索します（デフォルト）。 検索条件として [ソース行] の指定のみが有効となります。
関数名を指定して先頭アドレスの実行箇所を検索	指定した関数の実行箇所を検索します。 検索条件として [関数名] の指定のみが有効となります。
グローバル変数名を指定してアクセス箇所を検索	指定したグローバル変数をアクセスした箇所を検索します。 検索条件として [変数名] / [種類] / [変数値] の指定のみが有効となります。

(2) [検索条件の指定] エリア

(a) [ソース行]

この項目は“ソース行を指定して実行箇所を検索”が選択された場合のみ有効となります。

ここで指定した文字列を **トレースパネル [IECUBE] [IECUBE2] [シミュレータ]** 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) [変数値]

この項目は“[グローバル変数名を指定してアクセス箇所を検索](#)”が選択された場合のみ有効となります。

アクセスした変数値を 16 進数で指定します。

変数値をテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。

変数値の指定は範囲で指定することができます。この場合は、左右両方のテキスト・ボックスに変数値を指定することにより範囲を指定します。

右側のテキスト・ボックスが空欄の場合は、左側のテキスト・ボックスに指定された固定変数値でアクセス箇所を検索を行います。

(3) [検索範囲の指定] エリア

(a) [番号]

検索するトレース・データの範囲を、[トレースパネル【IECUBE】【IECUBE2】【シミュレータ】](#)の[\[番号\] エリア](#)に表示されている番号で指定します。

左右のテキスト・ボックスに、それぞれ開始番号と終了番号を指定します（デフォルトでは、“0”～“最終番号”が指定されます）。

10 進数で番号をテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。

左側のテキスト・ボックスが空欄の場合は、“0”の指定として扱われます。

右側のテキスト・ボックスが空欄の場合は、最終番号の指定として扱われます。

[機能ボタン]

ボタン	機能
前を検索	指定した範囲内で、番号の小さい方向に検索を行います。 検索結果箇所を トレースパネル【IECUBE】【IECUBE2】【シミュレータ】 上で選択状態にします。 ただし、不正な値を指定している場合メッセージを表示し、トレース・データの検索は行いません。また、トレースパネルが非表示の場合、または他のパネルにフォーカスがある状態からこのダイアログへフォーカスを移動した場合、このボタンは無効となります。
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【IECUBE】【IECUBE2】【シミュレータ】 上で選択状態にします。 ただし、不正な値を指定している場合メッセージを表示し、トレース・データの検索は行いません。また、トレースパネルが非表示の場合、または他のパネルにフォーカスがある状態からこのダイアログへフォーカスを移動した場合、このボタンは無効となります。
キャンセル	トレース・データの検索の設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

スクロール範囲設定 ダイアログ

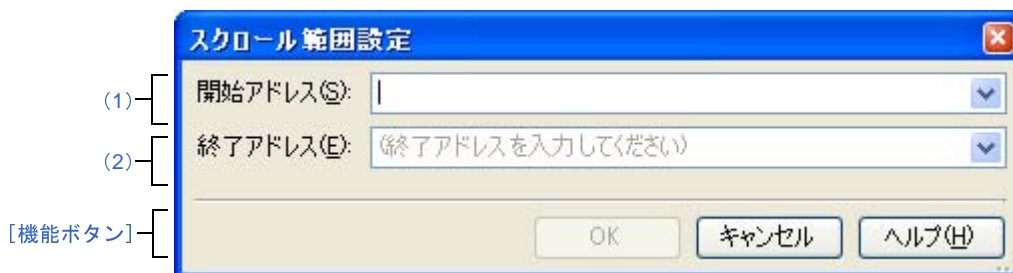
メモリ パネル／逆アセンブル パネルの垂直スクロール・バーのスクロール範囲の設定を行います。

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が表すアドレスに変更が生じても、スクロール範囲の修正は行いません。

備考 [Page Up] / [Page Down] / [↑] / [↓] キー、スクロール・バー端のボタン、またはジャンプ系のメニュー項目の選択による移動は、スクロール範囲外でも可能です。



図 A—74 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）。

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）。

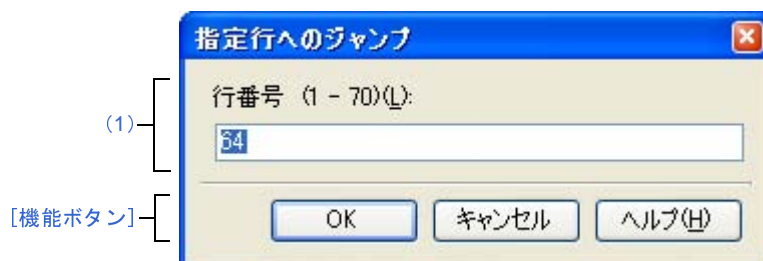
[機能ボタン]

ボタン	機能
OK	指定したスクロール範囲を対象パネルに設定し、開始アドレスを表示の先頭として対象パネルにキャレットを移動します。
キャンセル	設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

指定行へのジャンプ ダイアログ

指定したソース行にカーレットを移動します。

図 A—75 指定行へのジャンプ 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]

- [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)] エリア

“(有効な行の範囲)”に、現在のファイルの有効な行の範囲が表示されます。

カーレットを移動したい行番号を10進数で直接入力により指定します。

または、シンボルを入力することも可能です。

デフォルトでは、**エディタ パネル**上の現在のカーレット位置の行番号が表示されます。

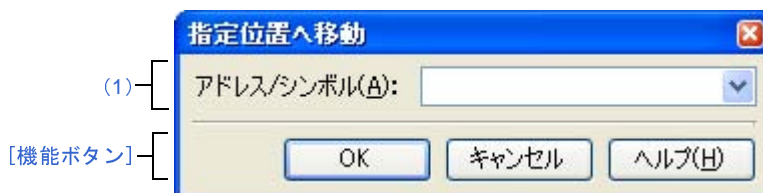
[機能ボタン]

ボタン	機能
OK	指定したソース行の先頭にカー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。

指定位置へ移動 ダイアログ

指定した位置にカーレットを移動します。

図 A—76 指定位置へ移動 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- [編集] メニュー→ [移動...] を選択
- 逆アセンブルパネルにおいて、コンテキスト・メニューの [移動...] を選択
- IORパネルにおいて、コンテキスト・メニューの [移動...] を選択

[各エリアの説明]

(1) [アドレス/シンボル] / [IOR] エリア

カー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、現在のカー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）。

[機能ボタン]

ボタン	機能
OK	指定した位置を表示の先頭として対象パネルにキャレットを移動します。
キャンセル	設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

データ保存 ダイアログ

逆アセンブルパネル／メモリパネル／トレースパネル【IECUBE】【IECUBE2】【シミュレータ】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—77 データ保存 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- 逆アセンブルパネルにフォーカスがある状態で、[ファイル]メニュー→[名前を付けて逆アセンブル・データを保存...]を選択
- メモリパネルにフォーカスがある状態で、[ファイル]メニュー→[名前を付けてメモリ・データを保存...]を選択
- トレースパネル【IECUBE】【IECUBE2】【シミュレータ】にフォーカスがある状態で、[ファイル]メニュー→[名前を付けてトレース・データを保存...]を選択
- [デバッグ]メニュー→[デバッグ・ツールからアップロード...]を選択

[各エリアの説明]

(1) [ファイル名] エリア

保存するファイル名を指定します。

テキスト・ボックスに直接入力するか（最大指定文字数：259文字）、またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10個）。

また、[...] ボタンをクリックすることでオープンするデータ保存ファイルを選択ダイアログにより、ファイルを選択することもできます。

なお、パス情報を含まずファイル名のみを指定した場合は、プロジェクト・フォルダが対象となります。

(2) [ファイルの種類] エリア

保存するファイルの形式を次のドロップダウン・リストにより選択します。

保存する対象により、選択できるファイルの形式が次のように異なります。

(a) パネルの表示内容を保存する場合

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

注 各データを“,”で区切り保存します。

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

(b) アップロード・データを保存する場合

選択できるファイル形式は、「表 2—5 アップロード可能なファイル形式」を参照してください。

(3) [保存範囲 xxx] エリア

ファイルに保存する際の保存範囲を指定します。

それぞれのテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目を選択します (最大履歴数: 10 個)。

保存する対象により、指定方法が次のように異なります。

保存対象	説明
逆アセンブル パネル	保存するアドレス範囲を、開始アドレスと終了アドレスで指定します。 16 進数の数値、またはアドレス式による入力が可能です。 パネル上で範囲選択している場合は、デフォルトでその選択範囲が指定されます。 範囲選択していない場合は、現在のパネルの表示範囲が指定されます。
メモリ パネル	保存するメモリ範囲を、開始アドレスと終了アドレスで指定します。 16 進数の数値、またはアドレス式による入力が可能です。 範囲選択していない場合は、現在のパネルの表示範囲が指定されます。
トレース パネル 【IECUBE】【IECUBE2】 【シミュレータ】	- 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。

保存対象	説明
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caret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）。

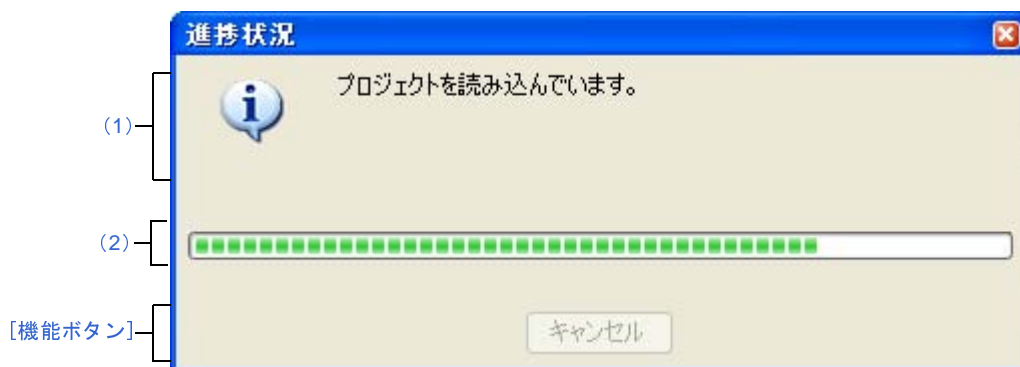
[機能ボタン]

ボタン	機能
保存	指定したファイルに、指定した形式でデータを保存します。
キャンセル	データ保存の設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

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—78 処理中表示 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]

- 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。

[機能ボタン]

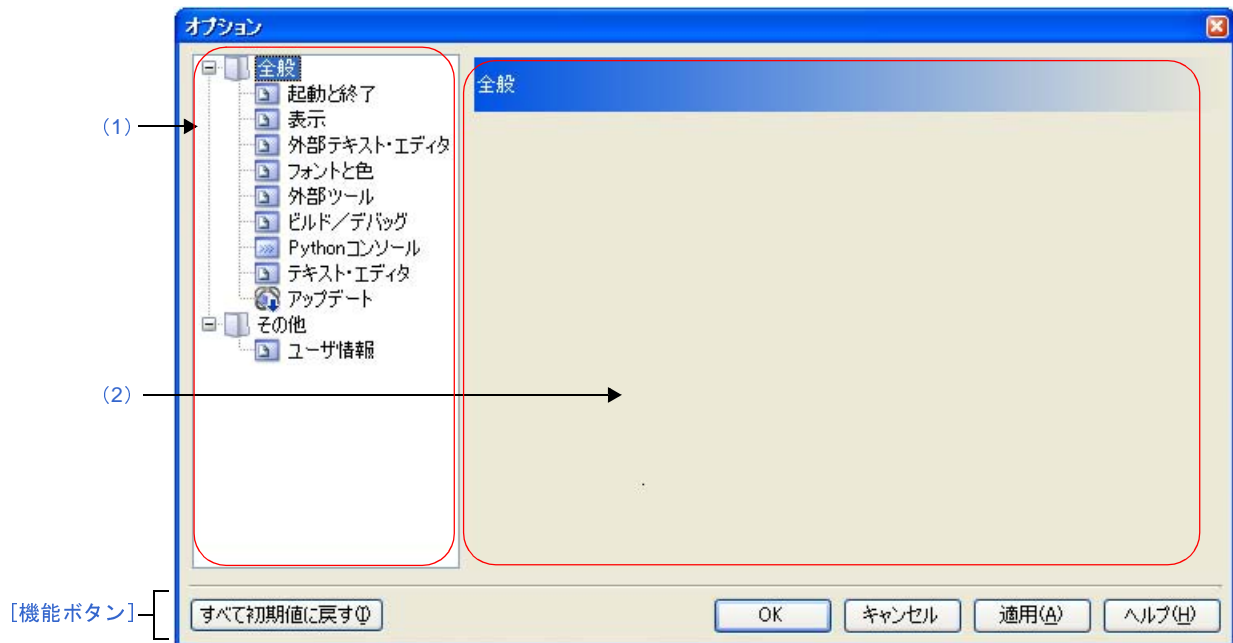
ボタン	機能
キャンセル	現在実行中の処理を中断し、このダイアログをクローズします。 ただし、実行中の処理の中断が不可能な場合、このボタンは無効となります。

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。

図 A—79 オプション 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]

- [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。
[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。

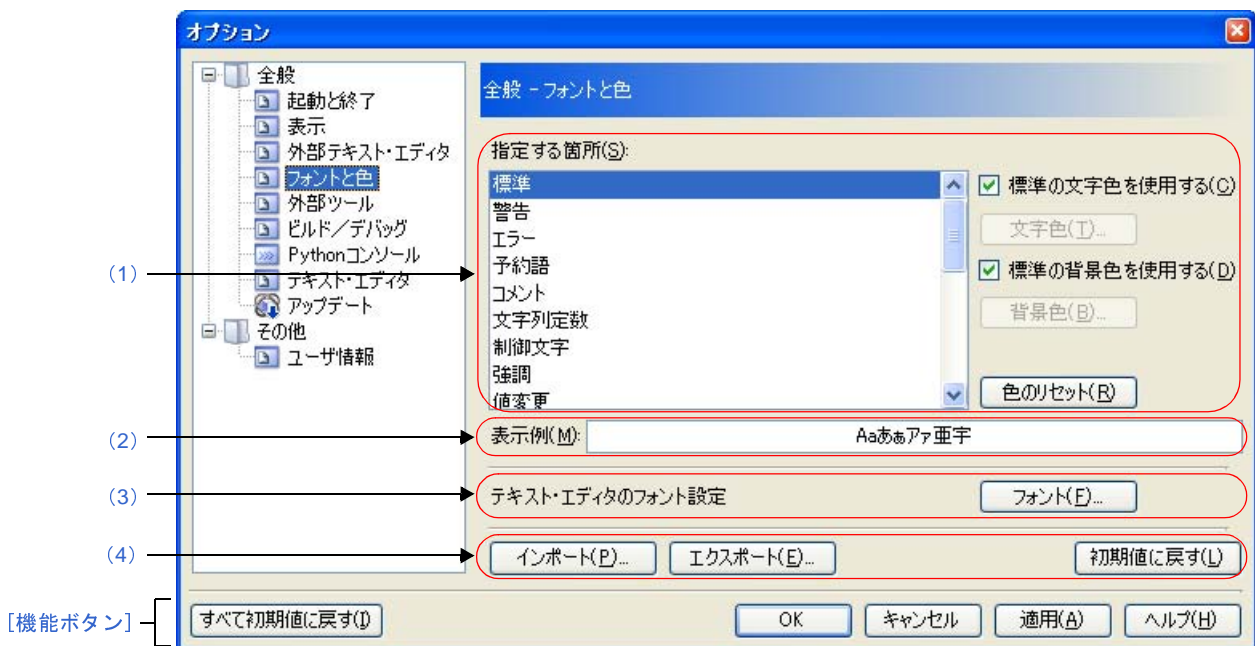
[機能ボタン]

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。
OK	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。

図 A—80 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]
- [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) 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一覧から選択します。

表示されるリスト一覧の項目と、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。

項目	表示例		説明	
標準 ^注	AaBbCc	文字色	黒	すべてのウィンドウ／パネルにおいて、標準となる表示色
		背景色	白	
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
		背景色	標準色	
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、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
		背景色	薄グレー	
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／アセンブラの予約語の表示色
		背景色	標準色	
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・ファイルの場合、“/*～*/”）の表示色
		背景色	標準色	
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	
値変更	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
		背景色	クリーム	
値編集	AaBbCc	文字色	青	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 / ウォッチパネルにおいて、ユーザが強制的に値を変更した箇所の表示色
		背景色	標準色	
PC位置	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、カレントPC位置のある行の表示色
		背景色	山吹	
ブレイクポイント	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、ブレイクポイントが設定されている行の表示色
		背景色	サーモンピンク	
リアルタイム更新中	AaBbCc	文字色	ピンク	メモリパネル / ウォッチパネルにおいて、リアルタイム表示更新に設定されている領域の表示色
		背景色	標準色	
リード／フェッチ	AaBbCc	文字色	標準色	メモリパネル / トレースパネル【IECUBE】【IECUBE2】【シミュレータ】において、リード、またはフェッチされた箇所の表示色
		背景色	薄緑	
ライト	AaBbCc	文字色	標準色	メモリパネル / トレースパネル【IECUBE】【IECUBE2】【シミュレータ】において、ライトされた箇所の表示色
		背景色	オレンジ	
リード＆ライト	AaBbCc	文字色	標準色	メモリパネル / トレースパネル【IECUBE】【IECUBE2】【シミュレータ】において、リードとライトされた箇所の表示色
		背景色	薄青	

項目	表示例		説明	
ロスト		文字色	白	解析ツールの解析グラフパネルにおいて、グラフ・データの取得に失敗した区間の表示色
		背景色	ライトグレー	
カバレッジ 100%		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、コード・カバレッジ率 100% の行の表示色
		背景色	ライトグリーン	
カバレッジ 1 ～ 99%		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、コード・カバレッジ率 1 ～ 99% の行の表示色
		背景色	ライトピンク	
カバレッジ 0%		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、コード・カバレッジ率 0% (未実行) の行の表示色
		背景色	ライトグレー	
無効		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
		背景色	標準色	

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]

<input checked="" type="checkbox"/>	[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。
<input type="checkbox"/>	[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]

<input checked="" type="checkbox"/>	[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。
<input type="checkbox"/>	[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—81 色の設定 ダイアログ



(2) [表示例] エリア

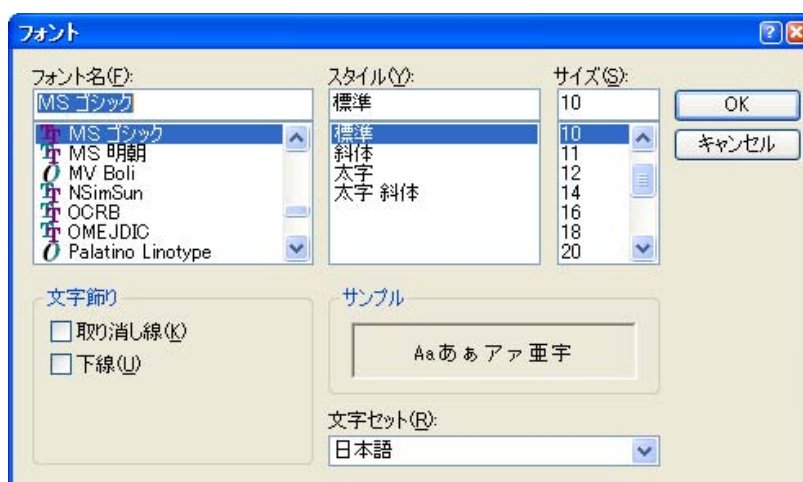
色の設定エリアで指定した色とフォントの表示例を示します。

デフォルトでは、文字列“Aa ああアア亜宇”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。

図 A—82 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。

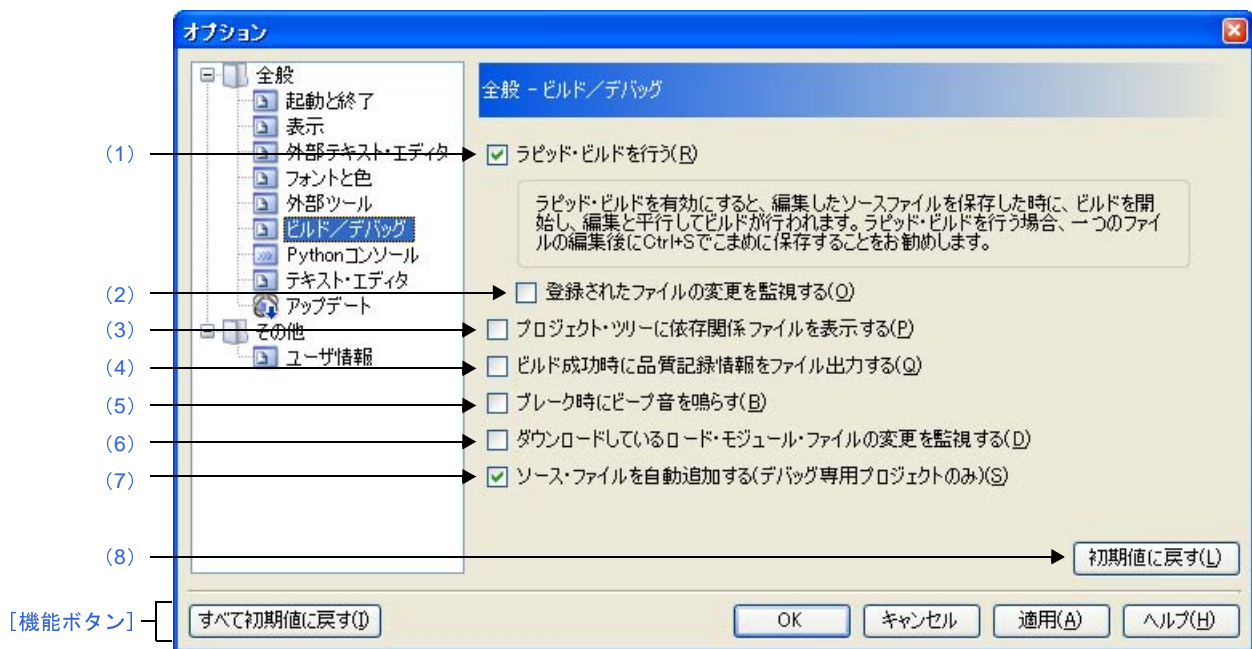
[機能ボタン]

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。
OK	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 - 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。

図 A—83 オプション ダイアログ ([全般 - ビルド／デバッグ] 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]
- [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) [ラピッド・ビルドを行う]

<input checked="" type="checkbox"/>	ラピッド・ビルド機能 ^注 を有効にします (デフォルト)。
<input type="checkbox"/>	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。

(2) [登録されたファイルの変更を監視する]

<input checked="" type="checkbox"/>	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。
<input type="checkbox"/>	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 [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。
2. 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。

(3) [プロジェクト・ツリーに依存関係ファイルを表示する]

<input checked="" type="checkbox"/>	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・ツリーに表示します。
<input type="checkbox"/>	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）。

(4) [ビルド成功時に品質記録情報をファイル出力する]

<input checked="" type="checkbox"/>	ビルド成功時に品質記録情報ファイルを出力します。
<input type="checkbox"/>	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
3. 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。

(5) [ブレーク時にビーブ音を鳴らす]

<input checked="" type="checkbox"/>	プログラムの実行が、ブレーク・イベント（ハードウェア・ブレーク/ソフトウェア・ブレーク）により停止した際、ビーブ音を鳴らします。
<input type="checkbox"/>	プログラムの実行が、ブレーク・イベント（ハードウェア・ブレーク/ソフトウェア・ブレーク）により停止した際、ビーブ音を鳴らしません（デフォルト）。

(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]

<input checked="" type="checkbox"/>	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。
<input type="checkbox"/>	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）]

<input checked="" type="checkbox"/>	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）。
<input type="checkbox"/>	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) 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。
--------	---------------------------

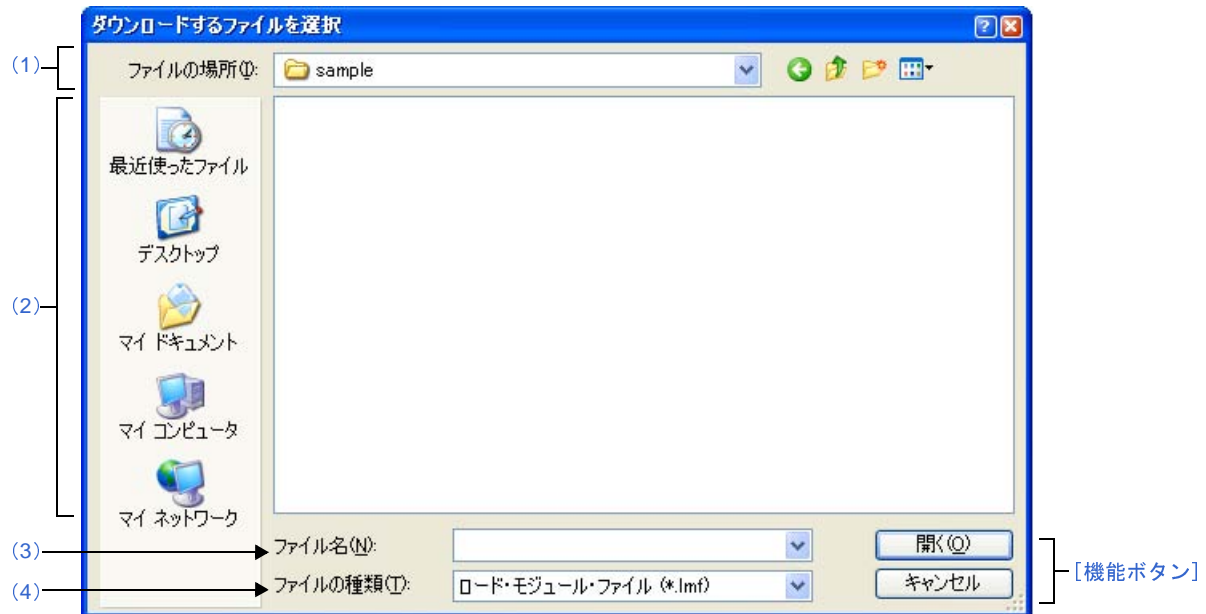
[機能ボタン]

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。
OK	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。

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—84 ダウンロードするファイルを選択 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]

- 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 (*.lmf, *.out)	ロード・モジュール・フォーマット ^{注1} (デフォルト)
ヘキサ・ファイル (*.hex)	ヘキサ・フォーマット ^{注2}
バイナリ・データ・ファイル (*.bin)	バイナリ・データ・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

注1. 【V850E2】

ロード・モジュール・フォーマットとして利用可能な拡張子は“*.lmf”のみです（“*.out”は表示されません）。

2. 【シミュレータ】以外

ID タグ付きフォーマットを使用する場合は、[ダウンロード・ファイル ダイアログ](#)の [ファイルの種類] において [ヘキサ・ファイル [ID タグ付]] を選択してください。

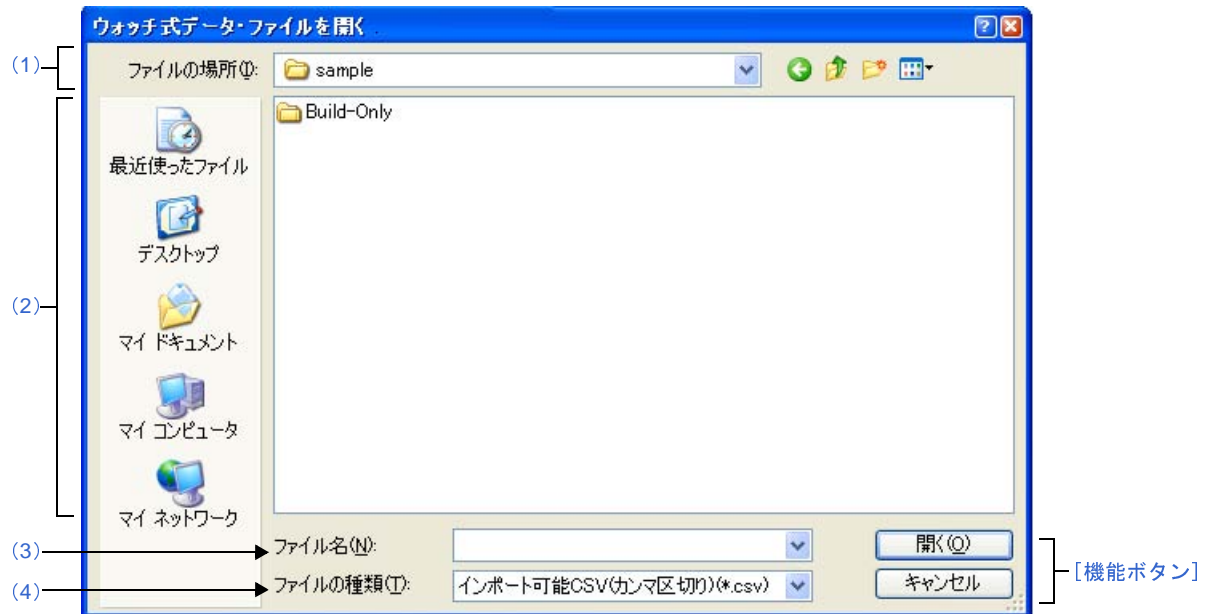
[機能ボタン]

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—85 ウォッチ式データ・ファイルを開く 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]

- 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
-----------------------------	------------------

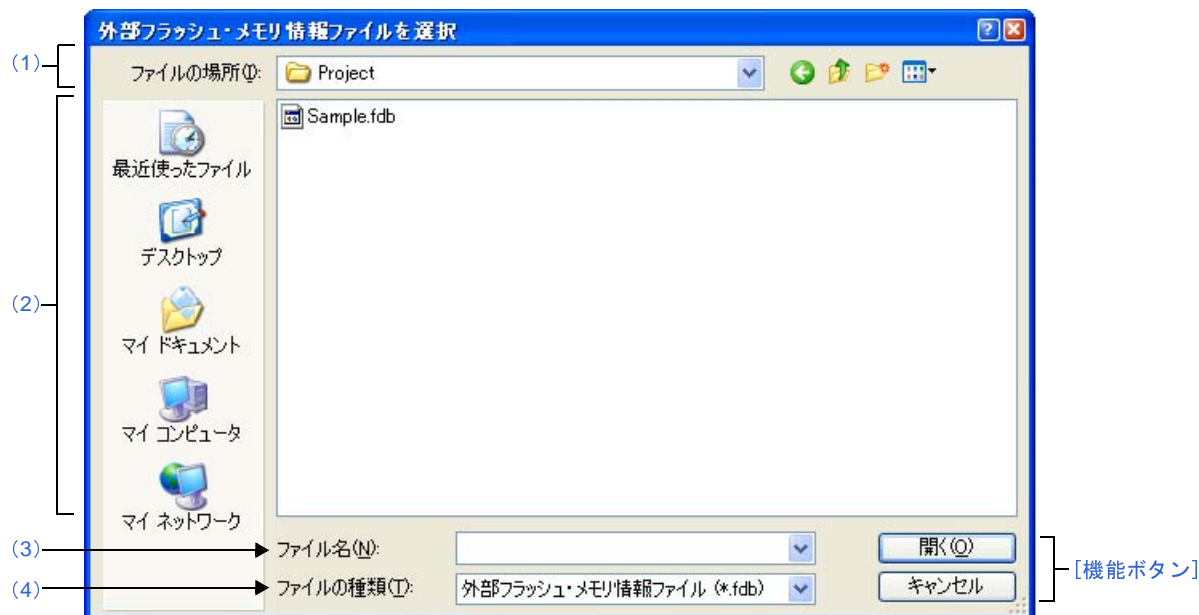
[機能ボタン]

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。

外部フラッシュ・メモリ情報ファイルを選択 ダイアログ

外部フラッシュ・メモリにダウンロードする際の外部フラッシュ・メモリ情報ファイル (*.fdb) の選択を行います。

図 A—86 外部フラッシュ・メモリ情報ファイルを選択 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]

- 外部フラッシュ・メモリ・ダウンロード ダイアログの [外部フラッシュ設定のプロパティ] エリアにおいて、[ファイル名] 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

外部フラッシュ・メモリ情報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

外部フラッシュ・メモリ情報ファイル名を指定します。

(4) [ファイルの種類] エリア

外部フラッシュ・メモリ情報ファイルの種類（ファイル・タイプ）を選択します。

外部フラッシュ・メモリ情報ファイル (*.fdb)	外部フラッシュ・メモリ情報ファイル・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

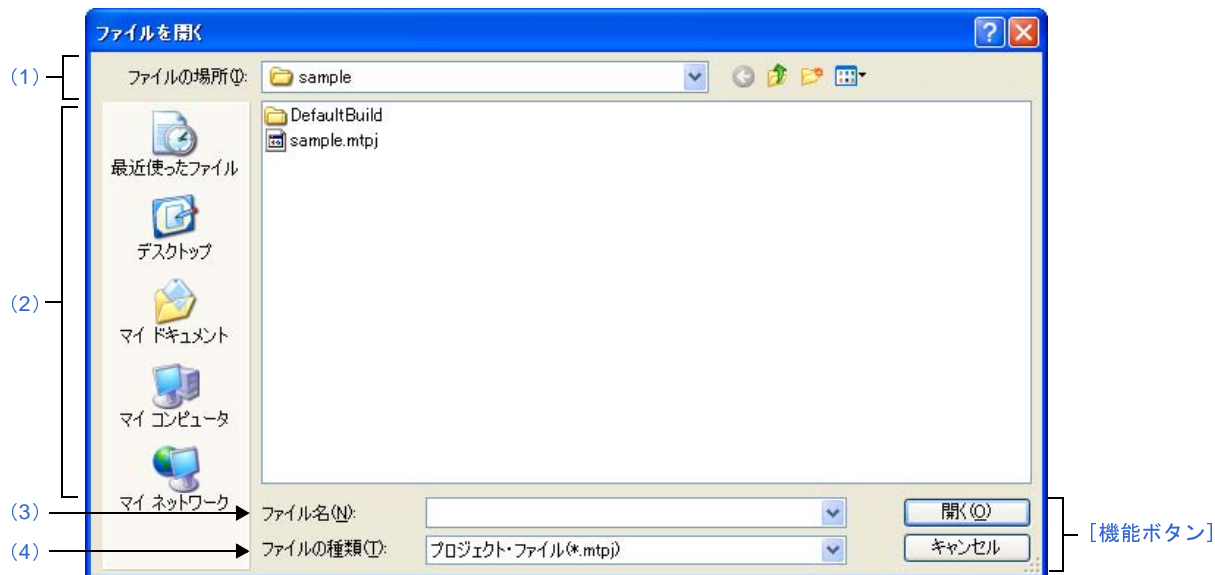
[機能ボタン]

ボタン	機能
開く	指定したファイルを外部フラッシュ・メモリ・ダウンロード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。

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—87 ファイルを開く 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→ [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 (*.mpj)	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
アセンブル・ファイル (*.s) 【CA850】	アセンブラ・ソース・ファイル
アセンブル・ファイル (*.asm; *.s) 【CX】	アセンブラ・ソース・ファイル
リンク・ディレクティブ・ファイル (*.dir; *.dr)	リンク・ディレクティブ・ファイル
セクション・ファイル (*.sf) 【CA850】	セクション・ファイル
シンボル情報ファイル (*.sfg) 【CX】	シンボル情報ファイル
マップ・ファイル (*.map)	マップ・ファイル
ヘキサ・ファイル (*.hex)	ヘキサ・ファイル
Python スクリプト・ファイル (*.py)	Python スクリプト・ファイル
テキスト・ファイル (*.txt)	テキスト形式

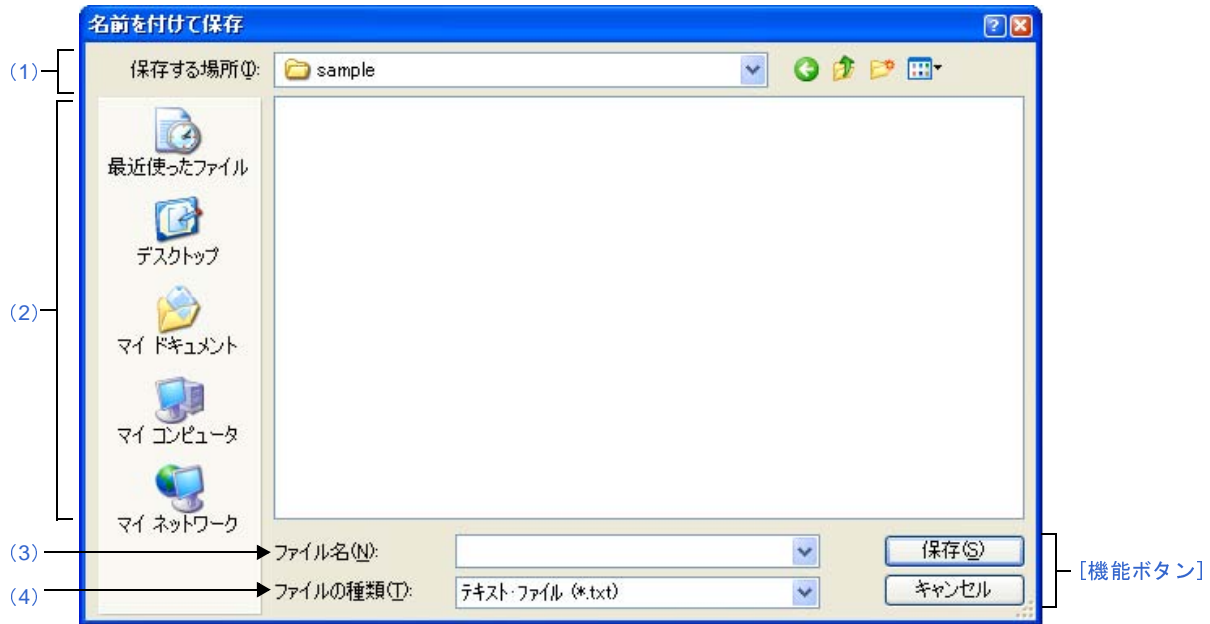
[機能ボタン]

ボタン	機能
開く	<ul style="list-style-type: none"> - [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 指定したファイルをオープンします。 - [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—88 名前を付けて保存 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- 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。

テキスト・ファイル (*.txt)	テキスト形式
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
アセンブル・ファイル (*.asm) 【CX】	アセンブラ・ソース・ファイル
アセンブル・ファイル (*.s)	アセンブラ・ソース・ファイル
リンク・ディレクティブ・ファイル (*.dir;*.dr)	リンク・ディレクティブ・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
セクション・ファイル (*.sf) 【CA850】	セクション・ファイル
シンボル情報ファイル (*.sfg) 【CX】	シンボル情報ファイル
マップ・ファイル (*.map)	マップ・ファイル
ヘキサ・ファイル (*.hex)	ヘキサ・ファイル

(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. このダイアログをウォッチパネルから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）
-------------------	---------------

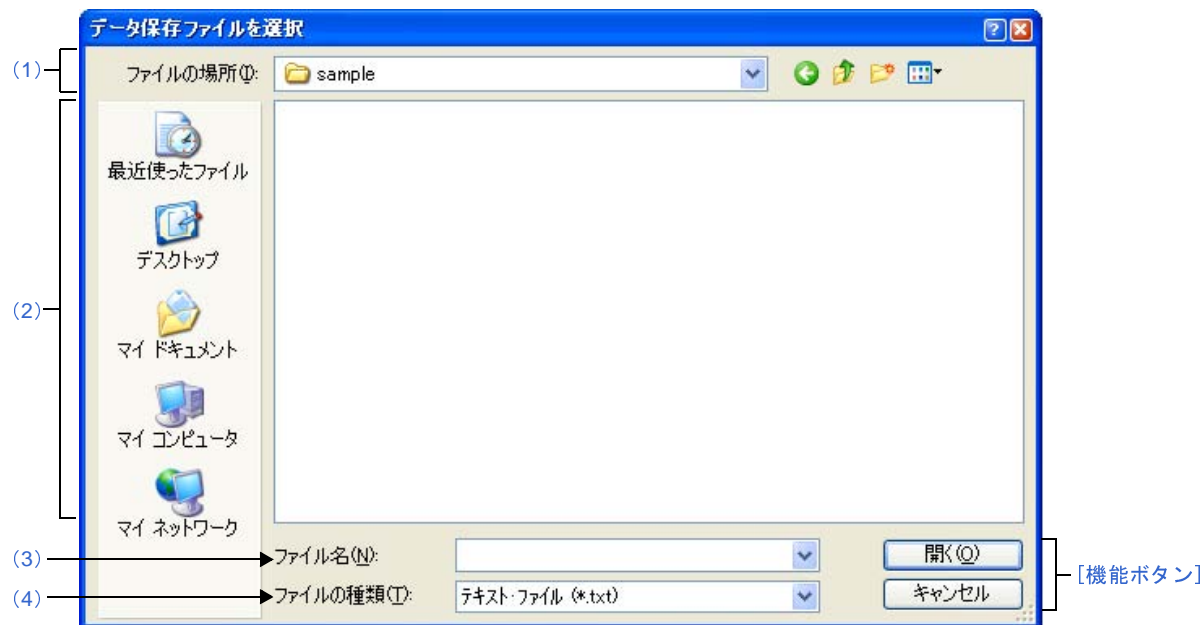
[機能ボタン]

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。

図 A—89 データ保存ファイルを選択 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]

- 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

選択できるファイル形式は、「[表 2—5 アップロード可能なファイル形式](#)」を参照してください。

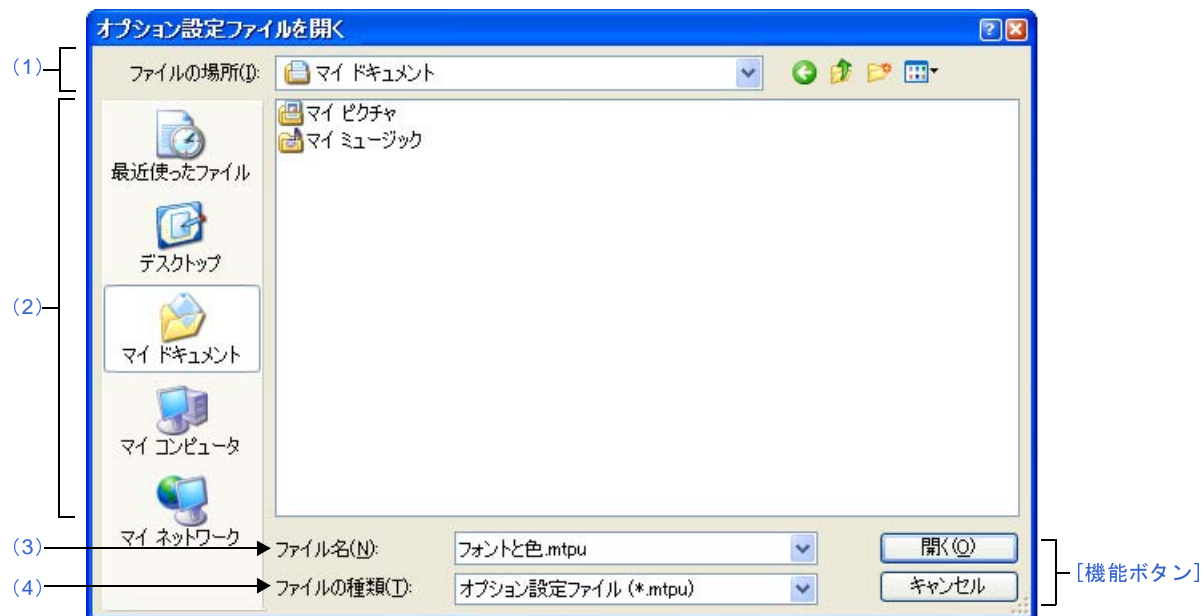
[機能ボタン]

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。

図 A—90 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
----------------------	-------------

[機能ボタン]

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。

図 A—91 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イル
----------------------	-------------

[機能ボタン]

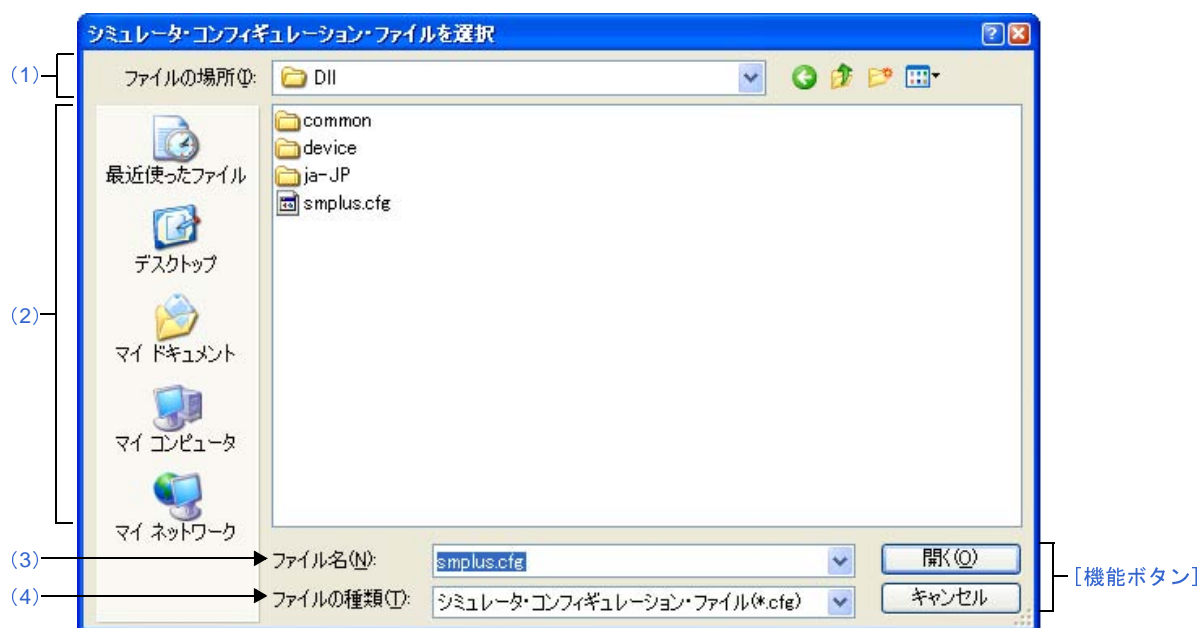
ボタン	機能
保存	指定したファイル名でオプション設定ファイルを保存します。
キャンセル	このダイアログをクローズします。

シミュレータ・コンフィギュレーション・ファイルを選択ダイアログ

【シミュレータ】

シミュレータを使用する場合において、ユーザ・カスタマイズ（ユーザ・モデルの追加）を行うためのシミュレータ・コンフィギュレーション・ファイルの選択を行います。

図 A—92 シミュレータ・コンフィギュレーション・ファイルを選択 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】

- プロパティ パネルの [接続用設定] タブにおいて、[コンフィギュレーション] カテゴリ内 [シミュレータ・コンフィギュレーション・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) [ファイルの場所] エリア

シミュレータ・コンフィギュレーション・ファイルが存在するフォルダをドロップダウン・リストにより選択します。

デフォルトでは空欄です。

(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

使用するシミュレータ・コンフィギュレーション・ファイル名を指定します。

(4) [ファイルの種類] エリア

ファイルの種類 (ファイル・タイプ) を選択します。

ただし、“シミュレータ・コンフィギュレーション・ファイル (*.cfg)” に固定です。

[機能ボタン]

ボタン	機能
開く	指定したシミュレータ・コンフィギュレーション・ファイルを使用します。
キャンセル	このダイアログをクローズします。

シミュレータ GUI ウィンドウ

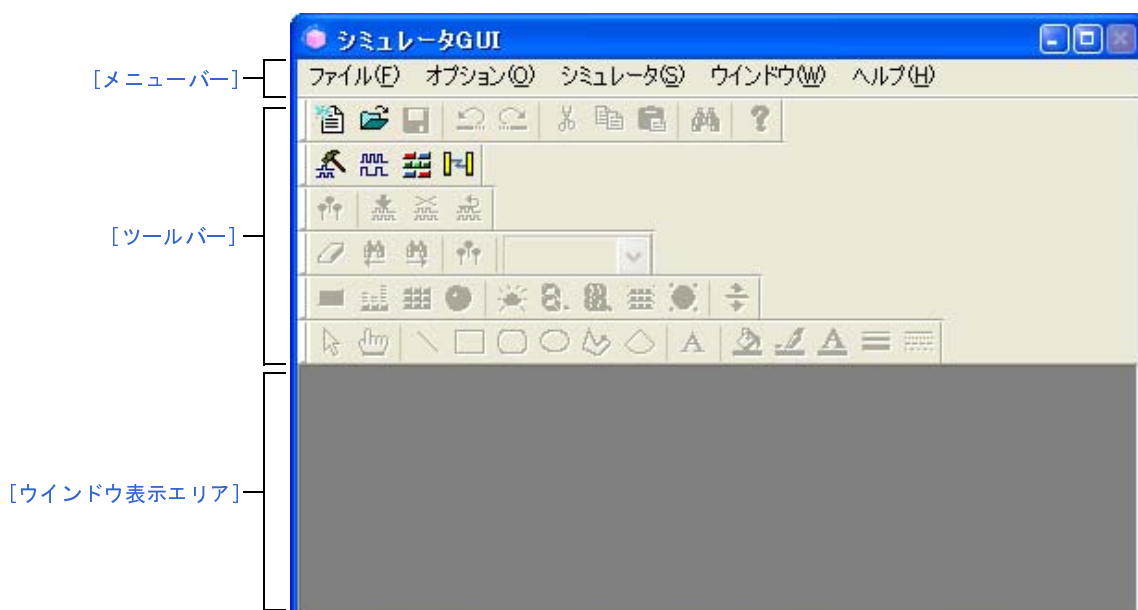
選択しているマイクロコントローラのシミュレータが周辺機器シミュレーションをサポートしている場合で、かつ使用するデバッグ・ツールに“シミュレータ”を選択している場合、デバッグ・ツールと接続すると自動的にオープンするウィンドウです（「2.17 シミュレータ GUI の使用【シミュレータ】」参照）。

シミュレータ GUI では、このウィンドウを中心に、各種ウィンドウ（信号データエディタ ウィンドウ／タイミングチャート ウィンドウ／入出力パネル ウィンドウ／シリアル ウィンドウ）を操作します。

- 注意 1. 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、このウィンドウをオープンすることはできません。
- このウィンドウ、およびこのウィンドウよりオープンする各種ウィンドウは、CubeSuite+ のメイン・ウィンドウとドッキング表示することはできません。
 - このウィンドウよりオープンする各種ウィンドウを何もオープンしていない状態で [F1] キーを押下してもこのウィンドウのヘルプは表示されません。このウィンドウのヘルプを表示する場合は、このウィンドウ上の [ヘルプ] メニュー→ [メイン・ウィンドウ] を選択してください。
 - このウィンドウ右上の [x] ボタンは無効です（Windows Vista の Aero 機能を有効にしている場合も無効）。このウィンドウを非表示にする場合は、プロパティ パネルの設定により行ってください（「2.17 シミュレータ GUI の使用【シミュレータ】」参照）。
- なお、[Alt] + [F4] キーにより、このウィンドウをクローズすることができますが、この操作は行わないでください。

備考 このウィンドウ、およびこのウィンドウよりオープンする各種ウィンドウのタイトルバー／メニューバーの表記は、使用するホスト・マシンの [コントロールパネル] → [地域と言語のオプション] 設定に依存します（この設定を日本以外／日本語以外に指定すると、タイトルバー／メニューバーが英語表記となります）。

図 A—93 シミュレータ GUI ウィンドウ






ここでは、次の項目について説明します。

- [メニューバー]
- [ツールバー]
- [ウィンドウ表示エリア]

[メニューバー]

- (1) [ファイル] メニュー
- (2) [編集] メニュー
- (3) [表示] メニュー
- (4) [部品] メニュー
- (5) [図形] メニュー
- (6) [オプション] メニュー
- (7) [シミュレータ] メニュー
- (8) [ウィンドウ] メニュー
- (9) [ヘルプ] メニュー

(1) [ファイル] メニュー

新規作成 ...	新規作成のための選択ダイアログをオープンします。  ボタンのクリックと同様です。
開く ...	シミュレータ GUI ウィンドウが扱うファイルを開きます。  ボタンのクリックと同様です。
閉じる	現在フォーカスのあるウィンドウをクローズします。
上書き保存	シミュレータ GUI ウィンドウが扱うファイルに、現在フォーカスのあるウィンドウの内容を上書き保存します。  ボタンのクリックと同様です。
名前を付けて保存 ...	指定したファイルに、現在フォーカスのあるウィンドウの内容を保存します。

(2) [編集] メニュー

このメニューの内容は、現在フォーカスのあるウィンドウの種類により異なります。

このメニューについての詳細は、[信号データエディタ ウィンドウ](#)/[タイミングチャート ウィンドウ](#)/[入出力パネル ウィンドウ](#)/[シリアル ウィンドウ](#)の [専用メニュー] の項を参照してください。

(3) [表示] メニュー

このメニューの内容は、現在フォーカスのあるウィンドウの種類により異なります。

このメニューについての詳細は、[信号データエディタ ウィンドウ](#)/[タイミングチャート ウィンドウ](#)/[入出力パネル ウィンドウ](#)/[シリアル ウィンドウ](#)の [専用メニュー] の項を参照してください。

(4) [部品] メニュー

このメニューは、[入出力パネル ウィンドウ](#)をオープンした際に追加されます。

このメニューについての詳細は、[\[部品\] メニュー](#)/[\[部品\] ツールバー](#)を参照してください。

(5) [図形] メニュー





このメニューは、[入出力パネル ウィンドウ](#)をオープンした際に追加されます。

このメニューについての詳細は、[\[図形\] メニュー](#) / [\[図形\] ツールバー](#)を参照してください。

(6) [オプション] メニュー

ツールバー	カスケード・メニューに対応するツールバーの表示／非表示を切り替えます。
シミュレータスタンダード	[シミュレータスタンダード] ツールバーの表示／非表示を切り替えます。
シミュレータツール	[シミュレータツール] ツールバーの表示／非表示を切り替えます。
信号データエディタ	[信号データエディタ] ツールバーの表示／非表示を切り替えます。
タイミングチャート	[タイミングチャート] ツールバーの表示／非表示を切り替えます。
部品	[部品] ツールバーの表示／非表示を切り替えます。
図形	[図形] ツールバーの表示／非表示を切り替えます。
ウィンドウのカスタマイズ ...	書式設定 ダイアログ をオープンします。

(7) [シミュレータ] メニュー

信号データエディタ	信号データエディタ ウィンドウ をオープンします。  ボタンのクリックと同様です。
タイミングチャート	タイミングチャート ウィンドウ をオープンします。  ボタンのクリックと同様です。
入出力パネル	入出力パネル ウィンドウ をオープンします。  ボタンのクリックと同様です。
シリアル	シリアル ウィンドウ をオープンします。  ボタンのクリックと同様です。

(8) [ウィンドウ] メニュー

すべてのウィンドウを閉じる	このウィンドウを除く、すべてのウィンドウをクローズします。
重ねて表示	このウィンドウ内のウィンドウをカスケード表示します。
並べて表示	このウィンドウ内のウィンドウを並べて表示します。
アイコンの整列	このウィンドウの下部にアイコンを並べて表示します。












(9) [ヘルプ] メニュー

メイン・ウィンドウ	このウィンドウのヘルプを表示します。
カレント ウィンドウ	カレント・ウィンドウのヘルプを表示します。





[ツールバー]

- (1) [シミュレータスタンダード] ツールバー
- (2) [シミュレータツール] ツールバー
- (3) [信号データエディタ] ツールバー
- (4) [タイミングチャート] ツールバー
- (5) [部品] ツールバー
- (6) [図形] ツールバー

(1) [シミュレータスタンダード] ツールバー

	シミュレータ GUI ウィンドウを新規にオープンします。
	シミュレータ GUI ウィンドウが扱うファイルを開きます。
	シミュレータ GUI ウィンドウが扱うファイルに、現在フォーカスのあるウィンドウの内容を上書き保存します。
	直前に行った操作を取り消し、元の状態に戻します。
	 ボタンで戻した状態を復帰します。
	選択範囲を切り取り、クリップボードに保存します。
	選択範囲をコピーし、クリップボードに保存します。
	クリップボードの内容を貼り付けます。
	データ検索 ダイアログをオープンします。
	ヘルプの目次を表示します。

(2) [シミュレータツール] ツールバー

	信号データエディタ ウィンドウをオープンします。
	タイミングチャート ウィンドウをオープンします。
	シリアル ウィンドウをオープンします。
	入出力パネル ウィンドウをオープンします。

(3) [信号データエディタ] ツールバー

このツールバーは、[信号データエディタ ウィンドウ](#)にフォーカスがある場合にのみ選択可能となります。
このツールバーについての詳細は、[\[\[信号データエディタ\] ツールバー\]](#) を参照してください。

(4) [タイミングチャート] ツールバー

このツールバーは、[タイミングチャート ウィンドウ](#)にフォーカスがある場合にのみ選択可能となります。
このツールバーについての詳細は、[\[\[タイミングチャート\] ツールバー\]](#) を参照してください。

(5) [部品] ツールバー

このツールバーは、[入出力パネル ウィンドウ](#)にフォーカスがある場合にのみ選択可能となります。
このツールバーについての詳細は、[\[部品\] メニュー / \[部品\] ツールバー](#)を参照してください。

(6) [図形] ツールバー

このツールバーは、[入出力パネル ウィンドウ](#)にフォーカスがある場合にのみ選択可能となります。

このツールバーについての詳細は、[\[図形\] メニュー](#) / [\[図形\] ツールバー](#)を参照してください。

[ウィンドウ表示エリア]

各種ウィンドウ ([信号データエディタ ウィンドウ](#) / [タイミングチャート ウィンドウ](#) / [入出力パネル ウィンドウ](#) / [シリアル ウィンドウ](#)) を表示するエリアです。

表示されたウィンドウは、このエリアの中でウィンドウ・サイズの変更、アイコン化などを行います。

書式設定 ダイアログ

信号データエディタ ウィンドウ、 タイミングチャート ウィンドウ、 およびシリアル ウィンドウの色やフォントの設定、 変更を行います。

図 A—94 書式設定 ダイアログ：[色] タブ（タイミングチャート ウィンドウの場合）

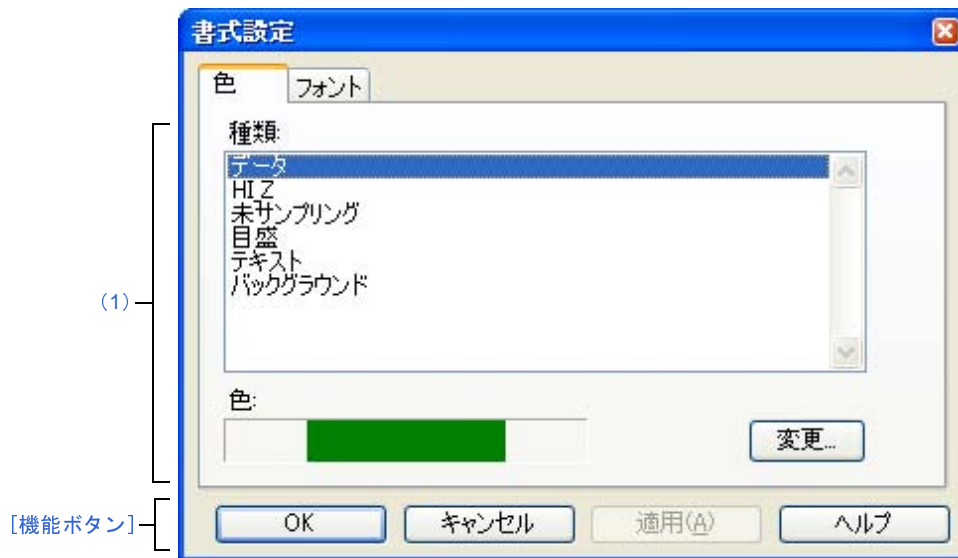


図 A—95 書式設定 ダイアログ：[フォント] タブ



ここでは、次の項目について説明します。

- [オープン方法]
- [[色] タブ]
- [[フォント] タブ]
- [機能ボタン]

[オープン方法]

- 信号データエディタ ウィンドウ／タイミングチャート ウィンドウ／シリアル ウィンドウのいずれかにフォーカスがある状態で、[オプション] メニュー→ [ウィンドウのカスタマイズ...] を選択

[[色] タブ]

(1) 色設定エリア

対象ウィンドウ内の各部位の色を設定／変更します。

種類	色の変更が可能な部位がリスト表示されます。 表示される部位は、対象となるウィンドウにより異なります。
色	リスト内を選択することにより、該当部位の現在の設定色が表示されます。
[変更...] ボタン	リスト内の該当部位の現在の設定色を変更します。

[[フォント] タブ]

(1) フォント設定エリア

対象ウィンドウ内で使用する各部位のテキストのフォントを設定／変更します。

種類	フォントの変更が可能な部位がリスト表示されます。
フォント	リスト内を選択することにより、該当部位の現在のフォント名が表示されます。
サイズ	リスト内を選択することにより、該当部位の現在のフォント・サイズが表示されます。
[変更...] ボタン	リスト内の該当部位の現在の設定フォントを変更します。

[機能ボタン]

ボタン	機能
OK	設定を有効にし、このダイアログをクローズします。
キャンセル	設定を無視し、このダイアログをクローズします。
適用	選択不可
ヘルプ	このダイアログのヘルプを表示します。

信号データエディタ ウィンドウ

入力端子に入力する信号データの作成、編集を行います。

作成した信号データは、[編集] メニュー→ [信号入力] → [開始] を選択することにより、シミュレーション中の入力端子へ入力されます。また、[ファイル] メニュー→ [上書き保存] / [名前を付けて保存...] の選択により、信号データ・ファイル (*.wvi)、およびプロジェクト・ファイルへ保存することができます。

なお、保存した信号データは、[ファイル] メニュー→ [開く...] の選択、またはプロジェクト・ファイルのロードにより復元することができます。

- 注意 1. 保存した信号データ・ファイルをオープンする際、またはプロジェクト・ファイルをオープンする際に、信号データ・ファイルを作成した時点でのマイクロコントローラとは異なるマイクロコントローラでシミュレータ GUI が起動されていた場合、そのマイクロコントローラに存在しない端子名の設定は復元されません。
2. このウィンドウから、メイン・クロック入力、およびサブ・クロック入力を行うことはできません。メイン・クロック、およびサブ・クロックの発振周波数の設定は、[プロパティ パネルの \[接続用設定\] タブ](#)で行ってください。
3. プログラムがブレークしている間に信号データの入力を開始した場合、実際に信号入力開始されるのはプログラムの実行直後になります。

備考 1. このウィンドウでは、次の信号データの表示、および編集を行うことができます。

- 新規に作成する信号データ
 - 以前に作成した信号データ・ファイル
 - 以前にシミュレーションし、出力信号データとして保存した信号データ・ファイル
2. このウィンドウのタイトルバー上には、プロジェクト・ファイルを読み込んだ場合、“プロジェクト・ファイル名+数字 (0 から連番) wvi” を表示します。

ただし、PM+ のプロジェクト・ファイルを読み込み、CubeSuite+ のプロジェクト・ファイルに保存した場合は、それ以降、“プロジェクト・ファイル名+ CS + 数字 (0 から連番) .wvi” と表示します。


図 A—96 信号データエディタ ウィンドウ

Unit	Mark	Wait	PO0	PO1	PO2	ANIO	AVREF
1	↓	100	0	0	0	0	5000
2		100	0	0	1	500	5000
3		100	0	1	0	1000	5000
4		100	0	1	1	1500	5000
5		100	1	0	0	2000	5000
6		100	1	0	1	2500	5000
7		100	1	1	1	3000	5000
8		100	0	0	0	3500	5000
9		100	0	0	1	4000	5000
10		100	0	1	0	4500	5000

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]
- [専用メニュー（信号データエディタ ウィンドウ）]
- [[信号データエディタ] ツールバー]
- [コンテキスト・メニュー]
- [操作方法]





[オープン方法]

-  ボタンをクリック
- [シミュレータ] メニュー → [信号データエディタ] を選択

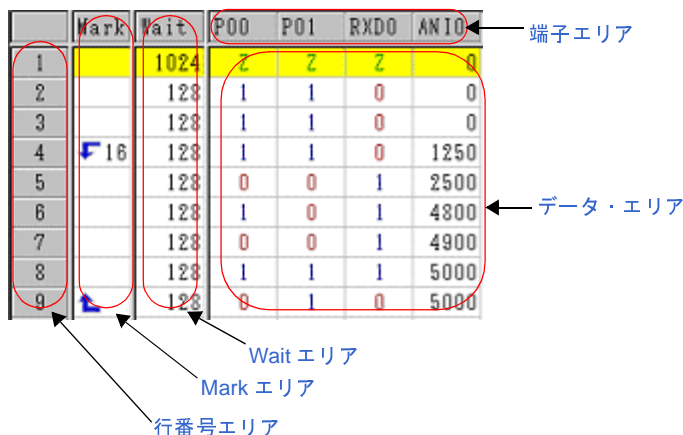
[各エリアの説明]

(1) インフォメーション・バー

このエリアは [表示] メニュー → [インフォメーションバー] により、表示/非表示の選択が可能です。

 MainClk	ウェイト時間の単位をドロップダウン・リストから変更します。 ウェイト時間の単位は、[編集] メニュー → [時間単位] から変更可能です。
	プログラム実行中に、このボタンをクリックすると信号入力を開始します。 プログラム停止中に、このボタンをクリックしておく、次回プログラム実行開始のタイミングで信号入力が自動的に開始されます。
	プログラム実行中に、このボタンをクリックすると信号入力を停止します。 プログラム停止中に、このボタンをクリックしておく、プログラム実行を開始しても信号入力が自動的にには行われません。
	入力カレント行（黄色の行）を先頭に戻します。

(2) クライアント・エリア



	Mark	Wait	P00	P01	RXD0	AN10
1		1024	2	2	2	0
2		128	1	1	0	0
3		128	1	1	0	0
4	16	128	1	1	0	1250
5		128	0	0	1	2500
6		128	1	0	1	4800
7		128	0	0	1	4900
8		128	1	1	1	5000
9		128	0	1	0	5000





端子エリア

データ・エリア

Wait エリア

Mark エリア

行番号エリア

端子エリア	<p>入力端子名を示します。</p> <p>使用する入力端子の選択は、ツールバーの  ボタンのクリック、または [編集] メニュー → [端子選択 ...] によりオープンする 端子選択 ダイアログ で行います。</p> <p>なお、[編集] メニュー → [端子状態] により端子へのデータ入力の有効/無効の制御が可能です。</p>		
行番号エリア	<p>行番号を示します。</p> <p>このエリアは行単位で編集する際に使用します。</p> <p>なお、信号データとして、最大 1,048,576 (= 1M) 行まで入力することができます。</p>		
Mark エリア	<p>設定した入力値に対するループ情報を示します。</p> <p>ループ情報の設定は、対象枠内におけるコンテキスト・メニューからの選択、または [編集] メニュー → [マーク設定] により行います。</p> <p>ループ情報を設定すると、次のマークが表示されます。</p>		
		ループの開始位置 (無限ループ)	
	 n	ループの開始位置 (ループカウント付き)	
		ループの終了位置	
Wait エリア	<p>設定した入力値が端子に入力されるタイミングをウェイト時間として示します。ウェイト時間の設定は、対象枠内に直接数値を書き込むことにより行います。</p> <p>0 ~ 4,294,967,295 までの整数値 (10 進数) が指定可能です (4,294,967,295 を越える数値を設定する場合は、もう 1 段を使用することで設定可)。</p> <p>なお、ウェイト時間の単位は、[編集] メニュー → [時間単位] により選択可能です。</p>		
データ・エリア	<p>端子に入力する入力値を示します。</p> <p>入力値の設定は、対象枠内に直接数値を書き込むことにより行います。</p> <p>なお、端子の種類によって次のように入力規則が異なります。</p>		
	デジタル端子	次のいずれかの 1 文字列	
		0	LOW 信号
		1	HIGH 信号
	Z	Hi-Z 信号 (大小文字不問)	
アナログ端子	0 ~ 5000 の範囲の値 (10 進数) (単位 : mV)		

[専用メニュー (信号データエディタ ウィンドウ)]

(1) [編集] メニュー

元に戻す	選択不可
やり直し	選択不可
切り取り	選択範囲を切り取りクリップボードに保存します。
コピー	選択範囲をコピーしクリップボードに保存します。
貼り付け	クリップボードの内容を選択位置に貼り付けます。
削除	選択範囲を削除します。
すべて選択	すべての表示データを選択します。
検索	選択不可

端子選択 ...	端子選択 ダイアログ をオープンします。データを作成／編集する端子を選択します。
時間単位	ウェイト時間の単位を選択します。
メインクロック	ウェイト時間の単位をメイン・クロックとします (デフォルト)。
マイクロ秒	ウェイト時間の単位をマイクロ秒とします。
ミリ秒	ウェイト時間の単位をミリ秒とします。
端子状態	選択した端子の入力状態を選択します。
有効	端子へのデータ入力を有効にします (デフォルト)。
無効	端子へのデータ入力を無効にします。
マーク設定	選択した Mark エリア にループ情報を設定します。
ループ開始	ループ開始マークを設定します。
ループ終了	ループ終了マークを設定します。
ループ詳細設定	ループ設定 ダイアログ をオープンします。ループ情報の詳細を設定します。
信号入力	信号データをシミュレータに入力します。
開始	信号入力を開始します。
停止	信号入力を停止します。
リセット	入力カレント行を先頭に戻します。





(2) [表示] メニュー

インフォメーションバー	インフォメーション・バーの表示／非表示を切り替えます。
-------------	-----------------------------

(3) [オプション] メニュー

ウィンドウのカスタマイズ ...	書式設定 ダイアログ をオープンします。
------------------	--------------------------------------

[[信号データエディタ] ツールバー]

	端子選択 ダイアログ をオープンします。データを作成／編集する端子を選択します。
	プログラム実行中に、このボタンをクリックすると信号入力を開始します。 プログラム停止中に、このボタンをクリックしておく、次回プログラム実行開始のタイミングで信号入力が自動的に開始されます。
	プログラム実行中に、このボタンをクリックすると信号入力を停止します。 プログラム停止中に、このボタンをクリックしておく、プログラム実行を開始しても信号入力が自動的にには行われません。
	入力カレント行 (黄色の行) を先頭に戻します。

[コンテキスト・メニュー]

クライアント・エリアの各エリアにおいて、次のコンテキスト・メニューを表示します。

(1) 端子エリア

有効	端子へのデータ入力を有効にします（デフォルト）。
無効	端子へのデータ入力を無効にします。
端子選択 ...	端子選択 ダイアログをオープンします。データを作成／編集する端子を選択します。

(2) 行番号

切り取り	選択範囲を切り取りクリップボードに保存します。
コピー	選択範囲をコピーしクリップボードに保存します。
貼り付け	クリップボードの内容を選択位置に貼り付けます。
削除	選択範囲を削除します。

(3) Mark エリア

切り取り	選択セルを切り取りクリップボードに保存します。
コピー	選択セルをコピーしクリップボードに保存します。
貼り付け	クリップボードの内容を選択位置に貼り付けます。
削除	選択セルを削除します。
ループ開始	ループ開始マークを設定します。
ループ終了	ループ終了マークを設定します。
ループ詳細設定	ループ設定 ダイアログをオープンします。ループ情報の詳細を設定します。

(4) Wait エリア

切り取り	選択セルのデータを切り取りクリップボードに保存します。切り取られたデータは“0”になります。
コピー	選択セルのデータをコピーしクリップボードに保存します。
貼り付け	クリップボードの内容を選択位置に貼り付けます。
削除	選択セルのデータを削除します。削除されたデータは“0”になります。

(5) データ・エリア

切り取り	選択セルのデータを切り取りクリップボードに保存します。切り取られたデータは“Z (Hi-Z)”になります。
コピー	選択セルのデータをコピーしクリップボードに保存します。
貼り付け	クリップボードの内容を選択位置に貼り付けます。
削除	選択セルのデータを削除します。削除されたデータは“Z (Hi-Z)”になります。
信号入力開始	信号入力を開始します。


信号入力停止	信号入力を停止します。
信号入力リセット	入力カレント行を先頭に戻します。

[操作方法]

- (1) 端子の選択
- (2) 信号データの作成
- (3) データのコピーと貼り付け編集
- (4) 行単位の編集
- (5) 信号の入力
- (6) CPU リセット時の動作

(1) 端子の選択

信号データを作成するためには、最初に使用する端子を選択する必要があります。

端子の選択は、ツールバーの  ボタンのクリック、または [編集] メニュー → [端子選択 ...] の選択によりオープンする [端子選択 ダイアログ](#)で行います。これにより、[端子エリア](#)に選択した端子名が表示されます。

(2) 信号データの作成

各端子へ入力する信号データを作成します。

(a) 入力値の設定

データ・エリアにおいて、各端子へ入力する値を設定します（「[データ・エリア](#)」参照）。

(b) 入力タイミングの設定

Wait エリアにおいて、各端子へ入力するタイミングをウエイト時間として設定します（「[Wait エリア](#)」参照）。

(c) ループ情報の設定

上記 (a) ~ (b) で設定した信号データをループ処理させたい場合は、ループ情報を設定します。

ループ情報の設定は、Mark エリアのループ開始位置で、コンテキスト・メニューの [ループ開始] を選択し、ループ終了位置で [ループ終了] を選択します。

なお、この際に、ループ・カウントを指定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 [ループ詳細設定 ...] を選択することでオープンする [ループ設定 ダイアログ](#)により、ループ・カウントの設定を行います。

ループ情報の設定が完了すると、該当するループ情報のマークが表示されます（「[Mark エリア](#)」参照）。

(3) データのコピーと貼り付け編集

[Mark エリア](#) / [Wait エリア](#) / [データ・エリア](#) 上の設定値は、コピーと貼り付けを行うことができます。

ただし、コピーしたデータは同一エリア内でのみの貼り付けとなります。

コピー	1つ、または複数（範囲）のセルを選択した状態で、[編集] メニュー → [コピー] の選択（または [Ctrl] キー + [C] の入力）により行います。
-----	--

貼り付け	1つ、または複数（範囲）のセルを選択した状態で、[編集]メニュー→[貼り付け]の選択（または [Ctrl] キー + [V] の入力）により行います。 複数（範囲）のセルを選択した場合は、コピーしたデータを繰り返して貼り付けます。
------	--

(4) 行単位の編集

行単位の編集は、[行番号エリア](#)を選択することにより行います。

方法は「(3) データのコピーと貼り付け編集」と同様です。




なお、行の貼り付け（挿入）時に貼り付けられるデータは選択行の直前に挿入されます。

(5) 信号の入力

作成した信号データをシミュレーション実行時にシミュレータの入力端子へ入力します。

この際、プログラムがブレイクすると、現在信号入力中の行が黄色く強調表示され（[書式設定ダイアログ](#)の[入力カレント行]で変更可能）信号入力の進捗状況を表示します。


信号データの入力操作には次の種類があります。

信号入力の開始	 ボタンのクリック、または [編集] メニュー → [信号入力] → [開始] を選択します。 これにより、入力カレント行（強調表示されている行）から信号入力が開始されます。
信号入力の停止	 ボタンのクリック、または [編集] メニュー → [信号入力] → [停止] を選択します。 これにより信号入力が停止されます。
信号のリセット	 ボタンのクリック、または [編集] メニュー → [信号入力] → [リセット] を選択します。 これにより入力カレント行が先頭に戻ります。なお、信号入力中に信号のリセットを行った場合には先頭から引き続き入力が継続されます。

備考 端子名を選択したのち、[編集]メニュー→[端子状態]→[有効] / [無効] を選択することにより、端子への信号データ入力を制御することができます。

(6) CPU リセット時の動作

CPU リセットが発生した場合、入力カレント行は先頭に戻ります。

信号入力中に CPU リセットが発生した場合は、先頭から引き続き入力が継続されます（ ボタンのクリックと同等）。

ループ設定 ダイアログ

信号データエディタ ウィンドウのループ情報に関する詳細設定（ループ開始／終了，ループ・カウントなど）を行います。

図 A—97 ループ設定 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]

信号データエディタ ウィンドウにおける次のいずれか

- Mark エリアをダブルクリック
- Mark エリアを選択したのち，[編集] メニュー→ [マーク設定] → [ループ詳細設定] を選択

[各エリアの説明]

(1) ループ情報設定エリア

ループ開始	ループ開始を設定する場合，チェックします。	
	無限ループ	無限ループを設定する場合，選択します。
	ループカウント	ループ・カウントを設定する場合，選択します。 スピン・ボタンにより，次のカウント値を指定します。
		0
	1 ~ 99	指定カウント分のループをします。
ループ終了	ループ終了を設定する場合，チェックします。	

[機能ボタン]

ボタン	機能
OK	設定を有効にし、このダイアログをクローズします。
キャンセル	設定を無視し、このダイアログをクローズします。

端子選択 ダイアログ

信号データエディタ ウィンドウ、およびタイミングチャート ウィンドウで表示する端子を設定します。

設定した端子情報は、[保存] ボタンにより端子情報ファイル (*.pin) として保存することができます。また、保存した端子情報は、[読み込み] ボタンにより復元することができます。

図 A—98 端子選択 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]

信号データエディタ ウィンドウ／タイミングチャート ウィンドウにフォーカスがある状態で次のいずれか

-  ボタンをクリック
- [編集] メニュー → [端子選択 ...] を選択

[各エリアの説明]

(1) 接続端子設定エリア

エリア右側のスクロール・バーを操作することにより、最大 256 端子まで設定することができます。

接続端子名	接続する端子名を指定します。 指定は、ドロップダウン・リストからの選択、または直接入力により行います。
アナログ	指定した端子をアナログ端子として使用する場合、対応するチェック・ボックスをチェックします。
機能名	指定した端子に機能名を設定します。 入力した文字列を端子名としてウィンドウ上に表示します。 何も指定しない場合、端子名を表示します。

備考 指定する端子名に関しては、使用するマイクロコントローラのユーザーズ・マニュアルを参照してください。

[機能ボタン]

ボタン	機能
OK	設定を有効にし、このダイアログをクローズします。 このダイアログを呼び出したウィンドウの Pin 欄に端子名（または表示名）が表示されます。
保存	表示内容を端子情報ファイル (*.pin) に保存します。
読み込み	指定した端子情報ファイル (*.pin) を読み込みます。
クリア	設定内容をすべて削除します。
キャンセル	設定を無視し、このダイアログをクローズします。

タイミングチャート ウィンドウ

端子に対する入力信号と出力信号をタイミング・チャートで表示します。

このウィンドウでは、メイン・クロック単位で時間計測を行います。

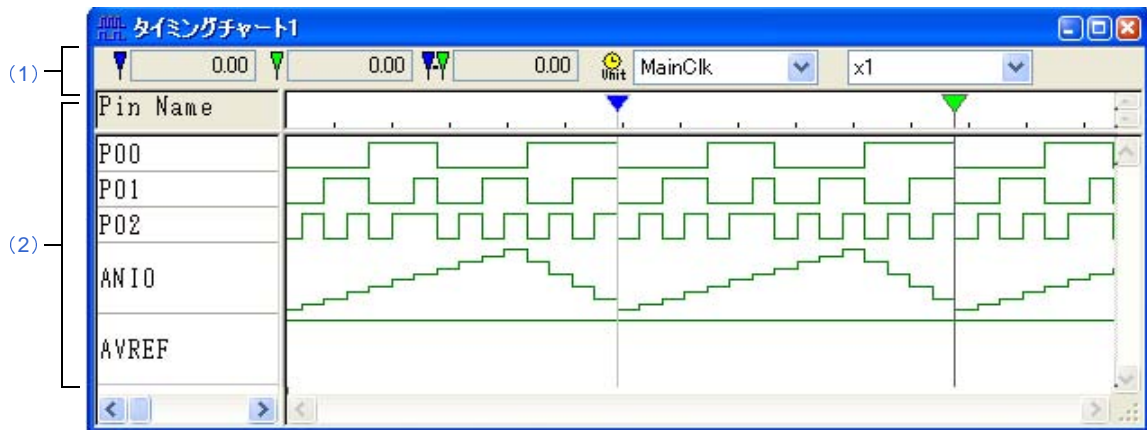
ブラウザした信号データは、[ファイル] メニュー→ [上書き保存] / [名前を付けて保存 ...] により、タイミングチャート・ファイル (*.wvo) として保存することができます。

保存した信号データは、[ファイル] メニュー→ [開く ...] の選択により復元することができます。

なお、プロジェクト・ファイルとして保存した場合、信号データは保存されませんが、設定した端子情報は保存されます（測定結果を保存する必要がない場合、この方法で問題ありません）。

- 注意 1. 保存したタイミングチャート・ファイルをオープンする際、またはプロジェクト・ファイルをオープンする際に、タイミングチャート・ファイルを保存した時点でのマイクロコントローラとは異なるマイクロコントローラでシミュレータ GUI が起動されていた場合、そのマイクロコントローラに存在しない端子名の設定は復元されません。
2. このウィンドウでメイン・クロック波形、およびサブ・クロック波形を表示することはできません。また、外部バス・インタフェース機能使用時に、外部バス・インタフェース機能で使用する端子の波形を表示することはできません。


図 A—99 タイミングチャート ウィンドウ



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]
- [専用メニュー（タイミングチャート ウィンドウ）]
- [[タイミングチャート] ツールバー]
- [コンテキスト・メニュー]
- [操作方法]






[オープン方法]

-  ボタンをクリック
- [シミュレータ] メニュー → [タイミングチャート] を選択

[各エリアの説明]

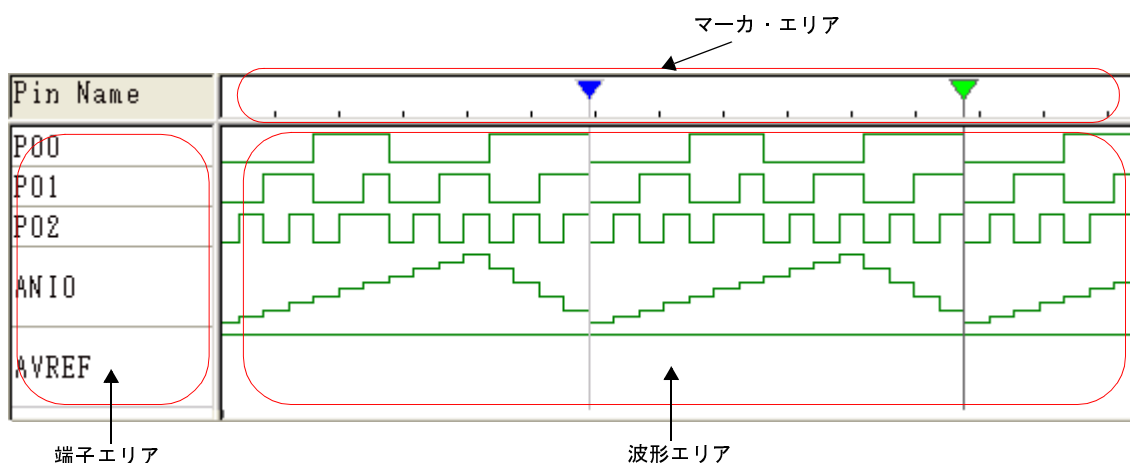
(1) インフォメーション・バー

このエリアは [表示] メニュー → [インフォメーションバー] により、表示/非表示の選択が可能です。



 5553364.00	シミュレーション開始からマーカ A の位置までの時間を表示します。
 5554860.00	シミュレーション開始からマーカ B の位置までの時間を表示します。
 1496.00	マーカ A, B 間の時間を絶対値で表示します。
 MainClk	マーカ A, B の位置情報の単位をドロップダウン・リストから選択します。 [編集] メニュー → [時間単位], およびコンテキスト・メニューの [時間単位] から変更可能です。
 x1/2	波形データの表示倍率をドロップダウン・リストから選択します。なお、表示倍率を変更する際、波形データの一部が消去される場合には、確認ダイアログが表示されます。

- 備考 1. シミュレーション開始から、最大 4,294,967,262 クロックまでカウントすることができます。カウントが最大値に達した場合 0 に戻り、再度カウントを開始します。
2. プログラム実行中は表示倍率の設定欄が淡色表示になり変更できません。

(2) クライアント・エリア



端子エリア	このウィンドウで表示する端子名を表示します。 端子の選択は、[編集] メニュー → [端子選択 ...] によりオープンする 端子選択 ダイアログ で行います。
-------	---

マーカエリア	マーカ A, B のヘッド部分を表示します。 これらのマーカはドラッグにより移動可能です。	
		マーカ A
		マーカ B
波形エリア	端子エリアに指定した端子のデータをタイミング・チャート表示します。 なお、信号の種類により、デフォルトで次のように色分けされます。	
	緑	端子の HIGH, LOW 信号
	赤	ハイ・インピーダンス信号
	青	未サンプリング状態の信号

備考 1. 端子データを格納するバッファはリング・バッファ形式のため、バッファがいっぱいになると最古のデータは、最新のデータによって上書きされます。

なお、バッファのサイズは、次のいずれかまでです。

- 端子変化点：4,096 箇所
- クロック数：2,147,483,631
- 描画横幅：134,217,711 ピクセル

2. [オプション] メニュー→ [ウィンドウのカスタマイズ...] の選択でオープンする [書式設定 ダイアログ](#) により、このエリア内の色/フォントを変更することができます。

[専用メニュー (タイミングチャート ウィンドウ)]

(1) [編集] メニュー

クリア	すべての波形データを削除します。
検索 ...	データ検索 ダイアログ がオープンします。 データの検索を行います。
後方検索	選択端子の変化点を後方 (左方向) に検索します。
前方検索	選択端子の変化点を前方 (右方向) に検索します。
端子選択 ...	端子選択 ダイアログ をオープンします。データを表示する端子を選択します。
時間単位	マーカ A, B の位置情報の単位を選択します。
メインクロック	ウェイト時間の単位をメイン・クロックとします (デフォルト)。
マイクロ秒	ウェイト時間の単位をマイクロ秒とします。
ミリ秒	ウェイト時間の単位をミリ秒とします。






(2) [表示] メニュー

波形表示	マーカ・エリアと波形エリアの表示／非表示を切り替えます。
インフォメーション バー	インフォメーション・バーの表示／非表示を切り替えます。
ズーム	波形データの表示倍率を選択します。なお、表示倍率を変更する際、波形データの一部が消去される場合には、確認ダイアログが表示されます。
x 1/32	表示倍率を 1/32 倍にします。
x 1/16	表示倍率を 1/16 倍にします。
x 1/8	表示倍率を 1/8 倍にします。
x 1/4	表示倍率を 1/4 倍にします。
x 1/2	表示倍率を 1/2 倍にします。
x 1	表示倍率を 1 倍にします。
x 2	表示倍率を 2 倍にします。
x 4	表示倍率を 4 倍にします。
x 8	表示倍率を 8 倍にします。
x 16	表示倍率を 16 倍にします。
x 32	表示倍率を 32 倍にします。

(3) [オプション] メニュー

ウィンドウのカスタマイズ ...	書式設定ダイアログをオープンします。
------------------	--------------------

[[タイミングチャート] ツールバー]

	すべての波形データを削除します。
	選択端子の変化点を後方（左方向）に検索します。
	選択端子の変化点を前方（右方向）に検索します。
	端子選択ダイアログをオープンします。 データを表示する端子を選択します。
	波形データの表示倍率をドロップダウン・リストから選択します。なお、表示倍率を変更する際、波形データの一部が消去される場合には、確認ダイアログが表示されます。

[コンテキスト・メニュー]

クライアント・エリアにおいて、次のコンテキスト・メニューを表示します。


クリア	すべての波形データを削除します。
検索	データ検索 ダイアログをオープンします。 データの検索を行います。
後方検索	選択端子の変化点を後方（左方向）に検索します。
前方検索	選択端子の変化点を前方（右方向）に検索します。
端子選択 ...	端子選択 ダイアログをオープンします。 データを表示する端子を選択します。
波形表示	マーカ・エリアと波形エリアの表示／非表示を切り替えます。
時間単位	マーカ A, B の位置情報の単位を選択します。
ズーム	波形データの表示倍率を選択します。なお、表示倍率を変更する際、波形データの一部が消去される場合には、確認 ダイアログが表示されます。
マーカ A の配置	マーカ A をマウス・カーソル位置に配置します。 [Shift] キー + クリックでも同様の操作が可能です。
マーカ B の配置	マーカ B をマウス・カーソル位置に配置します。 [Ctrl] キー + クリックでも同様の操作が可能です。

[操作方法]

- (1) 端子の選択
- (2) タイミング・チャートの表示
- (3) タイミング・チャートのクリア
- (4) タイミング・チャートのタイミング計測
- (5) データの検索
- (6) リセット時の動作

(1) 端子の選択

タイミング・チャートを表示するためには、最初に表示する端子を選択する必要があります。

端子の選択は、ツールバーの  ボタンのクリック、または [編集] メニュー → [端子選択 ...] の選択によりオープンする **端子選択 ダイアログ** で行います。これにより、**端子エリア** に選択した端子名が表示されます。

(2) タイミング・チャートの表示

プログラムを実行することにより、選択した端子の波形がタイミング・チャート形式で表示されます。

備考 タイミング・チャートを非表示にすることでシミュレーション速度を速めることができます。

非表示にするには、[表示] メニュー → [波形表示] を選択します（チェックなしにする）。

非表示時は、マーカ・エリア、および波形エリアは淡色表示され、中央に“Display OFF”と表示されます。

(3) タイミング・チャートのクリア

[編集] メニュー → [クリア] の選択により、タイミング・チャートの表示波形はすべてクリアされます。

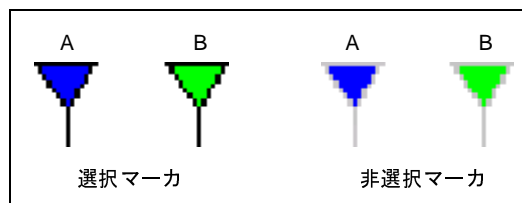
(4) タイミング・チャートのタイミング計測

マーカ A, B を使用し、2 箇所をマーキングすることで 2 点間のタイミング計測を行います。各マーカの時間、およびマーカ間の時間はインフォメーション・バーに表示されます。

マーカの配置は、マーカ・ヘッドをドラッグすることにより目的の位置への移動することができます。また、コンテキスト・メニュー → [マーカ A の配置] / [マーカ B の配置] の選択によっても、マーカが現在のマウス・カーソルの位置に移動します。

なお、最後にクリックしたマーカは選択マーカとなり、[データの検索](#)の対象となります。

図 A—100 マーカ A とマーカ B

**(5) データの検索**

タイミング・チャートのデータ検索機能には、次の 2 種類があります。

(a) 単純検索

単純検索は 1 つの端子の変化点を検索する機能です。

検索したい端子名を端子エリアで 1 つ選択し、[編集] メニュー → [後方検索] / [前方検索] を選択します。これにより、変化点が検索されたデータ位置に選択マーカが移動します。

(b) 詳細検索

詳細検索では複数端子の様々なデータの組み合わせによる検索を行います。

検索データの設定は、[編集] メニュー → [検索 ...] の選択によりオープンする[データ検索 ダイアログ](#)で行います。検索結果は単純検索と同様に、ヒットしたデータ位置に選択マーカが移動します。

(6) リセット時の動作

CPU リセット、またはシミュレータ GUI のリセットが発生した場合、タイミング・チャートの表示波形はすべてクリアされます。

データ検索 ダイアログ

タイミングチャート ウィンドウで表示されている信号データの検索を行います。

- 注意 1. アナログ入出力信号は検索できません。
 2. プログラム実行中に、このダイアログをオープンすることはできません。

図 A—101 データ検索 ダイアログ




ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

タイミングチャート ウィンドウにフォーカスがある状態で次のいずれか

-  ボタンをクリック
- [編集] メニュー → [検索 ...] を選択

[各エリアの説明]

(1) 検索端子設定エリア

検索条件を複数個指定した場合、すべての検索条件を満たす信号データを検索します。

右側のスクロール・バーを操作することにより、検索条件は最大 48 個指定できます。

検索端子	検索する端子名を指定します。 指定は、ドロップダウン・リストからの選択、または直接入力により行います。 空白を入力するとデータ検索の対象外となり [検索データ] を入力不可にします。	
検索データ	指定した端子で検索するデータをドロップダウン・リストから選択します。	
	-----	指定しません。
	Rising Edge	信号データの立ち上がりを検索します。
	Falling Edge	信号データの立ち下がりを検索します。
	Rise/Fall Edge	信号データの立ち上がり/立ち下がりを検索します。
	High	信号データが HIGH の状態を検索します。
	Low	信号データが LOW の状態を検索します。
検索する方向	データ検索する方向をオプション・ボタンにより選択します。 [次を検索] ボタンをクリックした際、このエリアで選択した方向へ検索を行います。	
	後方	後方（現在位置より古い時間）のデータを検索します。
	前方	前方（現在位置より新しい時間）のデータを検索します（デフォルト）。

[機能ボタン]

ボタン	機能
次を検索	選択している方向へ検索を行います。 検索終了後、再度クリックすることにより次のデータを検索します。
キャンセル	データ検索を中止し、このダイアログをクローズします。

入出力パネル ウィンドウ

疑似的なターゲット・システムの構築、および作成した接続部品の操作を行います。

このウィンドウでは、使用する接続部品（図形オブジェクト／部品オブジェクト）の作成／設定を行うことで疑似的なターゲット・システムを構築することができます。設定を行った接続部品はこのウィンドウ内の任意の位置へ配置することができます。シミュレーション中に信号処理としてそれらを操作することができます。

ウィンドウ内に配置した接続部品の情報は、[ファイル]メニュー→[上書き保存]／[名前を付けて保存...]の選択により、入出力パネル・ファイル (*.pnl)、またはプロジェクト・ファイルへ保存することができます。

なお、保存した部品情報ファイルの内容は [ファイル]メニュー→[開く...]の選択、またはプロジェクト・ファイルのロードにより復元することができます。

1. 保存した入出力パネル・ファイルを開く際、そのファイルを作成した時点のマイクロコントローラとは異なるマイクロコントローラでシミュレータ GUI が起動されていた場合、そのマイクロコントローラに存在しない端子に接続されていた部品の端子設定情報は復元されません（各部品の設定ダイアログ内の [接続端子] が空欄になります）。
2. プログラムがブレークしている間に信号を入力した場合（ボタンを押した場合など）、実際に信号が変化するのはプログラムの実行直後になります。

備考 このウィンドウのタイトルバー上には、プロジェクト・ファイルを読み込んだ場合、“プロジェクト・ファイル名+数字（0から連番）.pnl”を表示します。

ただし、PM+ のプロジェクト・ファイルを読み込み、CubeSuite+ のプロジェクト・ファイルに保存した場合は、それ以降、“プロジェクト・ファイル名+CS +数字（0から連番）.pnl”と表示します。


図 A—102 入出力パネル ウィンドウ



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]
- [専用メニュー／ツールバー（入出力パネル ウィンドウ）]
- [コンテキスト・メニュー]
- [操作方法]

[オープン方法]

-  ボタンをクリック
- [シミュレータ] メニュー→ [入出力パネル ...] を選択

[各エリアの説明]

(1) クライアント・エリア

疑似的なターゲット・システムを構築するために、使用する接続部品（図形オブジェクト／部品オブジェクト）の作成、および設定を行うエリアです（[\[操作方法\]](#) 参照）。

[専用メニュー／ツールバー（入出力パネル ウィンドウ）]

入出力パネル ウィンドウに関する操作を行うメニュー項目、ツールバー上のボタンは次のとおりです。

(1) [編集] メニュー

作成したオブジェクトに対して基本的な編集操作を行う場合に選択します。

元に戻す	オブジェクトの移動等、直前に行った操作を元に戻します。 変更内容の復帰は、5回前の状態まで可能です。
やり直し	[元に戻す] で戻した状態を復帰します。
切り取り	選択範囲を切り取りクリップボードに保存します。
コピー	選択範囲をコピーしクリップボードに保存します。
貼り付け	クリップボードの内容を貼り付けます。
削除	選択範囲を削除します。
すべて選択	ウィンドウ上のすべてのオブジェクトを選択します。
グループ化	選択しているオブジェクトをグループ化します。
グループ解除	選択しているオブジェクトのグループ状態を解除します。
最前面へ移動	選択しているオブジェクトをパネルの最前面に移動します。
最背面へ移動	選択しているオブジェクトをパネルの最背面へ移動します。
前面へ移動	選択しているオブジェクトを一つ前面へ移動します。
背面へ移動	選択しているオブジェクトを一つ背面へ移動します。

(2) [表示] メニュー



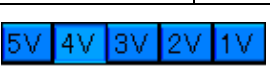



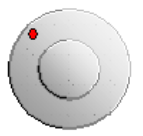



ツールバー、ステータスバーの表示状態の切り替え、およびウィンドウ内の各種情報を表示／非表示する場合に選択します。




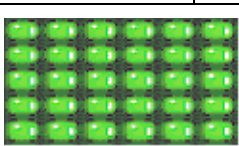



ツールバー	[図形] ツールバー／[部品] ツールバーの表示／非表示を切り替えます。
ステータスバー	ステータスバーの表示／非表示を切り替えます。
接続部品一覧 ...	部品一覧ダイアログをオープンします。 このウィンドウ上に存在するすべての図形オブジェクト／部品オブジェクトの一覧を表示します。
グリッド	グリッドの表示／非表示をします。
プロパティ	選択している図形オブジェクト／部品オブジェクトの設定ダイアログをオープンします。

(3) [部品] メニュー／[部品] ツールバー

シミュレータ GUI が提供する接続部品（部品オブジェクト）を新規作成／配置する際に選択します（「(3) 部品オブジェクトの作成」参照）。

なお、[部品] メニューの各項目は、[部品] ツールバーのボタンにより同様の動作を行うことができます。










メニュー項目	ボタン	機能
ボタン		デジタル入力用スイッチです。
例)		任意の端子に対する接続が可能で、表示されたボタンをクリックすることで接続端子へデジタル入力値を与えることができます。
アナログボタン		アナログ入力用スイッチです。
例)		任意の端子に対する接続が可能で、表示されたボタンをクリックすることで接続端子へアナログ入力値を与えることができます。
キーマトリクス		複数の端子をマトリクス状に接続し、その接点を各種のキーとみなし、キーをクリックするとある種の状態になる部品です。
例)		任意の端子に対する接続が可能で、複数のキーを使用した入力が可能です。
レベルゲージ		電圧源などのアナログ・データの入力用として、ある一定範囲のデータを加減に設定できる部品です。
例)		A/D コンバータを接続した端子に対して、指定した範囲内の任意の値を与えることができます。
LED		発光ダイオード (Light Emitting Diode) です。
例)		任意の端子に対する接続が可能で、端子の出力を LED の点灯／消灯で表示します。
7 セグメント LED		LED7 個を数字の字画に近似させ 1 つのパッケージにした部品です。
例)		桁信号に割り当てた端子の出力がアクティブ時に、対応する 7 セグメント LED を点灯／消灯で表示します。


メニュー項目	ボタン	機能
14 セグメント LED		LED14個をアルファベットの字面に近似して1つのパッケージにした部品です。
例)		桁信号に割り当てた端子の出力がアクティブ時に、対応する14セグメントLEDを点灯/消灯で表示します。
マトリクス LED		複数のLEDをマトリクス状に配置して1つのパッケージにした部品です。
例)		割り当てた端子の出力がアクティブ時に、対応する14セグメントLEDを点灯/消灯で表示します。
ブザー		ブザーです。
例)		端子と接続したブザーは、接続した端子からの出力情報をビットマップやブザー音で表します。
ブルアップ/ブルダウン設定 ...		ブルアップ/ブルダウン設定ダイアログをオープンします。 端子にブルアップ抵抗/ブルダウン抵抗を接続することができます。


(4) [図形] メニュー / [図形] ツールバー

このウィンドウの動作モードの設定、および接続部品（図形オブジェクト）を新規作成/配置する際に選択します（「(2) 図形オブジェクトの作成」参照）。

なお、[図形] メニューの各項目は、[図形] ツールバーのボタンにより同様の動作を行うことができます。

メニュー項目	ボタン	機能
選択		このウィンドウの動作モードを編集モードにします。 マウス・カーソルが矢印になり、オブジェクトの編集を可能にします。
入力シミュレーション		このウィンドウの動作モードを入力シミュレーション・モードにします。 マウス・カーソルが手の形になり、接続部品（部品オブジェクト）への入力操作を可能にします。
線		マウス・カーソルが十字 (+) になり、線の作画を可能にします。
四角形		マウス・カーソルが十字 (+) になり、四角形の作画を可能にします。
丸四角形		マウス・カーソルが十字 (+) になり、丸みを帯びた四角形の作画を可能にします。
楕円		マウス・カーソルが十字 (+) になり、楕円の作画を可能にします。
多角形		マウス・カーソルが十字 (+) になり、多角形の作画を可能にします。
扇型		マウス・カーソルが十字 (+) になり、扇型の作画を可能にします。
文字		マウス・カーソルが十字 (+) になり、文字の作成を可能にします。
ビットマップの貼り付け ...	—	選択しているビットマップ・ファイルをこのウィンドウに貼り付けます。

メニュー項目	ボタン	
線の色 ...		<p>次の色の設定 ダイアログをオープンします。 選択しているオブジェクトの線の色を、選択した色に変更します。</p> 
塗りつぶしの色 ...		<p>色の設定 ダイアログをオープンします。 選択しているオブジェクトの塗りつぶしの色を、選択した色に変更します。</p>
フォントの指定 ...	—	<p>次のフォント ダイアログをオープンします。 選択しているオブジェクトのフォントを、選択したフォントに変更します。</p> 
線のスタイル		<p>選択しているオブジェクトの線のスタイルを変更します。</p>
16pt		線の太さを 16pt に設定します。
12pt		線の太さを 12pt に設定します。
8pt		線の太さを 8pt に設定します。
4pt		線の太さを 4pt に設定します。
2pt		線の太さを 2pt に設定します。
1pt		線の太さを 1pt に設定します。
なし		線を描画しません。

メニュー項目	ボタン	
点線のスタイル		選択しているオブジェクトの線のスタイルを変更します。
実線		実線を描画します。
破線		破線を描画します。
点線		点線を描画します。
一点鎖線		一点鎖線を描画します。
二点鎖線		二点鎖線を描画します。

【コンテキスト・メニュー】

編集モード選択時、次のコンテキスト・メニューを表示します。

コピー	選択しているオブジェクトをコピーします。
貼り付け	クリップボードの内容を貼り付けます。
削除	選択しているオブジェクトを削除します。
グループ化	カスケード・メニューより選択します。
グループ化	選択しているオブジェクトをグループ化します。
グループ化解除	選択しているオブジェクトのグループ状態を解除します。
順序	カスケード・メニューより選択します。
最前面へ移動	選択しているオブジェクトをパネルの最前面に移動します。
最背面へ移動	選択しているオブジェクトをパネルの最背面へ移動します。
前面へ移動	選択しているオブジェクトを一つ前面へ移動します。
背面へ移動	選択しているオブジェクトを一つ背面へ移動します。
プロパティ	選択している図形／部品オブジェクトの設定ダイアログをオープンします。

【操作方法】


疑似的なターゲット・システムを構築するためのオブジェクト（図形オブジェクト／部品オブジェクト）の作成方法、およびそのシミュレーション方法は次のとおりです。

- (1) 編集モード
- (2) 図形オブジェクトの作成
- (3) 部品オブジェクトの作成
- (4) オブジェクトの配置
- (5) 文字の入力
- (6) オブジェクトの一覧表示
- (7) オブジェクトの詳細設定
- (8) 入力シミュレーション・モード

(1) 編集モード


オブジェクトの作成を行うためには、このウィンドウの動作モードを“編集モード”（デフォルト）に設定します。

編集モードの設定は、次のいずれかの方法により行います。

- [図形] メニュー→ [選択] を選択
- 図形ツールバーの  ボタンをクリック
- [編集] メニュー→ [すべて選択] を選択

(2) 図形オブジェクトの作成

(a) 線の描画




[図形] メニュー→ [線] を選択、またはツールバーの  ボタンをクリックします。

→マウス・カーソルが十字 (+) に変わり、線の描画が可能になります。

線の開始位置からドラッグし、終了位置でドロップします。

→線の開始位置と終了位置が直線で結ばれます（線の太さ、形状はデフォルトになります）。

(b) 四角形／丸四角形／楕円／扇型の描画

[図形] メニュー→ [四角形] / [丸四角] / [楕円] / [扇型] を選択、またはツールバーの  /  /  ボタンをクリックします。

→マウス・カーソルが十字 (+) に変わり、それぞれの描画が可能になります。


描画領域（長方形領域）の左上隅から右下隅の方向へドラッグします。

→マウス位置を右下隅とする描画領域に該当図形が表示されます。

ドロップすることで図形のサイズが確定されます。

→四角形は長方形領域と同じサイズに、その他の図形は長方形領域に納まるサイズで描画されます（線の太さ、形状はデフォルトになります）。

(c) 多角形の描画

[図形] メニュー→ [多角形] を選択、またはツールバーの  ボタンをクリックします。

→マウス・カーソルが十字 (+) に変わり、多角形の描画が可能になります。

多角形の各頂点を描画したい位置でクリックします。

→クリックした順番に各頂点が直線で結ばれます。

ダブルクリックすることで多角形の描画が終了します。

→多角形の線の太さ、形状はデフォルトになります。

(d) ビットマップの貼り付け





図形オブジェクトとして、任意のビットマップを使用することができます。

[図形] メニュー→ [ビットマップの貼り付け...] を選択したのち、貼り付けたいビットマップ・ファイル (*.bmp) を選択します。

→このウィンドウ上のデフォルト位置に該当ビットマップ・ファイルが貼り付けられます。

(e) 図形オブジェクトのスタイル変更

次のいずれかの方法により、作成した図形オブジェクトの色／線の種類などを変更することができます。

- 対象図形オブジェクトをダブルクリックすることによりオープンする [Object Properties ダイアログ](#) の [スタイル] タブ上で操作
- 対象図形オブジェクトを選択したのち、[図形] メニュー→ [線の色] / [塗りつぶしの色] / [線のスタイル] / [点線のスタイル] のいずれかを選択、またはツールバーの  /  /  /  ボタンをクリック

(3) 部品オブジェクトの作成

シミュレータ GUI が提供する接続部品を利用して、部品オブジェクトを作成することができます。

(a) 部品オブジェクトの選択

[部品] メニュー、またはツールバーから作成する部品オブジェクトを選択します。

→マウス・カーソルが十字 (+) に変わります。

任意の位置をクリックします。

→クリック位置を左上隅の位置として、該当部品オブジェクトが作成／配置されます (デフォルト・サイズ)。

(b) 部品オブジェクトのスタイル変更

対象部品オブジェクトをダブルクリックすることによりオープンする設定ダイアログの [スタイル] タブにより、作成した部品オブジェクトのスタイルを変更することができます。

なお、変更可能な項目についての詳細は、各部品オブジェクトに対応する設定ダイアログの項を参照してください (対象となる部品オブジェクトに依存します)。

(4) オブジェクトの配置

(a) グリッドの表示

[表示] メニュー→ [グリッド] の選択により、このウィンドウ上にグリッドが表示されます。

(b) オブジェクトの選択

次のいずれかの方法により、作成したオブジェクトが選択状態になります。

なお、選択状態となったオブジェクトは、周囲にトラッカーが表示されます。

- 個別選択

選択したいオブジェクトをクリック

- 複数選択

[Shift] キーを押しながら選択したいオブジェクトをクリック

- 範囲選択

選択したいオブジェクトを含む領域の左上隅からドラッグし、右下隅でドロップ

- すべてを選択

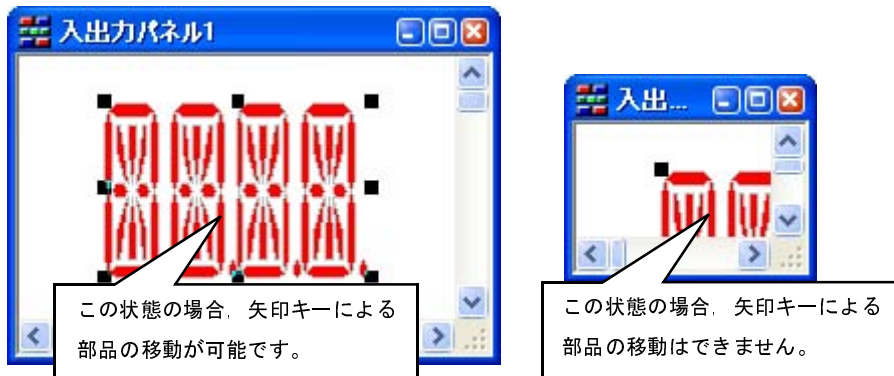
[編集] メニュー→ [すべて選択] を選択

(c) オブジェクトの移動

対象オブジェクトを選択したのち（複数選択可）、そのままドラッグし、移動先でドロップします。

備考 オブジェクトの移動は、矢印キーを使用することもできます。

ただし、ウィンドウを縮小し選択部品を半分以上隠した状態では、矢印キーでの選択部品の移動はできません。

**(d) オブジェクトのサイズ変更**

対象オブジェクトを選択したのち、表示されるトラッカーをそのままドラッグします。

(e) オブジェクトの切り取り／コピー／貼り付け／削除／グループ化／グループ解除

対象オブジェクトを選択したのち、[編集]メニューから該当項目を選択することで行います。

(f) オブジェクトの順序変更（最前面へ移動／最背面へ移動／前面へ移動／背面へ移動）

対象オブジェクトを選択したのち、[編集]メニューから該当項目を選択することで行います。

(5) 文字の入力

[図形]メニュー→[文字]を選択、またはツールバーの **A** ボタンをクリックします。

→マウス・カーソルが十字 (+) に変わります。

文字描画領域（長方形領域）の左上隅からドラッグし右下隅でドロップします。

→この長方形領域が文字描画領域となります。

文字描画領域内をクリックします。

→カーソルが表示され、文字入力が可能になります。

(6) オブジェクトの一覧表示

このウィンドウ上で作成した図形オブジェクト、および部品オブジェクトは、このウィンドウ上での表示以外に [表示]メニュー→[接続部品一覧...]の選択により、一覧表示することができます。

(7) オブジェクトの詳細設定

作成したオブジェクトには、使用するターゲット・システムに準じた詳細設定（端子接続情報など）が必要です。

(a) 図形オブジェクト

詳細設定は、図形オブジェクトをダブルクリックことによりオープンする [Object Properties ダイアログ](#) の [端子接続] タブで行います。

オブジェクトと出力端子を接続することにより、接続端子の出力状態により図形オブジェクトの表示／非表示を切り替えることができます。

(b) 部品オブジェクト

詳細設定は、部品オブジェクトをダブルクリックことによりオープンする設定ダイアログの [xxx 端子接続] タブで行います。


設定可能な項目についての詳細は、各部品オブジェクトに対応する設定ダイアログの項を参照してください（対象となる部品オブジェクトに依存します）。

(8) 入力シミュレーション・モード

詳細設定が完了した部品オブジェクトは、ユーザがシミュレーション中に操作することができるため（シミュレータに対して入力値を与えることができます）、その入出力結果をこのウィンドウ上で確認することができます。

部品オブジェクトの操作を行うためには、このウィンドウの動作モードを入力シミュレーション・モードに設定します。

入力シミュレーション・モードの設定は、次のいずれかの方法により行います（マウス・カーソルが手の形になります）。

- [図形] メニュー→ [入力シミュレーション] を選択
- 図形ツールバーの  ボタンをクリック

備考 入力操作についての詳細は、各部品オブジェクトに対応する設定ダイアログの項を参照してください。

Parts Button Properties ダイアログ

入出力パネル ウィンドウの接続部品の一つであるボタンの端子接続情報の設定、変更を行います。

入力シミュレーション・モード時、端子と接続したボタンからシミュレータに対して入力操作が可能になります。

なお、ボタンの表示スタイルには、図形とビットマップの2種類があり、これらスタイルの変更は [[スタイル] タブ] で行います。

図 A—103 Parts Button Properties ダイアログ : [ボタン端子接続] タブ



図 A—104 Parts Button Properties ダイアログ : [スタイル] 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]
- [[ボタン端子接続] タブ]
- [[スタイル] タブ]
- [機能ボタン]
- [表示例と操作方法]

[オープン方法]

入出力パネル ウィンドウにおける次のいずれか

- 部品オブジェクト “ボタン” をダブルクリック
- 部品オブジェクト “ボタン” のコンテキスト・メニューより [プロパティ ...] を選択
- 部品オブジェクト “ボタン” を選択したのち、[表示] メニュー→ [プロパティ ...] を選択

[[ボタン端子接続] タブ]


(1) ボタン端子接続設定エリア

ラベル	部品に名前を付ける際に指定します。 ここで指定した名前はボタン上に表示されます。また、 部品一覧 ダイアログ 上でラベルとして表示されます。	
端子接続	接続する端子名を指定するエリアです。 指定は、ドロップダウン・リストからの選択、または直接入力により行います。	
アクティブレベル	アクティブ状態をオプション・ボタンにより選択します。	
	LOW	アクティブ・レベルを LOW に設定します。
	HIGH	アクティブ・レベルを HIGH に設定します (デフォルト)。
種類	ボタンの種類をオプション・ボタンにより選択します。	
	プッシュ	プッシュ・ボタン にします (デフォルト)。 [保有時間] での指定が必要となります。
	トグル	トグル・ボタン にします。
	グループ	グループ・ボタン にします。 [グループ] での指定が必要となります。
グループ	ボタンのグループ名を入力します。 このエリアは、[種類] で [グループ] を選択した時のみ有効です。	
保有時間	入力した値を保持させる時間 (保有時間) を指定します (デフォルト : 0.5 ミリ秒)。 指定可能範囲は 0.001 ~ 999 ミリ秒です。 このエリアは、[種類] で [プッシュ] を選択した時のみ有効です。	
CPU リセット時	CPU リセット時のボタンの状態を指定します。	
	CPU リセット前を維持	CPU リセット時、ボタンの状態を維持します。
	インアクティブ	CPU リセット時、ボタンが押されていない状態にします (デフォルト)。
	アクティブ	CPU リセット時、ボタンが押された状態にします。

備考 指定する端子名に関しては、使用するマイクロコントローラのユーザーズ・マニュアルを参照してください。

[[スタイル] タブ]

(1) スタイル情報設定エリア

図形	ボタンを図形で表示する場合、このオプション・ボタンを選択します。		
	形状	図形の形状（四角、楕円のいずれか）を選択します。	
	影	選択不可	
	線	図形の線に関する指定、変更を行います。 プルダウン・ボタンをクリックすることにより色の指定が可能です。	
		太さ	線の太さを指定します。 スピン・ボタンでの選択、または直接入力により行います。 1～100 までの範囲での指定が可能です。
		アクティブ	アクティブ表示時の線の色を指定します。
		インアクティブ	インアクティブ表示時の線の色を指定します。
	塗りつぶし	図形の塗りつぶしに関する指定、変更を行います。 プルダウン・ボタンをクリックすることにより色の指定が可能です。	
アクティブ		アクティブ表示時の塗りつぶしの色を指定	
インアクティブ		インアクティブ表示時の塗りつぶしの色を指定	
ビットマップ	ボタンを指定したビットマップで表示する場合、このオプション・ボタンを選択します（デフォルト）。		
	選択リスト	使用するビットマップを選択します（リストには、現在選択可能なビットマップが表示されます）。	
	[追加] ボタン	次のビットマップの追加 ダイアログがオープンし、選択リストに新規にビットマップを追加します。[...] ボタンによるファイル選択、または直接入力によりファイルを指定します。 	
	[削除] ボタン	現在選択リストで選択しているビットマップを削除します。 ただし、ユーザにより追加されたビットマップのみ削除可能です。	

(2) プレビュー・エリア

現在設定しているボタンのスタイルを表示します。

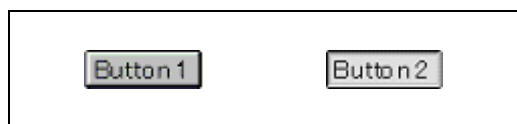
[機能ボタン]

ボタン	機能
OK	設定を有効にし、このダイアログをクローズします。
キャンセル	設定を無視し、このダイアログをクローズします。
適用	選択不可
ヘルプ	このダイアログのヘルプを表示します。

[表示例と操作方法]

入カシミュレーション・モード時、ボタンの押下により、接続した端子へのデータ入力が可能になります。
なお、ボタンの種類により入力形態が異なります。

図 A—105 接続部品表示例（ボタン）



プッシュ・ボタン	ボタンの押下により、接続した端子にはアクティブ値が取り込まれます。 アクティブ値は、保有時間の間、保持され、保有時間を過ぎると元に戻ります。
トグル・ボタン	ボタンの押下により、接続した端子にはアクティブ値が取り込まれます。 アクティブ値は、再度同一のボタンが押されるまでの間、保持されます。
グループ・ボタン	ボタンの押下により、接続した端子にはアクティブ値が取り込まれます。 同じグループ名を持つグループ・ボタンの値は元に戻ります。

Analog Button Properties ダイアログ

入出力パネル ウィンドウの接続部品の一つであるアナログ・ボタンの端子接続情報の設定、変更を行います。

入力シミュレーション・モード時、端子と接続したアナログ・ボタンからは、シミュレータに対して入力操作が可能になります。

なお、アナログ・ボタンの表示スタイルには、図形とビットマップの2種類があり、これらスタイルの変更は [[スタイル] タブ] で行います。

図 A—106 Analog Button Properties ダイアログ : [アナログボタン端子接続] タブ



図 A—107 Analog Button Properties ダイアログ : [スタイル] 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]
- [[アナログボタン端子接続] タブ]
- [[スタイル] タブ]
- [機能ボタン]
- [表示例と操作方法]

[オープン方法]

入出力パネル ウィンドウにおける次のいずれか

- 部品オブジェクト “アナログ・ボタン” をダブルクリック
- 部品オブジェクト “アナログ・ボタン” のコンテキスト・メニューより [プロパティ ...] を選択
- 部品オブジェクト “アナログ・ボタン” を選択したのち、[表示] メニュー→ [プロパティ ...] を選択

[[アナログボタン端子接続] タブ]

(1) アナログ・ボタン端子接続設定エリア

ラベル	部品に名前を付ける際に指定します。 ここで指定した名前はボタン上に表示されます。また、 部品一覧ダイアログ 上でラベルとして表示されます。	
端子接続	接続する端子名を指定するエリアです。 指定は、ドロップダウン・リストからの選択、または直接入力により行います。	
アクティブ	アクティブ状態を指定します。	
	チェック・ボックス	チェックした数だけアナログ・ボタンが作成されます。
	ラベル	各アナログ・ボタンに表示する名前を直接入力します。
	電圧 (mV)	各アナログ・ボタンをクリックした際、入力される電圧 (単位: mV) を直接入力します。
配置	ボタンの並びをオプション・ボタンで指定します。 上記アクティブ・エリアで作成したアナログ・ボタン数が2個以上の場合、この設定が有効になります。アナログ・ボタン数が1個以下の場合、この設定は無視されます。	
	横列	アナログ・ボタンを横並びに配置します (デフォルト)。
	縦列	アナログ・ボタンを縦並びに配置します。
CPU リセット時	CPU リセット時のアナログ・ボタンの状態を指定します。	
	CPU リセット前を維持	CPU リセット時、ボタンの状態を維持します。
	インアクティブ	CPU リセット時、ボタンが押されていない状態にします (デフォルト)。
	'xxx'yyy(mV) をアクティブ	“'xxx'yyy(mV)” で指定されたアナログ・ボタンが CPU リセット後に押された状態になります。
インアクティブ	すべてのアナログ・ボタンが押されていない場合の入力レベルを指定します (単位: mV)。	

備考 指定する端子名に関しては、使用するマイクロコントローラのユーザーズ・マニュアルを参照してください。

[[スタイル] タブ]

(1) スタイル情報設定エリア

図形	アナログ・ボタンを図形で表示する場合、このオプション・ボタンを選択します。	
	形状	図形の形状（四角、楕円のいずれか）を選択します。
	影	選択不可
	線	図形の線に関する指定、変更を行います。 プルダウン・ボタンをクリックすることにより色の指定が可能です。
	太さ	線の太さを指定します。 スピン・ボタンでの選択、または直接入力により行います。 1～100 までの範囲での指定が可能です。
	アクティブ	アクティブ表示時の線の色を指定します。
	インアクティブ	インアクティブ表示時の線の色を指定します。
	塗りつぶし	図形の塗りつぶしに関する指定、変更を行います。 プルダウン・ボタンをクリックすることにより色の指定が可能です。
アクティブ	アクティブ表示時の塗りつぶしの色を指定します。	
インアクティブ	インアクティブ表示時の塗りつぶしの色を指定します。	
ビットマップ	アナログ・ボタンを指定したビットマップで表示する場合、このオプション・ボタンを選択します（デフォルト）。	
	選択リスト	使用するビットマップを選択します（リストには、現在選択可能なビットマップが表示されます）。
	[追加] ボタン	次のビットマップの追加 ダイアログがオープンし、選択リストに新規にビットマップを追加します。[...] ボタンによるファイル選択、または直接入力によりファイルを指定します。
	[削除] ボタン	現在選択リストで選択しているビットマップを削除します。 ただし、ユーザにより追加されたビットマップのみ削除可能です。



(2) プレビュー・エリア

現在設定しているアナログ・ボタンのスタイルを表示します。

[機能ボタン]

ボタン	機能
OK	設定を有効にし、このダイアログをクローズします。
キャンセル	設定を無視し、このダイアログをクローズします。
適用	選択不可
ヘルプ	このダイアログのヘルプを表示します。

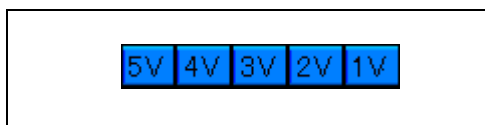
[表示例と操作方法]

入カシミュレーション・モード時、アナログ・ボタンを押下することにより、接続した端子へのアナログ電圧値入力が可能になります。

なお、一度に押すことができるボタンは1つのみです。

アナログ・ボタンが押されている間、設定した端子へ設定したアナログ電圧値が入力されます。押下状態のアナログ・ボタンは同じボタンを再度押すことで元の状態に戻ります。

図 A—108 接続部品表示例（アナログ・ボタン）



Parts Key Properties ダイアログ

入出力パネル ウィンドウの接続部品の一つであるキー・マトリクス端子接続情報の設定、変更を行います。
 入力シミュレーション・モード時、端子と接続したキーからは、シミュレータに対して入力操作が可能になります。
 入力端子と出力端子によるキー・マトリクスは、最大 16 × 16 まで設定可能です。
 なお、キー・マトリクスの表示スタイルには、図形とビットマップの 2 種類があり、これらスタイルの変更は [\[スタイル\] タブ](#) で行います。

注意 キー・マトリクスを端子に接続する際は、接続端子のプルアップ/プルダウン設定も合わせて行う必要があります。キーの押下時、キーと接続された入力端子には、該当キーに接続された出力端子の出力値が入力されます。なお、キーの非押下時の値は、[プルアップ/プルダウン設定 ダイアログ](#)で指定した値になります。プルアップ/プルダウンの設定を行わない場合、入力端子はハイ・インピーダンス状態になります。このため、入力端子に接続された機能の動作は不定となります。

図 A—109 Parts Key Properties ダイアログ : [キーマトリクス端子接続] タブ



図 A—110 Parts Key Properties ダイアログ: [スタイル] 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]
- [[キーマトリクス端子接続] タブ]
- [[スタイル] タブ]
- [機能ボタン]
- [表示例と操作方法]

[オープン方法]

入出力パネル ウィンドウにおける次のいずれか

- 部品オブジェクト “キー・マトリクス” をダブルクリック
- 部品オブジェクト “キー・マトリクス” のコンテキスト・メニューより [プロパティ ...] を選択
- 部品オブジェクト “キー・マトリクス” を選択したのち、[表示] メニュー→ [プロパティ ...] を選択

[[キーマトリクス端子接続] タブ]

(1) 端子接続情報設定エリア

ラベル	<p>部品に名前を付ける際に指定します。</p> <p>ここで指定した名前は部品一覧 ダイアログ上でラベルとして表示されます。</p>
-----	---


接続端子	接続する端子名（入力端子、出力端子）を指定するエリアです。 指定は、ドロップダウン・リストからの選択、または直接入力により行います。 このエリアは、スクロール・バーにより 16 × 16 端子の設定が可能です。	
	In0 ~ In15	入力端子を指定します。
	Out0 ~ Out15	出力端子を指定します。
	N00 ~ Nff	キー・マトリクス上に表示する文字列を直接入力により指定します。 任意の長さの文字列が指定可能です。 デフォルトの記述文字列（N 数字）は、キー上には表示されません。
保有時間	入力した値を保持させる時間（保有時間）を指定します（デフォルト：0.5 ミリ秒）。 指定可能範囲は 0.001 ~ 999 ミリ秒です。 なお、保有時間内に同じ入力端子に入力されるキーを複数個、押下した場合には、最後にクリックしたキーが有効になります。	
CPU リセット時	CPU リセット時のキー・マトリクスの動作を指定します。	
	CPU リセット前を維持	CPU リセット時、キー・マトリクスの状態が変化しません。
	インアクティブ	CPU リセット時、キー・マトリクスがすべて押されていない状態になります（デフォルト）。

備考 指定する端子名に関しては、使用するマイクロコントローラのユーザーズ・マニュアルを参照してください。

[[スタイル] タブ]

(1) スタイル情報設定エリア

図形	キー・マトリクスを図形で表示する場合、このオプション・ボタンを選択します。		
	形状	図形の形状（四角、楕円のいずれか）を選択します。	
	影	選択不可	
	線	図形の線に関する指定、変更を行います。 プルダウン・ボタンをクリックすることにより色の指定が可能です。	
		太さ	線の太さを指定します。 スピン・ボタンでの選択、または直接入力により行います。 1 ~ 100 までの範囲での指定が可能です。
		アクティブ	アクティブ表示時の線の色を指定します。
		インアクティブ	インアクティブ表示時の線の色を指定します。
	塗りつぶし	図形の塗りつぶしに関する指定、変更を行います。 プルダウン・ボタンをクリックすることにより色の指定が可能です。	
		アクティブ	アクティブ表示時の塗りつぶしの色を指定します。
		インアクティブ	インアクティブ表示時の塗りつぶしの色を指定します。

ビットマップ	キー・マトリクスを指定したビットマップで表示する場合、このオプション・ボタンを選択します（デフォルト）。
選択リスト	使用するビットマップを選択します（リストには、現在選択可能なビットマップが表示されます）。
[追加] ボタン	次のビットマップの追加 ダイアログがオープンし、選択リストに新規にビットマップを追加します。[...] ボタンによるファイル選択、または直接入力によりファイルを指定します。 
[削除] ボタン	現在選択リストで選択しているビットマップを削除します。 ただし、ユーザにより追加されたビットマップのみ削除可能です。

(2) プレビュー・エリア

現在設定しているキー・マトリクスのスタイルを表示します。

[機能ボタン]

ボタン	機能
OK	設定を有効にし、このダイアログをクローズします。
キャンセル	設定を無視し、このダイアログをクローズします。
適用	選択不可
ヘルプ	このダイアログのヘルプを表示します。

[表示例と操作方法]

入力シミュレーション・モード時、次の操作を行うことができます。

- (1) 同時に複数個のキーを操作する
- (2) キーの入力値をロックする

(1) 同時に複数個のキーを操作する

同時に入力したいキーの片方をマウスの右ボタンでクリックし、待ち状態にします。続いて残りのキーをクリックすることにより、先の待ち状態が解除され両方のキーを同時に入力することができます。複数のキーを待ち状態にすることにより、複数のキーの同時入力が可能になります。

ただし、同じ入力端子への入力になる場合には、後から入力したキーが有効となります。

(2) キーの入力値をロックする

任意のキーに対してマウスの右ボタンを押しながら、マウスの左ボタンをクリックすることにより、その時のキーの入力値がロックされます。ロック状態中に、ロックされたキーと同じ入力端子への入力になるキーがクリックされた場合には、後から入力したキーの入力値が有効になりますが、そのキーの保有時間が経過すると再度ロック状態時の入力値になります。

ロック状態のキーをクリックすることによりロックが解除され、右ボタンでクリックすることにより待ち状態になります。

図 A—111 接続部品表示例（キー・マトリクス）

1	2	3
4	5	6
7	8	9
10	11	12
+	-	予約
<<	再生	>>
多重	停止	録画
ポーズ	取消	電源

Parts Level Gauge Properties ダイアログ

入出力パネル ウィンドウの接続部品の一つであるレベル・ゲージの端子接続情報の設定、変更を行います。

入力シミュレーション・モード時、端子と接続したレベル・ゲージからは、シミュレータに対して入力操作が可能になります。ただし、接続する端子はアナログ入力用端子に限ります。

なお、レベル・ゲージの表示スタイルには、スライド式とダイヤル式の2種類があり、これらスタイルの変更は [[スタイル] タブ] で行います。

図 A—112 Parts Level Gauge Properties ダイアログ : [レベルゲージ端子接続] タブ



図 A—113 Parts Level Gauge Properties ダイアログ : [スタイル] 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]
- [[レベルゲージ端子接続] タブ]
- [[スタイル] タブ]
- [機能ボタン]
- [表示例と操作方法]

[オープン方法]

入出力パネル ウィンドウにおける次のいずれか

- 部品オブジェクト “レベル・ゲージ” をダブルクリック
- 部品オブジェクト “レベル・ゲージ” のコンテキスト・メニューより [プロパティ ...] を選択
- 部品オブジェクト “レベル・ゲージ” を選択したのち、[表示] メニュー→ [プロパティ ...] を選択

[[レベルゲージ端子接続] タブ]


(1) 端子接続情報設定エリア

ラベル	部品に名前を付ける際に指定します。 ここで指定した名前は部品一覧ダイアログ上でラベルとして表示されます。	
接続端子	接続する端子名を指定するエリアです。 指定は、ドロップダウン・リストからの選択、または直接入力により行います。	
最大入力値	レベル・ゲージ入力の最大値を指定します（デフォルト：5000 mV）。 指定は mV 単位で行ってください。 指定可能範囲は 0 ~ 65535 です。 この指定値により、入出力パネル ウィンドウに表示されたレベル・ゲージの動作範囲が決定します。	
CPU リセット時	CPU リセット時のレベル・ゲージの動作を指定します。	
	CPU リセット前を維持	CPU リセット直前の状態を CPU リセット後も維持します。
	初期電圧を指定	CPU リセット時、レベル・ゲージが指定した値に設定されます（デフォルト）。 指定は mV 単位で行ってください。 指定可能範囲は 0 ~ 最大入力値で指定した値です（デフォルト：0 mV）。

備考 指定する端子名に関しては、使用するマイクロコントローラのユーザーズ・マニュアルを参照してください。

[[スタイル] タブ]

(1) スタイル情報設定エリア

スライド式	レベル・ゲージをスライド式で表示する場合、このオプション・ボタンを選択します。	
	方向	スライド方向（垂直、水平）をドロップダウン・リストから選択します。
	色	スライドの色を指定、変更します。 プルダウン・ボタンをクリックすることにより色の指定が可能です。
ダイアル式	レベル・ゲージをダイアル式で表示する場合このオプション・ボタンを選択します（デフォルト）。	
	マークの色	動作点を示すマークの色を指定、変更します。 プルダウン・ボタンをクリックすることにより色の指定が可能です。
	選択リスト	使用するビットマップを選択します（リストには、現在選択可能なビットマップが表示されます）。
	[追加] ボタン	次のビットマップの追加 ダイアログがオープンし、選択リストに新規にビットマップを追加します。[...] ボタンによるファイル選択、または直接入力によりファイルを指定します。
		
[削除] ボタン	現在選択リストで選択しているビットマップを削除します。 ただし、ユーザにより追加されたビットマップのみ削除可能です。	

(2) プレビュー

現在設定しているレベル・ゲージのスタイルを表示します。

[機能ボタン]

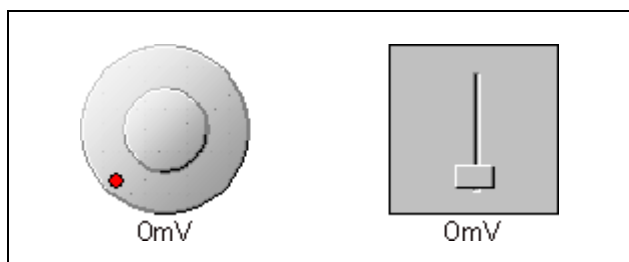
ボタン	機能
OK	設定を有効にし、このダイアログをクローズします。
キャンセル	設定を無視し、このダイアログをクローズします。
適用	選択不可
ヘルプ	このダイアログのヘルプを表示します。

[表示例と操作方法]

入カシミュレーション・モード時、表示されたスライダ、またはダイヤルを操作することにより、レベル・ゲージからのアナログ入力を行います。

[最大入力値] で指定した値が入力できる最大値になります。

図 A—114 接続部品表示例（レベル・ゲージ）



ダイヤル式レベル・ゲージ	設ダイヤル上の動作点（赤い丸）をドラッグすることにより、表示しているアナログ値が変化します。このアナログ値が入力したい値になった時、動作点からマウスを離します。これにより、表示しているアナログ値を入力が可能になります。なお、動作点の移動は任意の位置をクリックすることによっても可能です。
スライド式レベル・ゲージ	スライダのつまみをドラッグし動かすことにより表示しているアナログ値が変化します。このアナログ値が入力したい値になった時、つまみからマウスを離します。これにより、表示しているアナログ値の入力が可能になります。なお、つまみの移動は任意の位置をクリックすることによっても可能です。

注意 ダイヤル上の動作点（赤い丸）、またはスライダのつまみをドラッグしたのち、レベル・ゲージから離れた場所でドロップすると、レベル・ゲージに表示される電圧は変更されますが実際にレベル・ゲージから出力される電圧は変更されません。ドラッグ・アンド・ドロップは、必ずレベル・ゲージの上で行ってください。

Parts Led Properties ダイアログ

入出力パネル ウィンドウの接続部品の一つである LED の端子接続情報の設定、変更を行います。

入カシミュレーション・モード時、端子と接続した LED は、シミュレータからの出力情報を点灯／消灯で表示します。

なお、LED の表示スタイルには、図形とビットマップの2種類があり、これらスタイルの変更は [[スタイル] タブ]で行います。

図 A—115 Parts Led Properties ダイアログ : [LED 端子接続] タブ

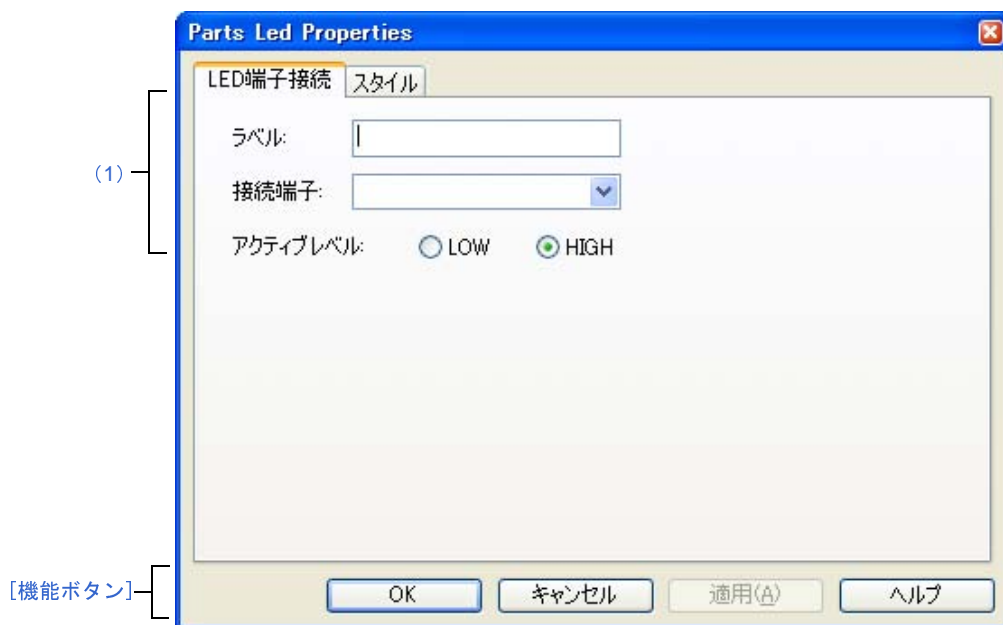


図 A—116 Parts Led Properties ダイアログ : [スタイル] タブ



ここでは、次の項目について説明します。

- [オープン方法]
- [[LED 端子接続] タブ]
- [[スタイル] タブ]
- [機能ボタン]
- [表示例と操作方法]

[オープン方法]

入出力パネル ウィンドウにおける次のいずれか

- 部品オブジェクト “LED” をダブルクリック
- 部品オブジェクト “LED” のコンテキスト・メニューより [プロパティ ...] を選択
- 部品オブジェクト “LED” を選択したのち、[表示] メニュー→ [プロパティ ...] を選択

[[LED 端子接続] タブ]


(1) 端子接続情報設定エリア

ラベル	部品に名前を付ける際に指定します。 ここで指定した名前は部品一覧ダイアログ上でラベルとして表示されます。	
接続端子	接続する端子名（出力端子）を指定するエリアです。 指定は、ドロップダウン・リストからの選択、または直接入力により行います。	
アクティブレベル	アクティブ状態をオプション・ボタンにより選択します。	
	LOW	アクティブ・レベルを LOW に設定します。
	HIGH	アクティブ・レベルを HIGH に設定します（デフォルト）。

備考 指定する端子名に関しては、使用するマイクロコントローラのユーザーズ・マニュアルを参照してください。

[[スタイル] タブ]

(1) スタイル情報設定エリア

図形	LED を図形で表示する場合、このオプション・ボタンを選択します。		
	形状	図形の形状（四角、楕円のいずれか）を選択します。	
	影	選択不可	
	線	図形の線に関する指定、変更を行います。 プルダウン・ボタンをクリックすることにより色の指定が可能です。	
		太さ	線の太さを指定します。 スピン・ボタンでの選択、または直接入力により行います。 1～100 までの範囲での指定が可能です。
		アクティブ	アクティブ表示時の線の色を指定します。
		インアクティブ	インアクティブ表示時の線の色を指定します。
	塗りつぶし	図形の塗りつぶしに関する指定、変更を行います。 プルダウン・ボタンをクリックすることにより色の指定が可能です。	
		アクティブ	アクティブ表示時の塗りつぶしの色を指定します。
		インアクティブ	インアクティブ表示時の塗りつぶしの色を指定します。
ビットマップ	LED を指定したビットマップで表示する場合、このオプション・ボタンを選択します（デフォルト）。		
	選択リスト	使用するビットマップを選択します（リストには、現在選択可能なビットマップが表示されます）。	
	[追加] ボタン	次のビットマップの追加 ダイアログがオープンし、選択リストに新規にビットマップを追加します。[...] ボタンによるファイル選択、または直接入力によりファイルを指定します。 	
	[削除] ボタン	現在選択リストで選択しているビットマップを削除します。 ただし、ユーザにより追加されたビットマップのみ削除可能です。	

(2) プレビュー・エリア

現在設定している LED のスタイルを表示します。

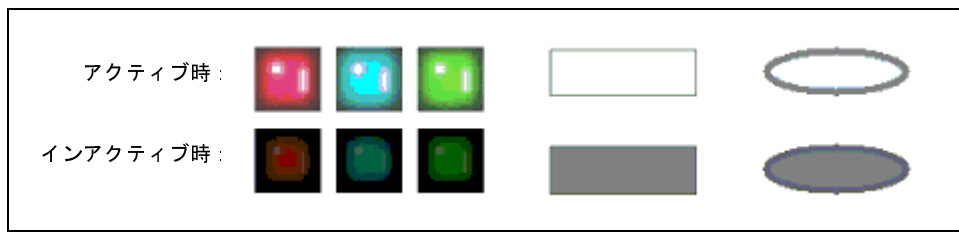
[機能ボタン]

ボタン	機能
OK	設定を有効にし、このダイアログをクローズします。
キャンセル	設定を無視し、このダイアログをクローズします。
適用	選択不可
ヘルプ	このダイアログのヘルプを表示します。

[表示例と操作方法]

入カシミュレーション・モード時、接続した端子の出力状態（アクティブ／インアクティブ）を2種類のビットマップ、または図形でリアルタイムに表示します。

図 A—117 接続部品表示例（LED）



Parts Segment LED Properties ダイアログ

入出力パネル ウィンドウの接続部品の一つである7セグメントLED、および14セグメントLEDの端子接続情報の設定、変更を行います。

入カシミュレーション・モード時、端子と接続した各LEDは、シミュレータからの出力情報を表示します。セグメントLEDの表示スタイルの変更は [[スタイル] タブ] で行います。

図 A-118 Parts Segment LED Properties ダイアログ : [セグメントLED 端子接続] タブ



図 A-119 Parts Segment LED Properties ダイアログ : [スタイル] 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]
- [[セグメント LED 端子接続] タブ]
- [[スタイル] タブ]
- [機能ボタン]
- [表示例と操作方法]

[オープン方法]

入出力パネル ウィンドウにおける次のいずれか

- 部品オブジェクト “7セグメント LED” / “14セグメント LED” をダブルクリック
- 部品オブジェクト “7セグメント LED” / “14セグメント LED” のコンテキスト・メニューより [プロパティ ...] を選択
- 部品オブジェクト “7セグメント LED” / “14セグメント LED” を選択したのち、[表示] メニュー → [プロパティ ...] を選択

[[セグメント LED 端子接続] タブ]

(1) 端子接続情報設定エリア

ラベル	部品に名前を付ける際に指定します。 ここで指定した名前は 部品一覧 ダイアログ 上でラベルとして表示されます。	
グリッドの割り当て	グリッド信号の割り当て方法を次の中から選択します。 この選択により、 桁信号設定エリア での設定が変化します。	
	桁	セグメント LED の 1 桁を 1 つのグリッド端子と接続します (デフォルト)。 桁信号設定エリア では、桁信号の設定を行います。グリッド信号は 16 桁の指定が可能で、1 つのセグメント LED 部品で最大 16 桁のセグメント LED を作成可能です。
	カスタマイズ	選択不可

(2) セグメント信号設定エリア

セグメント信号	7セグメントLED/14セグメントLEDのセグメント信号と接続する端子（出力端子）、およびアクティブ・レベルを指定するエリアです。	
図示	左上には7セグメントLED/14セグメントLEDのビットマップを表示します。 [接続端子]を入力する際、対応する位置を示します。	
接続端子	接続する端子をドロップダウン・リストからの選択、または直接入力により指定します。 接続するセグメント端子数は、7セグメントLEDでは8個、14セグメントLEDでは15個となります。 なお、右側にあるスクロール・バーを操作することにより、すべてのセグメント端子に接続可能です。	
アクティブレベル	アクティブ状態をオプション・ボタンにより選択します。	
	LOW	アクティブ・レベルをLOWに設定します。
	HIGH	アクティブ・レベルをHIGHに設定します（デフォルト）。

備考 指定する端子名に関しては、使用するマイクロコントローラのユーザーズ・マニュアルを参照してください。

(3) 桁信号設定エリア

桁信号	7セグメントLED/14セグメントLEDの桁、またはグリッド信号と接続する端子（出力端子）、およびアクティブ・レベルを指定するエリアです。 [グリッドの割り当て]での指定により、接続の仕方が次のように変化します。 - [桁] 選択時 桁信号の設定を行います。接続する桁端子の数は最大16です。 エリア右側にあるスクロール・バーを操作することにより、すべての桁端子に接続可能です。 - [カスタマイズ] 選択時 選択不可	
接続端子	接続する端子名をドロップダウン・リストからの選択、または直接入力により指定します。 なお、設定信号は、最下位桁からの連続端子を指定してください。	
アクティブレベル	アクティブ状態をオプション・ボタンにより選択します	
	LOW	アクティブ・レベルをLOWに設定します。
	HIGH	アクティブ・レベルをHIGHに設定します（デフォルト）。

備考 指定する端子名に関しては、使用するマイクロコントローラのユーザーズ・マニュアルを参照してください。

[[スタイル] タブ]

(1) スタイル情報設定エリア

セルの塗りつぶし	各セルの塗りつぶしに関する設定、変更を行うエリアです。 プルダウン・ボタンをクリックすることにより色の指定が可能です。	
	アクティブ色	アクティブ表示時の塗りつぶしの色を指定します。
	インアクティブ色	インアクティブ表示時の塗りつぶしの色を指定します。
セルの枠線	セルの枠線の形状に関する設定、変更を行うエリアです。	
	色	線の色を指定、変更します。 プルダウン・ボタンをクリックすることにより色の指定が可能です。
	点線／実線	線の形状（点線／実線）を指定、変更します。 ドロップダウン・リストからの選択により行います。 [線の太さ]での指定が“1”の時のみ指定可能です。
	線の太さ	線の太さを指定、変更します。 スピン・ボタンでの選択、または直接入力により行います。 0.1 ~ 100 までの範囲での指定が可能です。

[機能ボタン]

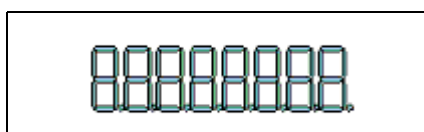
ボタン	機能
OK	設定を有効にし、このダイアログをクローズします。
キャンセル	設定を無視し、このダイアログをクローズします。
適用	選択不可
ヘルプ	このダイアログのヘルプを表示します。

[表示例と操作方法]

入カシミュレーション・モード時、1シミュレーションの結果、接続端子の出力情報を受け取りその値にしたがって表示します。

桁／グリッド信号、およびセグメント信号ともにアクティブ出力の際、対応する桁／グリッドのセグメント LED が点灯します。

図 A—120 接接続部品表示例 (7セグメントLED)



Parts Matrix Led Properties ダイアログ

入出力パネル ウィンドウの接続部品の一つであるマトリクス LED の端子接続情報の設定、変更を行います。

入力シミュレーション・モード時、端子と接続したマトリクス LED は、シミュレータからの出力情報を点灯／消灯で表示します。

なお、マトリクス LED の表示スタイルには、図形とビットマップの2種類があり、これらスタイルの変更は [[スタイル] タブ] で行います。

図 A—121 Parts Matrix Led Properties ダイアログ : [マトリクス LED 端子接続] タブ



図 A—122 Parts Matrix Led Properties ダイアログ : [スタイル] 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]
- [[マトリクス LED 端子接続] タブ]
- [[スタイル] タブ]
- [機能ボタン]
- [表示例と操作方法]

[オープン方法]

入出力パネル ウィンドウにおける次のいずれか

- 部品オブジェクト “マトリクス LED” をダブルクリック
- 部品オブジェクト “マトリクス LED” のコンテキスト・メニューより [プロパティ ...] を選択
- 部品オブジェクト “マトリクス LED” を選択したのち、[表示] メニュー→ [プロパティ ...] を選択

[[マトリクス LED 端子接続] タブ]

(1) ラベル設定エリア

ラベル	部品に名前を付ける際に指定します。 ここで指定した名前は 部品一覧 ダイアログ 上でラベルとして表示されます。
-----	--

(2) 行方向信号設定エリア

行方向信号	マトリクス LED の行方向の信号と接続する端子（出力端子）、およびアクティブ・レベルを指定するエリアです。	
接続端子	接続する端子名をドロップダウン・リストからの選択、または直接入力により指定します。 接続する端子数は最大 16 です。右側にあるスクロール・バーを操作することにより、すべての行方向信号に接続可能です。	
アクティブレベル	アクティブ状態をオプション・ボタンオプション・ボタンにより選択します。	
	LOW	アクティブ・レベルを LOW に設定します。
	HIGH	アクティブ・レベルを HIGH に設定します（デフォルト）。

備考 指定する端子名に関しては、使用するマイクロコントローラのユーザーズ・マニュアルを参照してください。

(3) 列方向信号設定エリア


列方向信号	マトリクス LED の列方向信号と接続する端子（出力端子）、およびアクティブ・レベルを指定するエリアです。		
接続端子	接続する端子名をドロップダウン・リストからの選択、または直接入力により指定します。接続する端子数は最大 16 です。右側にあるスクロール・バーを操作することにより、すべての列方向信号に接続可能です。		
アクティブレベル	アクティブ状態をオプション・ボタンにより選択します。		
	LOW	アクティブ・レベルを LOW に設定します。	
	HIGH	アクティブ・レベルを HIGH に設定します（デフォルト）。	

備考 指定する端子名に関しては、使用するマイクロコントローラのユーザーズ・マニュアルを参照してください。

[[スタイル] タブ]

(1) スタイル情報設定エリア

図形	マトリクス LED を図形で表示する場合、このオプション・ボタンを選択します。		
形状	図形の形状（四角、楕円のいずれか）を選択します。		
影	選択不可		
線	図形の線に関する指定、変更を行います。 プルダウン・ボタンをクリックすることにより色の指定が可能です。		
	太さ	線の太さを指定します。 スピン・ボタンでの選択、または直接入力により行います。 1 ~ 100 までの範囲での指定が可能です。	
	アクティブ	アクティブ表示時の線の色を指定します。	
	インアクティブ	インアクティブ表示時の線の色を指定します。	
	塗りつぶし	図形の塗りつぶしに関する指定、変更を行います。 プルダウン・ボタンをクリックすることにより色の指定が可能です。	
アクティブ		アクティブ表示時の塗りつぶしの色を指定します。	
インアクティブ		インアクティブ表示時の塗りつぶしの色を指定します。	

ビットマップ	マトリクス LED を指定したビットマップで表示する場合、このオプション・ボタンを選択します (デフォルト)。
選択リスト	使用するビットマップを選択します (リストには、現在選択可能なビットマップが表示されます)。
[追加] ボタン	次のビットマップの追加 ダイアログがオープンし、選択リストに新規にビットマップを追加します。[...] ボタンによるファイル選択、または直接入力によりファイルを指定します。 
[削除] ボタン	現在選択リストで選択しているビットマップを削除します。ただし、ユーザにより追加されたビットマップのみ削除可能です。

(2) プレビュー・エリア

現在設定しているマトリクス LED のスタイルを表示します。

[機能ボタン]

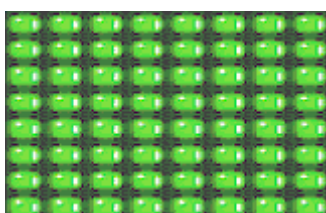
ボタン	機能
OK	設定を有効にし、このダイアログをクローズします。
キャンセル	設定を無視し、このダイアログをクローズします。
適用	選択不可
ヘルプ	このダイアログのヘルプを表示します。

[表示例と操作方法]

入力シミュレーション・モード時、1 シミュレーションの結果、接続端子の出力情報を受け取りその値にしたがって表示します。

行方向端子と列方向端子のマトリクス上での交点で両方の端子がアクティブの際、対応する LED が点灯します。

図 A—123 接続部品表示例 (マトリクス LED)



Parts Buzzer Properties ダイアログ

入出力パネル ウィンドウの接続部品の一つであるブザーの端子接続情報の設定、変更を行います。

入力シミュレーション・モード時、端子と接続したブザーは、接続した端子からの出力情報をビットマップで表示します（表示確認のみ）。

なお、ブザーの表示スタイルには、図形とビットマップの2種類があり、これらスタイルの変更は [[スタイル] タブ] で行います。

図 A—124 Parts Buzzer Properties ダイアログ：[ブザー端子接続] タブ



図 A—125 Parts Buzzer Properties ダイアログ：[スタイル] 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]
- [[ブザー端子接続] タブ]
- [[スタイル] タブ]
- [機能ボタン]
- [表示例と操作方法]

[オープン方法]

入出力パネル ウィンドウにおける次のいずれか

- 部品オブジェクト “ブザー” をダブルクリック
- 部品オブジェクト “ブザー” のコンテキスト・メニューより [プロパティ ...] を選択
- 部品オブジェクト “ブザー” を選択したのち、[表示] メニュー→ [プロパティ ...] を選択

[[ブザー端子接続] タブ]


(1) ブザー端子接続設定エリア

ラベル	部品に名前を付ける際に指定します。 ここで指定した名前は部品一覧ダイアログ上でラベルとして表示されます。	
接続端子	接続する端子名（出力端子）を指定するエリアです。 指定は、ドロップダウン・リストからの選択、または直接入力により行います。	
アクティブレベル	アクティブ状態をオプション・ボタンにより選択します。	
	LOW	アクティブ・レベルを LOW に設定します。
	HIGH	アクティブ・レベルを HIGH に設定します（デフォルト）。
出力形態	この項目は、変更不可です。	

備考 指定する端子名に関しては、使用するマイクロコントローラのユーザーズ・マニュアルを参照してください。

[[スタイル] タブ]

(1) スタイル情報設定エリア

図形	ブザーを図形で表示する場合、このオプション・ボタンを選択します。		
	形状	図形の形状（四角、楕円のいずれか）を選択します。	
	影	選択不可	
	線	図形の線に関する指定、変更を行います。 プルダウン・ボタンをクリックすることにより色の指定が可能です。	
		太さ	線の太さを指定します。 スピン・ボタンでの選択、または直接入力により行います。 1～100までの範囲での指定が可能です。
		アクティブ	アクティブ表示時の線の色を指定します。
		インアクティブ	インアクティブ表示時の線の色を指定します。
	塗りつぶし	図形の塗りつぶしに関する指定、変更を行います。 プルダウン・ボタンをクリックすることにより色の指定が可能です。	
		アクティブ	アクティブ表示時の塗りつぶしの色を指定します。
インアクティブ		インアクティブ表示時の塗りつぶしの色を指定します。	
ビットマップ	ブザーを指定したビットマップで表示する場合、このオプション・ボタンを選択します（デフォルト）。		
	選択リスト	使用するビットマップを選択します（リストには、現在選択可能なビットマップが表示されます）。	
	[追加] ボタン	次のビットマップの追加ダイアログがオープンし、選択リストに新規にビットマップを追加します。[...] ボタンによるファイル選択、または直接入力によりファイルを指定します。 	
	[削除] ボタン	現在選択リストで選択しているビットマップを削除します。 ただし、ユーザにより追加されたビットマップのみ削除可能です。	

(2) プレビュー・エリア

現在設定しているブザーのスタイルを表示します。

[機能ボタン]

ボタン	機能
OK	設定を有効にし、このダイアログをクローズします。
キャンセル	設定を無視し、このダイアログをクローズします。
適用	選択不可
ヘルプ	このダイアログのヘルプを表示します。

[表示例と操作方法]

入力シミュレーション・モード時、接続した端子のアクティブ・レベル出力をビットマップで表示します。端子の出力値（アクティブ/インアクティブ）により、次のようなビットマップとして表示されます。

図 A—126 接続部品表示例（ブザー）



プルアップ／プルダウン設定 ダイアログ

入出力パネル ウィンドウの接続部品の一つであるプルアップ／プルダウン抵抗の端子接続情報の設定、変更を行います。

この接続部品の設定方法は他の部品とは異なり、このダイアログで全端子の接続情報を一括管理します。

図 A-127 プルアップ／プルダウン設定 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]

入出力パネル ウィンドウにフォーカスがある状態で次のいずれか

-  ボタンのクリック
- [部品] メニュー → [プルアップ／プルダウン設定 ...] の選択

[各エリアの説明]

(1) 接続情報表示エリア

端子名	プルアップ／プルダウン抵抗と接続できる端子名を表示します。	
Pull Up/Pull Down	端子の接続状態を表示します。	
	Pull Up	プルアップ抵抗接続状態を示します。
	Pull Down	プルダウン抵抗接続状態を示します。
	—	プルアップ抵抗／プルダウン抵抗未接続状態を示します（デフォルト）。
ボタン	プルアップ／プルダウン抵抗の端子接続情報を変更します。	
	プルアップ	選択した端子をプルアップ抵抗と接続します。 接続すると“Pull Up”が表示されます。
	プルダウン	選択した端子をプルダウン抵抗と接続します。 接続すると“Pull Down”が表示されます。
	解除	選択した端子の接続状態を解除します。 解除が完了すると“-”が表示されます。

[機能ボタン]

ボタン	機能
OK	設定を有効にし、このダイアログをクローズします。
キャンセル	設定を無視し、このダイアログをクローズします。

Object Properties ダイアログ

入出力パネル ウィンドウの図形オブジェクト（文字／ビットマップを含む）の端子接続情報の設定、変更を行います。
 入力シミュレーション・モード時、端子と接続した各オブジェクトは、接続端子の出力状態により、表示／非表示を切り替えます。

なお、各信号のアクティブ状態は、アクティブ HIGH です。

表示スタイルの変更は [[スタイル] タブ] で行います。

図 A—128 Object Properties ダイアログ：[端子接続] タブ



図 A—129 Object Properties ダイアログ：[スタイル] 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]
- [[端子接続] タブ]
- [[スタイル] タブ]
- [機能ボタン]

[オープン方法]

入出力パネル ウィンドウにおける次のいずれか

- 図形オブジェクトをダブルクリック
- 図形オブジェクトのコンテキスト・メニューより [プロパティ ...] を選択
- 図形オブジェクトを選択したのち、[表示] メニュー→ [プロパティ ...] を選択

[[端子接続] タブ]

(1) 端子接続設定エリア

接続端子	オブジェクトと出力端子との接続方法をオプション・ボタンにより選択し、各出力端子名を指定します。 接続することにより、接続した出力端子の ON/OFF 状態で図形の表示が切り替わります。	
	接続なし	オブジェクトを端子と接続しません（デフォルト）。 端子に接続していないオブジェクトは、常に表示状態です。
	スタティック接続	オブジェクトを1つの出力端子と接続します。 [出力信号] に、接続する端子名をドロップダウン・リストからの選択、または直接入力により指定します。 シミュレーション中に、指定した端子の出力信号データがアクティブの際、オブジェクトが表示されます。
	ダイナミック接続	オブジェクトを2つの出力端子と接続します。 [出力信号 1] / [出力信号 2] に接続する端子名をドロップダウン・リストからの選択、または直接入力により指定します。 シミュレーション中に、指定した端子の出力信号データがともにアクティブの際、オブジェクトが表示されます。
アクティブレベル	各出力信号共通のアクティブ状態をオプション・ボタンにより選択します。	
	LOW	アクティブ・レベルを LOW に設定します。
	HIGH	アクティブ・レベルを HIGH に設定します（デフォルト）。

備考 指定する端子名に関しては、使用するマイクロコントローラのユーザーズ・マニュアルを参照してください。

[[スタイル] タブ]

(1) スタイル情報設定エリア

塗りつぶし	各オブジェクトの塗りつぶしに関する設定、変更を行うエリアです ^注 。 対象オブジェクトにより、塗りつぶしをする範囲は次のようになります。		
	<ul style="list-style-type: none"> - 線 <ul style="list-style-type: none"> 対象外 - 四角, 楕円, 丸四角 輪郭となる線で囲まれた範囲内 - 多角形 <ul style="list-style-type: none"> 各頂点を結ぶ線で囲まれた範囲内 - 文字 <ul style="list-style-type: none"> テキスト・ボックス内 - ビットマップ <ul style="list-style-type: none"> 図形描画領域内 		
色	塗りつぶしの色を指定、変更します。 プルダウン・ボタンをクリックすることにより色の指定が可能です。		
線	各オブジェクトの線の形状に関する設定、変更を行うエリアです。 各オブジェクトにより、線の定義は次のようになります。		
	<ul style="list-style-type: none"> - 線 <ul style="list-style-type: none"> すべて - 四角, 楕円, 丸四角 輪郭となる線 - 多角形 <ul style="list-style-type: none"> 各頂点を結ぶ線 - 文字 <ul style="list-style-type: none"> テキスト・ボックスの輪郭線 - ビットマップ <ul style="list-style-type: none"> 図形描画領域の輪郭線 		
	色	線の色を指定、変更します。 プルダウン・ボタンをクリックすることにより色の指定が可能です。	
	点線/実線	線の形状（点線/実線）をドロップダウン・リストにより指定、変更します。 ただし、[線の太さ]での指定が“1”の時のみ変更可能です。	
	線の太さ	線の太さを指定、変更します。 スピン・ボタンでの選択、または直接入力により行います。 1～100までの範囲での指定が可能です。	

注 対象オブジェクトがビットマップ・ファイルからの貼り付けの場合、表示していたビットマップは見えなくなります。

[機能ボタン]

ボタン	機能
OK	設定を有効にし、このダイアログをクローズします。
キャンセル	設定を無視し、このダイアログをクローズします。
適用	選択不可
ヘルプ	このダイアログのヘルプを表示します。

部品一覧 ダイアログ

入出力パネル ウィンドウ上に作成したすべての図形オブジェクト、および部品オブジェクトの端子接続状況を一覧表示します。

なお、各オブジェクトの端子接続の設定内容を変更するには、一覧内のオブジェクトをダブルクリック、または一覧内のオブジェクトを選択した状態で、[表示]メニュー→[プロパティ...]を選択することによりオープンする設定ダイアログで行います。

図 A—130 部品一覧 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]

- 入出力パネル ウィンドウにフォーカスがある状態で、[表示]メニュー→[接続部品一覧...]を選択

[各エリアの説明]

(1) 端子接続状況表示エリア

ラベル	オブジェクトに付けたラベル（名前）を表示します。ラベルがないオブジェクトでは何も表示されません。	
部品の種類	部品の種類を表示します。	
	rectangle	直線、四角形、楕円、丸四角、扇型
	polygon	多角形
	text	テキスト
	bitmap	ビットマップ
	button	プッシュ・ボタン、トグル・ボタン、グループ・ボタン
	analog button	アナログ・ボタン
	key	キー・マトリクス
	level gauge	レベル・ゲージ
	led	LED
	7segment led	7セグメント LED
	14segment led	14セグメント LED
	matrix led	マトリクス LED
	buzzer	ブザー
groups	グループ化された部品	
接続端子	部品と接続されている端子を表示します。 なお、複数端子と接続されている部品では“-”が、接続されていない部品では“(空白)”が表示されます。	
アクティブ値	部品に設定されているアクティブ値を表示します。 なお、複数端子と接続されている部品では“-”が、接続されていない部品では“(空白)”が表示されます。	

[機能ボタン]

ボタン	機能
閉じる	このダイアログをクローズします。

シリアル ウィンドウ

CPUに搭載されているシリアル・インタフェースと通信を行います。

このウィンドウはCPUの他局側のシリアル・インタフェースとして動作するため、CPUからの送信データがこのウィンドウでの受信データに、このウィンドウからの送信データがCPUでの受信データになります。

このウィンドウでは、次の2種類のファイルを扱うことができます。

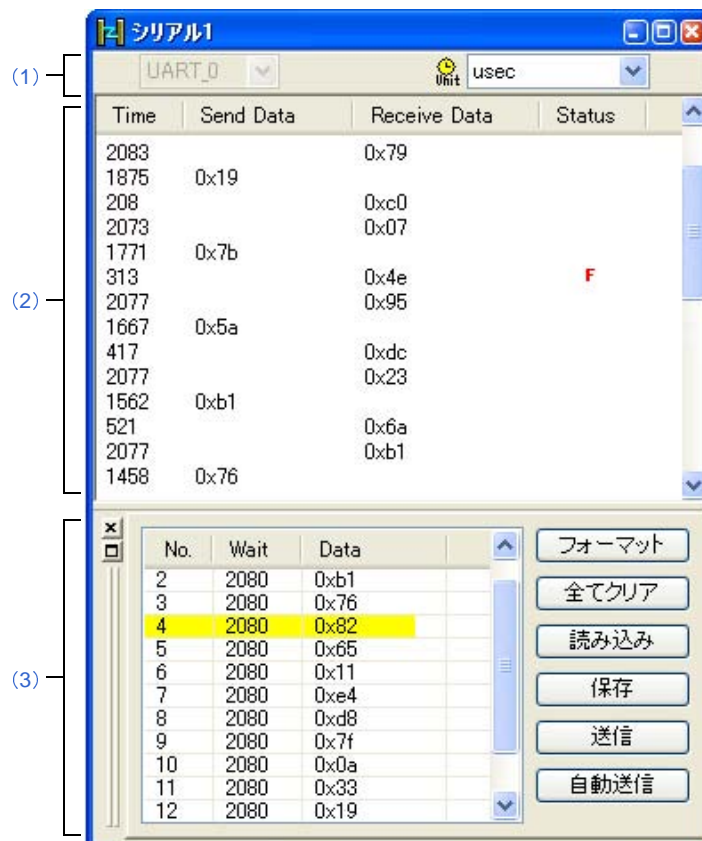
ウィンドウ上部エリアで表示された送受信データは、[ファイル]メニュー→[上書き保存] / [名前を付けて保存...]の選択により、シリアル・ログデータ・ファイル (*.log) (CSV形式)として保存することができます。

また、ウィンドウ下部エリアで作成した送信データは、[保存]ボタンによりシリアル送信データ・ファイル (*.ser) (CSV形式)として保存することができます ([読み込み]ボタンにより復元可能)。

なお、作成したデータの保存/復元はプロジェクト・ファイルの保存/ロードでも行うことができますが、この場合、送信データはCSV形式のテキスト・ファイルとして保存されず、プロジェクト・ファイル内に保存されます。

- 注意 1. 保存したシリアル送信データ・ファイルをオープンする際、またはプロジェクト・ファイルをオープンする際に、シリアル送信データ・ファイルを作成した時点のマイクロコントローラとは異なるマイクロコントローラでシミュレータ GUI が起動されていた場合、そのマイクロコントローラに存在しないシリアルの設定は復元されません。
2. このウィンドウは複数オープン可能です。ウィンドウのオープン後に、[シリアル選択エリア](#)で検証したいシリアル・インタフェースを選択してください。


図 A—131 シリアル ウィンドウ



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]
- [専用メニュー（シリアル ウィンドウ）]
- [コンテキスト・メニュー]



[オープン方法]

-  ボタンをクリック
- [シミュレータ] メニュー → [シリアル] を選択

[各エリアの説明]

(1) シリアル選択エリア

通信するシリアル・インタフェースを選択します。

	搭載されているシリアル・インタフェースの一覧（ドロップダウン・リスト）から通信するシリアル・インタフェースを選択します ^注 。
	シリアル・エディタ・エリアの [Wait] に使用される時間情報の単位をドロップダウン・リストから選択します。 なお、時間情報の単位は [編集] メニュー → [時間単位] から変更できます。

注 一度選択すると、変更することはできません。

変更する場合は、新たなシリアル ウィンドウをオープンしてください。

(2) ログ表示エリア

送受信データを表示します。

表示タイミングは、データを構成する全ビットの受信、または送信が完了した時点です。

このエリアには、スタート・ビット、ストップ・ビット、およびパリティ・ビットを削除したデータのみが表示されます。

データの表示方法は、[表示] メニュー → [数値表現] → [2進表示] / [16進表示] の選択により変更することができます。

なお、デバッグ、またはシミュレータのリセット発生によりログ表示はクリアされます。

Time	前データの送受信終了から今回の送受信終了までの時間を表示します。 時間情報の単位は、[編集] メニュー → [時間単位] により選択した単位となります。	
Send Data	このウィンドウが送信したデータ（CPU 側が受信したデータ）を表示します。	
Receive Data	このウィンドウが受信したデータ（CPU 側が送信したデータ）を表示します。	
Status	データ受信時の状態を表示します。 エラーが発生時には次のマークを表示します。正常時は何も表示しません。	
	P	パリティ・エラー（パリティ・ビットの不一致）
	F	フレーミング・エラー（ストップ・ビットが検出されない）

(3) シリアル・エディタ・エリア

送信データを作成するエリアです。

このエリアは、[表示]メニュー→[シリアル・エディタ]により表示/非表示の選択が可能です。

No.	先頭からシーケンシャルに付けられている行番号です。直接書き込みはできません。 最大 9999 行まで設定可能です。	
Wait	ひとつ前のデータ送信完了から次のデータを送信開始するまでの間の時間を指定します。[自動送信]ボタンでの送信時に有効です。 時間情報の単位は、[編集]メニュー→[時間単位]により選択した単位となります。 Wait 値の入力は、編集を行う Wait 欄にカーソルを置きダブルクリックすることにより行います。 1回の操作で1つ分のWait値の書き込みが可能です。	
Data	送信データを編集するエリアです。 Data 欄にカーソルを置きダブルクリックすることにより、直接書き込み可能です。 先頭に“0x”と付与すると16進数として、“0b”を付与すると2進数として扱われます。デフォルト進数は16進数です。 フォーマット (UART) ダイアログ、またはフォーマット (CSI) ダイアログで設定したビット長と異なるビット長を指定した場合、下位ビットからのデータが有効になります。 1回の操作で1つ分のデータ書き込みが可能です。	
ボタン	フォーマット	フォーマット (UART) ダイアログ、またはフォーマット (CSI) ダイアログをオープンします。
	全てクリア	シリアル・エディタ・エリアをすべて空白にします。
	読み込み	以前に保存したシリアル送信データ・ファイル (*.ser) の内容を読み込み、シリアル・エディタ・エリアに再現します。 ただし、UART 用として作成されたファイルを CSI 用として読み込むことはできません。また、CSI 用として作成されたファイルを UART 用として読み込むこともできません。
	保存	シリアル・エディタ・エリアで設定した送信データの内容を指定したシリアル送信データ・ファイル (*.ser) に保存します。
	送信	シリアル・エディタ・エリアで選択されているデータを1つ送信します。 送信の完了により次のデータが選択状態になります。 データが選択されてない場合、先頭のデータを送信します。
	自動送信	シリアル・エディタ・エリアで選択されているデータを先頭に、下方向にデータを自動送信します。データ送信の時間間隔は Wait に指定した時間となります。

注意 このエリア内にカーソルがある場合、[F1] キーを押下してもこのウィンドウのヘルプは表示されません。

備考 このウィンドウの CSI をマスタ・モードとして設定する場合、受信時にクロックが必要になるため、受信動作を行うためにダミー・データを送信する必要があります。

[専用メニュー (シリアル ウィンドウ)]

(1) [編集] メニュー

挿入	選択行の直前に新しい行を挿入します。
切り取り	選択範囲を切り取りクリップボードに保存します。
コピー	選択範囲をコピーしクリップボードに保存します。
貼り付け	クリップボードの内容を選択位置に貼り付けます。
削除	選択範囲を削除します。
時間単位	時間の単位を選択します。
メインクロック	ウエイト時間の単位をメイン・クロックとします (デフォルト)。
マイクロ秒	ウエイト時間の単位をマイクロ秒とします。
ミリ秒	ウエイト時間の単位をミリ秒とします。
フォーマット設定 ...	フォーマット (UART) ダイアログ, またはフォーマット (CSI) ダイアログをオープンします。

(2) [表示] メニュー

シリアル・エディタ	シリアル・エディタ・エリアの表示/非表示を切り換えます。
数値表現	ログ表示エリアの表示方法を変更します。
2進数表示	2進数表示します。
16進数表示	16進数表示します。

(3) [オプション] メニュー

ウィンドウのカスタマイズ ...	書式設定 ダイアログをオープンします。
------------------	---------------------

[コンテキスト・メニュー]

シリアル・エディタ・エリアにおいて、次に示すコンテキスト・メニューを表示します。

挿入	選択行の直前に新しい行を挿入します。
切り取り	選択範囲を切り取りクリップボードに保存します。
コピー	選択範囲をコピーしクリップボードに保存します。
貼り付け	クリップボードの内容を選択位置に貼り付けます。
削除	選択範囲を削除します。

フォーマット (UART) ダイアログ

アシンクロナス・シリアル・インタフェース (UART) 用のシリアル・フォーマットの設定, 変更を行います。

図 A—132 フォーマット (USRT) 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]

UART 系シリアル・インタフェースを選択している [シリアル ウィンドウ](#) における次のいずれか

- [フォーマット] ボタンをクリック
- [編集] メニュー → [フォーマット設定 ...] を選択

[各エリアの説明]

(1) フォーマット (UART) 設定エリア

ボー・レート	シリアルのボー・レート値 (単位 : bps) を整数で直接入力します。	
転送方向	転送方向を選択します。	
	MSB ファースト	転送方向を MSB ファーストとします。
	LSB ファースト	転送方向を LSB ファーストとします (デフォルト)。
データ・ビット長	送信データのビット長をドロップダウン・リストからの選択、または直接入力により指定します (デフォルト : 7)。	
ストップ・ビット長	ストップ・ビット長をドロップダウン・リストから選択します (デフォルト : 1)。	
パリティ	パリティ情報 (パリティなし (デフォルト) / 奇数パリティ / 偶数パリティ / 0 パリティ) を選択します。	
繰り返し	シリアル ウィンドウの [自動送信] ボタンをクリックした際に、データ転送の繰り返しを行う場合、チェックします。	
	<input checked="" type="checkbox"/>	自動送信で最後のデータを送信した後、データの先頭に戻って自動送信を続けます。
	<input type="checkbox"/>	自動送信で最後のデータを送信した後、送信を停止します (デフォルト)。

備考 選択可能な範囲については、使用するマイクロコントローラのユーザーズ・マニュアルを参照してください。

[機能ボタン]

ボタン	機能
OK	設定を有効にし、このダイアログをクローズします。
キャンセル	設定を無視し、このダイアログをクローズします。

フォーマット (CSI) ダイアログ

3 線式シリアル・インタフェース (CSI) 用のシリアル・フォーマットの設定, 変更を行います。

図 A-133 フォーマット (CSI) 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]
- [3 線式シリアル・インタフェース (CSI) 選択時の送受信動作に関して]

[オープン方法]

CSI 系シリアル・インタフェースを選択しているシリアル ウィンドウにおける次のいずれか

- [フォーマット] ボタンをクリック
- [編集] メニュー → [フォーマット設定 ...] を選択

[各エリアの説明]

(1) シリアル・フォーマット設定エリア

マスタ、スレーブ	転送モードを選択します。	
	マスタ	このウィンドウ側をマスタとして動作します。 通信の際クロックを生成するため [転送クロック] の設定が必要です。
	スレーブ	このウィンドウ側をスレーブとして動作します (デフォルト)。 CPU 搭載のシリアル・インタフェースのクロックで通信します。
転送クロック	転送クロック値を直接入力します (単位: kHz)。 小数点付き数値の設定も可能です。 マスタを選択時には必ず設定します。	
転送方向	転送方向を選択します。	
	MSB ファースト	転送方向を MSB ファーストとします (デフォルト)。
	LSB ファースト	転送方向を LSB ファーストとします。
データ・ビット長	送信データのビット長をドロップダウン・リストからの選択、または直接入力により指定します (デフォルト: 8)。	
データ位相	送受信のタイミングを選択することでデータ位相を設定します。 [クロック位相] との組み合わせにより、「表 A—23 データ・クロック位相設定表」で示すデータ・クロック位相となります。	
	通常	3 線式シリアルの通常の送受信タイミングでデータの送受信を行います (デフォルト)。
	先行	3 線式シリアルの通常の送受信タイミングより、動作クロックの半クロック分先行したタイミングでデータの送受信を行います。
クロック位相	送受信のクロック波形を選択することでクロック位相を設定します。 [データ位相] との組み合わせにより、「表 A—23 データ・クロック位相設定表」で示すデータ・クロック位相となります。	
	通常	3 線式シリアルの通常のクロックで動作します (クロックの立ち下がりを転送開始とします) (デフォルト)。
	反転	3 線式シリアルの通常のクロックを反転したクロックで動作します (クロックの立ち上がりを転送開始とします)。
繰り返し	シリアル ウィンドウの [自動送信] ボタンをクリックした際に、データ転送の繰り返しを行う場合、チェックします。	
	<input checked="" type="checkbox"/>	自動送信で最後のデータを送信したのち、データの先頭に戻って自動送信を続けます。
	<input type="checkbox"/>	自動送信で最後のデータを送信したのち、送信を停止します (デフォルト)。

備考 選択可能な範囲については、使用するマイクロコントローラのユーザーズ・マニュアルを参照してください。

表 A—23 データ・クロック位相設定表

データ 位相設定	クロック 位相設定	データ・クロック位相
通常	通常	
先行	通常	
通常	反転	
先行	反転	

[機能ボタン]

ボタン	機能
OK	設定を有効にし、このダイアログをクローズします。
キャンセル	設定を無視し、このダイアログをクローズします。

[3 線式シリアル・インタフェース (CSI) 選択時の送受信動作に関して]

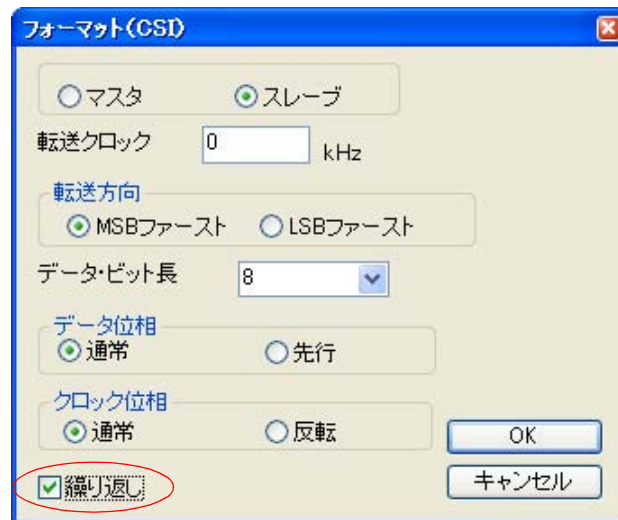
CSI系シリアル・インタフェース選択時のシリアルウィンドウは、このダイアログで [マスタ]、または [スレーブ] のどちらを選択していても、次のように常に送受信モードで動作します。

[マスタ] 選択時	シリアル・エディタ・エリアの [送信] ボタン、または [自動送信] ボタンの押下後すぐにデータの送受信が行われます。
[スレーブ] 選択時	シリアル・エディタ・エリアの [送信] ボタン、または [自動送信] ボタンの押下によりデータ送受信レディ状態となります。 データ送受信レディ状態では、CSIクロック信号を受け取ることでデータの送受信が開始され、データの送受信の終了によりこの状態が解除されます (データ送受信レディ状態以外の状態では、CSIクロック信号を受け取ってもデータの送受信は行われません)。

このため、[スレーブ] 選択時に、**シリアル ウィンドウ**でデータ受信のみを行いたい場合には、次の手順で操作してください（CSI シリアル・スレーブ選択時の受信設定方法）。

(1) [繰り返し] の指定

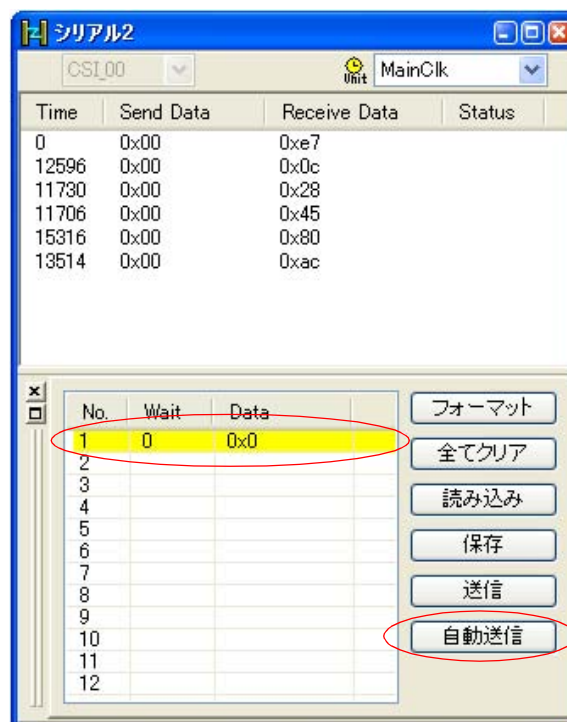
[繰り返し] ボタンをチェックします。



(2) Wait 時間の設定

シリアル ウィンドウで Wait 時間 0 のダミー設定を行います。

(3) [自動送信] ボタンのクリック



付録B ユーザ・オープン・インタフェース

この付録では、シミュレータ GUI が提供する機能の 1 つである、ユーザ・オープン・インタフェースについての詳細を説明します。

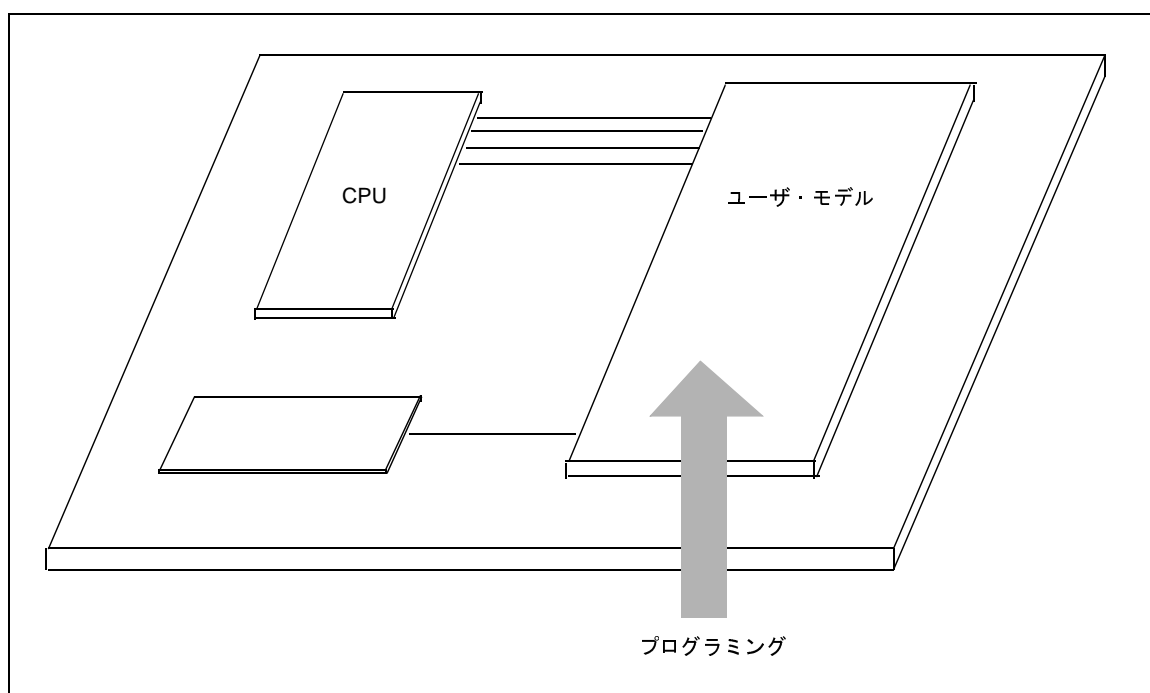
B.1 概要

シミュレータ GUI では、CPU（CPU コア+内蔵周辺）のシミュレーションに加え、次に挙げる 2 つのターゲット・システムのシミュレーション環境構築手段を用意しています。

1 つは、**入出力パネル ウィンドウ**を使用する方法で、標準的な接続部品とその操作環境を提供することにより、GUI 操作による簡単なシミュレーション環境を構築することができます。

もう 1 つが、この付録で説明するユーザ・オープン・インタフェースを使用する方法で、ユーザが外部のユーザ・モデルをプログラミングすることにより、**入出力パネル ウィンドウ**では実現不可能なシミュレーション環境を構築することができます。

図 B—1 ユーザ・モデルのプログラミング・イメージ



B. 1.1 インタフェース関数の種類

シミュレータ GUI のユーザ・オープン・インタフェースでは、次の種類のインタフェース関数を提供しています。各関数についての詳細は、「B. 4 提供インタフェース関数」を参照してください。

表 B—1 提供インタフェース関数の種類

種類	概要
基本インタフェース関数	シミュレータの基本機能 - 初期化通知 - リセット通知など
時間インタフェース関数	ユーザ・モデルの時系列処理を行うための周期タイマ機能 - タイマの設定 - タイマの解除 - タイマの時間通知など
端子インタフェース関数	端子の入出力機能 - 端子への信号出力 - 端子への信号入力通知など
外部バス・インタフェース関数 ^注	外部バスのスレーブ機能 - 外部バス・リード・アクセス通知 - 外部バス・ライト・アクセス通知など
シリアル・インタフェース関数	シリアルの送受信機能 - シリアル・データの送信 - シリアル・データの受信通知など
信号出力器インタフェース関数	信号データ・ファイルに従った信号出力機能 - 信号データ・ファイルに従った信号出力など

注 外部バス・インタフェース関数を使用する場合は、使用する外部メモリ領域を、[メモリ・マッピングダイアログ](#)上の [メモリ種別] エリアにおいて [ターゲット・メモリ領域] に指定する必要があります。

B. 1.2 インタフェース方式

シミュレータ GUI が提供するユーザ・オープン・インタフェースのインタフェース方式は、次のとおりです。

(1) C 言語インタフェース

C 言語の API (Application Program Interface) 関数セットで構成されています。
したがって、ユーザ・モデルのプログラミングは、C 言語で行います。

(2) コールバック関数方式

システム側からプログラムを呼び出す手段として、コールバック関数方式を採用しています。

コールバック関数方式とは、プログラムで作成した関数へのポインタをあらかじめシステム側に通知しておき、システム側は、その関数へのポインタを使用してプログラムで作成した関数を呼び出す方式です。

プログラムからシステム側を呼び出す API 関数に対し、コールバック関数は、システム側からプログラムを呼び出す場合（たとえば、端子への信号入力など）に使用します。

(3) イベント・ドリブン方式

事象（イベント）の発生に従い処理を記述する、イベント・ドリブン方式を採用しています。

したがって、シミュレータ GUI 側で、初期化 /CPU リセット /端子への信号入力 /外部バス・アクセスなどの事象が発生した時点で、ユーザ・モデル側で用意したコールバック関数を呼び出します。

また、ユーザ・モデルにおいて時系列処理を行うために設けている“時間インタフェース”（タイマ機能）でも、設定した時間に到達した時点で、ユーザ・モデル側で容易したコールバック関数を呼び出します。

B.1.3 開発環境

シミュレータ GUI が提供するユーザ・オープン・インタフェースでプログラミングを行い、DLL ファイルを作成する際は、次の開発ツールを使用してください。

- Microsoft Visual C++ (Ver. 6.00 以上)

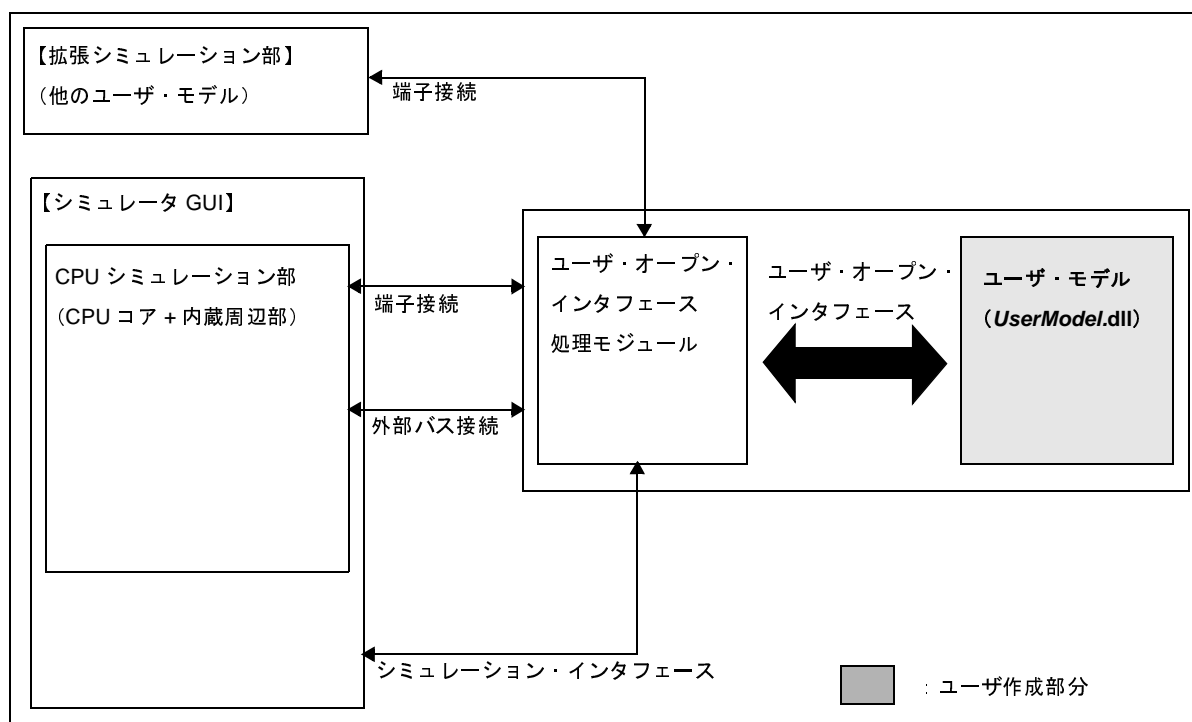
B.2 ユーザ・モデルの作成

この節では、ユーザ・モデル（UserModel.dll）の作成方法について説明します。

B.2.1 プログラム構成

次に、シミュレータ GUI が提供するユーザ・オープン・インタフェースを使用し、システムを拡張する際のプログラム構成を示します。

図 B-2 プログラム構成図



システムを拡張するためには、まず、ユーザ・モデルを作成する必要があります。

ユーザ・モデルは、シミュレーション・システムと連携動作をするため、ユーザ・オープン・インタフェース処理モジュールとインタフェースをとります。このインタフェースが、ユーザ・オープン・インタフェースとなります。

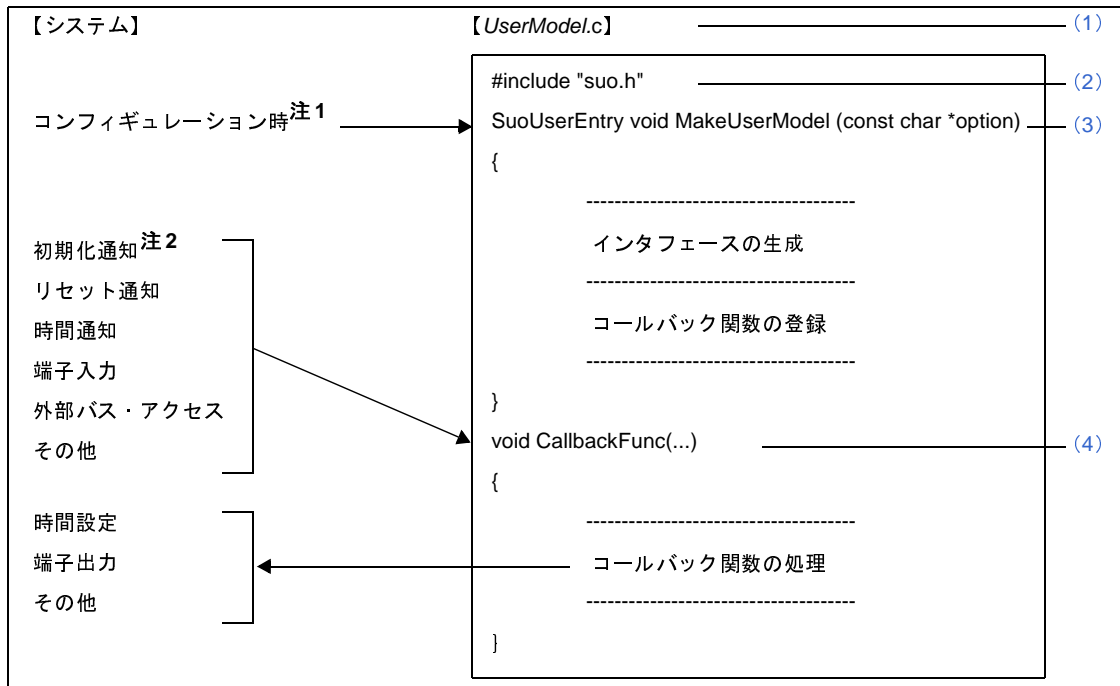
ユーザ・モデルは、ユーザ・オープン・インタフェースを介することで、コンフィギュレーション（シミュレータ GUI 起動時に行うシミュレータ構築処理）時に端子や外部バス・スレーブなどの資源を生成します。これにより生成された端子や外部バス・スレーブを CPU シミュレーション部の端子や外部バス・マスタに接続することにより、CPU シミュレーション部の端子の入出力、および CPU シミュレーション部からの外部バス・アクセスの処理が可能となります。

なお、生成された端子や外部バス・スレーブは、CPU シミュレーション部だけではなく拡張シミュレーション部（他のユーザ・モデル）とも接続することができます。

B. 2.2 ユーザ・モデルのプログラミング

ユーザ・モデルは、WIN32 ダイナミック・リンク・ライブラリ（DLL）形式でプログラミングします。
次に、プログラム・ファイル（*UserModel.c*）のテンプレートを示します。

図 B—3 プログラム・ファイルのテンプレート



注1. シミュレータ GUI 起動の際に行うシミュレータ構築処理時

2. シミュレータ構築処理が完了し、シミュレータ GUI が起動した直後の1回のみ通知されます。

(1) ファイル名

作成するユーザ・モデルのファイル名を示します。

ファイル名は任意に指定することができます。

ただし、C ソース・ファイルであるため、拡張子は“*.c”固定です。

(2) インクルード・ファイル

インクルード・ファイルを示します。

ユーザ・オープン・インタフェースを使用するためには、システム・ヘッダ・ファイル“suo.h”をインクルードする必要があります。

(3) MakeUserModel 関数

MakeUserModel 関数（シミュレータ GUI のコンフィギュレーション時にシステム側から呼び出される関数）を示します。関数名は、“MakeUserModel”である必要があります。

【指定形式】

```
SuoUserEntry void MakeUserModel(const char *option);
```

なお、この関数の中では、次の2つの処理を記述します。

(a) インタフェースの生成

シミュレータ GUI 起動時のコンフィギュレーション処理において端子やバス接続を行うため、コンフィギュレーションのタイミングで、接続する端子やバスなどの資源を生成しておく必要があります。

これを行うために、`MakeUserModel` 関数の中でインタフェースの生成関数を呼び出し、インタフェースの生成を行います（「B.4 提供インタフェース関数」参照）。これに伴い必要な資源の生成も行われます。

(b) コールバック関数の登録

必要に応じ、コールバック関数を登録することができます。

注意 初期化のコールバック関数を記述する場合は必ずこのタイミングで登録してください。このタイミングで登録しない場合、コールバックが機能しません。

これは、初期化通知が、`MakeUserModel` 関数呼び出しの次のタイミングであるためです。

(4) コールバック関数

コールバック関数を示します。

ここでは、初期化通知、リセット通知、時間通知、端子入力、外部バス・アクセスなどの複数のコールバック関数を作成することができます。コールバック関数の中ではコールバック内容に応じた処理を記述します（「B.5 ユーザ定義関数」参照）。

なお、作成したコールバック関数は、システムから呼び出しできるように事前の登録が必要です（「B.4 提供インタフェース関数」参照）。コールバック関数の名前は任意に決定ことができ、その関数の形式はコールバックの種類によって異なります。

B. 2.3 プログラム・ファイル (UserModel.c) の記述例

```
#include "suo.h"
#include <memory.h>

void Init(void);
void InputP00(SuoHandle handle, int pinValue);
void ReadBUS1(SuoHandle handle, unsigned long addr, int accessSize, unsigned char data[]);
void WriteBUS1(SuoHandle handle, unsigned long addr, int accessSize, const unsigned char data[]);

SuoHandle p00;
SuoHandle p01;
SuoHandle bus1;
unsigned char mem[0x100];

/* MakeUserModel */
SuoUserEntry void MakeUserModel(const char *option)
{
    SuoCreatePin("P00", &p00);
    SuoCreatePin("P01", &p01);
    SuoCreateExtbus("BUS1", 0x200000, 0x100, &bus1);

    SuoSetInitCallback(Init);
    SuoSetInputDigitalPinCallback(p00, InputP00);
    SuoSetReadExtbusCallback(bus1, ReadBUS1);
    SuoSetWriteExtbusCallback(bus1, WriteBUS1);
}

/* callbacks */
void Init(void)
{
    memset(mem, 0, 0x100);
}

void InputP00(SuoHandle handle, int pinValue)
{
    SuoOutputDigitalPin(p01, pinValue);
}

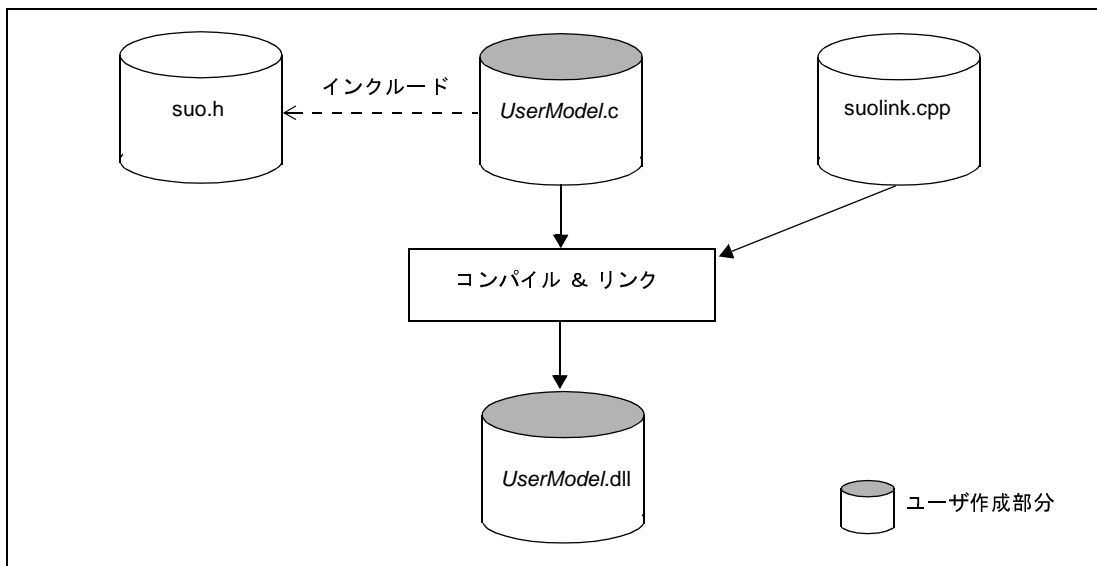
void ReadBUS1(SuoHandle handle, unsigned long addr, int accessSize, unsigned char data[])
{
    memcpy(data, &mem[addr-0x200000], accessSize);
}

void WriteBUS1(SuoHandle handle, unsigned long addr, int accessSize, const unsigned char data[])
{
    memcpy(&mem[addr-0x200000], data, accessSize);
}
```

B. 2.4 コンパイルとリンク

作成した *UserModel.c* と *suolink.cpp* をコンパイル、およびリンクすることにより、*UserModel.dll* を作成します。

図 B—4 コンパイルとリンク



ファイル名	説明
suo.h	ユーザ・オープン・インタフェース用のシステム・ヘッダ・ファイルです。 プログラム (<i>UserModel.c</i>) がインクルードするだけで、コンパイル対象ではありません。 なお、suo.h は、デフォルトで次のフォルダに格納されています。 【V850E1】【V850ES】 - インストール・フォルダ¥ CubeSuite+ ¥ DebugTools ¥ DebugToolV850Simulator ¥ useropen ¥ sys 【V850E2】 - インストール・フォルダ¥ CubeSuite+ ¥ DebugTools ¥ DebugToolV850E2RSimulator ¥ useropen ¥ sys
suolink.cpp	システムのユーザ・オープン・インタフェース処理モジュールとの動的リンク処理を行うファイルです。 なお、suolink.cpp は、デフォルトで次のフォルダに格納されています。 【V850E1】【V850ES】 - インストール・フォルダ¥ CubeSuite+ ¥ DebugTools ¥ DebugToolV850Simulator ¥ useropen ¥ sys 【V850E2】 - インストール・フォルダ¥ CubeSuite+ ¥ DebugTools ¥ DebugToolV850E2RSimulator ¥ useropen ¥ sys
<i>UserModel.c</i>	作成するユーザ・モデルの C ソース・ファイルです。 ファイル名は任意に指定することができます。
<i>UserModel.dll</i>	ユーザ・モデルのバイナリ・ファイル (DLL ファイル) です。 ファイル名は任意に指定することができます。

注意 Microsoft Visual C++ がインストールされていない環境で DLL ファイルを動作させるには、DLL ファイルをリリース版で作成する必要があります。

B.3 ユーザ・モデルの組み込み

この節では、作成したユーザ・モデル (*UserModel.dll*) をシミュレータ GUI に組み込む方法について説明します。

シミュレータ GUI への組み込みは、シミュレータ・コンフィギュレーション・ファイル (*.cfg) を使用します。

したがって、作成したユーザ・モデルを組み込んでシミュレータ GUI を使用する場合は、[プロパティパネル](#)の [接続設定] タブ上 [コンフィギュレーション] カテゴリ内 [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] プロパティにおいて、[はい] を指定したのち、同カテゴリ内の [シミュレータ・コンフィギュレーション・ファイル] プロパティにおいて、使用するシミュレータ・コンフィギュレーション・ファイルを指定する必要があります。

B.3.1 シミュレータ・コンフィギュレーション・ファイルへの記述

シミュレータ・コンフィギュレーション・ファイルには、作成したユーザ・モデルの生成処理、および端子/外部バスの接続処理などを記述します。

- (1) ユーザ・モデルの生成処理
- (2) 端子の接続処理
- (3) 外部バスの接続処理
- (4) その他の処理

(1) ユーザ・モデルの生成処理

```
UserModel1 = Device("USEROPEN", "UserModel1.dll UserOption1");
```

(a) *UserModel1*

生成したユーザ・モデルを表す変数です。変数名は任意に指定することができます。

(b) Device 関数

ユーザ・モデルを生成する関数です。

(c) "USEROPEN"

ユーザ・オープン・インタフェース処理モジュール (システム・モジュール) です。

(d) *UserModel1.dll*

「[B.2 ユーザ・モデルの作成](#)」で作成したユーザ・モデルのバイナリ・ファイル (DLL 形式) です。ファイル名は任意に指定することができます。

なお、ファイルのパスは、シミュレータ・コンフィギュレーション・ファイルが存在するフォルダからの相対パス、または絶対パスを指定します。

注意 パスの指定に、半角スペースを使用することはできません。

半角スペースを使用した場合、ユーザ・モデルを生成することはできません。

(e) UserOption1

UserModel1.dll に対するオプション文字列です。このオプションは [MakeUserModel](#) 関数の引数 “option” にそのまま渡されます。

(2) 端子の接続処理

```
wire1 = Wire(1); --- (a)
wire1 += cpu.Port("PinName1"); --- (b)
wire1 += UserModel1.Port("UserPinName1"); --- (c)
```

(a) ワイヤの生成

Wire 関数を使用し、ワイヤ（端子同士を接続する線）を生成します。

Wire 関数の引数には、必ず “1” を指定してください。

なお、*wire1* は、生成したワイヤを表す変数です。変数名は任意に指定することができます。

(b) ワイヤと CPU の接続

ワイヤの一方を CPU の端子に接続します。

“PinName1” には、接続したい CPU の外部端子名を大文字アルファベットで指定します（小文字アルファベットは使用不可）。必ず “” で囲んで指定してください。

(c) ワイヤとユーザ・モデルの接続

ワイヤのもう一方をユーザ・モデルの端子に接続します。

“UserPinName1” には、接続したいユーザ・モデルの端子名（[MakeUserModel](#) 関数の中で生成した端子名）を指定します。必ず “” で囲んで指定してください。

なお、ユーザ・モデルの複数の端子を同一のワイヤに接続する場合は、この行を追加します。

(3) 外部バスの接続処理

```
extbus1 = BUS(n); --- (a)
extbus1 += cpu.BusMasterIF("EXTBUS"); --- (b)
extbus1 += UserModel1.BusSlaveIF("UserExtbusName1"); --- (c)
```

(a) バスの生成

BUS 関数を使用し、バスを生成します。

BUS 関数の引数 *n* は、データ・バス・ビット幅を示し、8/16/32 のいずれかを指定することができます。

なお、*extbus1* は、生成したバスを表す変数です。変数名は任意に指定することができます。

(b) バスと CPU の接続

バス的一方を CPU の外部バス・マスタに接続します。

引数には、外部バス・マスタ “EXTBUS” を指定してください。

(c) バスとユーザ・モデルの接続

バスのもう一方をユーザ・モデルの外部バスに接続します。

"UserExtbusName1"には、接続したいユーザ・モデルの外部バス名（MakeUserModel関数の中で生成した外部バス名）を指定します。必ず"で囲んで指定してください。

なお、ユーザ・モデルの複数の外部バスを接続する場合は、この行を追加します。

(4) その他の処理

上記の処理以外に、ユーザ・オープン・インタフェースを動作させるためには、メイン・クロック通知端子とリセット通知端子の接続が必要となります。

```
clock1 = Wire(1); --- (a)
clock1 += cpu.DebuggerPseudoPort("debugger_pseudo_pin_main_clkout"); --- (b)
clock1 += UserModel1.Port("gui_pseudo_pin_clock_notice"); --- (c)
reset1 = Wire(1); --- (d)
reset1 += cpu.DebuggerPseudoPort("debugger_pseudo_pin_reset_notice"); --- (e)
reset1 += UserModel1.Port("gui_pseudo_pin_reset_notice"); --- (f)
```

(a) ワイヤの生成

Wire関数を使用し、ワイヤ（端子同士を接続する線）を生成します。

Wire関数の引数には、必ず"1"を指定してください。

なお、clock1は、生成したワイヤを表す変数です。変数名は任意に指定することができます。

(b) ワイヤとメイン・クロック通知端子の接続

ワイヤの一方をシミュレータ GUI のメイン・クロック通知端子に接続します。

引数には、"debugger_pseudo_pin_main_clkout"を指定してください。

(c) ワイヤとユーザ・モデルの接続

ワイヤのもう一方をユーザ・モデルの端子に接続します。

引数には、"gui_pseudo_pin_clock_notice"を指定してください。

(d) ワイヤの生成

Wire関数を使用し、ワイヤ（端子同士を接続する線）を生成します。

Wire関数の引数には、必ず"1"を指定してください。

なお、reset1は、生成したワイヤを表す変数です。変数名は任意に指定することができます。

(e) ワイヤとリセット通知端子の接続

ワイヤの一方をシミュレータ GUI のリセット通知端子に接続します。

引数には、"debugger_pseudo_pin_reset_notice"を指定してください。

(f) ワイヤとユーザ・モデルの接続

ワイヤのもう一方をユーザ・モデルの端子に接続します。

引数には、"gui_pseudo_pin_reset_notice"を指定してください。

B.3.2 シミュレータ・コンフィギュレーション・ファイルの記述例

次に、シミュレータ・コンフィギュレーション・ファイルの記述例を示します。

この例では、次表に示す接続処理を行っています。

接続の種類	CPU		ユーザ・モデル (SampleModel.dll)	
端子	"P00/INTP0"	P00 端子	"P00"	P00 の操作端子
	"P30/TXD1"	シリアル出力端子	"RXD"	シリアル入力端子
	"P31/RXD1"	シリアル入力端子	"TXD"	シリアル出力端子
外部バス	"EXTBUS"	外部バス・マスタ	"EXTBUS1"	外部バス・スレーブ 1
	"EXTBUS"	外部バス・マスタ	"EXTBUS2"	外部バス・スレーブ 2

```

cpu = CPU('a');
# -----
# SampleModel description
# -----

# Generate SampleModel.dll
model = Device("USEROPEN", "SampleModel.dll -a -b");

# Connect PIN (CPU.P00-MODEL.P00)
wire_P00 = Wire(1);
wire_P00 += cpu.Port("P00/INTP0");
wire_P00 += model.Port("P00");

# Connect PIN (CPU.TXD1-MODEL.RXD)
wire_RXD = Wire(1);
wire_RXD += cpu.Port("P30/TXD1");
wire_RXD += model.Port("RXD");

# Connect PIN (CPU.RXD1-MODEL.TXD)
wire_TXD = Wire(1);
wire_TXD += cpu.Port("P31/RXD1");
wire_TXD += model.Port("TXD");

# Connect BUS (CPU.EXTBUS-MODEL.EXTBUS1)
extbus = BUS(32);
extbus += cpu.BusMasterIF("EXTBUS");
extbus += model.BusSlaveIF("EXTBUS1");
extbus += model.BusSlaveIF("EXTBUS2");

# Connect Pseudo PIN
wire_clock = Wire(1);
wire_clock += cpu.DebuggerPseudoPort("debugger_pseudo_pin_main_clkout");
wire_clock += model.Port("gui_pseudo_pin_clock_notice");
wire_reset = Wire(1);
wire_reset += cpu.DebuggerPseudoPort("debugger_pseudo_pin_reset_notice");
wire_reset += model.Port("gui_pseudo_pin_reset_notice");

```

B.4 提供インタフェース関数

この節では、ユーザ・オープン・インタフェースとしてシミュレータ GUI が提供するインタフェース関数について説明します。

B.4.1 概要

次に、シミュレータ GUI が提供するインタフェース関数の一覧を示します。

表 B—2 提供インタフェース関数一覧

種類	関数名	機能概要
基本インタフェース関数	SuoSetInitCallback	初期化処理のコールバック登録
	SuoSetResetCallback	リセット処理のコールバック登録
	SuoGetMainClock	シミュレーションのメイン・クロック周期の取得
時間インタフェース関数	SuoCreateTimer	タイマ・インタフェースの生成
	SuoGetTimerHandle	タイマ・インタフェースのハンドルの取得
	SuoSetTimer	周期タイマの設定
	SuoKillTimer	周期タイマの停止
	SuoSetNotifyTimerCallback	タイマの時間通知処理のコールバック登録
端子インタフェース関数	SuoCreatePin	端子インタフェースの生成
	SuoGetPinHandle	端子インタフェースのハンドルの取得
	SuoOutputDigitalPin	端子のデジタル・データの出力
	SuoOutputAnalogPin	端子のアナログ・データ出力
	SuoOutputHighImpedance	端子のハイ・インピーダンス出力
	SuoSetInputDigitalPinCallback	端子のデジタル値入力処理のコールバック登録
	SuoSetInputAnalogPinCallback	端子のアナログ値入力処理のコールバック登録
	SuoSetInputHighImpedanceCallback	端子のハイ・インピーダンス状態通知処理のコールバック登録
外部バス・インタフェース関数 ^注	SuoCreateExtbus	外部バス・インタフェースの生成
	SuoGetExtbusHandle	外部バス・インタフェースのハンドルの取得
	SuoSetReadExtbusCallback	外部バスのリード・アクセスのコールバック登録
	SuoSetWriteExtbusCallback	外部バスのライト・アクセスのコールバック登録

種類	関数名	機能概要
シリアル・インタフェース関数	SuoCreateSerialUART	シリアル・インタフェース (UART タイプ) の生成
	SuoCreateSerialCSI	シリアル・インタフェース (CSI タイプ) の生成
	SuoGetSerialHandle	シリアル・インタフェースのハンドルの取得
	SuoSetSerialParameterUART	シリアルのパラメータ (UART タイプ) の設定
	SuoSetSerialParameterCSI	シリアルのパラメータ (CSI タイプ) の設定
	SuoGetSerialParameterUART	シリアルのパラメータ (UART タイプ) の取得
	SuoGetSerialParameterCSI	シリアルのパラメータ (CSI タイプ) の取得
	SuoSendSerialData	シリアル・データの送信 (1 データ)
	SuoSendSerialDataList	シリアル・データの送信 (複数データ)
	SuoSendSerialFile	シリアル・データの送信 (シリアル送信データ・ファイル)
	SuoSetNotifySentSerialCallback	シリアルの送信完了通知のコールバック登録
	SuoSetReceiveSerialCallback	シリアルの受信のコールバック登録
信号出力器インタフェース関数	SuoCreateWave	信号出力器インタフェースの生成
	SuoGetWaveHandle	信号出力器インタフェースのハンドルの取得
	SuoSendWaveFile	信号出力器による信号データの送信
	SuoSetNotifySentWaveCallback	信号出力器の送信完了通知のコールバック登録

注 外部バス・インタフェース関数を使用する場合は、使用する外部メモリ領域を、[メモリ・マッピングダイアログ](#)上の [メモリ種別] エリアにおいて [ターゲット・メモリ領域] に指定する必要があります。

B. 4.2 基本インタフェース関数

基本インタフェース関数には、次のものがあります。

関数名	機能概要
SuoSetInitCallback	初期化処理のコールバック登録
SuoSetResetCallback	リセット処理のコールバック登録
SuoGetMainClock	シミュレーションのメイン・クロック周期の取得

SuoSetInitCallback

初期化処理のコールバック登録を行います。

注意 この関数を **MakeUserModel** 関数内で呼び出さない場合、コールバック関数は機能しません。

[指定形式]

```
#include    "suo.h"
void      SuoSetInitCallback(SuoInitCallback func);
```

[引数]

引数	説明
<i>func</i>	初期化処理を行うユーザ定義関数へのポインタ（「InitFunc」参照）

[戻り値]

なし

[詳細説明]

- 初期化処理を行うユーザ定義関数を登録します。
- ここで登録された関数は、シミュレータ GUI 起動時の 1 回のみ呼び出されます。
- *func* に NULL を指定した場合は、登録を解除します。

[使用例]

```
#include    "suo.h"
void InitFunc(void);

/* MakeUserModel */
SuoUserEntry void MakeUserModel(const char *option)
{
    .....
    SuoSetInitCallback(InitFunc);          /* Set initialize function */
}

/* Initialize function */
void InitFunc(void) {
    .....
}
```

SuoSetResetCallback

リセット処理のコールバック登録を行います。

[指定形式]

```
#include "suo.h"
void SuoSetResetCallback(SuoResetCallback func);
```

[引数]

引数	説明
<i>func</i>	リセット処理を行うユーザ定義関数へのポインタ（「ResetFunc」参照）

[戻り値]

なし

[詳細説明]

- リセット処理を行うユーザ定義関数を登録します。
- ここで登録された関数は、CPU リセット時に呼び出されます。
- *func* に NULL を指定した場合は、登録を解除します。

[使用例]

```
#include "suo.h"
void ResetFunc(void);

void func1(void)
{
    .....
    SuoSetResetCallback(ResetFunc);    /* Set reset function */
}

/* Reset function */
void ResetFunc(void){
    .....
}
```

SuoGetMainClock

シミュレーションのメイン・クロック周期を取得します。

注意 この関数を **MakeUserModel** 関数内で呼び出すことはできません。コールバック関数内でのみ呼び出すことができます。

[指定形式]

```
#include    "suo.h"
int        SuoGetMainClock(unsigned long* time);
```

[引数]

引数	説明
<i>time</i>	メイン・クロックの周期（単位：ピコ秒）の格納場所

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「B.4.8 エラー番号一覧」参照）

[詳細説明]

- 現在動作中のシミュレーション環境のメイン・クロック周期を取得します。

[使用例]

```
#include    "suo.h"
unsigned long time;

void func1(void)
{
    .....
    SuoGetMainClock(&time);    /* Get main clock */
}
```

B. 4.3 時間インタフェース関数

時間インタフェース関数には、次のものがあります。

関数名	機能概要
SuoCreateTimer	タイマ・インタフェースの生成
SuoGetTimerHandle	タイマ・インタフェースのハンドルの取得
SuoSetTimer	周期タイマの設定
SuoKillTimer	周期タイマの停止
SuoSetNotifyTimerCallback	タイマの時間通知処理のコールバック登録

SuoCreateTimer

タイマ・インタフェースを生成します。

注意 この関数は、[MakeUserModel](#) 関数内でのみ呼び出すことができます。他のタイミングで呼び出した場合はエラーになります。

[指定形式]

```
#include "suo.h"
int SuoCreateTimer(const char* timerName, SuoHandle* handle);
```

[引数]

引数	説明
<i>timerName</i>	タイマの名前
<i>handle</i>	タイマ・インタフェースのハンドルの格納場所

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「 B.4.8 エラー番号一覧 」参照）

[詳細説明]

- タイマ・インタフェースを生成します。
- 生成したタイマ・インタフェースは、*timerName* で指定した名前と関連付けられます。
- この関数が正常終了すると、生成したタイマ・インタフェースのハンドルを取得できます。後のタイマ・インタフェースに対する制御は、ここで取得したハンドルを指定して行います。
なお、ハンドルは、[SuoGetTimerHandle](#) 関数により取得することもできます。

[使用例]

```
#include    "suo.h"
SuoHandle hTim1;

SuoUserEntry void MakeUserModel(const char *option)
{
    .....
    SuoCreateTimer("TIM1", &hTim1);      /* Create "TIM1" */
}
```


SuoGetTimerHandle

タイマ・インタフェースのハンドルを取得します。

[指定形式]

```
#include    "suo.h"
SuoHandle  SuaGetTimerHandle(const char* timerName);
```

[引数]

引数	説明
<i>timerName</i>	タイマの名前

[戻り値]

マクロ	説明
タイマ・インタフェースのハンドル	正常終了
NULL	不正終了

[詳細説明]

- 指定したタイマ・インタフェースのハンドルを取得します。
- *timerName* には、[SuoCreateTimer](#) 関数で指定した名前を指定してください（異なる名前を指定した場合は“NULL”が返ります）。

[使用例]

```
#include    "suo.h"
SuoHandle  hTim1;

void func1(void)
{
    .....
    hTim1 = SuaGetTimerHandle("TIM1");      /* Get handle of "TIM1" */
}
```

SuoSetTimer

周期タイマの設定を行います。

注意 この関数を **MakeUserModel** 関数内で呼び出すことはできません。コールバック関数内でのみ呼び出すことができます。

[指定形式]

```
#include "suo.h"
int SuoSetTimer(SuoHandle handle, int timeUnit, unsigned long timeValue);
```

[引数]

引数	説明
<i>handle</i>	タイマ・インタフェースのハンドル
<i>timeUnit</i>	次のいずれかの時間単位 -SUO_MAINCLK: メイン・クロック周期の単位 -SUO_USEC: マイクロ秒単位
<i>timeValue</i>	タイマ周期時間

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「B.4.8 エラー番号一覧」参照）

[詳細説明]

- 指定したタイマ・インタフェースに対して周期タイマの設定を行います。
- 周期時間は、*timeUnit* の単位で *timeValue* の値で指定します（*timeValue* に 0 は指定できません）。
- この関数の呼び出し直後からタイマは動作を開始し、**SuoKillTimer** 関数でタイマ動作を停止するまでタイマは動作し続けます。
- **SuoSetNotifyTimerCallback** 関数でタイマ通知関数が登録されていれば、1 周期毎にタイマ通知関数が呼び出されます。
- 現在動作しているタイマに対してこの関数を呼び出した場合は、タイマはリセットされ新たに指定した周期時間で動作を開始します。

[使用例]

```
#include    "suo.h"
SuoHandle hTim1;

void func1(void)
{
.....
    SuoSetTimer(hTim1, SUO_USEC, 20);    /* Invoke 20us cyclic timer */
}
```

SuoKillTimer

周期タイマの停止を行います。

注意 この関数を **MakeUserModel** 関数内で呼び出すことはできません。コールバック関数内でのみ呼び出すことができます。

[指定形式]

```
#include    "suo.h"
int        SuoKillTimer(SuoHandle handle);
```

[引数]

引数	説明
<i>handle</i>	タイマ・インタフェースのハンドル

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「B.4.8 エラー番号一覧」参照）

[詳細説明]

- 指定したタイマ・インタフェースに対して周期タイマの停止を行います。
- タイマが動作している場合はタイマ動作を停止し、タイマが停止している場合は何もしません（エラーにはなりません）。

[使用例]

```
#include    "suo.h"
SuoHandle hTim1;

void func1(void)
{
    .....
    SuoKillTimer(hTim1);    /* Stop timer */
}
```

SuoSetNotifyTimerCallback

タイマの時間通知処理のコールバック登録を行います。

[指定形式]

```
#include "suo.h"
int SuoSetNotifyTimerCallback(SuoHandle handle, SuoNotifyTimerCallback func);
```

[引数]

引数	説明
<i>handle</i>	タイマ・インタフェースのハンドル
<i>func</i>	タイマの時間通知を行うユーザ定義関数へのポインタ（「 NotifyTimerFunc 」参照）

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「 B.4.8 エラー番号一覧 」参照）

[詳細説明]

- タイマの時間通知処理を行うユーザ定義関数を登録します。
- ここで登録された関数は、指定したタイマ・インタフェースのタイマ周期毎に呼び出されます。
- *func* に NULL を指定した場合は、登録を解除します。

[使用例]

```
#include    "suo.h"
void NotifyTimerFunc(SuoHandle handle);
SuoHandle hTim1;
void func1(void)
{
    .....
    SuoSetNotifyTimerCallback(hTim1, NotifyTimerFunc);    /* Set notify-timer function */
}

/* Notify-timer function */
void NotifyTimerFunc(SuoHandle handle)
{
    .....
}
```

B. 4.4 端子インタフェース関数

端子インタフェース関数には、次のものがあります。

関数名	機能概要
SuoCreatePin	端子インタフェースの生成
SuoGetPinHandle	端子インタフェースのハンドルの取得
SuoOutputDigitalPin	端子のデジタル・データの出力
SuoOutputAnalogPin	端子のアナログ・データの出力
SuoOutputHighImpedance	端子のハイ・インピーダンス出力
SuoSetInputDigitalPinCallback	端子のデジタル・データ入力処理のコールバック登録
SuoSetInputAnalogPinCallback	端子のアナログ・データ入力処理のコールバック登録
SuoSetInputHighImpedanceCallback	端子のハイ・インピーダンス状態通知処理のコールバック登録

SuoCreatePin

端子インタフェースを生成します。

注意 この関数は、[MakeUserModel](#) 関数内でのみ呼び出すことができます。他のタイミングで呼び出した場合はエラーになります。

[指定形式]

```
#include "suo.h"
int SuoCreatePin(const char* pinName, SuoHandle* handle);
```

[引数]

引数	説明
<i>pinName</i>	端子の名前
<i>handle</i>	端子インタフェースのハンドルの格納場所

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「 B.4.8 エラー番号一覧 」参照）

[詳細説明]

- 端子インタフェースを生成します。
- 生成した端子インタフェースは、*pinName* で指定した名前と関連付けられます。また、*pinName* で指定した端子が生成されます。
- この関数が正常終了すると、生成した端子インタフェースのハンドルを取得できます。後の端子インタフェースに対する制御は、ここで取得したハンドルを指定して行います。
なお、ハンドルは、[SuoGetPinHandle](#) 関数により取得することもできます。

[使用例]

```
#include    "suo.h"
SuoHandle hPinP00;
SuoHandle hPinABC;

SuoUserEntry void MakeUserModel(const char *option)
{
    .....
    SuoCreatePin("P00", &hPinP00);      /* Create "P00" */
    SuoCreatePin("ABC", &hPinABC);      /* Create "ABC" */
}
```

SuoGetPinHandle

端子インタフェースのハンドルを取得します。

[指定形式]

```
#include    "suo.h"
SuoHandle  SuaGetPinHandle(const char* pinName);
```

[引数]

引数	説明
<i>pinName</i>	端子の名前

[戻り値]

マクロ	説明
端子インタフェースのハンドル	正常終了
NULL	不正終了

[詳細説明]

- 指定した端子インタフェースのハンドルを取得します。
- *pinName* には [SuoCreatePin](#) 関数で指定した名前を指定してください（異なる名前を指定した場合は NULL が返ります）。

[使用例]

```
#include    "suo.h"
SuoHandle  hPinP00;

void func1(void)
{
    .....
    hPinP00 = SuaGetPinHandle("P00");      /* Get handle of "P00" */
}
```

SuoOutputDigitalPin

端子インタフェースに対し、デジタル・データを出力します。

注意 この関数を **MakeUserModel** 関数内で呼び出すことはできません。コールバック関数内でのみ呼び出すことができます。

[指定形式]

```
#include "suo.h"
int SuoOutputDigitalPin(SuoHandle handle, int pinValue);
```

[引数]

引数	説明
<i>handle</i>	端子インタフェースのハンドル
<i>pinValue</i>	次のいずれかの出力値（デジタル・データ） - SUO_HIGH (=1)： HIGH 値 - SUO_LOW (=0)： LOW 値

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「B.4.8 エラー番号一覧」参照）

[詳細説明]

- 指定した端子インタフェースに対して、*pinValue* で指定したデジタル・データを出力します。
- アナログ・データの出力は、**SuoOutputAnalogPin** 関数を使用してください。

[使用例]

```
#include    "suo.h"
SuoHandle hPinP00;

void func1(void)
{
    .....
    SuoOutputDigitalPin(hPinP00, SUO_HIGH);    /* Output HIGH */
}
```

SuoOutputAnalogPin

端子インタフェースに対し、アナログ・データを出力します。

注意 この関数を **MakeUserModel** 関数内で呼び出すことはできません。コールバック関数内でのみ呼び出すことができます。

[指定形式]

```
#include    "suo.h"
int        SuoOutputAnalogPin(SuoHandle handle, double pinValue);
```

[引数]

引数	説明
<i>handle</i>	端子インタフェースのハンドル
<i>pinValue</i>	出力値（アナログ・データ）（単位：V）

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「B.4.8 エラー番号一覧」参照）

[詳細説明]

- 指定した端子インタフェースに対して、*pinValue* で指定したアナログ・データを出力します。
- *pinValue* の単位は“V”で、浮動小数点データで指定してください。
- デジタル・データの出力は、**SuoOutputDigitalPin** 関数を使用してください。

[使用例]

```
#include    "suo.h"
SuoHandle hPinP00;

void func1(void)
{
    .....

    SuoOutputAnalogPin(hPinP00, 3.5);    /* Output 3.5V */
}
```

SuoOutputHighImpedance

端子インタフェースに対し、ハイ・インピーダンスを出力します。

注意 この関数を **MakeUserModel** 関数内で呼び出すことはできません。コールバック関数内でのみ呼び出すことができます。

[指定形式]

```
#include    "suo.h"
int        SuoOutputHighImpedance(SuoHandle handle);
```

[引数]

引数	説明
handle	端子インタフェースのハンドル

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「B.4.8 エラー番号一覧」参照）

[詳細説明]

- 指定したデジタル／アナログ端子インタフェースに対して、ハイ・インピーダンスを出力します。

[使用例]

```
#include    "suo.h"
SuoHandle hPinP00;

void func1(void)
{
    .....
    SuoOutputHighImpedance(hPinP00);    /* Output High Impedance */
}
```

SuoSetInputDigitalPinCallback

端子のデジタル・データ入力処理のコールバック登録を行います。

[指定形式]

```
#include "suo.h"
int SuoSetInputDigitalPinCallback(SuoHandle handle, SuoInputDigitalPinCallback func);
```

[引数]

引数	説明
<i>handle</i>	端子インタフェースのハンドル
<i>func</i>	端子のデジタル・データ入力処理を行うユーザ定義関数へのポインタ（「 InputDigitalPinFunc 」参照）

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「 B.4.8 エラー番号一覧 」参照）

[詳細説明]

- 端子のデジタル・データ入力処理を行うユーザ定義関数を登録します。
- ここで登録された関数は、指定した端子に信号が入力された時に呼び出されます。
- *func* に NULL を指定した場合は、登録を解除します。

[使用例]

```
#include    "suo.h"
void InputDigitalPinFunc(SuoHandle handle, int pinValue);
SuoHandle hPinP00;

void func1(void)
{
    .....
    SuoSetInputDigitalPinCallback(hPinP00, InputDigitalPinFunc); /* Set input-digital-pin function */
}

/* Input-digital-pin function */
void InputDigitalPinFunc(SuoHandle handle, int pinValue)
{
    .....
}
```


SuoSetInputAnalogPinCallback

端子のアナログ・データ入力処理のコールバック登録を行います。

[指定形式]

```
#include "suo.h"
int SuoSetInputAnalogPinCallback(SuoHandle handle, SuoInputAnalogPinCallback func);
```

[引数]

引数	説明
<i>handle</i>	端子インタフェースのハンドル
<i>func</i>	端子のアナログ・データ入力処理を行うユーザ定義関数へのポインタ（「 InputAnalogPinFunc 」参照）

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「 B.4.8 エラー番号一覧 」参照）

[詳細説明]

- 端子のアナログ・データ入力処理を行うユーザ定義関数を登録します。
- ここで登録された関数は、指定した端子に信号が入力された時に呼び出されます。
- *func* に NULL を指定した場合は、登録を解除します。

[使用例]

```
#include    "suo.h"
void InputAnalogPinFunc(SuoHandle handle, double pinValue);
SuoHandle hPinP00;

void func1(void)
{
    .....
    SuoSetInputAnalogPinCallback(hPinP00, InputAnalogPinFunc); /* Set input-analog-pin function */
}

/* Input-analog-pin function */
void InputAnalogPinFunc(SuoHandle handle, double pinValue)
{
    .....
}
```

SuoSetInputHighImpedanceCallback

端子のハイ・インピーダンス状態通知処理のコールバック登録を行います。

[指定形式]

```
#include "suo.h"
int SuoSetInputHighImpedanceCallback(SuoHandle handle, SuoInputHighImpedanceCallback func);
```

[引数]

引数	説明
<i>handle</i>	端子インタフェースのハンドル
<i>func</i>	接続されたすべての端子がハイ・インピーダンス状態になった際の処理を行う ユーザ定義関数へのポインタ（「 InputHighImpedanceFunc 」参照）

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「 B.4.8 エラー番号一覧 」参照）

[詳細説明]

- デジタル／アナログ端子に接続されたすべての端子がハイ・インピーダンス状態になった際の処理を行うユーザ定義関数を登録します。
- *func* に NULL を指定した場合は、登録を解除します。

[使用例]

```
#include "suo.h"
void InputHighImpedanceFunc(SuoHandle handle);
SuoHandle hPinP00;

void func1(void)
{
    .....
    SuoSetInputHighImpedanceCallback(hPinP00, InputHighImpedanceFunc);
                                     /* Set input-high-impedance function */
}

/* Input-high-impedance function */
void InputHighImpedanceFunc(SuoHandle handle)
{
    .....
}
```

B. 4.5 外部バス・インタフェース関数

外部バス・インタフェース関数には、次のものがあります。

関数名	機能概要
SuoCreateExtbus	外部バス・インタフェースの生成
SuoGetExtbusHandle	外部バス・インタフェースのハンドルの取得
SuoSetReadExtbusCallback	外部バスのリード・アクセスのコールバック登録
SuoSetWriteExtbusCallback	外部バスのライト・アクセスのコールバック登録

注意 外部バス・インタフェース関数を使用する場合は、使用する外部メモリ領域を、[メモリ・マッピングダイアログ](#)上の [メモリ種別] エリアにおいて [ターゲット・メモリ領域] に指定する必要があります。

SuoCreateExtbus

外部バス・インタフェースを生成します。

- 注意 1. この関数は、[MakeUserModel](#) 関数内でのみ呼び出すことができます。他のタイミングで呼び出した場合はエラーになります。
2. 外部バス・インタフェース関数を使用する場合は、使用する外部メモリ領域を、[メモリ・マッピングダイアログ](#)上の [メモリ種別] エリアにおいて [ターゲット・メモリ領域] に指定する必要があります。

[指定形式]

```
#include "suo.h"

int SuoCreateExtbus(const char* extbusName, unsigned long addr, unsigned long size,
SuoHandle* handle);
```

[引数]

引数	説明
<i>extbusName</i>	外部バスの名前
<i>addr</i>	外部メモリ領域の先頭アドレス
<i>size</i>	外部メモリ領域のサイズ
<i>handle</i>	外部バス・インタフェースのハンドルの格納場所

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「 B.4.8 エラー番号一覧 」参照）

[詳細説明]

- 外部バス・インタフェースを生成します。
 - 生成した外部バス・インタフェースは、*extbusName* で指定した名前と関連付けられます。
 - この関数が正常終了すると、生成した外部バス・インタフェースのハンドルを取得できます。後の外部バス・インタフェースに対する制御は、ここで取得したハンドルを指定して行います。
- なお、ハンドルは、[SuoGetExtbusHandle](#) 関数により取得することもできます。

[使用例]

```
#include    "suo.h"
SuoHandle hExtbus1;

SuoUserEntry void MakeUserModel(const char *option)
{
    .....
    SuoCreateExtbus("EXTBUS1", 0x200000, 0x1000, &hExtbus1);    /* Create "EXTBUS1" */
}
```

SuoGetExtbusHandle

外部バス・インタフェースのハンドルを取得します。

[指定形式]

```
#include    "suo.h"
SuoHandle  SuaGetExtbusHandle(const char* extbusName);
```

[引数]

引数	説明
<i>extbusName</i>	外部バスの名前

[戻り値]

マクロ	説明
外部バス・インタフェースのハンドル	正常終了
NULL	不正終了

[詳細説明]

- 指定した外部バス・インタフェースのハンドルを取得します。
- *extbusName* には [SuoCreateExtbus](#) 関数で指定した名前を指定してください（異なる名前を指定した場合は NULL が返ります）。

[使用例]

```
#include    "suo.h"
SuoHandle  hExtbus1;

void func1(void)
{
    .....
    hExtbus1 = SuaGetExtbusHandle("EXTBUS1");    /* Get handle of "EXTBUS1" */
}
```


SuoSetReadExtbusCallback

外部バスのリード・アクセス処理のコールバック登録を行います。

[指定形式]

```
#include "suo.h"
int SuoSetReadExtbusCallback(SuoHandle handle, SuoReadExtbusCallback func);
```

[引数]

引数	説明
<i>handle</i>	外部バス・インタフェースのハンドル
<i>func</i>	外部バスのリード・アクセス処理を行うユーザ定義関数へのポインタ（「 ReadExtbusFunc 」参照）

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「 B.4.8 エラー番号一覧 」参照）

[詳細説明]

- 外部バスのリード・アクセス処理を行うユーザ定義関数を登録します。
- ここで登録された関数は、指定した外部バス（登録したアドレス範囲）に対してリード要求が発生した時に呼び出されます。
- *func* に NULL を指定した場合は、登録を解除します。

[使用例]

```
#include    "suo.h"
void ReadExtbusFunc(SuoHandle handle, unsigned long addr, int accessSize, unsigned char data[]);
SuoHandle hExtbus1;

void func1(void)
{
    .....
    SuoSetReadExtbusCallback(hExtbus1, ReadExtbusFunc);    /* Set read-external-bus function */
}

/* Read-external-bus function */
void ReadExtbusFunc(SuoHandle handle, unsigned long addr, int accessSize, unsigned char data[])
{
    .....
}
```

SuoSetWriteExtbusCallback

外部バスのライト・アクセス処理のコールバック登録を行います。

[指定形式]

```
#include "suo.h"
int SuoSetWriteExtbusCallback(SuoHandle handle, SuoWriteExtbusCallback func);
```

[引数]

引数	説明
<i>handle</i>	外部バス・インタフェースのハンドル
<i>func</i>	外部バスのライト・アクセス処理を行うユーザ定義関数へのポインタ（「WriteExtbusFunc」参照）

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「B.4.8 エラー番号一覧」参照）

[詳細説明]

- 外部バスのライト・アクセス処理を行うユーザ定義関数を登録します。
- ここで登録された関数は、指定した外部バス（登録したアドレス範囲）に対してライト要求が発生した時に呼び出されます。
- *func* に NULL を指定した場合は、登録を解除します。

[使用例]

```
#include    "suo.h"
void WriteExtbusFunc(SuoHandle handle, unsigned long addr, int accessSize, const unsigned char data[]);
SuoHandle hExtbus1;

void func1(void)
{
    .....
    SuoSetWriteExtbusCallback(hExtbus1, WriteExtbusFunc); /* Set write-external-bus function */
}

/* Write-external-bus function */
void WriteExtbusFunc(SuoHandle handle, unsigned long addr, int accessSize, const unsigned char data[])
{
    .....
}
```

B. 4.6 シリアル・インタフェース関数

シリアル・インタフェース関数には、次のものがあります。

関数名	機能概要
SuoCreateSerialUART	シリアル・インタフェース（UART タイプ）の生成
SuoCreateSerialCSI	シリアル・インタフェース（CSI タイプ）の生成
SuoGetSerialHandle	シリアル・インタフェースのハンドルの取得
SuoSetSerialParameterUART	シリアルのパラメータ（UART タイプ）の設定
SuoSetSerialParameterCSI	シリアルのパラメータ（CSI タイプ）の設定
SuoGetSerialParameterUART	シリアルのパラメータ（UART タイプ）の取得
SuoGetSerialParameterCSI	シリアルのパラメータ（CSI タイプ）の取得
SuoSendSerialData	シリアル・データの送信（1 データ）
SuoSendSerialDataList	シリアル・データの送信（複数データ）
SuoSendSerialFile	シリアル・データの送信（シリアル・ファイル）
SuoSetNotifySentSerialCallback	シリアルの送信完了通知のコールバック登録
SuoSetReceiveSerialCallback	シリアルの受信のコールバック登録

SuoCreateSerialUART

シリアル・インタフェース（UART タイプ）を生成します。

注意 この関数は、[MakeUserModel](#) 関数内でのみ呼び出すことができます。他のタイミングで呼び出した場合はエラーになります。

[指定形式]

```
#include "suo.h"
int SuoCreateSerialUART(const char* serialName, const char* pinNameTXD, const char* pinNameRXD, SuoHandle* handle);
```

[引数]

引数	説明
<i>serialName</i>	シリアルの名前
<i>pinNameTXD</i>	シリアルで使用する送信データ用端子の名前
<i>pinNameRXD</i>	シリアルで使用する受信データ用端子の名前
<i>handle</i>	シリアル・インタフェースのハンドルの格納場所

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「 B.4.8 エラー番号一覧 」参照）

[詳細説明]

- シリアル・インタフェース（UART タイプ）を生成します。
- 生成したシリアル・インタフェースは、*serialName* で指定した名前と関連付けられます。また、*pinNameTXD*、および *pinNameRXD* で指定した端子が生成されます。
- この関数が正常終了すると、生成したシリアル・インタフェースのハンドルを取得できます。後の外部バス・インタフェースに対する制御は、ここで取得したハンドルを指定して行います。
なお、ハンドルは、[SuoGetSerialHandle](#) 関数により取得することもできます。

[使用例]

```
#include    "suo.h"
SuoHandle hUart1;

SuoUserEntry void MakeUserModel(const char *option)
{
    .....
    SuoCreateSerialUART("UART1", "TXD1", "RXD1", &hUart1);    /* Create "UART1" */
}
```

SuoCreateSerialCSI

シリアル・インタフェース（CSIタイプ）を生成します。

注意 この関数は、[MakeUserModel](#) 関数内でのみ呼び出すことができます。他のタイミングで呼び出した場合はエラーになります。

[指定形式]

```
#include "suo.h"

int SuoCreateSerialCSI(const char* serialName, const char* pinNameSO, const char*
pinNameSI, const char* pinNameSCK, SuoHandle* handle);
```

[引数]

引数	説明
<i>serialName</i>	シリアルの名前
<i>pinNameSO</i>	シリアルで使用する送信データ用端子の名前
<i>pinNameSI</i>	シリアルで使用する受信データ用端子の名前
<i>pinNameSCK</i>	シリアルで使用するクロック用端子の名前
<i>handle</i>	シリアル・インタフェースのハンドルの格納場所

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「 B.4.8 エラー番号一覧 」参照）

[詳細説明]

- シリアル・インタフェース（CSIタイプ）を生成します。
- 生成したシリアル・インタフェースは、*serialName* で指定した名前と関連付けられます。また、*pinNameSO*、*pinNameSI*、および *pinNameSCK* で指定した端子が生成されます。
- この関数が正常終了すると、生成したシリアル・インタフェースのハンドルを取得できます。後の外部バス・インタフェースに対する制御は、ここで取得したハンドルを指定して行います。
なお、ハンドルは、[SuoGetSerialHandle](#) 関数により取得することもできます。

[使用例]

```
#include    "suo.h"
SuoHandle hCsi1;

SuoUserEntry void MakeUserModel(const char *option)
{
    .....
    SuoCreateSerialCSI("CSI1", "S01", "SI1", "SCK1", &hCsi1);    /* Create "CSI1" */
}
```

SuoGetSerialHandle

シリアル・インタフェースのハンドルを取得します。

[指定形式]

```
#include    "suo.h"
SuoHandle  SuoGetSerialHandle(const char* serialName);
```

[引数]

引数	説明
<i>serialName</i>	シリアルの名前

[戻り値]

マクロ	説明
シリアル・インタフェースのハンドル	正常終了
NULL	不正終了

[詳細説明]

- 指定したシリアル・インタフェースのハンドルを取得します。
- *serialName* には、[SuoCreateSerialUART](#) 関数、または [SuoCreateSerialCSI](#) 関数で指定した名前を指定してください（異なる名前を指定した場合は NULL が返ります）。

[使用例]

```
#include    "suo.h"
SuoHandle  hSerial1;

void func1(void)
{
    .....
    hSerial1 = SuoGetSerialHandle("SERIAL1");    /* Get handle of "SERIAL1" */
}
```

SuoSetSerialParameterUART

シリアルのパラメータ (UART タイプ) を設定します。

注意 この関数を **MakeUserModel** 関数内で呼び出すことはできません。コールバック関数内でのみ呼び出すことができます。

[指定形式]

```
#include "suo.h"
int SuoSetSerialParameterUART(SuoHandle handle, const SuoSerialParameterUART* param);
```

[引数]

引数	説明
<i>handle</i>	シリアル・インタフェースのハンドル
<i>param</i>	シリアルのパラメータ (UART タイプ) の格納場所を示す SuoSerialParameterUART 構造体 ^注 へのポインタ

注 SuoSerialParameterUART 構造体の構成は次のとおりです。

```
typedef struct {
    unsigned long    baudrate;           /* ボー・レート値 */
    int              direction;         /* 転送方向 */
    int              dataLength;        /* データ・ビット長 */
    int              stopLength;        /* ストップ・ビット長 */
    int              parity;            /* パリティ */
} SuoSerialParameterUART;
```

パラメータ (UART タイプ)	値	説明
ボー・レート	ボー・レート値	単位: bps
転送方向	SUO_MSBFIRST	MSB ファースト
	SUO_LSBFIRST	LSB ファースト
データ・ビット長	1 ~ 32	—
ストップ・ビット長	1, または 2	—
パリティ	SUO_NONEPARITY	パリティなし
	SUO_ZEROPARITY	0 パリティ (送信時: パリティ 0, 受信時: パリティ・チェックなし)
	SUO_ODDPARITY	奇数パリティ
	SUO_EVENPARITY	偶数パリティ

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「B.4.8 エラー番号一覧」参照）

[詳細説明]

- 指定したシリアル・インタフェースに対してシリアル動作に関するパラメータ（UART タイプ）を設定します。デフォルトの設定値は次の通りです。
 - ボー・レート： 9600 bps
 - 転送方向： LSB ファースト
 - データ・ビット長： 7 ビット
 - ストップ・ビット長： 1 ビット
 - パリティ： なし

[使用例]

```
#include "suo.h"
SuoHandle hUart1;

void func1(void)
{
    SuoSerialParameterUART param;
    .....
    param.baudrate = 19200; /* 19200 bps */
    param.direction = SUO_LSBFIRST; /* LSB First */
    param.dataLength = 8; /* databit 8 bit */
    param.stopLength = 1; /* stopbit 1 bit */
    param.parity = SUO_EVENPARITY; /* even parity */
    SuoSetSerialParameterUART(hUart1, &param); /* Set parameter of UART1 */
}
```

SuoSetSerialParameterCSI

シリアルのパラメータ（CSI タイプ）を設定します。

注意 この関数を **MakeUserModel** 関数内で呼び出すことはできません。コールバック関数内でのみ呼び出すことができます。

[指定形式]

```
#include "suo.h"
int SuoSetSerialParameterCSI(SuoHandle handle, const SuoSerialParameterCSI* param);
```

[引数]

引数	説明
<i>handle</i>	シリアル・インタフェースのハンドル
<i>param</i>	シリアルのパラメータ（CSI タイプ）の格納場所を示す SuoSerialParameterCSI 構造体 ^注 へのポインタ

注 SuoSerialParameterCSI 構造体の構成は次のとおりです。

```
typedef struct {
    int mode; /* 動作モード */
    unsigned long frequency; /* 転送クロックの周波数 */
    int phase; /* 位相 */
    int direction; /* 転送方向 */
    int datalength; /* データ・ビット長 */
} SuoSerialParameterCSI;
```

パラメータ（CSI タイプ）	値	説明	
動作モード	SUO_MASTER	マスタ動作	
	SUO_SLAVE	スレーブ動作	
転送クロックの周波数	<i>周波数</i>	単位：Hz ただし、マスタ動作の場合は“0”の指定不可	
位相	0	通常の位相	表 B—3 参照
	SUO_PRECEDEDATA	データ先行出力	
	SUO_REVERSELOCK	クロック反転	
	SUO_PRECEDEDATA SUO_REVERSELOCK	データ先行出力、およびクロック反転の両方を指定	
転送方向	SUO_MSBFIRS	MSB ファースト	
	SUO_LSBFIRST	LSB ファースト	

パラメータ (CSI タイプ)	値	説明
データ・ビット長	1 ~ 32	-

表 B—3 CSI 位相タイプ (SuoSetSerialParameterCSI 関数)

値	位相
0	
SUO_PRECEDEDATA	
SUO_REVERSELOCK	
SUO_PRECEDEDATA SUO_REVERSELOCK	

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了 (「B.4.8 エラー番号一覧」参照)

[詳細説明]

- 指定したシリアル・インタフェースに対してシリアル動作に関するパラメータ（CSI タイプ）を設定します。デフォルトの設定値は次の通りです。
 - 動作モード： スレーブ動作
 - 転送クロック： 0
 - 位相： 通常の位相
 - 転送方向： MSB ファースト
 - データ・ビット長： 8 ビット
- CSI をマスタ動作で使用する場合、受信を行うためにはダミー・データの送信を行う必要があります（CSI はマスタ側がクロックを出力することで送受信を行う通信方式であるため）。

備考 CSI の通信が行われていない状態で、[タイミングチャート ウィンドウ](#)で CSI の端子波形を確認した場合、次のように本来の期待値と異なったレベルが観測されます。

なお、通信が開始すると期待通りのレベルになるため、実際の動作には影響しません。

- SCK 端子（スレーブ動作時）： ハイ・レベル（本来はハイ・インピーダンス）
- SO 端子： ハイ・レベル（本来はロー・レベル）

[使用例]

```
#include    "suo.h"

SuoHandle hCsi1;
void func1(void)
{
    SuoSerialParameterCSI param;
    .....
    param.mode          = SUO_SLAVE;          /* slave */
    param.frequency     = 1000000;          /* 1MHz */
    param.phase         = 0;                 /* normal */
    param.direction     = SUO_LSBFIRST;     /* LSB First */
    param.dataLength    = 8;                /* databit 8bit */
    SuoSetSerialParameterCSI(hCsi1, &param); /* Set parameter of CSI1 */
}
```

SuoGetSerialParameterUART

シリアルのパラメータ（UART タイプ）を取得します。

注意 この関数を **MakeUserModel** 関数内で呼び出すことはできません。コールバック関数内でのみ呼び出すことができます。

[指定形式]

```
#include "suo.h"
int SuoGetSerialParameterUART(SuoHandle handle, SuoSerialParameterUART* param);
```

[引数]

引数	説明
<i>handle</i>	シリアル・インタフェースのハンドル
<i>param</i>	シリアルのパラメータ（UART タイプ）の格納場所を示す SuoSerialParameterUART 構造体 ^注 へのポインタ

注 SuoSerialParameterUART 構造体についての詳細は、[SuoSetSerialParameterUART](#) 関数を参照してください。

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「 B.4.8 エラー番号一覧 」参照）

[詳細説明]

- 指定したシリアル・インタフェースに対して、シリアル動作に関するパラメータ（UART タイプ）を取得します。

[使用例]

```
#include    "suo.h"
SuoHandle hUart1;

void func1(void)
{
    SuoSerialParameterUART param;
    .....
    SuoGetSerialParameterUART(hUart1, &param);    /* Get parameter of UART1 */
    .....
}
```

SuoGetSerialParameterCSI

シリアルのパラメータ（CSI タイプ）を取得します。

注意 この関数を **MakeUserModel** 関数内で呼び出すことはできません。コールバック関数内でのみ呼び出すことができます。

[指定形式]

```
#include "suo.h"
int SuoGetSerialParameterCSI(SuoHandle handle, SuoSerialParameterCSI* param);
```

[引数]

引数	説明
<i>handle</i>	シリアル・インタフェースのハンドル
<i>param</i>	シリアルのパラメータ（CSI タイプ）の格納場所を示す SuoSerialParameterCSI 構造体 ^注 へのポインタ

注 SuoSerialParameterCSI 構造体についての詳細は、[SuoSetSerialParameterCSI](#) 関数を参照してください。

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「 B.4.8 エラー番号一覧 」参照）

[詳細説明]

- 指定したシリアル・インタフェースに対して、シリアル動作に関するパラメータ（CSI タイプ）を取得します。

[使用例]

```
#include    "suo.h"
SuoHandle hCsi1;

void func1(void)
{
    SuoSerialParameterCSI param;
    .....
    SuoGetSerialParameterCSI(hCsi1, &param);    /* Get parameter of CSI1 */
    .....
}
```

SuoSendSerialData

シリアル・データ（1 データ）の送信を行います。

- 注意 1. UART の連続送信を行う場合は、必ず [SuoSendSerialDataList](#) 関数（複数データ用シリアル送信関数）を使用してください。
2. この関数を [NotifySentSerialFunc](#) 関数（シリアル送信完了のコールバック関数）で呼び出した場合、送信開始が UART のボー・レート半周期分遅れます。
3. この関数を [MakeUserModel](#) 関数内で呼び出すことはできません。コールバック関数内でのみ呼び出すことができます。

[指定形式]

```
#include    "suo.h"
int        SuoSendSerialData(SuoHandle handle, unsigned long data);
```

[引数]

引数	説明
<i>handle</i>	シリアル・インタフェースのハンドル
<i>data</i>	送信データ（1 データ）

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「 B.4.8 エラー番号一覧 」参照）

[詳細説明]

- 1 つのシリアル・データの送信を開始します。
- シリアル・データの送信が完了するまで時間がかかります。送信完了のタイミングを通知する場合は、[SuoSetNotifySentSerialCallback](#) 関数で送信完了通知関数を設定してください。
- 現在送信中のシリアル・インタフェースに対してこの関数を呼び出した場合はエラーとなります。

[使用例]

```
#include    "suo.h"
SuoHandle hSerial1;

void func1(void)
{
    .....
    SuoSendSerialData(hSerial1, 0x80);    /* Send 0x80 */
}
```

SuoSendSerialDataList

シリアル・データ（複数データ）の送信を行います。

注意 この関数を **MakeUserModel** 関数内で呼び出すことはできません。コールバック関数内でのみ呼び出すことができます。

[指定形式]

```
#include "suo.h"
int SuoSendSerialDataList(SuoHandle handle, long count, unsigned long dataList[]);
```

[引数]

引数	説明
<i>handle</i>	シリアル・インタフェースのハンドル
<i>count</i>	送信データの個数（1～32767）
<i>dataList[]</i>	送信データ（配列によりデータ個数分指定）

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「B.4.8 エラー番号一覧」参照）

[詳細説明]

- 複数個のシリアル・データの送信を開始します。
- シリアル・データの送信が完了するまで時間がかかります。送信完了のタイミングを通知する場合は、[SuoSetNotifySentSerialCallback](#) 関数で送信完了通知関数を設定してください。
- 現在送信中のシリアル・インタフェースに対してこの関数を呼び出した場合はエラーとなります。

[使用例]

```
#include    "suo.h"
SuoHandle hSerial1;

void func1(void)
{
    unsigned long dataList[6] = {0x73, 0x65, 0x72, 0x69, 0x61, 0x6c};
    .....
    SuoSendSerialDataList(hSerial1, 6, dataList);      /* Send dataList */
}
```

SuoSendSerialFile

シリアル・データ（シリアル送信データ・ファイル）の送信を行います。

注意 この関数を **MakeUserModel** 関数内で呼び出すことはできません。コールバック関数内でのみ呼び出すことができます。

[指定形式]

```
#include "suo.h"
int SuoSendSerialFile(SuoHandle handle, const char* serialFile);
```

[引数]

引数	説明
<i>handle</i>	シリアル・インタフェースのハンドル
<i>serialFile</i>	シリアル ウィンドウで編集したのち保存したシリアル送信データ・ファイル名 なお、ファイル名を相対パスで指定した場合、ユーザ・モデル（UserModel.dll）のパスからの相対として扱われます。

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「B.4.8 エラー番号一覧」参照）

[詳細説明]

- シリアル ウィンドウで編集したのち保存したシリアル送信データ・ファイル（*.ser）に記録されているシリアル・データの送信を開始します。
- シリアル・データの送信が完了するまで時間がかかります。送信完了のタイミングを通知する場合は、**SuoSetNotifySentSerialCallback** 関数で送信完了通知関数を設定してください。
- 現在送信中のシリアル・インタフェースに対してこの関数を呼び出した場合はエラーとなります。

[使用例]

```
#include    "suo.h"
SuoHandle hSerial1;

void func1(void)
{
    .....
    SuoSendSerialFile(hSerial1, "foo.ser");    /* Send serial data on "foo.ser" */
}
```

SuoSetNotifySentSerialCallback

シリアルを送信完了通知のコールバック登録を行います。

[指定形式]

```
#include "suo.h"
int SuoSetNotifySentSerialCallback(SuoHandle handle, SuoNotifySentSerialCallback func);
```

[引数]

引数	説明
<i>handle</i>	シリアル・インタフェースのハンドル
<i>func</i>	シリアルを送信完了時の処理を行うユーザ定義関数へのポインタ（「 NotifySentSerialFunc 」参照）

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「 B.4.8 エラー番号一覧 」参照）

[詳細説明]

- シリアルの送信完了時の処理を行うユーザ定義関数を登録します。
- ここで登録された関数は、送信指定した1つ、または複数のシリアル・データがすべて送信完了した際に呼び出されます。
- *func* に NULL を指定した場合は、登録を解除します。

[使用例]

```
#include    "suo.h"
void NotifySentSerialFunc(SuoHandle handle);
SuoHandle hSerial1;

void func1(void)
{
    .....
    SuoSetNotifySentSerialCallback(hSerial1, NotifySentSerialFunc);
                                     /* Set notify-sent-serial function */
}
/* Notify-sent-serial function */
void NotifySentSerialFunc(SuoHandle handle)
{
    .....
}
```

SuoSetReceiveSerialCallback

シリアルを受信のコールバック登録を行います。

[指定形式]

```
#include "suo.h"
int SuoSetReceiveSerialCallback(SuoHandle handle, SuoReceiveSerialCallback func);
```

[引数]

引数	説明
<i>handle</i>	シリアル・インタフェースのハンドル
<i>func</i>	シリアルの受信時の処理を行うユーザ定義関数へのポインタ（「 ReceiveSerialFunc 」参照）

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「 B.4.8 エラー番号一覧 」参照）

[詳細説明]

- シリアルの受信時の処理を行うユーザ定義関数を登録します。
- ここで登録された関数は、シリアル・データを1つ受信した際に呼び出されます。
- *func* に NULL を指定した場合は、登録を解除します。

[使用例]

```
#include    "suo.h"
void ReceiveSerialFunc(SuoHandle handle, unsigned long data, int status);
SuoHandle hSerial1;

void func1(void)
{
    .....
    SuoSetReceiveSerialCallback(hSerial1, ReceiveSerialFunc);
                                     /* Set receive-serial function */
}
/* Receive-serial function */
void ReceiveSerialFunc(SuoHandle handle, unsigned long data, int status)
{
    .....
}
```

B. 4.7 信号出力器インタフェース関数

信号出力器インタフェース関数には、次のものがあります。

関数名	機能概要
SuoCreateWave	信号出力器インタフェースの生成
SuoGetWaveHandle	信号出力器インタフェースのハンドルの取得
SuoSendWaveFile	信号出力器による信号データの送信
SuoSetNotifySentWaveCallback	信号出力器の送信完了通知のコールバック登録

SuoCreateWave

信号出力器インタフェースを生成します。

注意 この関数は、[MakeUserModel](#) 関数内でのみ呼び出すことができます。他のタイミングで呼び出した場合はエラーになります。

[指定形式]

```
#include "suo.h"
int SuoCreateWave(const char* waveName, int count, const char* pinNameList[], SuoHandle* handle);
```

[引数]

引数	説明
<i>waveName</i>	信号出力器の名前
<i>count</i>	信号出力器で使用する端子の個数
<i>pinNameList[]</i>	信号出力器で使用する端子の名前（配列により端子数分指定）
<i>handle</i>	信号出力器インタフェースのハンドルの格納場所

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「 B.4.8 エラー番号一覧 」参照）

[詳細説明]

- 信号出力器インタフェースを生成します。
- 生成した信号出力器インタフェースは、*waveName* で指定した名前と関連付けられます。また、*count*、および *pinNameList* で指定した端子が生成されます。
- この関数が正常終了すると、生成した信号出力器インタフェースのハンドルを取得できます。後の信号出力器インタフェースに対する制御は、ここで取得したハンドルを指定して行います。
なお、ハンドルは、[SuoGetWaveHandle](#) 関数により取得することもできます。

[使用例]

```
#include    "suo.h"
SuoHandle hWave1;

SuoUserEntry void MakeUserModel(const char *option)
{
    .....
    char* pinNameList[4] = {"P00", "P01", "P02", "P03"};
    SuoCreateWave("WAVE1", 4, pinNameList, &hWave1);    /* Create "WAVE1" */
}
```


SuoGetWaveHandle

信号出力器インタフェースのハンドルを取得します。

[指定形式]

```
#include    "suo.h"
SuoHandle  SuoGetWaveHandle(const char* waveName);
```

[引数]

引数	説明
waveName	信号出力器の名前

[戻り値]

マクロ	説明
信号出力器インタフェースのハンドル	正常終了
NULL	不正終了

[詳細説明]

- 指定した信号出力器インタフェースのハンドルを取得します。
- waveName には [SuoCreateWave](#) 関数で指定した名前を指定してください（異なる名前を指定した場合は NULL が返ります）。

[使用例]

```
#include    "suo.h"
SuoHandle  hWave1;

void func1(void)
{
    .....
    hWave1 = SuoGetWaveHandle("WAVE1");      /* Get handle of "WAVE1" */
}
```

SuoSendWaveFile

信号出力器による信号データ（信号データ・ファイル）の送信を行います。

注意 この関数を **MakeUserModel** 関数内で呼び出すことはできません。コールバック関数内でのみ呼び出すことができます。

[指定形式]

```
#include "suo.h"
int SuoSendWaveFile(SuoHandle handle, const char* waveFile);
```

[引数]

引数	説明
<i>handle</i>	信号出力器インタフェースのハンドル
<i>waveFile</i>	信号データエディタ ウィンドウで編集したのち保存した信号データ・ファイル名 なお、ファイル名を相対パスで指定した場合、ユーザ・モデル（UserModel.dll）のパスからの相対として扱われます。

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「B.4.8 エラー番号一覧」参照）

[詳細説明]

- 信号データエディタ ウィンドウで編集したのち保存した信号データ・ファイル（*.wvi）に記録されている信号データの送信を開始します。
- 信号データの送信が完了するまで時間がかかります。送信完了のタイミングを通知する場合は、[SuoSetNotifySentWaveCallback](#) 関数で送信完了通知関数を設定してください。
- 現在送信中の信号出力器インタフェースに対してこの関数を呼び出した場合は、送信中の信号データはキャンセルされ、新たに指定した信号データが送信されます。

[使用例]

```
#include    "suo.h"
SuoHandle hWave1;

void func1(void)
{
    .....
    SuoSendWaveFile(hSerial1, "foo.wvi");    /* Send pin data on "foo.wvi" */
}
```

SuoSetNotifySentWaveCallback

信号出力器の送信完了通知のコールバック登録を行います。

[指定形式]

```
#include "suo.h"
int SuoSetNotifySentWaveCallback(SuoHandle handle, SuoNotifySentWaveCallback func);
```

[引数]

引数	説明
<i>handle</i>	信号出力器インタフェースのハンドル
<i>func</i>	信号出力器の送信完了時の処理を行うユーザ定義関数へのポインタ（「 NotifySentWaveFunc 」参照）

[戻り値]

マクロ	説明
SUO_NOERROR	正常終了
エラー番号	不正終了（「 B.4.8 エラー番号一覧 」参照）

[詳細説明]

- 信号出力器の送信完了時の処理を行うユーザ定義関数を登録します。
- ここで登録された関数は、送信指定した信号データがすべて送信完了した際に呼び出されます。
- *func* に NULL を指定した場合は、登録を解除します。

[使用例]

```
#include    "suo.h"
void NotifySentWaveFunc (SuoHandle handle);
SuoHandle hWave1;

void func1(void)
{
    .....
    SuoSetNotifySentWaveCallback(hWave1, NotifySentWaveFunc);
                                     /* Set notify-sent-wave function */
}
/* Notify-sent-wave function */
void NotifySentWaveFunc (SuoHandle handle)
{
    .....
}
```

B.4.8 エラー番号一覧

提供インタフェース関数からの戻り値として返されるエラー番号（マクロ）の意味は、次のとおりです。
 なお、エラー番号は、ヘッダ・ファイル（suo.h）で定義されているマクロ名称で示されます。

表 B—4 エラー番号（マクロ）一覧

エラー番号（マクロ）	説明
SUO_NOERROR	正常終了
SUO_CANTALLOC	メモリが確保できませんでした。
SUO_ILLIFNAME	インタフェース名が不正です。 インタフェース名に NULL、または "" が指定されました。 または、ハンドル取得関数において、未生成のインタフェース名が指定されました。
SUO_ILLHANDLE	ハンドルが不正です。 生成したインタフェースのハンドル以外のハンドルが指定されました。
SUO_ILLPARAM	引数が不正です。 引数として指定できる値以外が指定されました。
SUO_CANTCALL	関数呼び出しができません。 MakeUserModel 関数でのみ呼び出すことができる関数が、MakeUserModel 関数以外の関数で呼び出されました。 または、MakeUserModel 関数以外で呼び出すことができない関数が、MakeUserModel 関数で呼び出されました。
SUO_CONFLICTRES	生成する資源が競合しています。 MakeUserModel 関数内で生成したインタフェース名、または端子名の中に同名のも存在します。
SUO_ILLFILENAME	ファイル名が不正です。 ファイル名に NULL、または無効な文字を含む名前が指定されました。
SUO_CANTOPENFILE	信号データ・ファイルをオープンすることができませんでした。 指定した信号データ・ファイルが存在しません。 または、指定した信号データ・ファイルがリード許可されていません。
SUO_ILLFILEFMT	【シリアル送信データ・ファイルの場合】 ファイルをオープンすることができませんでした。 ファイルが存在しない、ファイルがリード許可されていない、またはファイル名が不正であるかのいずれかです。 【信号データ・ファイルの場合】 ファイルの形式が不正です。 ファイル名に NULL、または無効な文字を含む名前が指定されました。
SUO_ILLFILECONT	ファイルの内容が不正です。 ファイルに記録されているデータの内容に矛盾がある、またはデータが1つも存在しません。
SUO_ILLPINNAME	端子名が不正です。 端子名に NULL、または "" が指定されました。
SUO_ILLADDRRANGE	アドレス範囲が不正です。 有効なアドレス範囲ではありません。

エラー番号 (マクロ)	説明
SUO_UNDERSENDING	すでに送信中です。 すでに送信中のため、新たに送信開始することができません。

B.5 ユーザ定義関数

この節では、ユーザが作成するユーザ定義関数について説明します。

次に、ユーザ定義関数（ユーザ・モデルのエントリ関数、およびコールバック関数）の一覧を示します。

表 B—5 ユーザ定義関数一覧

関数名	機能概要
MakeUserModel	ユーザ・モデルのエントリ関数
InitFunc	初期化のコールバック関数
ResetFunc	リセットのコールバック関数
NotifyTimerFunc	タイマの時間通知のコールバック関数
InputDigitalPinFunc	端子のデジタル・データ入力のコールバック関数
InputAnalogPinFunc	端子のアナログ・データ入力のコールバック関数
InputHighImpedanceFunc	端子のハイ・インピーダンス状態通知のコールバック関数
ReadExtbusFunc	外部バスのリード・アクセスのコールバック関数
WriteExtbusFunc	外部バスのライト・アクセスのコールバック関数
NotifySentSerialFunc	シリアルを送信完了通知のコールバック関数
ReceiveSerialFunc	シリアルを受信のコールバック関数
NotifySentWaveFunc	信号出力器の送信完了通知のコールバック関数

MakeUserModel

ユーザ・モデルのエントリ関数として、使用する資源の生成を行います。

注意 **MakeUserModel** はユーザ・モデルの静的エントリ関数であるため、この関数名を使用する必要があります。

[指定形式]

```
#include "suo.h"
SuoUserEntry void MakeUserModel(const char *option);
```

[引数]

引数	説明
<i>option</i>	シミュレータ・コンフィギュレーション・ファイルで指定したオプション文字列 なお、シミュレータ・コンフィギュレーション・ファイルでオプションを指定しなかった場合、NULL 文字 (" ") が入ります。

[戻り値]

なし

[詳細説明]

- この関数では、ユーザ・モデルで使用する資源の生成を行う必要があります（この関数以外の関数で資源の生成を行うことができません）。
- この関数では、必要に応じてコールバック関数の登録をする必要があります。
特に初期化のコールバック関数の登録は、この関数内で登録する必要があります（この関数以外での登録では、初期化タイミングを過ぎているため意味を持ちません）。

[使用例]

```
#include    "suo.h"
SuoHandle hTim1;
SuoHandle hPinP00;
SuoHandle hExtbus1;

void InitFunc(void);
void ResetFunc(void);

SuoUserEntry void MakeUserModel(const char *option)
{
    /* Create source */
    SuoCreateTimer("TIM1", &hTim1);                /* Create "TIM1" */
    SuoCreatePin("P00", &hPinP00);                /* Create "P00" */
    SuoCreateExtbus("EXTBUS1", 0x200000, 0x1000, &hExtbus1); /* Create "EXTBUS1" */

    /* Set callbacks */
    SuoSetInitCallback(InitFunc);                /* Set initialize function */
    SuoSetResetCallback(ResetFunc);            /* Set reset function */
}
```

InitFunc

コールバック関数として、初期化処理を行います。

注意 InitFunc はユーザ定義の関数名のプレース・ホルダであるため、この関数名を使用する必要はありません。

[指定形式]

```
#include "suo.h"
void InitFunc (void);
```

[引数]

なし

[戻り値]

なし

[詳細説明]

- InitFunc では、初期化処理を記述します。
- InitFunc をコールバック関数として登録するためには、[SuoSetInitCallback](#) 関数を使用します。

ResetFunc

コールバック関数として、リセット処理を行います。

注意 ResetFunc はユーザ定義の関数名のプレース・ホルダであるため、この関数名を使用する必要はありません。

[指定形式]

```
#include "suo.h"
void ResetFunc (void);
```

[引数]

なし

[戻り値]

なし

[詳細説明]

- ResetFunc では、リセット処理を記述します。
- ResetFunc をコールバック関数として登録するためには、[SuoSetResetCallback](#) 関数を使用します。

NotifyTimerFunc

コールバック関数として、タイマの時間通知の処理を行います。

注意 **NotifyTimerFunc** はユーザ定義の関数名のプレース・ホルダであるため、この関数名を使用する必要はありません。

[指定形式]

```
#include "suo.h"
void NotifyTimerFunc (SuoHandle handle);
```

[引数]

引数	説明
<i>handle</i>	タイマ・インタフェースのハンドル

[戻り値]

なし

[詳細説明]

- NotifyTimerFunc では、タイマの時間通知時の処理を記述します。
- NotifyTimerFunc をコールバック関数として登録するためには、[SuoSetNotifyTimerCallback](#) 関数を使用します。

InputDigitalPinFunc

コールバック関数として、端子のデジタル入力処理を行います。

注意 InputDigitalPinFunc はユーザ定義の関数名のプレース・ホルダであるため、この関数名を使用する必要はありません。

[指定形式]

```
#include "suo.h"
void InputDigitalPinFunc (SuoHandle handle, int pinValue);
```

[引数]

引数	説明
<i>handle</i>	端子インタフェースのハンドル
<i>pinValue</i>	次のいずれかの端子への入力値（デジタル・データ） - SUO_HIGH (=1) : HIGH 値 - SUO_LOW (=0) : LOW 値

[戻り値]

なし

[詳細説明]

- InputDigitalPinFunc では、端子のデジタル入力処理を記述します。
- InputDigitalPinFunc をコールバック関数として登録するためには、[SuoSetInputDigitalPinCallback](#) 関数を使用します。

InputAnalogPinFunc

コールバック関数として、端子のアナログ入力処理を行います。

注意 InputAnalogPinFunc はユーザ定義の関数名のプレース・ホルダであるため、この関数名を使用する必要はありません。

[指定形式]

```
#include "suo.h"
void InputAnalogPinFunc (SuoHandle handle, double pinValue);
```

[引数]

引数	説明
<i>handle</i>	端子インタフェースのハンドル
<i>pinValue</i>	端子への入力値（アナログ・データ）（単位：V）

[戻り値]

なし

[詳細説明]

- InputAnalogPinFunc では、端子のアナログ入力処理を記述します。
- InputAnalogPinFunc をコールバック関数として登録するためには、[SuoSetInputAnalogPinCallback](#) 関数を使用します。

InputHighImpedanceFunc

コールバック関数として、端子のハイ・インピーダンス状態通知の処理を行います。

注意 InputHighImpedanceFunc はユーザ定義の関数名のプレース・ホルダであるため、この関数名を使用する必要はありません。

[指定形式]

```
#include "suo.h"
void InputHighImpedanceFunc (SuoHandle handle);
```

[引数]

引数	説明
<i>handle</i>	端子インタフェースのハンドル

[戻り値]

なし

[詳細説明]

- InputHighImpedanceFunc では、デジタル／アナログ端子に接続されたすべての端子がハイ・インピーダンス状態になった際の処理を記述します。
- InputHighImpedanceFunc をコールバック関数として登録するためには、[SuoSetInputHighImpedanceCallback](#) 関数を使用します。

ReadExtbusFunc

コールバック関数として、外部バスのリード・アクセスの処理を行います。

注意 ReadExtbusFunc はユーザ定義の関数名のプレース・ホルダであるため、この関数名を使用する必要はありません。

[指定形式]

```
#include "suo.h"
void ReadExtbusFunc (SuoHandle handle, unsigned long addr, int accessSize, unsigned char data []);
```

[引数]

引数	説明
<i>handle</i>	外部バス・インタフェースのハンドル
<i>addr</i>	アドレス
<i>accessSize</i>	アクセス・サイズ
<i>data[]</i>	データ格納領域 なお、アクセス・サイズ分のデータを格納する必要があります。

[戻り値]

なし

[詳細説明]

- ReadExtbusFunc では、外部バスのリード・アクセスの処理を記述します。
- *data[]* へのデータ格納処理を行う必要があります。
- ReadExtbusFunc をコールバック関数として登録するためには、[SuoSetReadExtbusCallback](#) 関数を使用します。

WriteExtbusFunc

コールバック関数として、外部バスのライト・アクセスの処理を行います。

注意 WriteExtbusFunc はユーザ定義の関数名のプレース・ホルダであるため、この関数名を使用する必要はありません。

[指定形式]

```
#include "suo.h"

void WriteExtbusFunc (SuoHandle handle, unsigned long addr, int accessSize, const unsigned char data []);
```

[引数]

引数	説明
<i>handle</i>	外部バス・インタフェースのハンドル
<i>addr</i>	アドレス
<i>accessSize</i>	アクセス・サイズ
<i>data[]</i>	データ格納領域 なお、アクセス・サイズ分のデータを格納する必要があります。

[戻り値]

なし

[詳細説明]

- WriteExtbusFunc では、外部バスのライト・アクセスの処理を記述します。
- WriteExtbusFunc をコールバック関数として登録するためには、[SuoSetWriteExtbusCallback](#) 関数を使用します。

NotifySentSerialFunc

コールバック関数として、シリアルを送信完了通知の処理を行います。

注意 **NotifySentSerialFunc** はユーザ定義の関数名のプレース・ホルダであるため、この関数名を使用する必要はありません。

[指定形式]

```
#include "suo.h"
void NotifySentSerialFunc (SuoHandle handle);
```

[引数]

引数	説明
<i>handle</i>	シリアル・インタフェースのハンドル

[戻り値]

なし

[詳細説明]

- **NotifySentSerialFunc** では、シリアルを送信完了時の処理を記述します。
- **NotifySentSerialFunc** をコールバック関数として登録するためには、[SuoSetNotifySentSerialCallback](#) 関数を使用します。

ReceiveSerialFunc

コールバック関数として、シリアル受信時の処理を行います。

注意 ReceiveSerialFunc はユーザ定義の関数名のプレース・ホルダであるため、この関数名を使用する必要はありません。

[指定形式]

```
#include "suo.h"
void ReceiveSerialFunc (SuoHandle handle, unsigned long data, int status);
```

[引数]

引数	説明
<i>handle</i>	シリアル・インタフェースのハンドル
<i>data</i>	受信したシリアル・データ
<i>status</i>	次のいずれかの受信ステータス - 0 : 正常受信 - SUO_PARITYERR : パリティ・エラー (パリティ・ビットの不一致の場合) - SUO_FRAMINGERR : フレーミング・エラー (ストップ・ビットが検出されない場合)

[戻り値]

なし

[詳細説明]

- ReceiveSerialFunc では、シリアル受信時の処理を記述します。
- ReceiveSerialFunc をコールバック関数として登録するためには、[SuoSetReceiveSerialCallback](#) 関数を使用します。

NotifySentWaveFunc

コールバック関数として、信号出力器の送信完了通知の処理を行います。

注意 **NotifySentWaveFunc** はユーザ定義の関数名のプレース・ホルダであるため、この関数名を使用する必要はありません。

[指定形式]

```
#include "suo.h"
void NotifySentWaveFunc (SuoHandle handle);
```

[引数]

引数	説明
<i>handle</i>	信号出力器インタフェースのハンドル

[戻り値]

なし

[詳細説明]

- NotifySentWaveFunc では、信号出力器の送信完了通知の処理を記述します。
- NotifySentWaveFunc をコールバック関数として登録するためには、[SuoSetNotifySentWaveCallback](#) 関数を使用します。

B.6 サンプル・プログラム (Timer モデル)

この節では、シミュレータ GUI のユーザ・オープン・インタフェースを使用したユーザ・モデルのサンプル・プログラム (Timer モデル) について説明します。

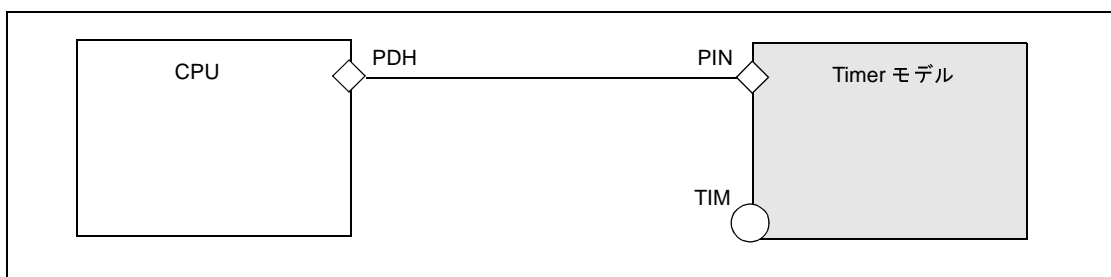
B.6.1 概要

Timer モデルは、タイマ・インタフェースを使用し、決められた時間間隔で端子に対して値を出力させるサンプル・プログラムです。

B.6.2 構成

Timer モデルは、PIN 端子と TIM タイマを生成します。生成した PIN 端子は CPU の PDH 端子に接続します。

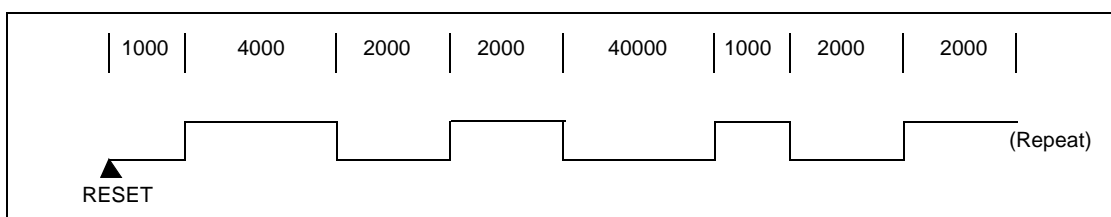
図 B—5 Timer モデルの構成



B.6.3 動作

タイマ・インタフェースを使用して決められた時間を計測し、端子 PDH に対して LOW/HIGH を交互に出力します。出力する値と時間の関係は次のとおりです。

図 B—6 Timer モデルの動作



B. 6.4 プロジェクト・ファイル

Timer モデルのプロジェクト・ファイル（Microsoft Visual C++）の設定情報は次のとおりです。

表 B—6 Timer モデルの設定情報

設定情報	内容
プロジェクト・タイプ	Win32 Dynamic-Link Library
構成ソース・ファイル	suolink.c, uo_timer.c
インクルード・ファイルのパス	suo.h の格納フォルダ

B. 6.5 プログラム詳細

次に、Timer モデルのサンプル・プログラムを示します。

- (1) Timer モデルのソース・ファイル (uo_timer.c)
- (2) シミュレータ・コンフィギュレーション・ファイル (smplus.cfg)
- (3) ターゲット・プログラムのソース・ファイル (lm_timer.c)

(1) Timer モデルのソース・ファイル (uo_timer.c)

```
#include <windows.h>
#include "suo.h"

/* Handle */
SuoHandle hTIM;
SuoHandle hPIN;

/* Wave-Table */
#define MAXWAVE 8
struct _WaveTable {
    unsigned longtime;           /* Wait Time [usec] */
    int pinValue;                /* Pin Value (SUO_HIGH or SUO_LOW) */
} waveTable[MAXWAVE] = {
    1000, SUO_HIGH,
    4000, SUO_LOW,
    2000, SUO_HIGH,
    2000, SUO_LOW,
    4000, SUO_HIGH,
    1000, SUO_LOW,
    2000, SUO_HIGH,
    2000, SUO_LOW
};
int waveIndex;

/* Declare */
void Reset(void);
void NotifyTimer(SuoHandle handle);
void puterr(int error);

/* MakeUserModel */
void SuoUserEntry MakeUserModel(const char *option)
{
    int error;
```



```
/* Create interface */
if((error = SuoCreateTimer("TIM", &hTIM)) != SUO_NOERROR){
    puterr(error);
    return;
}
if((error = SuoCreatePin("PIN", &hPIN)) != SUO_NOERROR){
    puterr(error);
    return;
}

/* Set callback */
SuoSetResetCallback(Reset);
SuoSetNotifyTimerCallback(hTIM, NotifyTimer);
}

/* Reset callback */
void Reset(void)
{
    int error;

    /* Initialize Wave-Table index */
    waveIndex = 0;

    /* Output LOW(initial value) to PIN */
    if((error = SuoOutputDigitalPin(hPIN, SUO_LOW)) != SUO_NOERROR){
        puterr(error);
        return;
    }

    /* Set wait time */
    if((error = SuoSetTimer(hTIM, SUO_USEC, waveTable[waveIndex].time)) != SUO_NOERROR){
        puterr(error);
        return;
    }
}

/* NotifyTimer callback */
void NotifyTimer(SuoHandle handle)
{
    int error;

    /* Output value to PIN */
    if((error = SuoOutputDigitalPin(hPIN, waveTable[waveIndex].pinValue)) != SUO_NOERROR){
        puterr(error);
        return;
    }
}
```

```

/* Set next Wave-Table index */
waveIndex++;
if(waveIndex >= MAXWAVE){
    waveIndex = 0;
}

/* Set wait time */
if((error = SuoSetTimer(hTIM, SUO_USEC, waveTable[waveIndex].time)) != SUO_NOERROR){
    puterr(error);
    return;
}
}

/* Report error */
void puterr(int error)
{
    char message[80];
    wsprintf(message, "The user open interface error (0x%04x) occurred.", error);
    MessageBox(NULL, message, "ERROR", MB_OK|MB_ICONERROR);
}

```

(2) シミュレータ・コンフィギュレーション・ファイル (smplus.cfg)

```

cpu = CPU('a');

#=====
# UO_TIMER description (CPU=uPD70F3261Y)
#=====

#---- Create UserOpen -----
uo_timer = Device("USEROPEN", "Release\uo_timer.dll");

#---- Pseudo Pin connection -----
wire_clock = Wire(1);
wire_clock += cpu.DebuggerPseudoPort("debugger_pseudo_pin_main_clkout");
wire_clock += uo_timer.Port("gui_pseudo_pin_clock_notice");
wire_reset = Wire(1);
wire_reset += cpu.DebuggerPseudoPort("debugger_pseudo_pin_reset_notice");
wire_reset += uo_timer.Port("gui_pseudo_pin_reset_notice");

#---- PIN connection -----
# UO_TIMER.PIN <--> CPU.PDH0
wire_PIN = Wire(1);
wire_PIN += uo_timer.Port("PIN");
wire_PIN += cpu.Port("PDH0");

```

(3) ターゲット・プログラムのソース・ファイル (lm_timer.c)

```
/* Target Program for UO_TIMER */

#pragma ioreg

void main( )
{
    unsigned char value;

    PMDH.0 = 1;          /* set port-input mode */
    PMDH.1 = 0;          /* set port-output mode */

    while( 1 ){
        value = PDH.0;   /* input signal from "PDH0" */
        PDH.1 = value;   /* output signal to "PDH1" */
    }
}
```

付録C 索引

【A】

Analog Button Properties ダイアログ … 561
Auto 変数 … 388

【C】

CPU レジスタ パネル … 372

【D】

DCU … 44, 60, 76, 279
DMA 機能 … 415

【E】

EEPROM ライブラリ … 35

【H】

HALT モード … 26, 50, 66, 82

【I】

I/O 保護領域 … 434
ID タグ … 356
InitFunc … 695
InputAnalogPinFunc … 699
InputDigitalPinFunc … 698
InputHighImpedanceFunc … 700
IOR パネル … 380

【M】

MakeUserModel … 693

【N】

NotifySentSerialFunc … 703
NotifySentWaveFunc … 705
NotifyTimerFunc … 697

【O】

Object Properties ダイアログ … 592

【P】

Parts Button Properties ダイアログ … 557
Parts Buzzer Properties ダイアログ … 586

Parts Key Properties ダイアログ … 565
Parts Led Properties ダイアログ … 574
Parts Level Gauge Properties ダイアログ … 570
Parts Matrix Led Properties ダイアログ … 582
Parts Segment LED Properties ダイアログ … 578
PC をここに設定 … 146
Print Preview ウィンドウ … 472
[Printf イベント] タブ … 454
Printf イベント … 230, 421, 422, 454

【R】

ReadExtbusFunc … 701
ReceiveSerialFunc … 704
Register 変数 … 388
ResetFunc … 696
ROM レス品 … 436
RRM 機能 … 165
Run-Beak 時間 … 218
Run-Break タイマ … 421, 422
Run-Break タイマ・イベント … 218

【S】

SuoCreateExtbus … 650
SuoCreatePin … 636
SuoCreateSerialCSI … 660
SuoCreateSerialUART … 658
SuoCreateTimer … 627
SuoCreateWave … 683
SuoGetExtbusHandle … 652
SuoGetMainClock … 625
SuoGetPinHandle … 638
SuoGetSerialHandle … 662
SuoGetSerialParameterCSI … 670
SuoGetSerialParameterUART … 668
SuoGetTimerHandle … 629
SuoGetWaveHandle … 685
SuoKillTimer … 632
SuoOutputAnalogPin … 641

- SuoOutputDigitalPin … 639
- SuoOutputHighImpedance … 642
- SuoSendSerialData … 672
- SuoSendSerialDataList … 674
- SuoSendSerialFile … 676
- SuoSendWaveFile … 686
- SuoSetInitCallback … 623
- SuoSetInputAnalogPinCallback … 645
- SuoSetInputDigitalPinCallback … 643
- SuoSetInputHighImpedanceCallback … 647
- SuoSetNotifySentSerialCallback … 678
- SuoSetNotifySentWaveCallback … 688
- SuoSetNotifyTimerCallback … 633
- SuoSetReadExtbusCallback … 653
- SuoSetReceiveSerialCallback … 680
- SuoSetResetCallback … 624
- SuoSetSerialParameterCSI … 665
- SuoSetSerialParameterUART … 663
- SuoSetTimer … 630
- SuoSetWriteExtbusCallback … 655
- 【W】**
- WriteExtbusFunc … 702
- 【あ行】**
- アクション・イベント … 230
- アクション・イベント ダイアログ … 452
- [Printf イベント] タブ … 454
- アクセス幅 … 435
- アップロード … 109, 123
- アップロード可能なファイル形式 … 123
- アドレス・オフセット設定 ダイアログ … 464
- アドレス範囲 … 435
- イベント・エリア … 327, 362
- イベント種別 … 421
- イベントの管理 … 233
- イベントのアドレスにジャンプする … 235
- イベントの設定状態を変更する … 234
- 実行中に設定／削除可能なイベント種別 … 238
- 設定したイベントを削除する … 235
- 特定のイベント種別のみ表示する … 234
- 有効イベント数の制限 … 236
- イベントの設定状態 … 234
- イベントパネル … 419
- イベント・マーク … 421
- 印刷アドレス範囲設定 ダイアログ … 470
- インタフェース関数 … 620
- 外部バス・インタフェース関数 … 649
- 基本インタフェース関数 … 622
- 時間インタフェース関数 … 626
- シリアル・インタフェース関数 … 657
- 信号出力器インタフェース関数 … 682
- 端子インタフェース関数 … 635
- ウインドウ・リファレンス … 252
- ウォッチ式 … 183, 393
- ウォッチ式データ・ファイルを開く ダイアログ … 504
- ウォッチ式として登録可能な対象 … 185
- ウォッチ式の登録 … 393
- ウォッチ式の登録方法 … 184
- ウォッチ式の入力形式 … 396
- ウォッチ式の表示形式 … 399
- ウォッチパネル … 393
- エディタパネル … 322
- エンコード … 458
- オープン・ブ레이크機能 … 291
- オプション設定ファイルを開く ダイアログ … 515
- オプション設定ファイルを保存 ダイアログ … 517
- オプションダイアログ … 492
- [全般 - ビルド／デバッグ] カテゴリ … 499
- [全般 - フォントと色] カテゴリ … 494
- オプション・バイト … 447
- オフセット値 … 364, 413
- オンチップ・デバッグ・セキュリティID … 46, 62, 78, 91
- 【か行】**
- 改行コード … 458
- 外部バス・インタフェース関数 … 649
- SuoCreateExtbus … 650
- SuoGetExtbusHandle … 652
- SuoSetReadExtbusCallback … 653

- SuoSetWriteExtbusCallback … 655
- 外部フラッシュ・メモリ … 120
- 外部フラッシュ・メモリ情報ファイル … 120, 442
- 外部フラッシュ・メモリ情報ファイルを選択 ダイアログ … 506
- 外部フラッシュ・メモリ・ダウンロード ダイアログ … 442
- カバレッジ測定 … 224
 - カバレッジ測定結果を表示する … 227
 - カバレッジ測定の設定をする … 224
- カレント PC 位置 … 329, 363
- カレント PC マーク … 329, 363
- 関数のエピローグ … 388
- 関数のプロローグ … 388
- 基本インタフェース関数 … 622
 - SuoGetMainClock … 625
 - SuoSetInitCallback … 623
 - SuoSetResetCallback … 624
- 逆アセンブル結果の表示 … 134
- 逆アセンブル・テキスト … 134, 360
- 逆アセンブル パネル … 360
- 逆アセンブル表示モード … 209, 414
- 共用体 … 388, 395
- グローバル変数 … 181
- クロックの設定 … 21, 43, 59, 75, 89, 99
- 構造体 … 388, 395
- コード … 364
- コード・カバレッジ測定 … 224
- コール・スタック情報 … 407
- コール・スタック パネル … 407
- コールバック関数 … 609
- ここまで実行 … 146
- 混合表示モード … 126, 208, 414
- コンフィギュレーション … 100
- 【さ行】**
- サンプル・プログラム … 706
- 時間インタフェース関数 … 626
 - SuoCreateTimer … 627
 - SuoGetTimerHandle … 629
 - SuoKillTimer … 632
- SuoSetNotifyTimerCallback … 633
- SuoSetTimer … 630
- 実行時間の計測 … 218
 - 実行開始から停止までの実行時間を計測する … 218
 - 測定可能時間の範囲 … 222
 - 任意区間の実行時間を計測する … 219
- 実行履歴の収集 … 193
 - 実行開始から停止までの実行履歴を収集する … 201
 - 実行履歴の収集を停止／再開する … 207
 - 実行履歴の表示内容を保存する … 216
 - 実行履歴を表示する … 208
 - 条件を満たした場合の実行履歴を収集する … 205
- トレース・データを検索する … 210
- トレース動作の設定をする … 193
- トレース・メモリをクリアする … 210
- 任意区間の実行履歴を収集する … 202
- 指定アドレスへ移動 … 136
- 指定位置へ移動 ダイアログ … 486
- 指定行へ移動 … 131
- 指定行へのジャンプ ダイアログ … 485
- シミュレータ GUI ウィンドウ … 521
- シミュレータ・コンフィギュレーション・ファイル … 616
- シミュレータ・コンフィギュレーション・ファイルを選択ダイアログ … 519
- 出力パネル … 429
- 初期化データ … 467
- 書式設定 ダイアログ … 526
- 処理中表示 ダイアログ … 491
- シリアル・インタフェース関数 … 657
 - SuoGetSerialHandle … 662
 - SuoCreateSerialCSI … 660
 - SuoCreateSerialUART … 658
 - SuoGetSerialParameterCSI … 670
 - SuoGetSerialParameterUART … 668
 - SuoSendSerialData … 672
 - SuoSendSerialDataList … 674
 - SuoSendSerialFile … 676

- SuoSetNotifySentSerialCallback … 678
- SuoSetReceiveSerialCallback … 680
- SuoSetSerialParameterCSI … 665
- SuoSetSerialParameterUART … 663
- シリアル ウィンドウ … 598
- シリアル送信データ・ファイル … 676
- 信号出力器インタフェース関数 … 682
 - SuoCreateWave … 683
 - SuoGetWaveHandle … 685
 - SuoSendWaveFile … 686
 - SuoSetNotifySentWaveCallback … 688
- 信号データエディタ ウィンドウ … 528
- 信号データ・ファイル … 528
- シンボル定義箇所へ移動 … 137
- スクロール範囲設定 ダイアログ … 483
- スコープの指定 … 181
- スタックからの関数呼び出し情報の表示 … 191
 - コール・スタック情報を表示する … 191
 - コール・スタック情報を保存する … 192
- ステータスバー … 260
- ステップ・イン実行 … 147, 260
- ステップ・オーバ実行 … 147, 260
- ステップ実行 … 146
- スマート・エディット機能 … 332
- 制御レジスタ … 176, 372
- セキュリティ ID … 282, 447
- セキュリティ・フラグ … 32, 305
- 接続方法 … 106
- 切断方法 … 106
- [全般 - ビルド/デバッグ] カテゴリ … 499
- [全般 - フォントと色] カテゴリ … 494
- ソース表示モード … 209, 414
- [ソース・レベル] タブ … 480
- ソース・レベル・デバッグ … 125, 322
- ソフトウェア・ブレーク … 150, 421, 422
- 【た行】**
- ターゲット・ボード … 22
- タイマ … 104
- タイマ開始 … 421
- タイマ開始イベント … 219, 422
- タイマ計測 … 421, 422
- タイマ計測イベント … 219
- タイマ終了 … 421
- タイマ終了イベント … 219, 422
- タイミングチャート ウィンドウ … 539
- タイミングチャート・ファイル … 539
- ダウンロード … 109, 316
- ダウンロード可能なファイル形式 … 113
- ダウンロード条件 … 438, 442
- ダウンロード条件の変更 … 116
- ダウンロードするファイルを選択 ダイアログ … 502
- ダウンロード・ファイル ダイアログ … 438
- タグ・ジャンプ … 133, 331, 430
- 端子インタフェース関数 … 635
 - SuoCreatePin … 636
 - SuoGetPinHandle … 638
 - SuoOutputAnalogPin … 641
 - SuoOutputDigitalPin … 639
 - SuoOutputHighImpedance … 642
 - SuoSetInputAnalogPinCallback … 645
 - SuoSetInputDigitalPinCallback … 643
 - SuoSetInputHighImpedanceCallback … 647
- 端子選択 ダイアログ … 537
- データ・カバレッジ測定 … 224
- データ検索 ダイアログ … 545
- データフラッシュ・エミュレーション機能 … 35
- データフラッシュ・メモリ … 20, 42, 58, 74, 88
- データ保存 ダイアログ … 488
- データ保存ファイルを選択 ダイアログ … 513
- テキスト編集 ダイアログ … 451
- デバッグ・コントロール・ユニット … 44, 60, 76, 279
- デバッグ情報 … 318
- デバッグ専用プロジェクト … 109
- デバッグ・ツールとの接続/切断 … 106
- デバッグ・ツールバー … 259
- デバッグ・マネージャパネル … 264
- 動作環境設定 … 17
- 特長 … 8
- トレース・イベント … 202
- トレース開始 … 421
- トレース開始イベント … 202, 422

- トレース検索 ダイアログ … 474
 - [ソース・レベル] タブ … 480
 - [命令レベル] タブ … 476
- トレース終了 … 421
- トレース終了イベント … 202, 422
- トレース動作の設定 … 193
- トレースパネル … 411
- トレース番号 … 412
- トレース・フレーム … 201
- トレース・メモリ … 195, 199, 295
- トレース・タイム・タグ … 296
- 【な行】**
- 内部スタティック変数 … 388
- 名前を付けて保存 ダイアログ … 510
- 入出力パネル ウィンドウ … 547
- 入力信号のマスク … 29, 30, 52, 68, 69, 84, 95
- 【は行】**
- ハードウェア・ブレーク … 150, 421, 422
- バイナリ・データ・フォーマット … 116
- 配列 … 388, 395
- 汎用レジスタ … 372
- 表示桁数設定 ダイアログ … 462
- ビルトイン・イベント … 201, 422
- ファイル・エンコードの選択 ダイアログ … 457
- ファイルの監視機能 … 332
- ファイルの保存設定 ダイアログ … 458
- ファイルを開く ダイアログ … 508
- ブート・スワップ … 33, 306
- ブート・ブロック・クラスタ … 34
- フェイルセーフ・ブレーク機能 … 28, 159
- フォーマット (CSI) ダイアログ … 604
- フォーマット (UART) ダイアログ … 602
- 不正なアクセス … 159
- フック処理 … 320
- フック処理を設定する … 242
- ブックマーク ダイアログ … 460
- 部品一覧 ダイアログ … 596
- フラッシュ … 45, 62, 77, 90
- フラッシュ・オプションの設定 ダイアログ … 445
- 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 … 31, 303
- フラッシュ・マクロ・サービス … 304
- プルアップ/プルダウン設定 ダイアログ … 590
- ブレーク … 51, 67, 83, 94, 103
- ブレーク・イベント … 153
- ブレークポイント … 149
- ブレークポイントの削除 … 153, 155
- ブレークポイントの種別 … 149, 290
- ブレーク要因 … 160, 413
- プログラマブルI/O領域 … 23, 47, 63, 79, 91, 101, 283
- プログラム内にアクションを挿入する … 230
 - printf を挿入する … 230
- プログラムの実行 … 143
 - プログラムを実行する … 144
 - プログラムをステップ実行する … 146
- プログラムの停止 … 149
 - 任意の場所で停止する (ブレーク・イベント) … 153
 - 任意の場所で停止する (ブレークポイント) … 149
 - プログラムの実行を手動で停止する … 149
 - 変数へのアクセスで停止する … 155
- プログラムの表示と変更 … 125
 - 逆アセンブルを表示する … 134
 - ソース・ファイルを表示する … 125
 - 他の処理と平行してビルドを実行する … 139
 - ライン・アセンブルを行う … 140
- プログラム・レジスタ … 176, 372
- プロジェクト・ツリーパネル … 266
- プロセッサ・エレメント … 264
- プロパティパネル … 268
 - [接続用設定] タブ … 271
 - [ダウンロード・ファイル設定] タブ … 316
 - [データフラッシュ・エミュレーション設定] タブ … 312
 - [デバッグ・ツール] タブ … 284
 - [フック処理設定] タブ … 320
 - [フラッシュ・オプション設定] タブ … 319
 - [フラッシュ・セルフ・エミュレーション設定] タブ … 303

- ヘキサ・フォーマット … 116
- ベリファイ … 288
- 変数値のポップアップ表示 … 331
- 変数へのアクセス … 155
- ポインタ型変数 … 388, 395
- ポイント・トレース … 422
- ポイント・トレース・イベント … 205, 347, 369
- ホット・プラグイン接続 … 107
- ホット・プラグイン … 257

- 【ま行】**
- マクロ … 690
- マクロ・サービス・エラー … 31
- 無条件トレース … 422
- 無条件トレース・イベント … 201, 234
- [命令レベル] タブ … 476
- 命令レベル・デバッグ … 125, 322
- メイン・ウインドウ … 255
- メニューバー … 256
- メモリ・アクセス … 25, 49, 65, 80, 93, 103
- メモリ検索 ダイアログ … 468
- メモリ種別 … 433
- メモリ初期化 ダイアログ … 466
- メモリの設定 … 24, 102
- メモリ・マッピング … 24, 48, 64, 80, 92, 102, 433
- メモリ・マッピング ダイアログ … 432
- メモリ, レジスタ, 変数の表示/変更 … 161
 - CPU レジスタを表示/変更する … 176
 - IOR を表示/変更する … 178
 - ウォッチ式を表示/変更する … 183
 - グローバル変数・スタティック変数を表示/変更する … 181
 - メモリを表示/変更する … 161
 - ローカル変数を表示/変更する … 181

- 【や行】**
- 有効イベント数の制限 … 236
- ユーザ定義関数 … 692
- ユーザ定義関数関数
 - InitFunc … 695
 - InputAnalogPinFunc … 699
 - InputDigitalPinFunc … 698
 - InputHighImpedanceFunc … 700
 - MakeUserModel … 693
 - NotifySentSerialFunc … 703
 - NotifySentWaveFunc … 705
 - NotifyTimerFunc … 697
 - ReadExtbusFunc … 701
 - ReceiveSerialFunc … 704
 - ResetFunc … 696
 - WriteExtbusFunc … 702
- ユーザ・オープン・インタフェース … 608
- ユーザ・モデル … 611
- 読み込み保護対象 … 383

- 【ら行】**
- ライン・アセンブル … 140, 364
- ラピッド・ビルド機能 … 139
- ラベル名 … 364, 413
- リアルタイム表示更新機能 … 165, 188
- リサイクル・モード … 128
- リセット … 144
- リセット・ベクタ … 33, 306
- リターン・アウト実行 … 148, 260
- ループ設定 ダイアログ … 535
- ローカル変数 … 181
- ローカル変数 パネル … 387
- ロード・モジュール・ファイル … 109

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2013.02.01	－	初版発行

CubeSuite+ V2.00.00 ユーザーズマニュアル
V850 デバッグ編

発行年月日 2013年2月1日 Rev.1.00

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



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス 販売株式会社 〒100-0004 千代田区大手町2-6-2 (日本ビル)

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。

総合お問合せ窓口：<http://japan.renesas.com/contact/>

CubeSuite+ V2.00.00