User's Manual

CubeSuite+ V2.01.00

統合開発環境

ユーザーズマニュアル　解析編

対象デバイス

78K0マイクロコントローラ

RL78ファミリ

78K0Rマイクロコントローラ

V850ファミリ

RXファミリ

RH850ファミリ

ルネサス エレクトロニクス
www.renesas.com

Rev.1.00　2013.09

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ものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、
   各品質水準は、以下に示す用途に製品が使用されることを意図しております。

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

   高品質水準：　輸送機器（自動車、電車、船舶等）、交通用信号機器、
   防災・防犯装置、各種安全装置等

   当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。 たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。 なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

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

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

このマニュアルの使い方

このマニュアルは，RH850 ファミリ，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アプリケーション・システムを開発する際の統合開発環境である CubeSuite+ について説明します。

CubeSuite+ は，RH850 ファミリ，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の統合開発環境（ソフトウエア開発における，設計，実装，デバッグなどの各開発フェーズに必要なツールをプラットフォームである IDE に統合）です。統合することで，さまざまなツールを使い分ける必要がなく，本製品のみを使用して開発のすべてを行うことができます。

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

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

構　　成　このマニュアルは，大きく分けて次の内容で構成しています。

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	：	$\overline{\text{XXX}}$（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	R20UT2682J	R20UT2682E
	RX 設計編	R20UT2683J	R20UT2683E
	V850 設計編	R20UT2134J	R20UT2134E
	RL78 設計編	R20UT2684J	R20UT2684E
	78K0R 設計編	R20UT2137J	R20UT2137E
	78K0 設計編	R20UT2138J	R20UT2138E
	RH850 コーディング編	R20UT2584J	R20UT2584E
	RX コーディング編	R20UT2470J	R20UT2470E
	V850 コーディング編	R20UT0553J	R20UT0553E
	コーディング編（CX コンパイラ）	R20UT2659J	R20UT2659E
	RL78,78K0R コーディング編	R20UT2140J	R20UT2140E
	78K0 コーディング編	R20UT2141J	R20UT2141E
	RH850 ビルド編	R20UT2585J	R20UT2585E
	RX ビルド編	R20UT2472J	R20UT2472E
	V850 ビルド編	R20UT0557J	R20UT0557E
	ビルド編（CX コンパイラ）	R20UT2142J	R20UT2142E
	RL78,78K0R ビルド編	R20UT2143J	R20UT2143E
	78K0 ビルド編	R20UT0783J	R20UT0783E
	RH850 デバッグ編	R20UT2685J	R20UT2685E
	RX デバッグ編	R20UT2702J	R20UT2702E
	V850 デバッグ編	R20UT2446J	R20UT2446E
	RL78 デバッグ編	R20UT2445J	R20UT2445E
	78K0R デバッグ編	R20UT0732J	R20UT0732E
	78K0 デバッグ編	R20UT0731J	R20UT0731E
	解析編	このマニュアル	R20UT2686E
	メッセージ編	R20UT2687J	R20UT2687E

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

目　　次

第1章 概 説

この章では，プログラム解析（解析ツール）の概要について説明します。

1.1 概 要

解析ツールは，CubeSuite+ が提供しているコンポーネントの一種で，RH850 ファミリ，RX ファミリ，V850 ファミリ，RL78 ファミリ，78K0R マイクロコントローラ，78K0 マイクロコントローラ用に開発されたプログラムの動作解析を支援します。

1.1.1 解析対象

解析ツールが解析対象とするプログラムは，プロジェクト・ツリー パネルにおいて，アクティブ・プロジェクトに設定されているプロジェクト内の C/C++ ソース・ファイル（ヘッダ・ファイルを含む）に限られます。

したがって，アセンブル・ソース・ファイルは解析対象外となります。

ただし，アセンブル・ソース・ファイルで定義されているシンボルのうち，C/C++ ソース・ファイルから参照されているものについては解析対象となります。

注意 1. C++ ソース・ファイルは，CC-RX でのみサポートしています。

2. 解析対象のプロジェクトが次のいずれかに該当する場合，解析ツールを使用することはできません。

- デバッグ専用プロジェクト

- マルチコア用ブートローダ・プロジェクト【RH850】

- 非サポートのビルド・ツールが登録されているプロジェクト

- ビルド・ツールが登録されていないプロジェクト

3. 【CA850】

K&R 形式の記述には対応していません。

備考 情報ファイル（関数一覧ファイル（*.mtfl）／変数一覧ファイル（*.mtvl））をインポートすることにより，アクティブ・プロジェクト以外の C/C++ ソース・ファイル／ヘッダ・ファイルにおける関数／変数の情報を強制的に表示することも可能です（「2.12 情報ファイルをインポート／エクスポートする」参照）。

なお，上記の解析対象ファイルのうち，より効率的に解析作業を行うために，任意のファイルを解析対象外ファイル／解析対象ファイルに指定することができます。

解析対象外ファイルに指定されたファイル内の関数情報／変数情報は，解析ツールが提供する各パネル（解析グラフ パネルを除く）では非表示となります。

また，解析対象ファイルに指定されたファイル内の関数情報／変数情報のみが，解析ツールが提供する各パネル（解析グラフ パネルを除く）で表示されます。

解析対象外ファイル／解析対象ファイルの指定方法は次のとおりです（解析目的に従って，より選択しやすい方法で指定してください）。

(1) 解析対象外ファイルを指定する場合（デフォルト）

- プロパティ パネルの［設定］タブ上の［解析対象］カテゴリ内［解析対象ファイルの指定方法］プロパティを［解析対象外ファイル］に指定します（デフォルト）。
- 同カテゴリ内の［解析対象外ファイル］プロパティを選択することにより欄内右端に表示される［...］ボタンをクリックします。
- オープンする解析対象外ファイルを指定 ダイアログ上において，解析対象外とするファイルを指定します。

(2) 解析対象ファイルを指定する場合

- プロパティ パネルの［設定］タブ上の［解析対象］カテゴリ内［解析対象ファイルの指定方法］プロパティを［解析対象ファイル］に指定します。
- 同カテゴリ内の［解析対象ファイル］プロパティを選択することにより欄内右端に表示される［...］ボタンをクリックします。
- オープンする解析対象ファイルを指定 ダイアログ上において，解析対象とするファイルを指定します。

1. 1. 2 解析情報の種別

解析ツールが取得／解析／表示する情報には，次の 2 つの種別があります。

(1) 静的解析情報

コンパイル・エラー，またはアセンブル・エラーが発生することなくビルドが完了した時点で表示可能となり，ビルド・ツールが生成したロード・モジュール，およびクロス・リファレンス情報を解析した関数情報／変数情報です。

したがって，静的解析情報を取得するためには，使用するビルド・ツールにおいて，ビルドの際にクロス・リファレンス情報を生成する設定となっている必要があります。

ただし，解析ツールでは，クロス・リファレンス情報をビルド・ツールに強制的に生成させる次のプロパティを用意しています。

- プロパティ パネル→［設定］タブ→［全般］カテゴリ→［静的解析を有効にする］プロパティ

このプロパティを［はい］に設定することにより，ビルド・ツールの設定に依存することなく，クロス・リファレンス情報を生成することができます（この設定が［いいえ］（デフォルト）の場合，現在のビルド・ツールの設定が優先されるため注意が必要です）。

(2) 動的解析情報

プログラムの実行によりデバッグ・ツールが取得したトレース・データ，リアルタイム RAM モニタ結果，またはカバレッジ結果を基に解析した関数情報／変数情報です。

したがって，動的解析情報を取得するためには，デバッグ・ツールのトレース機能，RRM 機能／疑似 RRM（RAM モニタ）機能，またはカバレッジ機能【IECUBE】【IECUBE2】【シミュレータ】が有効化されている必要があります。

ただし，解析ツールでは，デバッグ・ツールのこれらの機能を強制的に有効化する次のプロパティを用意しています。

- プロパティ パネル→［設定］タブ→［全般］カテゴリ→［動的解析を有効にする］プロパティ

このプロパティを［はい］に設定することにより，デバッグ・ツールの設定に依存することなく，上記のデバッグ・ツール機能を使用することができます（この設定が［いいえ］（デフォルト）の場合，現在のデバッグ・ツールの設定が優先されるため注意が必要です）。

注意 1.　上記の設定を行ったのち，デバッグ・ツールと接続する必要があります。
または，デバッグ中にこの設定を変更した場合，デバッグ・ツールと再接続する必要があります。
2.　使用するデバッグ・ツールがサポートしていない機能を有効化することはできません。
また，排他使用の機能を持つデバッグ・ツールの場合では，次の優先順位で機能を有効化します。
トレース機能＞ RRM 機能／疑似 RRM（RAM モニタ）機能＞カバレッジ機能
3.　リアルタイム OS「RI シリーズ」が提供しているプログラム解析ツール，またはタスク・アナライザ・ツールを使用している場合，動的解析情報を取得することはできません。

なお，デバッグ・ツールの各機能が解析ツールに提供する動的解析情報は次のとおりです。

（a）トレース機能

プログラムの実行履歴をトレース・データとして収集する機能です。
トレース機能は，次の動的解析情報を提供します。

- 実行時間（実行時間の割合／平均実行時間を含む）
- 実行回数（アクセス系（リード回数／ライト回数など）を含む）
- 変数値（最大値／最小値を含む）
- グラフ・データ（グラフ・データの取得方法がトレース・データ解析方式の場合）

注意 1.　【RH850】

- **【Full-spec emulator】【E1/E20】**
  デバッグ・ツールのプロパティ パネル→［デバッグ・ツール設定］タブ→［トレース］カテゴリ→［トレースの取得対象設定］プロパティの設定により，上記の動的解析情報の内容が異なります。
  - **［デバッグ対象のコアのみ］を選択している場合（デフォルト）**
    デバッグ・ツールは，デバッグ・マネージャ パネルで選択している PE*n* のみを対象にトレース・データを収集します。したがって，解析ツールも PE*n* のみを対象とした動的解析情報を表示します。
  - **［全てのコア］を選択している場合**
    デバッグ・ツールは，全 PE を対象にトレース・データを収集します。したがって，トレース・データ収集後，デバッグ・マネージャ パネルで PE*n* を切り替えることによって，解析ツールは対応した動的解析情報を表示します。
- **【シミュレータ】**
  デバッグ・ツールは，デバッグ・マネージャ パネルで選択している PE*n* のみを対象にトレース・データを収集します。したがって，解析ツールも PE*n* のみを対象とした動的解析情報を表示します。

2. 【Full-spec emulator】【IECUBE】【IECUBE2】【シミュレータ】
   実行時間を正しく取得するために，デバッグ・ツールのプロパティ パネルにおいて，次の指定を行ってください。
   - ［デバッグ・ツール設定］タブ→［トレース］カテゴリ→［実行前にトレース・メモリをクリアする］プロパティ→［はい］（デフォルト）
3. 【IECUBE【V850E1】【V850ES】】【E20【RX】】
   プログラム実行中にトレース パネル上のコンテキスト・メニューの［トレース停止］／［トレース開始］を選択すると，実行時間が不正な値となります。
4. 【IEUCBE【78K0】】【E1/E20【RH850】】【E1/E20【RX】】【EZ Emulator【RX】】
   トレース・タイム・タグ機能をサポートしていないため，関数の実行時間の表示，およびトレース・データ解析方式によるグラフ表示を行うことはできません。
5. 【E1/E20【RL78】】【EZ Emulator【RL78】】
   トレース・データから“分岐元アドレス”のみが取得できるため，関数一覧 パネル／コール・グラフ パネルにおける［実行回数］項目のみのサポートとなります。

備考 1. 【IECUBE【V850E1】【V850ES】】
トレース機能／カバレッジ機能 /RRM 機能は，一部が排他使用の機能であるため，［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優先的に使用する機能を指定する必要がありますが，この際に，同プロパティに［トレース］以外の［カバレッジ］／［RRM］を指定している場合でも，解析ツールが動的解析情報を取得するために必要となるトレース機能は有効となります。

2. 【IECUBE2【V850E2】】
   トレース機能／タイマ機能／カバレッジ機能は，一部が排他使用の機能であるため，［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優先的に使用する機能を指定する必要がありますが，この際に，同プロパティに［トレース］以外の［タイマ］／［カバレッジ］を指定している場合でも，解析ツールが動的解析情報を取得するために必要となるトレース機能は有効となります。

（b）RRM 機能／疑似 RRM（RAM モニタ）機能

プログラム実行中にリアルタイムにメモリ（変数／レジスタ／アドレス等）の内容を読み込む機能です。
RRM 機能／疑似 RRM（RAM モニタ）機能は，次の動的解析情報を提供します。
- グラフ・データ（グラフ・データの取得方法がリアルタイム・サンプリング方式の場合[注]）

注 【E1/E20【RL78】】
選択しているマイクロコントローラが Smart Analog IC 搭載品では，デバッグ・ツールがデータ収集モードに設定されている場合を除きます。
この場合は，Smart Analog 専用のサンプリング方式によりデータ収集を行いグラフを表示します。

注意 【RH850】
- 変数値／アドレス
デバッグ・マネージャ パネルで選択している PE*n* の PC 値を基にアドレスと値を決定します。

- レジスタ値

デバッグ・マネージャ パネルで選択している PE*n* の値を取得します。

- 対象領域

全 PE のアクセスを対象に読み込み可能です。

ただし，Local RAM self 領域については，デバッグ・マネージャ パネルで選択している PE*n* のみが対象となります。

備考 RRM 機能／疑似 RRM（RAM モニタ）機能により読み込みが可能な対象領域は，プロジェクトで選択しているマイクロコントローラ，および使用するデバッグ・ツールの種類により異なります（RRM 機能／疑似 RRM（RAM モニタ）機能とその対象領域の関係についての詳細は，「CubeSuite+ 統合開発環境 ユーザーズマニュアル デバッグ編」を参照してください）。

（c）カバレッジ機能【IECUBE】【IECUBE2】【シミュレータ】

カバレッジ測定を行う機能です。

カバレッジ機能は，次の動的解析情報を提供します。

- コード・カバレッジ
- データ・カバレッジ

注意 1. 【IECUBE【V850】】【IECUBE2【V850E2】】

データ・カバレッジはサポートしていません。

2. 【シミュレータ【RH850】】

全 PE のアクセスを対象にカバレッジ測定を行います。

ただし，Local RAM self 領域については，デバッグ・マネージャ パネルで選択している PE*n* のアクセスのみを対象に測定を行います。

1.2 特 長

次に，解析ツールの特長を示します。

- 関数情報の表示

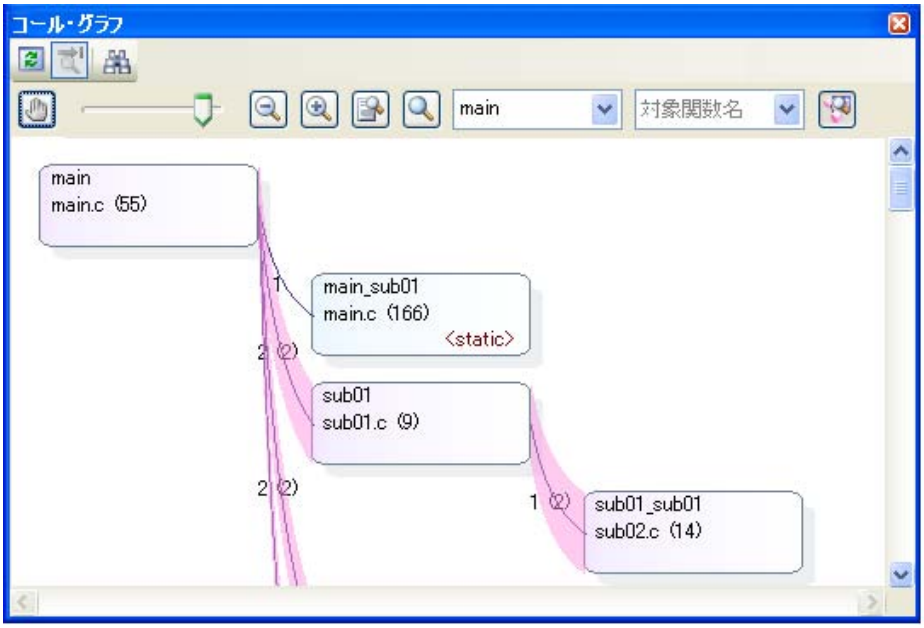
関数に関する情報として，静的解析情報（関数名／ファイル名／属性／戻り値の型／参照回数など），および動的解析情報（実行回数／実行時間／コード・カバレッジ率など）を表示します。

- 変数情報の表示

変数に関する情報として，静的解析情報（変数名／ファイル名／属性／型／参照回数など），および動的解析情報（リード回数／ライト回数／データ・カバレッジ率など）を表示します。

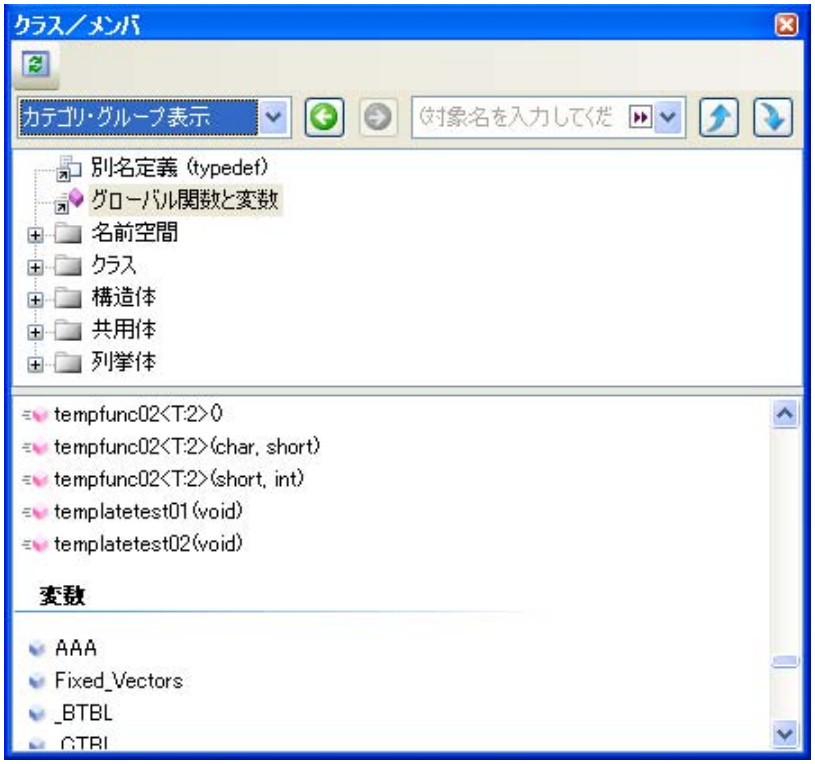
- コール・グラフの表示

プログラム中に存在する関数間の呼び出し関係をツリー構造の図（コール・グラフ）で表示します。



- クラス情報／関数情報／変数情報の表示

プログラム中のクラス情報【CC-RX】／関数情報／変数情報をツリー形式で表示します。



- デバッグ・ツールとの連携

解析ツールの各パネルから，デバッグ・ツールに対して次の操作を行うことができます。

- 指定した関数／変数が定義されている箇所へのジャンプ

  関数一覧 パネル／変数一覧 パネル／コール・グラフ パネル／クラス／メンバ パネル

- 指定した関数／変数に対するブレーク・イベントの設定

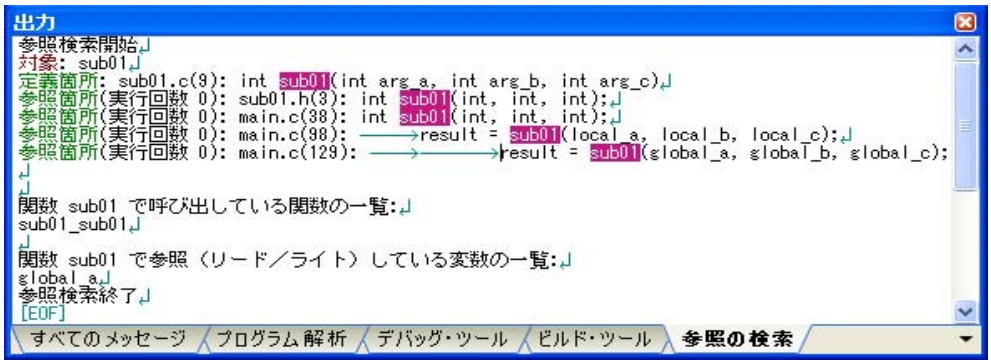
  関数一覧 パネル／変数一覧 パネル

- 指定した変数をウォッチ式に登録

  変数一覧 パネル

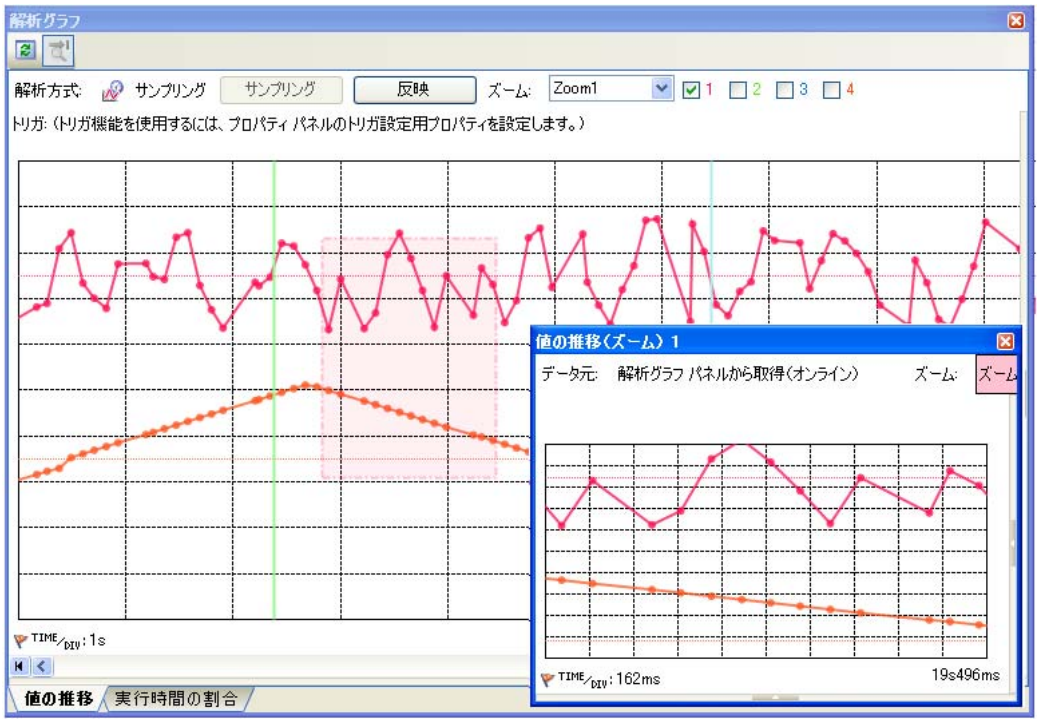
- 関数の参照箇所／変数の参照箇所の一覧表示

指定した関数／変数を参照している箇所を検索し，参照箇所の一覧を表示します。

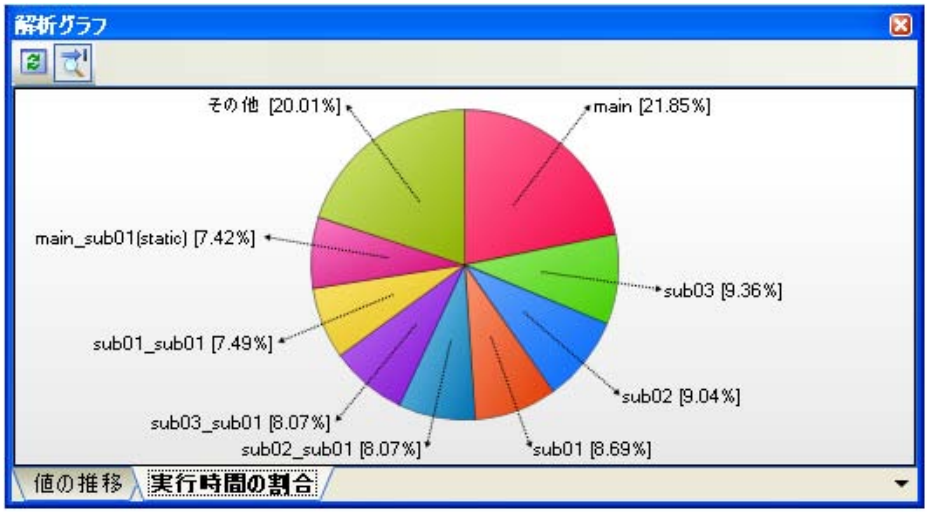


- 解析情報のグラフ化

- 変数，または Smart Analog 用に収集したデータ【E1/E20【RL78】】の値の推移を折れ線グラフで表示します。



- 関数の実行時間率を円グラフで表示します。



第 2 章 機 能

この章では，解析ツールの主な機能を操作手順とともに説明します。

2.1 概 要

解析ツールが取得した解析情報は，次のパネルで表示されます。

表 2―1 解析情報を表示するパネル

パネル	説明	
関数一覧 パネル	関数情報の表示	
変数一覧 パネル	変数情報の表示	
解析グラフ パネル	関数情報／変数情報のグラフ表示	
	［値の推移］タブ	変数，または Smart Analog 用に収集したデータ【E1/E20【RL78】】注 1 の値の推移（折れ線グラフ）
	［実行時間の割合］タブ	関数の実行時間の割合（円グラフ）
コール・グラフ パネル	関数間の呼び出し関係（コール・グラフ）の表示	
クラス／メンバ パネル	クラス情報【CC-RX】注 2 ／関数情報／変数情報のツリー表示	
値の推移（ズーム）パネル	［値の推移］タブで表示しているグラフのズーム表示	

注 1. 【E1/E20【RL78】】
Smart Analog 用のデータ収集は，選択しているマイクロコントローラが Smart Analog IC 搭載品の場合のみサポートされる機能です。

2. 【CC-RX】
クラス情報は，C++ ソース・ファイルを対象とする場合のみ提供される情報です。

また，各パネルでは，次の種類の関数／変数を解析情報の対象とします。

表 2―2 解析情報の対象となる関数／変数の種類

種類	パネル				
	関数一覧	変数一覧	解析グラフ 値の推移（ズーム）	コール・グラフ	クラス／メンバ
グローバル関数	○	－	－	○	○
スタティック関数	○	－	－	○	○
メンバ関数【CC-RX】注 1	○	－	－	○	○
グローバル変数	－	○	○	○	○

種類	パネル				
	関数一覧	変数一覧	解析グラフ 値の推移（ズーム）	コール・グラフ	クラス／メンバ
ファイル内スタティック変数	－	○	○	○	○
関数内スタティック変数	－	○	○	○	－
ローカル変数	－	－	－	－	－
IOR/SFR 注2	－	○	○	○	－
クラス変数【CC-RX】注1	－	○	○	○	○
インスタンス変数【CC-RX】注1	－	－	○	－	○

注 1. 【CC-RX】
C++ ソース・ファイルを対象とする場合のみ存在する関数／変数です。

2. 【RH850】【RX】【V850】 ：IOR
【RL78】【78K0R】【78K0】 ：SFR

これらの解析情報を検証することにより，未使用関数／未使用変数／ボトル・ネックとなっている処理の検出，およびコード・サイズ削減に有効なメモリ配置などを行うことが可能となります。

解析ツールの基本的な操作手順は，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。

注意 ビルド・ツールの設定に依存することなく，解析ツールに必要なクロス・リファレンス情報を生成するために，プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［静的解析を有効にする］プロパティを［はい］に指定したのち，ビルドを実行してください（「(1) 静的解析情報」参照）。

備考 1. “ロード・モジュールを作成する”についての詳細は，使用するコンパイラの「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。

2. コンパイル・エラー／アセンブル・エラーが発生することなくビルドが完了した場合，この時点で，関数一覧 パネル／変数一覧 パネル／コール・グラフ パネル／クラス／メンバ パネルにおいて，静的解析情報が表示可能となります。

（4）ダウンロードを実行する

使用するデバッグ・ツールの動作環境設定を行ったのち，CubeSuite+ とデバッグ・ツールを接続し，（3）で作成したロード・モジュールのダウンロードを実行します。

注意 デバッグ・ツールの設定に依存することなく，解析ツールに必要な動的解析情報を取得するために，プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［動的解析を有効にする］プロパティを［はい］に指定したのち，デバッグ・ツールと接続してください（「(2) 動的解析情報」参照）。

備考 “ダウンロードを実行する”についての詳細は，使用するマイクロコントローラの「CubeSuite+ 統合開発環境 ユーザーズマニュアル デバッグ編」を参照してください。

（5）プログラムを実行する

デバッグ・ツールにおいて，プログラムを実行します。

備考 “プログラムを実行する”についての詳細は，使用するマイクロコントローラの「CubeSuite+ 統合開発環境 ユーザーズマニュアル デバッグ編」を参照してください。

（6）解析対象のファイルを指定する

必要な場合，解析ツールが解析の対象とするファイルを指定します（「1. 1. 1 解析対象」参照）。

（7）関数情報を表示する

取得した関数情報を検証するために，関数一覧 パネルを表示します。

関数一覧 パネルでは，目的に応じて次の操作を行うことができます。

なお，デバッグ作業を進め，プログラムに変更を加えた場合は，上記（3）より操作を繰り返します。

（a）表示方法をカスタマイズする

パネル上の表示方法（表示項目の設定／並び替えなど）を変更します。

（b）定義箇所へジャンプする

パネル上の関数が定義されている箇所へジャンプします。

（c）関数にブレークポイントを設定する

パネル上の関数に対して，ブレークポイントを設定します。

（d）参照箇所を一覧表示する

パネル上の関数を参照している箇所を検索し，参照箇所一覧を表示します。

備考 アクティブ・プロジェクト以外で定義されている関数情報を表示することもできます。
表示方法についての詳細は，「2. 12 情報ファイルをインポート／エクスポートする」を参照してください。

(8) 変数情報を表示する

取得した変数情報を検証するために，変数一覧 パネルを表示します。
変数一覧 パネルでは，目的に応じて次の操作を行うことができます。
なお，デバッグ作業を進め，プログラムに変更を加えた場合は，上記（3）より操作を繰り返します。

(a) 表示方法をカスタマイズする

パネル上の表示方法（表示項目の設定／並び替えなど）を変更します。

(b) 定義箇所へジャンプする

パネル上の変数が定義されている箇所へジャンプします。

(c) 変数にブレーク・イベントを設定する

パネル上の変数に対して，アクセス系のブレーク・イベントを設定します。

(d) ウォッチ式に登録する

パネル上の変数をウォッチ パネル（ウォッチ 1）のウォッチ式として登録します。

(e) 参照箇所を一覧表示する

パネル上の変数を参照している箇所を検索し，参照箇所一覧を表示します。

備考 アクティブ・プロジェクト以外で定義されている変数情報を表示することもできます。
表示方法についての詳細は，「2. 12 情報ファイルをインポート／エクスポートする」を参照してください。

(9) 関数間の呼び出し関係（コール・グラフ）を表示する

関数間の呼び出し関係を検証するために，コール・グラフ パネルを表示します。
コール・グラフ パネルでは，目的に応じて次の操作を行うことができます。

(a) 定義箇所へジャンプする

コール・グラフ上の関数／変数が定義されている箇所へジャンプします。

(b) 関数／変数を検索する

コール・グラフ内に存在する関数／変数を検索します。

(10) クラス／関数／変数の情報を表示する

クラス情報【CC-RX】／関数情報／変数情報を検証するために，クラス／メンバ パネルを表示します。
クラス／メンバ パネルでは，目的に応じて次の操作を行うことができます。

(a) 定義箇所へジャンプする

ツリー上のクラス／関数／変数が定義されている箇所へジャンプします。

(b) 宣言箇所へジャンプする

ツリー上のクラス／関数／変数が宣言されている箇所へジャンプします。

備考 【CC-RX】

クラス情報は，C++ ソース・ファイルを対象とする場合のみ提供される情報です。

(11) 解析情報をグラフ化して表示する

取得した関数情報／変数情報をグラフ化して表示します。

なお，デバッグ作業を進め，プログラムに変更を加えた場合は，上記（3）より操作を繰り返します。

(a) 値の推移をグラフ化する

変数／レジスタ／アドレス等の値，または Smart Analog 用に収集したデータ値[注]と時間の関係を折れ線グラフで表示します。

注 【E1/E20【RL78】】

選択しているマイクロコントローラが Smart Analog IC 搭載品の場合のみサポートしている機能です。

(b) 関数の実行時間率をグラフ化する

関数の実行時間の割合を円グラフで表示します。

(12) 解析情報をファイルに保存する

取得した解析情報をファイルに保存します。

(13) 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　関数情報を表示する

プログラム中の関数情報（関数名／ファイル名／属性／参照回数／実行回数／コード・カバレッジ率など）を表示します。

メイン・ウイ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，現在表示可能な（「1. 1. 2　解析情報の種別」参照）最新の関数情報が関数一覧 パネルに表示されます。

ただし，解析対象外に指定されているファイル内の関数情報は表示されません（「1. 1. 1　解析対象」参照）。

なお，表示される関数情報の各項目についての詳細は，関数一覧 パネルを参照してください。

注意　【CA850】【CA78K0R】【CA78K0】

ビルド・ツールにおいてクリーンを実行すると，現在このパネルに表示している内容をすべてクリアします。

図 2—1　関数情報の表示（関数一覧 パネル）

備考 1. プログラムの実行が停止するごとに表示内容が更新されます（デフォルト）。
ただし，プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［プログラム停止時に更新を行う］プロパティの指定を［はい］（デフォルト）以外に変更した場合，プロパティ パネルでの設定に従った表示内容の更新を行います。

2. プログラムの実行により値が変化した情報は強調表示されます（強調表示の際の文字色／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）。強調表示は，ツールバーの  ボタンをクリックすることにより解除することができます。

3. カレント行マーク（▶）は，該当行がカレント行であることを示します。
カレント行マークのある関数を対象として，次の操作を行うことができます。
- 定義箇所へジャンプする
- ブレーク・イベントを設定する
- 参照箇所を一覧表示する

4. アクティブ・プロジェクト以外で定義され，かつアクティブ・プロジェクトから一度も参照されていない関数は，関数一覧 パネルでその情報が表示されません。この場合の関数情報の表示方法については，「2. 12　情報ファイルをインポート／エクスポートする」を参照してください。

2. 3 変数情報を表示する

プログラム中の変数情報（変数名／ファイル名／属性／リード回数／ライト回数／データ・カバレッジ率など）を表示します。

メイン・ウイ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，現在表示可能な（「1. 1. 2 解析情報の種別」参照）最新の変数情報が変数一覧 パネルに表示されます。

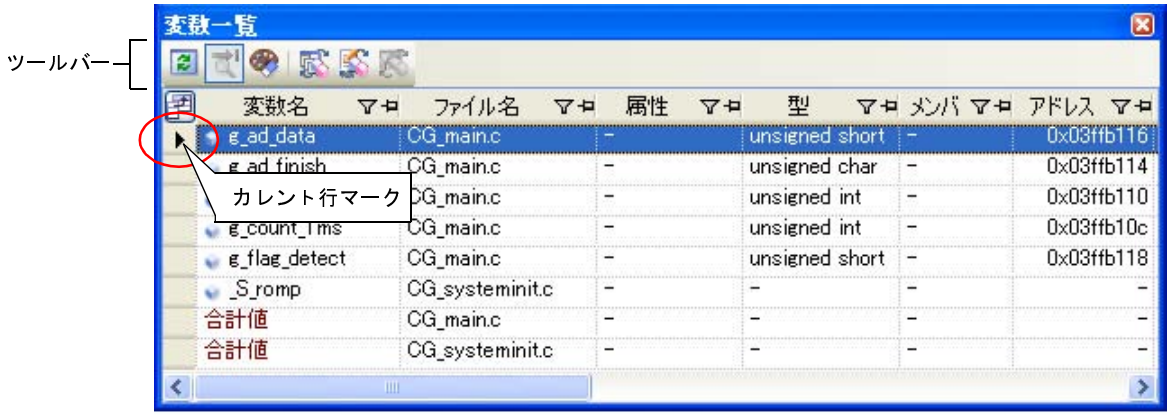
ただし，解析対象外に指定されているファイル内の変数情報は表示されません（「1. 1. 1 解析対象」参照）。

なお，表示される変数情報の各項目についての詳細は，変数一覧 パネルを参照してください。

注意 【CA850】【CA78K0R】【CA78K0】

ビルド・ツールにおいてクリーンを実行すると，現在このパネルに表示している内容をすべてクリアします。

図 2—2 変数情報の表示（変数一覧 パネル）



備考 1. プログラムの実行が停止するごとに表示内容が更新されます（デフォルト）。
ただし，プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［プログラム停止時に更新を行う］プロパティの指定を［はい］（デフォルト）以外に変更した場合，プロパティ パネルでの設定に従った表示内容の更新を行います。

2. プログラムの実行により値が変化した情報は強調表示されます（強調表示の際の文字色／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）。強調表示は，ツールバーの  ボタンをクリックすることにより解除することができます。

3. カレント行マーク（▶）は，該当行がカレント行であることを示します。
カレント行マークのある変数を対象として，次の操作を行うことができます。
- 定義箇所へジャンプする
- ブレーク・イベントを設定する
- 参照箇所を一覧表示する
- 値の推移をグラフ化する

4. アクティブ・プロジェクト以外で定義され，かつアクティブ・プロジェクトから一度も参照されていない変数は，変数一覧 パネルでその情報が表示されません。この場合の変数情報の表示方法については，「2. 12 情報ファイルをインポート／エクスポートする」を参照してください。

2.4 関数間の呼び出し関係（コール・グラフ）を表示する

取得した関数情報を基に，プログラム中に存在する関数間の呼び出し関係をツリー構造の図（コール・グラフ）で表示することができます。

メイン・ウイ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，現在表示可能な（「1.1.2 解析情報の種別」参照）最新のコール・グラフがコール・グラフ パネルに表示されます。

ただし，解析対象外に指定されているファイル内の関数情報／変数情報は表示されません（「1.1.1 解析対象」参照）。

なお，表示されるコール・グラフについての詳細は，コール・グラフ パネルを参照してください。

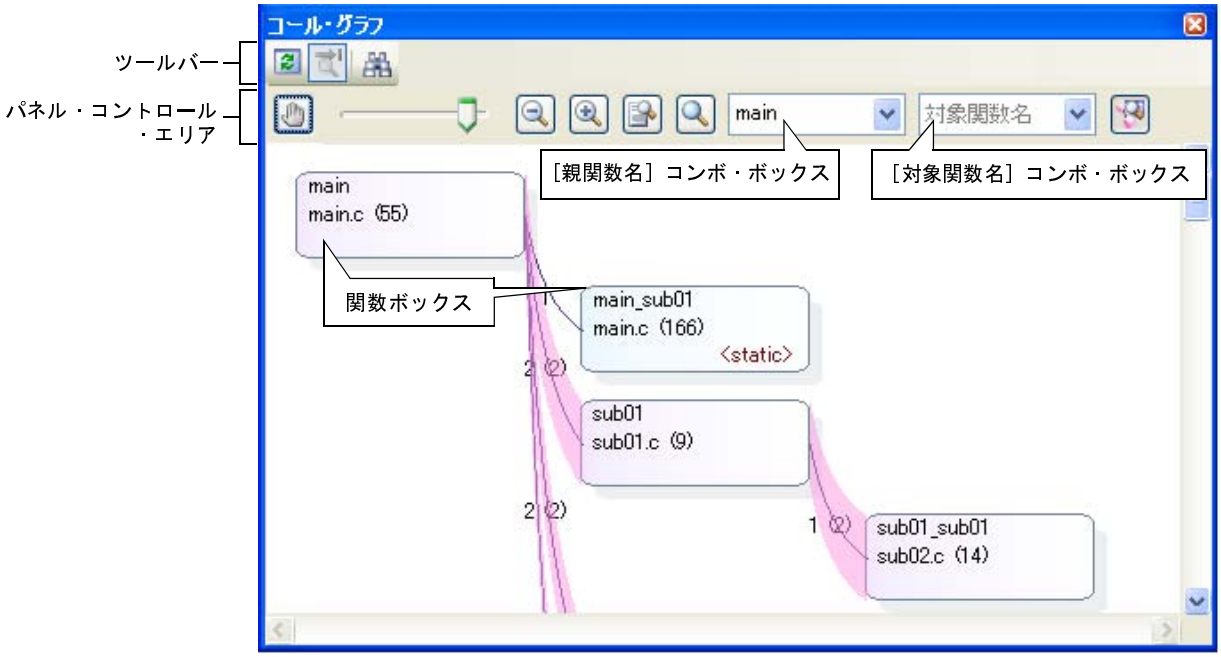
注意 1. デバッグ・ツールがトレース機能をサポートしていない場合，またはデバッグ・ツールのトレース機能を有効化していない場合，コール・グラフにおいて，動的解析情報（実行回数／リード回数／ライト回数）を表示することはできません。

2. 【RH850】
コール・グラフ内の動的解析情報（実行回数／リード回数／ライト回数）と現在デバッグ・マネージャ・パネルで選択してる PE*n* との関係についての詳細は，「(a) トレース機能」を参照してください。

3. 【CC-RX】
C++ ソース・ファイルを対象としている場合，暗黙的に呼び出されるクラス型のコンストラクタ／デストラクタの呼び出しは，コール・グラフに表示されません。

4. 【CA850】【CA78K0R】【CA78K0】
ビルド・ツールにおいてクリーンを実行すると，現在表示しているコール・グラフは消失します。

図 2—3 関数間の呼び出し関係の表示（コール・グラフ パネル：全体表示）



表示されるコール・グラフに対して，次の操作を行うことができます。

(1) 親関数を変更して表示する

デフォルトでは，“main”／“reset”[注]，またはそれを含む関数名のうち最初に出現した関数を親関数とみなし，その関数がコール・グラフ内の最左端に，ボックス形式（関数ボックス）で配置されます。

表示するコール・グラフの親関数を変更する場合は，［親関数名］コンボ・ボックスのドロップダウン・リストにより指定します。

注　選択しているマイクロコントローラにより関数名は異なります。

- 【RH850】【V850】【RL78】【78K0R】【78K0】: main
- 【RX】　: reset

備考　【RH850】【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では，［親関数名］コンボ・ボックスにおいて“PM*n*”【RH850】／“PE*n*”【V850E2】を指定することができます。この場合は，該当 PM*n*/PE*n* で実行される関数のみを対象としたコール・グラフが表示されます。

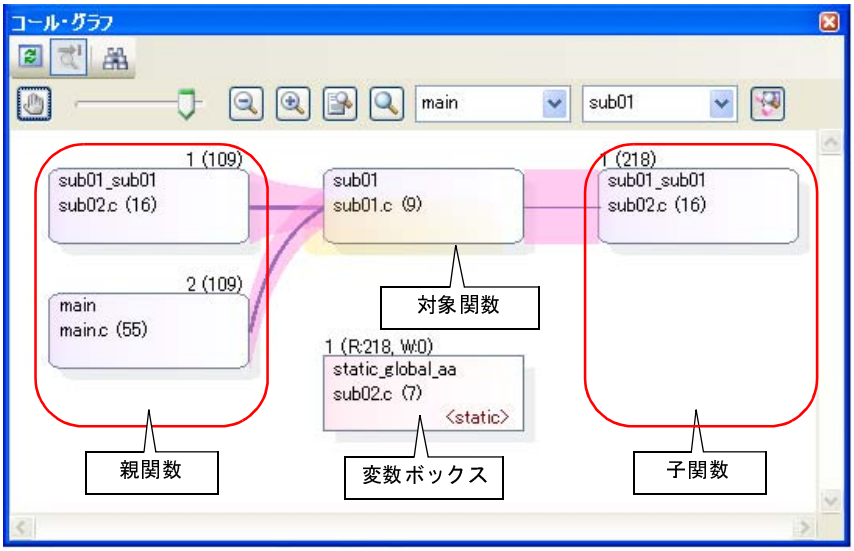
(2) 指定した関数の親関数／子関数を表示する

パネル・コントロール・エリアの  ボタンをクリックすることにより，現在，［対象関数名］コンボ・ボックスで指定している関数に対する親関数と子関数を表示する詳細表示にコール・グラフが切り替わります。

また，詳細表示では，対象関数からアクセスしているグローバル変数／ファイル内スタティック変数／関数内スタティック変数が存在する場合，その変数についても変数ボックスとして表示されます。

なお，コール・グラフを全体表示（デフォルト）に戻す場合は，再び  ボタンをクリックします。

図 2—4　親関数／子関数の表示（コール・グラフ パネル：詳細表示）



(3) 定義箇所へジャンプする

任意の関数ボックス／変数ボックスをダブルクリックすることにより，対象関数／変数が定義されているソース・テキスト箇所へジャンプすることができます（「2.7　定義箇所へジャンプする」参照）。

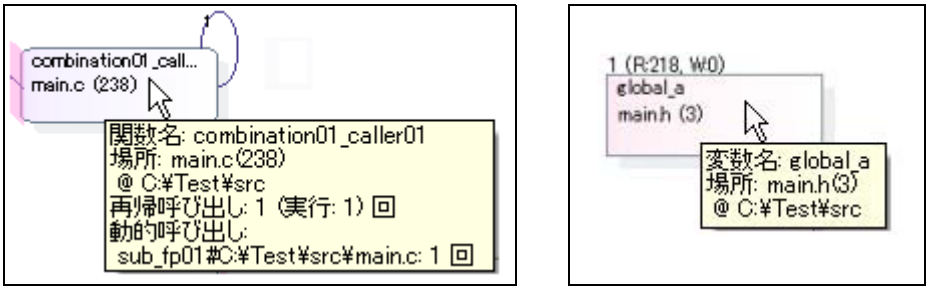
(4) 関数／変数の情報をポップアップ表示する

コール・グラフ内の関数ボックス／変数ボックスにマウス・カーソルを重ねることにより，対象関数／変数の情報がポップアップ表示されます。

注意　パネル・コントロール・エリアの  ボタンの設定により，マウスのドラッグによるスクロールを許可している場合，この機能を使用することはできません。

図 2—5　関数情報／変数情報のポップアップ表示例

【関数ボックス】　　【変数ボックス】



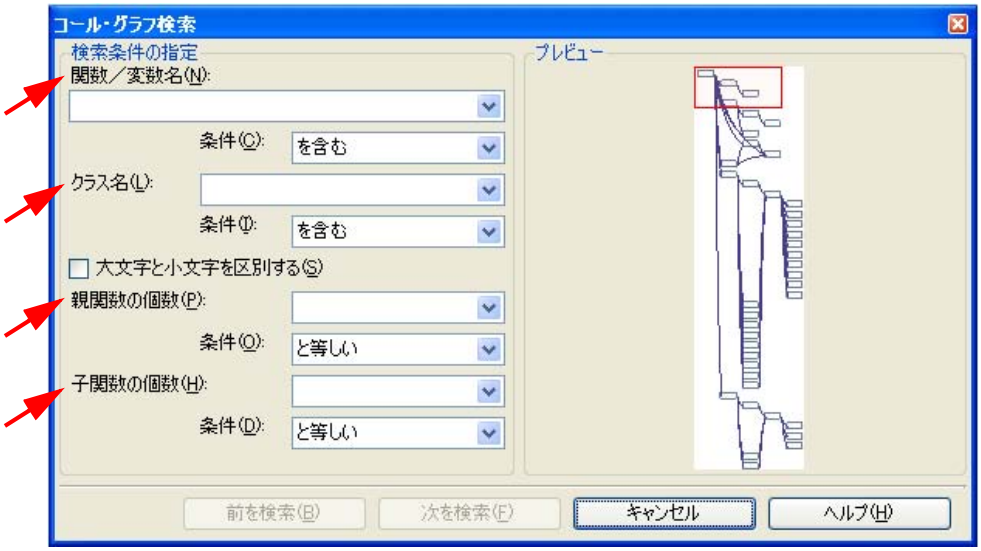
(5) 関数／変数を検索する

コール・グラフ内に存在する関数／変数を検索することができます。

操作は，コール・グラフ パネルのツールバーの  ボタンをクリックすることでオープンする，コール・グラフ検索 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—6　関数の検索（コール・グラフ検索 ダイアログ）



(a) ［関数／変数名］の指定

検索対象となる関数名／変数名を指定します。

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：2046 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）。

次に，指定した文字列に対して条件を指定する場合は，直下の［条件］コンボ・ボックスにおいて任意の条件を選択します（デフォルトでは，［を含む］が指定されます）。

なお，［大文字と小文字を区別する］チェック・ボックスを選択した場合は，大文字と小文字を区別して検索します。

備考 関数名／変数名は，次の操作によっても指定することができます。

- 関数一覧 パネル／変数一覧 パネルの任意の行をこのエリアにドラッグ・アンド・ドロップ
- 任意の文字列をこのエリアにドラッグ・アンド・ドロップ

(b)［クラス名］の指定【CC-RX】

関数／変数の検索条件の 1 つとして必要な場合，検索対象のメンバ関数／メンバ変数が属しているクラス名を指定します。

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：2046 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）。

次に，指定した文字列に対して条件を指定する場合は，直下の［条件］コンボ・ボックスにおいて任意の条件を選択します（デフォルトでは，［を含む］が指定されます）。

なお，［大文字と小文字を区別する］チェック・ボックスを選択した場合は，大文字と小文字を区別して検索します。

備考 クラス名は，次の操作によっても指定することができます。

- 任意の文字列をこのエリアにドラッグ・アンド・ドロップ

(c)［親関数の個数］の指定

関数の検索条件の 1 つとして必要な場合，検索対象の関数の親関数の個数を指定することができます。

キーボードより数値を直接入力するか（指定可能範囲：0 ～ 65535），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）。

次に，指定した数値に対して条件を指定する場合は，直下の［条件］コンボ・ボックスにおいて任意の条件を選択します（デフォルトでは，［と等しい］が指定されます）。

(d)［子関数の個数］の指定

関数の検索条件の 1 つとして必要な場合，検索対象の関数の子関数の個数を指定することができます。

キーボードより数値を直接入力するか（指定可能範囲：0 ～ 65535），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）。

次に，指定した数値に対して条件を指定する場合は，直下の［条件］コンボ・ボックスにおいて任意の条件を選択します（デフォルトでは，［と等しい］が指定されます）。

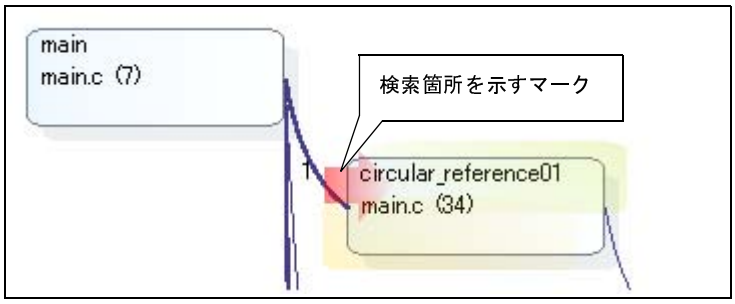
(e)［前を検索］／［次を検索］ボタンのクリック

［前を検索］ボタンをクリックすると，指定した条件でコール・グラフ内の最下段から上段方向へ関数名／変数名の検索を行い，該当関数ボックス／変数ボックスを強調表示します。

［次を検索］ボタンをクリックすると，指定した条件でコール・グラフ内の最上段から下段方向へ関数名／変数名の検索を行い，該当関数ボックス／変数ボックスを強調表示します。

なお，コール・グラフ パネル上で関数ボックス／変数ボックスを選択している場合は，対象関数／変数から上段方向／下段方向へ検索を開始します。

図 2—7 関数の検索箇所



備考 1. プログラムの実行が停止するごとに表示内容が更新されます（デフォルト）。
ただし，プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［プログラム停止時に更新を行う］プロパティの指定を［はい］（デフォルト）以外に変更した場合，プロパティ パネルでの設定に従った表示内容の更新を行います。

2. プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［定義箇所がない関数／変数をコール・グラフの表示対象とする］プロパティの指定を［はい］に変更した場合，ソース・ファイルが存在しない関数／変数をコール・グラフに含めることができます。

2.5 クラス／関数／変数の情報を表示する

プログラム中のクラス情報【CC-RX】／関数情報／変数情報をツリー形式で表示することができます。

これらの情報は，メイン・ウイ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，クラス／メンバ パネルに表示されます。

ただし，解析対象外に指定されているファイル内のクラス情報【CC-RX】／関数情報／変数情報は表示されません（「1.1.1 解析対象」参照）。

なお，表示されるツリーについての詳細は，クラス／メンバ パネルを参照してください。

注意 1. 【CC-RX】

クラス情報は，C++ ソース・ファイルを対象とする場合のみ提供される情報です。

2. 【CA850】【CA78K0R】【CA78K0】

ビルド・ツールにおいてクリーンを実行すると，現在このパネルに表示しているツリーの内容をすべてクリアします。

図 2—8 クラス情報の表示（クラス／メンバ パネル）

取得した情報は，クラス／カテゴリ用ツリー・エリア（上段）と関数／変数用リスト・エリア（下段）の 2 つのエリアで表示されます。

クラス／カテゴリ用ツリー・エリアにおいて表示されるツリー上の任意のノードを選択すると，そのノードに関する情報（該当ノードで定義されている関数名／変数名など）が関数／変数用リスト・エリアに一覧表示されます。

なお，クラス／カテゴリ用ツリーは，パネル・コントロール・エリアの［ビュー設定］コンボ・ボックスにより，次の 5 つのグループに分類して表示することができます。

表 2—3 クラス／メンバ パネルにおけるツリーの分類方法

リスト項目	説明
カテゴリ・グループ表示（デフォルト）	クラスやインタフェースなどの種類で分類して表示します。
アクセス・グループ表示	クラスなどに設定されたアクセス指定子で分類して表示します。
名前空間グループ表示	クラスなどが定義されている名前空間で分類して表示します。
ファイル・グループ表示	ファイルで分類して表示します。
アルファベット・グループ表示	アルファベット順で分類して表示します。

備考 1. 現在選択しているノードを対象として，次の操作を行うことができます。

- 定義箇所へジャンプする
- 宣言箇所へジャンプする

2. 【CC-RX】

ノード（カテゴリ・ノードを除く）にマウス・カーソルを重ねることにより，属している“*名前空間名*”をポップアップ表示します。

2. 6 表示方法をカスタマイズする

関数一覧 パネル／変数一覧 パネル上の各項目（列），およびその解析情報は，次の操作により表示方法をカスタマイズすることができます。

なお，操作はいずれも各パネル上のヘッダ・エリア（項目名が表示されている箇所）を対象に行います。

- 表示項目を設定する
- 表示項目を並び替える
- 特定項目を固定表示に設定する
- 解析情報をソート表示する
- 解析情報をフィルタ表示する

備考 列の選択 ダイアログ上の［デフォルト］ボタンをクリックすることにより，上記の操作により行ったカスタマイズをすべてデフォルトの状態に戻すことができます。

2. 6. 1 表示項目を設定する

関数一覧 パネル／変数一覧 パネル上に表示する項目（列）の表示／非表示を任意に設定することができます。

（1）表示項目を限定する場合
（2）表示項目を追加する場合

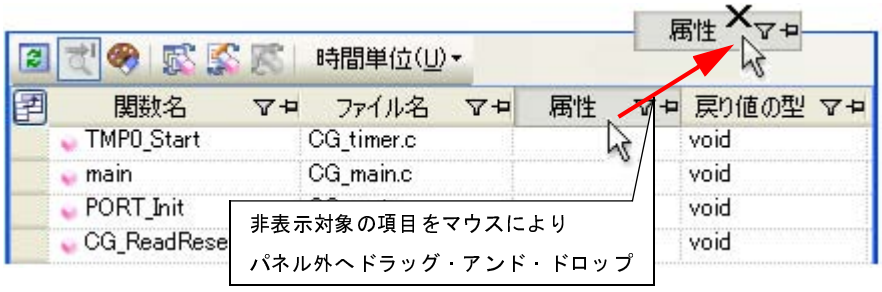
（1）表示項目を限定する場合

操作方法は，次の 2 通りです。

（a）パネル内での操作

非表示とする対象の項目名を，マウスによりパネル外へドラッグ・アンド・ドロップします。

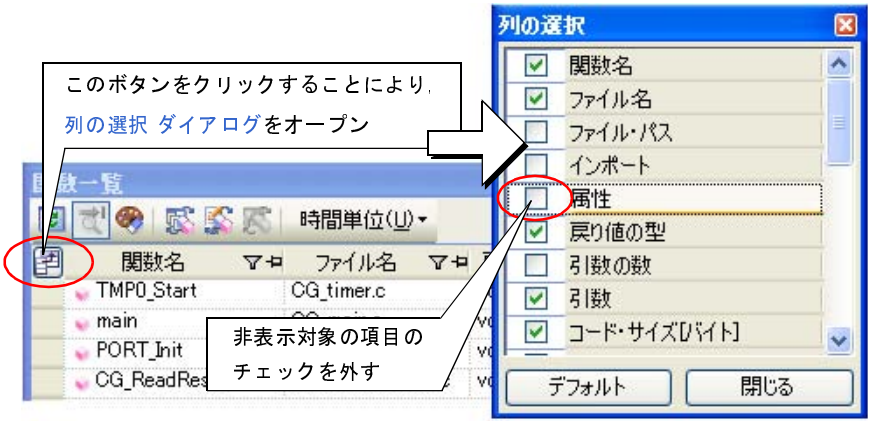
図 2—9 表示項目の限定（パネル内での操作）



（b）列の選択 ダイアログによる操作

パネルのヘッダ・エリア左端の ボタンをクリックすることによりオープンする列の選択 ダイアログにおいて，非表示とする対象項目名のチェック・ボックスのチェックを外します。

図 2—10 表示項目の限定（列の選択 ダイアログによる操作）



（2）表示項目を追加する場合

パネルのヘッダ・エリア左端の ボタンをクリックすることによりオープンする列の選択 ダイアログにおいて，表示する対象項目名のチェック・ボックスをチェックするか，または対象項目名をマウスによりパネル上の情報表示エリアへ直接ドラッグ・アンド・ドロップします。

図 2—11 表示項目の追加（列の選択 ダイアログのチェック・ボックスでの操作）

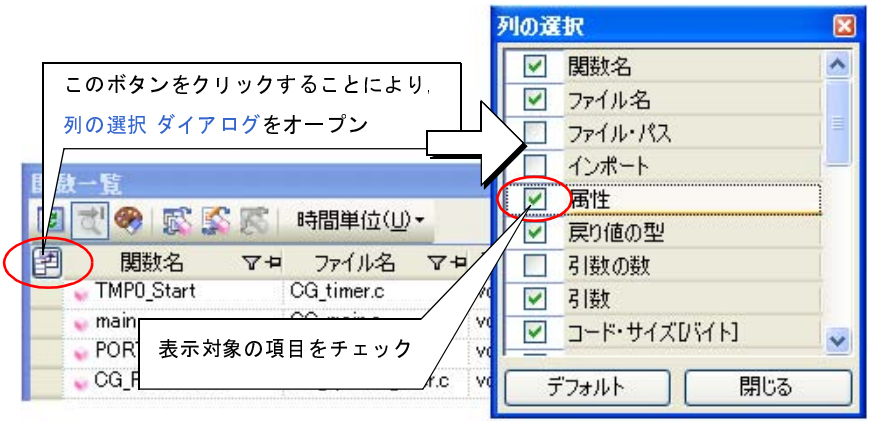
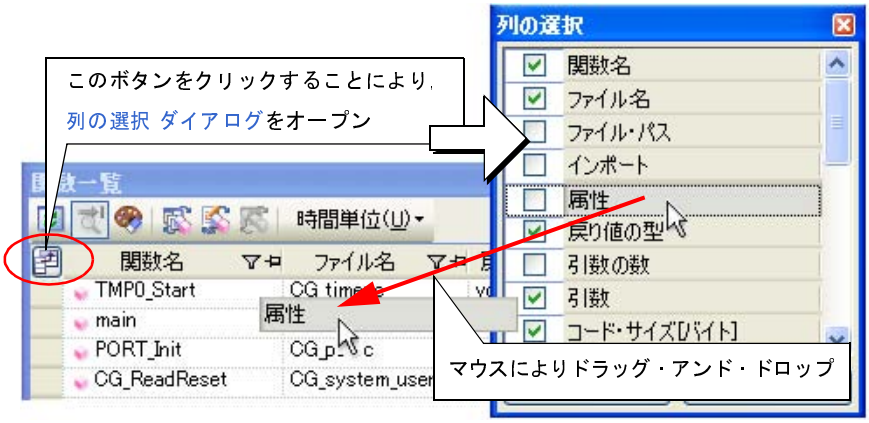


図 2—12 表示項目の追加（列の選択 ダイアログからのドラッグ・アンド・ドロップ操作）



2. 6. 2　表示項目を並び替える

関数一覧 パネル／変数一覧 パネル上の表示項目（列）の順序を並び替えることができます。

操作方法は，次の 2 通りです。

（1）パネル内での操作

（2）列の選択 ダイアログからの操作

（1）パネル内での操作

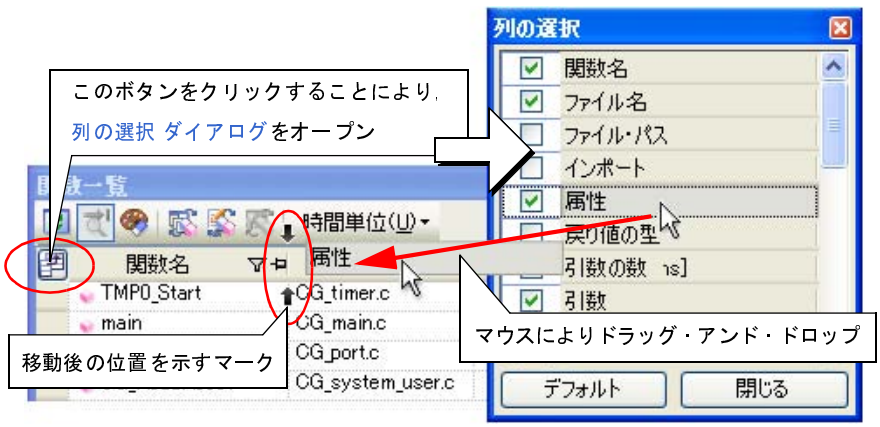
移動対象の項目名を，マウスにより任意の列（ヘッダ・エリア）へ直接ドラッグ・アンド・ドロップします。

図 2—13　表示項目の並び替え（パネル内での操作）

（2）列の選択 ダイアログからの操作

パネルのヘッダ・エリア左端の  ボタンをクリックすることによりオープンする列の選択 ダイアログにおいて，移動対象の項目名を，マウスによりパネル上の任意の列（ヘッダ・エリア）へ直接ドラッグ・アンド・ドロップします。

図 2—14　表示項目の並び替え（列の選択 ダイアログからの操作）



2. 6. 3 特定項目を固定表示に設定する

関数一覧 パネル／変数一覧 パネルにおいて，表示画面を横スクロールしても，指定した項目（列）が常に表示されるように設定することができます。

操作は，対象の項目名の固定表示アイコン（／）をマウスによりクリックすることで行います（クリックの繰り返しにより，固定表示の設定／解除が切り替わります）。

固定表示に設定された項目は最左列に移動したのち固定され，横スクロールを行ってもスクロールの対象とはなりません。

なお，固定表示アイコンの意味は次のとおりです。

表 2―4 固定表示アイコン

アイコン	説明
	固定表示を設定していないことを示します（デフォルト）。
	固定表示に設定されていることを示します。

備考 1. 固定表示に設定されていない項目名を，固定表示に設定されている項目名の間，またはその最右列にドラッグ・アンド・ドロップすることでも，固定表示の設定をすることができます。

2. 固定表示に設定されている項目名を，固定表示に設定されていない項目名の間にドラッグ・アンド・ドロップすることでも，固定表示の設定を解除することができます。

2. 6. 4 解析情報をソート表示する

関数一覧 パネル／変数一覧 パネル上の情報値は，各項目ごとに，その内容による昇順／降順で表示することができます。

操作は，対象の項目名をマウスによりクリックすることで行います（クリックの繰り返しにより，昇順表示／降順表示が切り替わります）。

ソート方法は，ソート対象の項目の情報値が数値（10 進数 /16 進数）の場合は数値の大小により行われ，それ以外の場合（文字列など）は，文字コード順に行われます。

なお，ソート表示を行っている項目名には次のマークが表示されます。

表 2―5 ソート表示マーク

マーク	説明
△	昇順表示を行っていることを示します。 再びマウスによりクリックすることで，降順表示に切り替わります。
▽	降順表示を行っていることを示します。 再びマウスによりクリックすることで，昇順表示に切り替わります。

備考 ［Shift］キーを押下しながらマウスによりクリックすることで，複数の項目に対してソート表示を行うことができます。

2. 6. 5 解析情報をフィルタ表示する

関数一覧 パネル／変数一覧 パネル上の情報値は，フィルタを設定して表示することができます。
設定可能なフィルタの種類は次のとおりです。

（1）項目ごとのカスタム設定によるフィルタ表示
（2）パネルと連携したフィルタ表示

注意 項目ごとのカスタム設定によるフィルタ表示とパネルと連携したフィルタ表示は排他使用の機能です。このため，ここで説明する 2 つのフィルタ表示機能を同時に有効化することはできません（どちらか一方のフィルタ表示を行っている際に，もう一方のフィルタ表示の設定を行った場合，それまで行っていたフィルタ表示は解除されます）。

（1）項目ごとのカスタム設定によるフィルタ表示

各項目ごとにカスタムなフィルタの設定を行い，取得した情報値を表示します。

操作は，対象の項目名のフィルタ・アイコン（／）をクリックすることで表示される，次のメニュー項目を選択することで行います。

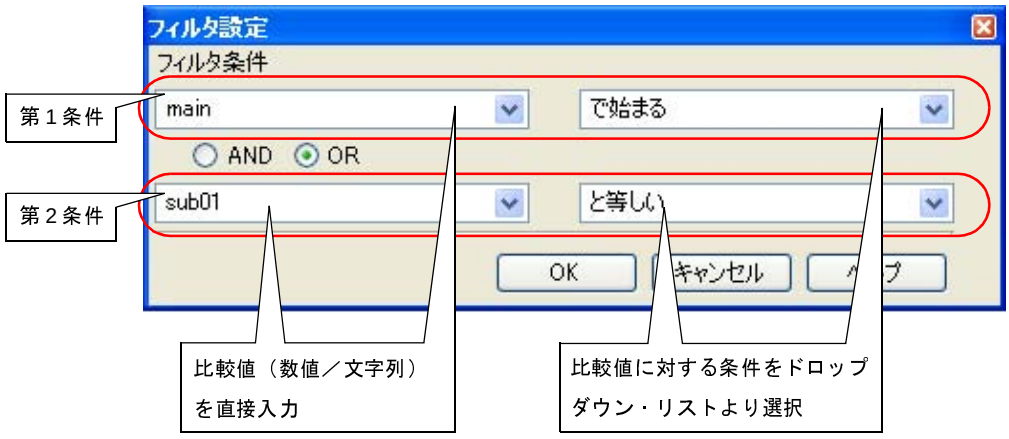
表 2―6 フィルタの設定項目

項目	説明
(すべて)	フィルタを設定しません（フィルタ表示を解除します）。 すべての情報値を表示します。
(カスタム)	詳細なフィルタ条件の設定を行うためのフィルタ設定 ダイアログがオープンし，このダイアログで指定した条件に合致する情報値のみ表示します。
(空白)	空欄（“-”表示）を表示します。
(空白以外)	空欄（“-”表示）を表示しません。
情報値一覧	取得したすべての情報値が文字列としてリスト表示されます。 リスト内より選択した文字列と合致する情報値のみ表示します。

[(カスタム)] を選択することでオープンする次のフィルタ設定 ダイアログでは，第 1 条件と第 2 条件の 2 つまでの条件を指定することができ，論理条件指定ボタン（[AND] ／ [OR]）の選択により，両条件を 1 つのフィルタ条件として設定することができます。

ただし，1 つの条件のみでフィルタ条件を設定する場合は，第 1 条件指定エリア（上段）において条件の指定を行ってください。

図 2—15　項目ごとのフィルタ設定（フィルタ設定 ダイアログ）



なお，フィルタ・アイコンの意味は次のとおりです。

表 2—7　フィルタ・アイコン

アイコン	説明
	フィルタ表示を行っていないことを示します（デフォルト）。
	フィルタ表示を行っていることを示します。

(2) パネルと連携したフィルタ表示

(a) プロジェクト・ツリー パネルとの連携
(b) エディタ パネルとの連携
(c) デバッグ・マネージャ パネルとの連携【RH850】【V850E2】

注意　ここで説明するフィルタ表示を行っている際に，「(1) 項目ごとのカスタム設定によるフィルタ表示」の設定を行うと，それまで行っていたパネルと連携したフィルタ表示は解除されます。

備考　ここで説明する次の３つのフィルタ表示は，すべて同時に使用することができます。

(a) プロジェクト・ツリー パネルとの連携

プロジェクト・ツリー パネルで選択したファイル／カテゴリ内の関数／変数の情報値のみを表示します。

操作は，関数一覧 パネル／変数一覧 パネルのツールバーの  ボタンをクリックすることでこのフィルタ機能を有効にしたのち，プロジェクト・ツリー パネル上で任意のファイル／カテゴリを選択します（この機能を解除する場合は，再び同ボタンをクリックします）。

なお，プロジェクト・ツリー パネルでの選択対象と，フィルタ表示の対象となる関数／変数の関係は次のとおりです。

表 2—8 プロジェクト・ツリー パネルと連携したフィルタ表示

選択対象	表示対象
アクティブ・プロジェクト内の単一ファイル	単一ファイル内に定義されている関数／変数
アクティブ・プロジェクト内の複数のファイル	複数ファイル内に定義されている関数／変数
アクティブ・プロジェクト内の単一のカテゴリ	単一カテゴリ下のファイル内に定義されている関数／変数
アクティブ・プロジェクト内の複数のカテゴリ	複数カテゴリ下のファイル内に定義されている関数／変数
アクティブ・プロジェクト内のファイルとカテゴリの組み合わせ	ファイル内に定義されている関数／変数，およびカテゴリ下のファイル内に定義されている関数／変数
上記以外	アクティブ・プロジェクトに含まれるすべてのファイル内に定義されている関数／変数

注意 アセンブラ・ソース・ファイルは，解析対象外です。

備考 ヘッダ・ファイルを選択した場合，該当ヘッダ・ファイル内で定義されている関数／変数が表示対象となります。

(b) エディタ パネルとの連携

エディタ パネル上のキャレット位置の単語で始まる関数／変数の情報値のみを表示します。

操作は，関数一覧 パネル／変数一覧 パネルのツールバーの  ボタンをクリックすることでこのフィルタ機能を有効にしたのち，エディタ パネル上で任意の関数名／変数名へキャレットを移動します（この機能を解除する場合は，再び同ボタンをクリックします）。

ただし，キャレット位置に単語が存在しない場合（空白やタブ記号の場合など），エディタ パネルと連携したフィルタ表示は行いません。

(c) デバッグ・マネージャ パネルとの連携【RH850】【V850E2】

現在デバッグ・マネージャ パネルで選択している PE*n* と，共通領域（Common）に割り当てられている関数／変数のみを表示します。

操作は，関数一覧 パネル／変数一覧 パネルのツールバーの  ボタンをクリックすることでこのフィルタ機能を有効します（この機能を解除する場合は，再び同ボタンをクリックします）。

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対応版でない場合，この機能は無効となります。

2.7 定義箇所へジャンプする

関数一覧 パネル／変数一覧 パネル／コール・グラフ パネル／クラス／メンバ パネル上の関数／変数／クラス【CC-RX】などが定義されている箇所へジャンプすることができます。

注意 1. 【CC-RH】【CC-RX】

列挙型のメンバを対象とした場合，列挙型の定義箇所へジャンプします。

ただし，無名の列挙型の場合は，メンバのノードからメンバの定義箇所へジャンプすることはできません。

2. 【CC-RX】

C++ ソース・ファイルを対象とする場合，テンプレート関数／テンプレート・クラス中に定義されているメンバ関数の定義位置情報を取得することができないため，この機能を使用することはできません。

ただし，static 宣言付きで，かつクラス外で定義されているテンプレート関数については，この限りではありません。

3. 【CX】

構造体／共用体／列挙体のノードから型の定義箇所へジャンプすることはできません。

また，構造体／共用体の場合は，メンバのノードからメンバの定義箇所へジャンプすることはできません。

4. 【CA850】

ソース・テキストにおいて，“#include 文”以降に次の例のような“#pragma 指令”の記述がある場合，ジャンプ位置が不正となります。この場合は，“#pragma 指令”の記述を“#include 文”以前に移動してください。

例 1. #pragma task TASK_A

2. #pragma interrupt INTP0 functionA

(1) エディタ パネルへのジャンプ

次の操作により，対象が定義されているソース・ファイルをエディタ パネル上にオープンし，定義記述のある行にキャレットが移動します。

パネル	対象	操作
関数一覧 パネル	関数行	ダブルクリック
変数一覧 パネル	変数行	
コール・グラフ パネル	- 関数ボックス - 変数ボックス	ダブルクリック[注]

パネル	対象	操作
クラス／メンバ パネル	クラス／カテゴリ用ツリー - *名前空間名ノード*【CC-RX】 - *クラス名ノード*【CC-RX】 - *インタフェース名ノード*【CC-RX】 - *構造体名ノード* - *共用体名ノード* - *列挙体名ノード*	コンテキスト・メニューの［ソースへジャンプ］を選択
	関数／変数用リスト - *関数名ノード* - *変数名ノード* - *マクロ名ノード* - *別名定義名ノード* - *列挙体メンバ名ノード*	ダブルクリック

注 パネル・コントロール・エリアの  ボタンにより，マウスのドラッグによるスクロールを許可している場合はこの操作を使用することはできません。この場合は，対象を選択したのちコンテキスト・メニューの［ソースへジャンプ］を選択するか，またはスクロールの許可をいったん解除したのち上記操作を行ってください。

(2) 逆アセンブル パネル／メモリ パネルへのジャンプ

次の操作により，対象の開始アドレスに対応する逆アセンブル・データ／メモリ・リストを逆アセンブル パネル（逆アセンブル 1）／メモリ パネル（メモリ 1）上にオープンし，該当箇所にキャレットが移動します（ただし，デバッグ・ツールと接続中の場合のみ）。

パネル	対象	操作
関数一覧 パネル	関数行	- 逆アセンブル パネルへのジャンプの場合 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 - メモリ パネルへのジャンプの場合 コンテキスト・メニューの［メモリへジャンプ］を選択
変数一覧 パネル	変数行**注1**	
コール・グラフ パネル	- 関数ボックス - 変数ボックス	
クラス／メンバ パネル	関数／変数用リスト**注2** - *関数名ノード* - *変数名ノード* - *マクロ名ノード* - *別名定義名ノード* - *列挙体メンバ名ノード*	

注 1. メモリ パネルへのジャンプのみサポートしています。

2. クラス／カテゴリ用ツリー上のノードはジャンプの対象となりません。
また，インスタンス変数を示すノードはジャンプの対象となりません。

2.8 宣言箇所へジャンプする

クラス／メンバ パネル上の関数／変数／クラス【CC-RX】などが宣言されているソース・テキスト箇所へジャンプすることができます。

注意 【CC-RX】

C++ ソース・ファイルを対象とする場合，テンプレート関数／テンプレート・クラス中に定義されているメンバ関数内で参照されている関数／変数の参照情報を取得することができないため，この機能を使用することはできません。

ただし，static 宣言付きで，かつクラス外で定義されているテンプレート関数については，この限りではありません。

次の操作により，対象が宣言されているソース・ファイルをエディタ パネル上にオープンし，関数のプロトタイプ宣言行（C ソース・ファイルを対象とする場合），またはクラス宣言内の関数の宣言行（C++ ソース・ファイルを対象とする場合）にキャレットが移動します。

対象	操作
クラス／カテゴリ用ツリー - *名前空間名ノード*【CC-RX】 - *クラス名ノード*【CC-RX】 - *インタフェース名ノード*【CC-RX】 - *構造体名ノード* - *共用体名ノード* - *列挙体名ノード*	コンテキスト・メニューの［ソースの宣言へジャンプ］を選択
関数／変数用リスト - *関数名ノード* - *変数名ノード* - *マクロ名ノード* - *別名定義名ノード* - *列挙体メンバ名ノード*	

備考 ジャンプ先の行が，［ソースへジャンプ］の選択によるジャンプ先（「2.7 定義箇所へジャンプする」参照）と同じ箇所となる場合があります。

2.9 ブレーク・イベントを設定する

関数一覧 パネル／変数一覧 パネル上の関数／変数に対して，デバッグ・ツールにブレーク・イベントを設定することができます。

- 関数にブレークポイントを設定する
- 変数にブレーク・イベントを設定する

2.9.1 関数にブレークポイントを設定する

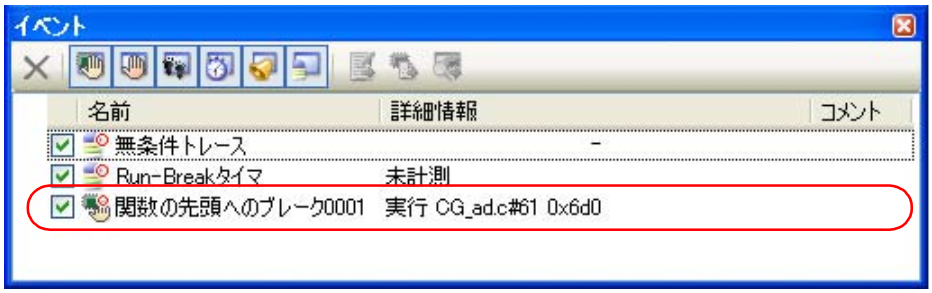
関数一覧 パネル上の関数の先頭行（関数の一番最初の実行可能行）にブレークポイントを設定します。

操作は，対象となる関数の表示行を選択したのち（選択行左端にカレント行マーク（▶）が表示されていることを確認），コンテキスト・メニューの［関数の先頭にブレークを設定］を選択することで行います。

注意 この操作は，デバッグ・ツールと接続時のみ行うことができます。

なお，この操作により設定されたブレークポイントは，イベント名を“関数の先頭へのブレーク”としてイベント パネル上で管理されます。

図 2—16 イベント パネルのブレークポイントの設定例



備考 1. 該当箇所にすでにブレークポイントが設定されている場合は，次の動作となります。

- 有効状態のブレークポイントが設定されている場合： 何もしません
- 無効状態のブレークポイントが設定されている場合： 有効状態にします

2. 【CC-RX】

対象関数が，テンプレート関数で，かつテンプレート関数に対応しているアドレスが複数存在している場合，テンプレート関数に対応しているすべてのアドレスにブレークポイントを設定します。

2.9.2 変数にブレーク・イベントを設定する

変数一覧 パネル上の変数にアクセス系のブレーク・イベントを設定します。

操作は，対象となる変数の表示行を選択したのち（選択行左端にカレント行マーク（▶）が表示されていることを確認），コンテキスト・メニューより次のいずれかを選択し，［Enter］キーを押下します。

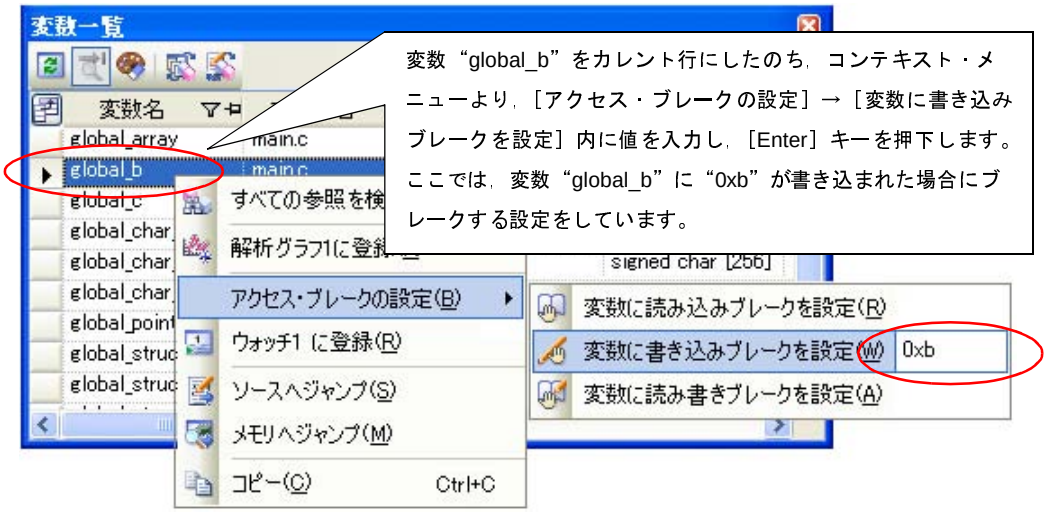
- ［アクセス・ブレークの設定］→［変数に読み込みブレークを設定］： リード・アクセスによるブレーク
- ［アクセス・ブレークの設定］→［変数に書き込みブレークを設定］： ライト・アクセスによるブレーク

- ［アクセス・ブレークの設定］→［変数に読み書きブレークを設定］：　リード／ライト・アクセスによるブレーク

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することができます。

この場合，指定した値でアクセスした場合のみブレークします。テキスト・ボックス内に値を指定せず［Enter］キーを押下した場合は，値にかかわらず対象変数に指定アクセスがあった場合にブレークします。

図 2—17　変数一覧 パネルにおけるブレーク・イベントの設定例



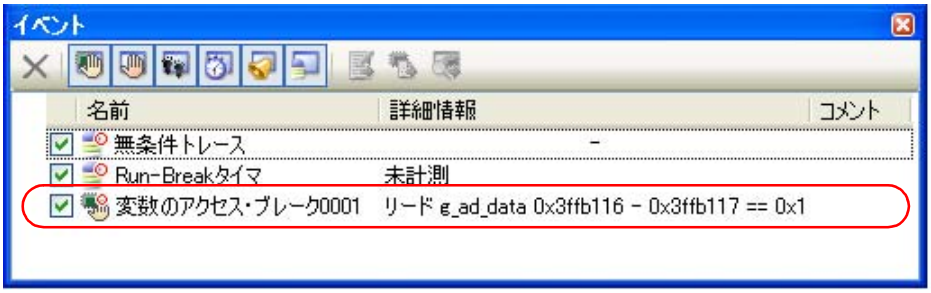
注意 1.　この操作は，デバッグ・ツールと接続時のみ行うことができます。

2.　【RX】

組み合わせブレークの場合，組み合わせ条件が“OR”の場合のみ有効となります。

なお，この操作により設定されたブレーク・イベントは，イベント名を“変数のアクセス・ブレーク”としてイベント パネル上で管理されます。

図 2—18　イベント パネルのブレーク・イベント（アクセス系）の設定例



備考　該当箇所にすでにブレーク・イベントが設定されている場合は，次の動作となります。

- 有効状態のブレーク・イベントが設定されている場合：　何もしません
- 無効状態のブレーク・イベントが設定されている場合：　有効状態にします

2. 10 ウォッチ式に登録する

変数一覧 パネル上の変数をウォッチ パネル（ウォッチ 1）のウォッチ式として登録することができます。

操作は，対象となる変数の行（複数行選択可）をウォッチ パネル（ウォッチ 1）へ直接ドラッグ・アンド・ドロップすることで行います。

注意 1. この操作は，デバッグ・ツールと接続時のみ行うことができます。

2. ［合計値］の行に対してこの操作は無効です。

備考 対象となる変数の種別により，ウォッチ パネル上に登録されるウォッチ式の名称は次のように異なります。

- グローバル変数 : “*変数名*”
- ファイル内スタティック変数 : “*ファイル名*#*変数名*”
- 関数内スタティック変数 : “*ファイル名*#*関数名*#*変数名*”
- クラス変数 : “*クラス名*::*変数名*”

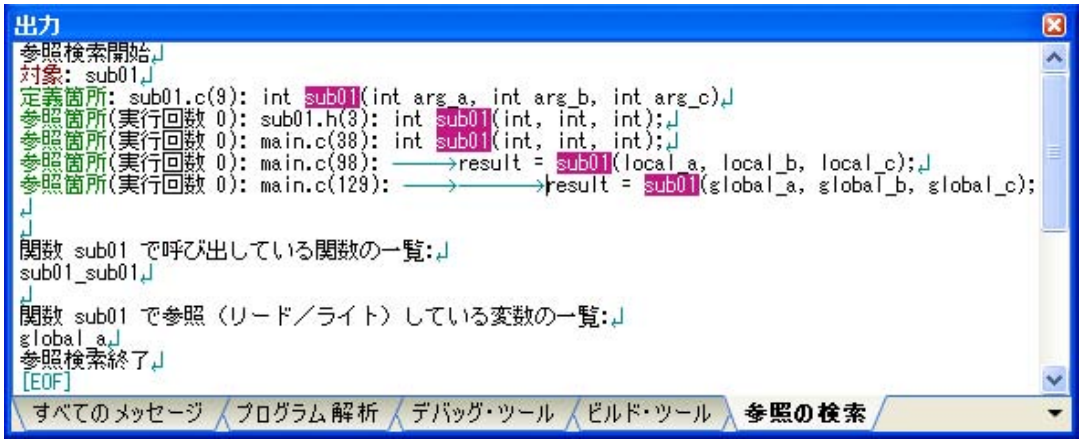
2. 11 参照箇所を一覧表示する

関数一覧 パネル／変数一覧 パネル上の関数／変数を参照している箇所を検索し，その結果として参照箇所一覧を表示することができます。

操作は，対象となる関数／変数の表示行を選択したのち（複数行選択可），コンテキスト・メニューの［すべての参照を検索］を選択することで行います。

検索結果は，次の出力 パネルの［参照の検索］タブ上に出力されます。

図 2—19 関数／変数の参照箇所一覧の出力例（出力 パネル）



参照箇所一覧では，検索結果として次の情報を出力します。

なお，出力フォーマットについての詳細は，出力 パネルの［参照の検索］タブを参照してください。

- 関数一覧 パネルより操作を行った場合
  - 対象関数の定義箇所
  - 対象関数を参照している箇所の一覧
  - 対象関数内で呼び出している関数の一覧
  - 対象関数内で参照（リード／ライト）している変数の一覧

- 変数一覧 パネルより操作を行った場合
  - 対象変数の定義箇所
  - 対象変数を参照している箇所の一覧

注意 C/C++ ソース・ファイル中の“#if”／“#ifdef”などで，コンパイル時にプリプロセッサにより除外されるコードにおいて参照されている箇所は参照箇所として出力されません。

備考 1. 出力結果上の関数名／変数名をダブルクリックすることにより，エディタ パネルをオープンし，該当関数／変数が定義されているソース・テキスト箇所へジャンプすることができます。

2. 出力 パネルの［参照の検索］タブ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 出力 - 参照の検索 を保存 ...］を選択することにより，参照箇所一覧をテキスト・ファイル（*.txt）に保存することができます。

2. 12 情報ファイルをインポート／エクスポートする

アクティブ・プロジェクト以外で定義され，かつアクティブ・プロジェクトから一度も参照されていない関数／変数は，関数一覧 パネル／変数一覧 パネルでその情報が表示されません。

この場合，表示したい関数／変数の情報を記録した情報ファイルをインポートすることにより，これらの情報を強制的に表示することができます。

情報ファイルには次の種類があります。

表 2—9 情報ファイルの種類

情報ファイル名	内容
関数一覧ファイル（*.mtfl)	関数一覧 パネルにおける次の項目の情報を記録します。 ［関数名］／［クラス名］【CC-RX】／［名前空間］【CC-RX】／［ファイル名］／［ファイル・パス］／［PM 情報］【RH850 注】／［PE 情報］【V850E2】注／［アクセス指定子］【CC-RX】／［属性］／［戻り値の型］／［引数の数］／［引数］／［コード・サイズ［バイト］］／［スタック・サイズ［バイト］］
変数一覧ファイル（*.mtvl)	変数一覧 パネルにおける次の項目の情報を記録します。 ［変数名］／［クラス名］【CC-RX】／［名前空間］【CC-RX】／［ファイル名］／［関数名］／［ファイル・パス］／［PM 情報］【RH850】注／［PE 情報］【V850E2】注／［アクセス指定子］【CC-RX】／［属性］／［型］／［メンバ］／［サイズ［バイト］］

注 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のみ対象となる項目です。

次の手順により操作を行ってください。

(1) 情報ファイルを生成（エクスポート）する

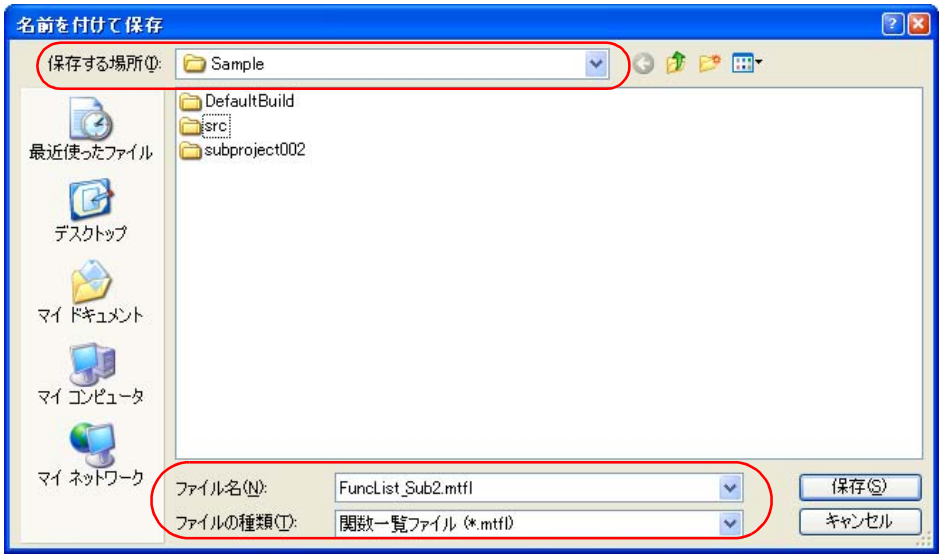
(a) 手動でエクスポートを行う場合

インポートする必要のある関数／変数についてのみの情報ファイルを生成します。

操作は，まず，対象となる関数／変数が定義されているプロジェクトをアクティブ・プロジェクトに変更したのち，関数一覧 パネル／変数一覧 パネルに必要な情報を表示させます。

次に，同パネルにおいて，必要となる対象関数／変数のすべての表示行を選択したのち（［Shift］キー／［Ctrl］キーを押下しながら表示行を選択することにより複数行の選択可），［ファイル］メニュー→［名前を付けて 関数一覧データ を保存］／［名前を付けて 変数一覧データ を保存］を選択し，名前を付けて保存 ダイアログをオープンします。

図 2—20　情報ファイルの生成（関数一覧ファイルを生成する場合の例）



上記ダイアログの［保存する場所］エリアにおいて，生成する情報ファイルを格納する任意のフォルダを選択したのち，［ファイルの種類］エリアのドロップダウン・リストにより，“関数一覧ファイル（*.mtfl）”または“変数一覧ファイル（*.mtvl）”を選択します。

次に，［ファイル名］エリアにおいて，生成する情報ファイル名を入力します（拡張子は，関数一覧ファイルの場合は“mtfl”，変数一覧ファイルの場合は“mtvl”に限ります）。

［保存］ボタンをクリックすることにより，指定したフォルダに指定したファイル名で情報ファイルが生成されます。

備考　指定したアクティブ・プロジェクト以外で定義されている関数／変数についての情報ファイルも必要な場合は，同様の操作により，別途エクスポートを行ってください。

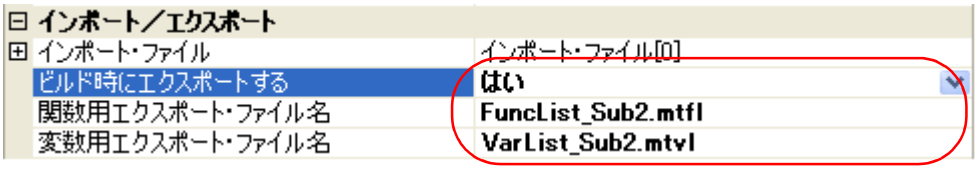
（b）自動でエクスポートを行う場合

ビルド・ツールによるビルド／リビルドを実行するごとに，関数一覧 パネル／変数一覧 パネルで表示されるすべての関数／変数についての情報ファイルを自動的に生成します。

操作は，プロパティ パネルの［設定］タブ上の［インポート／エクスポート］カテゴリ内［ビルド時にエクスポートする］プロパティを［はい］に指定したのち，同カテゴリ内の［関数用エクスポート・ファイル名］／［変数用エクスポート・ファイル名］プロパティにおいて，生成する情報ファイル名を指定します（拡張子は，関数一覧ファイルの場合は“mtfl”，変数一覧ファイルの場合は“mtvl”に限ります）。

相対パスによる指定の場合は，プロジェクト・フォルダを基点として指定してください。

図 2—21　［インポート／エクスポート］カテゴリ



以上の設定により，情報ファイルの自動エクスポートの設定は完了です。

ビルド／リビルドを実行するごとに，指定したフォルダに指定したファイル名で情報ファイルが自動的に生成されます。

注意 エクスポートの対象となるのは，現在のアクティブ・プロジェクトのみです。

備考 ［関数用エクスポート・ファイル名］／［変数用エクスポート・ファイル名］プロパティでは，次のプレースフォルダに対応しています。

- %ProjectName% ： プロジェクト名に置換します。
- %ActiveProjectName% ： アクティブ・プロジェクト名に置換します。

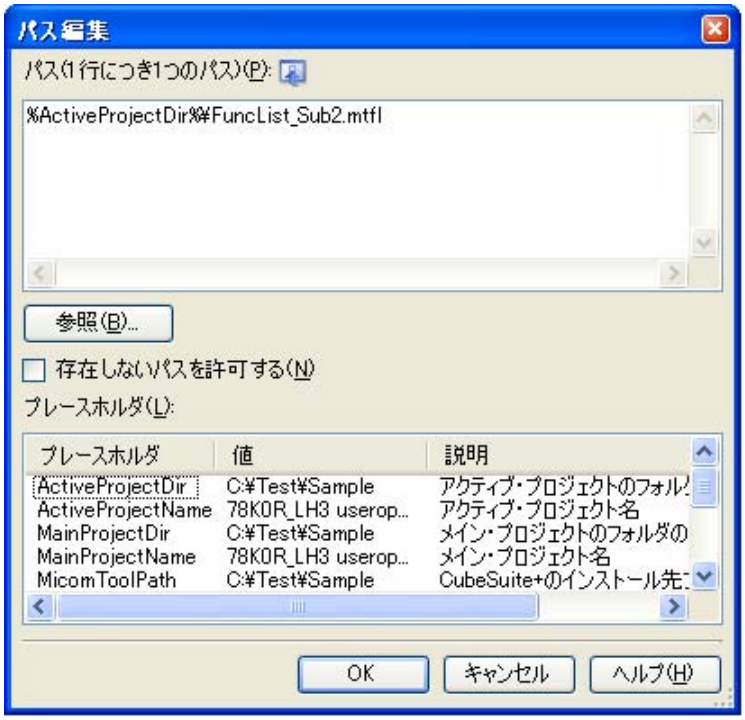
(2) 情報ファイルをインポートする

生成（エクスポート）した情報ファイルをインポートする設定を行います。

操作は，プロパティ パネルの［設定］タブ上の［インポート／エクスポート］カテゴリ内［インポート・ファイル］プロパティで行います。［インポート・ファイル］プロパティを選択すると表示される［...］ボタンをクリックすると，次のパス編集 ダイアログがオープンします。

図 2—22 ［インポート／エクスポート］カテゴリ内［インポート・ファイル］プロパティ

図 2—23 インポート・ファイル名の設定（パス編集 ダイアログ）

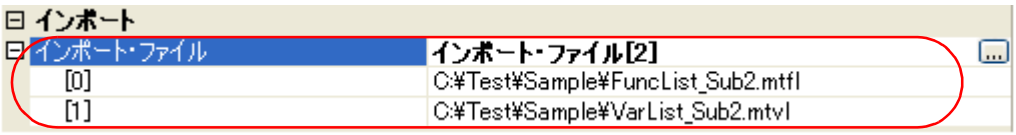


パス編集 ダイアログの［パス (1 行につき 1 つのパス)］エリアにおいて，（1）において生成した情報ファイル名のすべてを 1 行に 1 つずつ，パスを含めて指定します（1 行に 259 文字 /64 行まで指定可）。相対パスによる指定の場合は，プロジェクト・フォルダを基点として指定してください。

また，情報ファイルの指定は，［参照 ...］ボタンから情報ファイルを指定することもできます。

インポートするすべてのファイル名の入力が完了したのち，［OK］ボタンをクリックすると，指定したパスが［インポート・ファイル］プロパティのサブプロパティとして表示されます。

図 2—24 ［インポート・ファイル］プロパティ（インポート・ファイル追加後）



以上の設定により，情報ファイルのインポートの設定は完了です。

備考 1. パス編集 ダイアログでは，次のプレースフォルダに対応しています。

- %ProjectName% ： プロジェクト名に置換します。
- %MicomToolPath% ： CubeSuite+ のインストール・フォルダの絶対パスに置換します。

2. CubeSuite でエクスポートした情報ファイル（CubeSuite 関数一覧ファイル（*.csfl）/CubeSuite 変数一覧ファイル（*.csvl））をインポートすることができます。

3. インポート・ファイルとアクティブ・プロジェクト内に同名の関数／変数が存在する場合は，次の決定規則に従います。

- **C ソース・ファイルの場合**
  - ファイル名が異なり，かつインポート・ファイル側の属性に“static”（小文字のみ）が含まれている場合は，スタティック関数／スタティック変数とみなし，異なる関数情報／変数情報として取り込みます。
  - ファイル名が同名で関数名が異なり，かつインポート・ファイル側の属性に“static”（小文字のみ）が含まれている場合は，関数内スタティック変数とみなし，異なる変数情報として取り込みます。
  - 上記以外の関数／変数については，グローバル関数／グローバル変数，または同一ファイル内の同名スタティック関数／スタティック変数とみなし，同一の関数情報／変数情報としてマージして取り込みます。
- **C++ ソース・ファイルの場合【CC-RX】**
  - ［クラス名］，［名前空間］，［ファイル名］，および［引数］を比較して，１つでも一致していない場合，異なる関数情報として取り込みます。
  - ［クラス名］，［名前空間］，［関数名］，［ファイル名］，および［引数］を比較して，１つでも一致していない場合，異なる変数情報として取り込みます。
  - 上記以外の関数／変数については，同一の関数情報／変数情報としてマージして取り込みます。

なお，同一の関数情報／変数情報としてマージする場合の決定規則は次のとおりです。

項目		規則
関数情報	変数情報	
[クラス名]【CC-RX】 [名前空間]【CC-RX】 [ファイル名] [ファイル・パス] [PM 情報]【RH850】注 [PE 情報]【V850E2】注 [スタック・サイズ[バイト]] [引数の数] [引数] [戻り値の型] [アクセス指定子]【CC-RX】 [属性]	[クラス名]【CC-RX】 [名前空間]【CC-RX】 [ファイル名] [関数名] [ファイル・パス] [PM 情報]【RH850】注 [PE 情報]【V850E2】注 [アクセス指定子]【CC-RX】 [属性] [型] [メンバ]	次の優先順位に従います。 “アクティブ・プロジェクトの値”＞“インポート・ファイルの値” ただし，複数のインポート・ファイルにマージ対象となる関数情報／変数情報が存在する場合は，最後にインポートされたファイルの関数情報／変数情報を取り込みます。 なお，アクティブ・プロジェクト内／インポート・ファイル内のどちらにも値が存在しない場合は空欄（“-”）となります。
[コード・サイズ[バイト]]	[サイズ]	次の優先順位に従います。 “インポート・ファイルの値”＞“アクティブ・プロジェクトの値” ただし，複数のインポート・ファイルにマージ対象となる関数情報／変数情報が存在する場合は，最後にインポートされたファイルの関数情報／変数情報を取り込みます。 なお，アクティブ・プロジェクト内／インポート・ファイル内のどちらにも値が存在しない場合は空欄（“-”）となります。

注 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のみ表示される項目です。

(3) 情報ファイルのインポートを中止する

情報ファイルのインポートを中止する場合は，パス編集 ダイアログにおいて，指定したインポート・ファイルを消去してください。

2. 13 解析情報をグラフ化して表示する

取得した関数情報／変数情報（Smart Analog 用のデータ情報を含む[注]）をグラフ化して表示することができます。

グラフの表示は，メイン・ウインドウのツールバーの  ボタンをクリックすることでオープンする，解析グラフ パネルで行います。

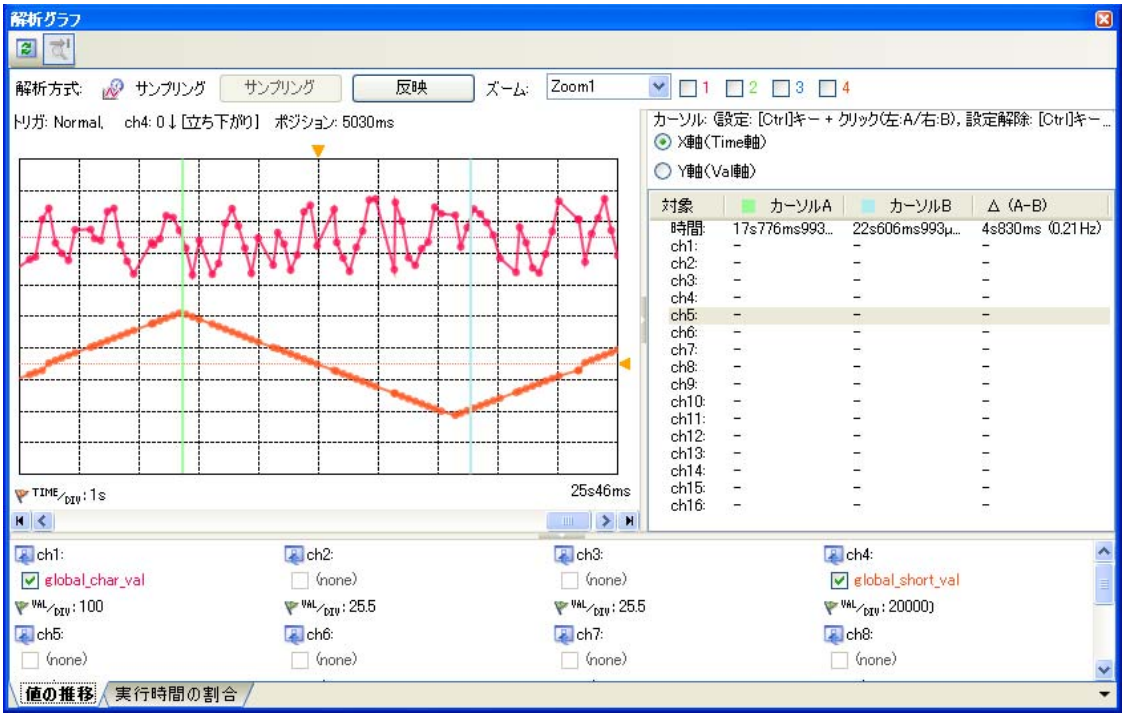
解析グラフ パネルでは，次の内容をグラフ化することができます。

- 値の推移をグラフ化する
- 関数の実行時間率をグラフ化する

注 【E1/E20【RL78】】

選択しているマイクロコントローラが Smart Analog IC 搭載品の場合のみサポートしている機能です。

図 2—25 解析情報のグラフ化（解析グラフ パネル）



2. 13. 1 値の推移をグラフ化する

登録した変数／レジスタ／アドレス等の値と時間の関係を折れ線グラフで表示します。

また，選択しているマイクロコントローラが Smart Analog IC 搭載品の場合では，デバッグ・ツールをデータ収集モードに設定することにより，Smart Analog 用に収集したデータをグラフで表示することができます【E1/E20【RL78】】。

グラフ表示は，解析グラフ パネルの［値の推移］タブで行います。

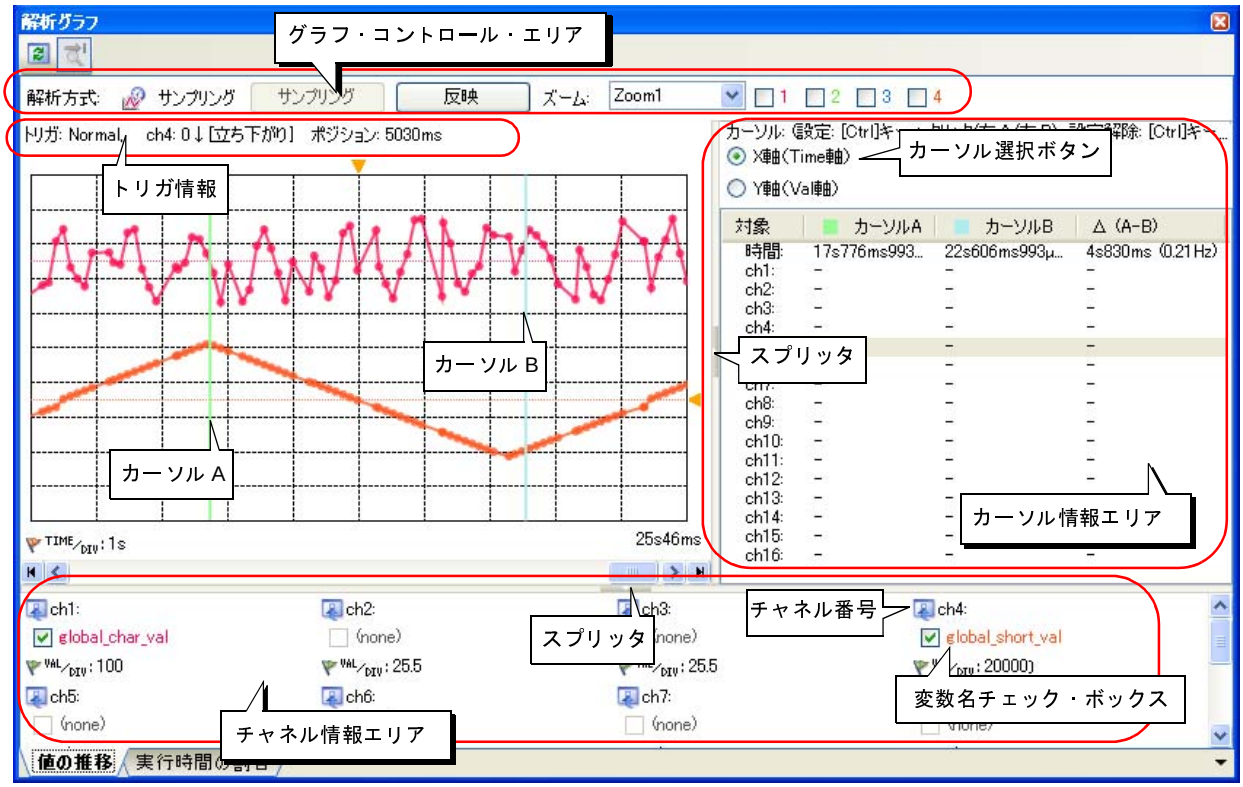
なお，表示される各エリアについての詳細は，［値の推移］タブを参照してください。

注意 1. デバッグ・ツールが取得したトレース・データ，またはリアルタイム RAM モニタ結果を基にグラフ表示を行います。グラフ表示を行うためには，「(a) トレース機能」／「(b) RRM 機能／疑似 RRM（RAM モニタ）機能」の注意も参照してください。

2. **【E1/E20【RL78】】**

Smart Analog 用に収集したデータをグラフ表示するためには，実行プログラムにデータ収集用のモニタ・プログラムをリンクする必要があります。

図 2—26　変数値の推移のグラフ化



備考 1. カーソル情報エリア，およびチャネル情報エリアは，スプリッタを移動することにより表示領域を変更することができます。また，これらのエリアは，各スプリッタ上の中央のマークをクリックすることにより，表示／非表示を切り替えることができます。

2. データ収集モード

選択しているマイクロコントローラが Smart Analog IC 搭載品の場合のみサポートするデバッグ・ツールの機能です【E1/E20【RL78】】。

デバッグ・ツールをデータ収集モードに設定することにより，プログラム実行時に Smart Analog 用のデータを収集します。

このモードは，デバッグ・ツールのプロパティ パネルにおける次の指定により設定されます。

［デバッグ・ツール設定］タブ→［Smart Analog］カテゴリ→［実行中にデータ収集を行う］→［はい］

グラフ表示の操作手順は，次のとおりです。

(1) グラフ化対象を登録する

グラフ化する対象を登録します。

（a）変数／レジスタ／アドレスの値をグラフ化する場合
（b）Smart Analog 用に収集したデータをグラフ化する場合【E1/E20【RL78】】

注意 次の状態の場合，グラフ化対象を登録することはできません（ここでの操作は無効となります）。
- プログラム実行中の場合
- プロパティ パネルの［値の推移］タブ上の［全般］カテゴリ内［解析方式］プロパティにおいて，［ファイルから読み込み］を指定している場合（解析グラフ・データ・ファイル（*.mtac）からグラフを復帰した場合）

(a) 変数／レジスタ／アドレスの値をグラフ化する場合

登録可能なグラフ化対象の種別は次のとおりです。
- グローバル変数
- ファイル内スタティック変数
- 関数内スタティック変数
- クラス変数（C++ ソース・ファイルを対象とする場合）
- CPU レジスタ
- IOR【RH850】【RX】【V850】
- SFR【RL78】【78K0R】【78K0】
- アドレス

グラフ化対象の登録方法には，次の 3 通りがあります。
なお，登録が完了すると，チャネル情報エリアの各チャネル番号に対応した変数名チェック・ボックスに，登録したグラフ化対象名が表示されます。

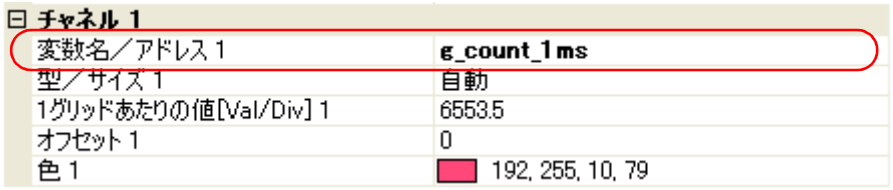
- 対象を個別に登録する場合（プロパティ パネル上での登録）
- 対象を個別に登録する場合（他のパネルからの登録）
- ウォッチ パネルから反映する場合（自動登録）

注意 グラフ化対象は，1 チャネルにつき 1 個，最大 16 チャネル（16 個）まで登録することができます。ただし，使用するデバッグ・ツール，およびグラフ・データの取得方法により，グラフ化が可能な対象の数，およびサイズに限りがある場合があります（「(2) グラフ・データの取得方法を選択する」参照）。

- 対象を個別に登録する場合（プロパティ パネル上での登録）

次のプロパティ パネルの［値の推移］タブ上の［チャネル *1* ～ *16*］カテゴリ内［変数名／アドレス *1* ～ *16*］プロパティにおいて，登録する対象名を直接キーボードより入力します。

図 2—27　［チャネル *1* ～ *16*］カテゴリ



なお，［チャネル *1* ～ *16*］カテゴリでは，各対象ごとに，次の詳細条件を指定することができます（「(4) グラフを表示する」参照）。

- ［型／サイズ *1* ～ *16*］　：型／サイズ
- ［1 グリッドあたりの値 [Val/Div] *1* ～ *16*］　：グラフにおける単位グリッドあたりの数値
- ［オフセット *1* ～ *16*］　：グラフにおけるオフセット値
- ［色 *1* ～ *16*］　：グラフの描画色

備考 1.　登録したグラフ化対象を削除する場合は，上記［変数名／アドレス *1* ～ *16*］プロパティの値を空欄にします。

2.　登録したグラフ化対象名は，［値の推移］タブの変数名チェック・ボックスに表示されます。

- 対象を個別に登録する場合（他のパネルからの登録）

次のパネル上の対象を，このタブ上のチャネル番号，または変数名チェック・ボックスへ直接ドラッグ・アンド・ドロップします。

- 変数一覧 パネル[注]
- エディタ パネル
- CPU レジスタ パネル
- IOR パネル【RH850】【RX】【V850】
- SFR パネル【RL78】【78K0R】【78K0】
- ウォッチ パネル

注　対象となる変数の種別により，グラフ化対象として表示される名称は次のように異なります。

- グローバル変数　：“*変数名*”
- ファイル内スタティック変数　：“*ファイル名*＃*変数名*”
- 関数内スタティック変数　：“*ファイル名*＃*関数名*＃*変数名*”
- クラス変数　：“*クラス名*::*変数名*”

備考 1.　上記のほか，変数一覧 パネル／エディタ パネルでは，コンテキスト・メニューの［解析グラフに登録］を選択することによっても，選択している変数をグラフ化対象として登録することができます。

2.　登録したグラフ化対象を削除する場合は，プロパティ パネルの［値の推移］タブ上の［チャネル *1* ～ *16*］カテゴリ内［変数名／アドレス *1* ～ *16*］プロパティの値を空欄にします。

- ウォッチ パネルから反映する場合（自動登録）

グラフ・コントロール・エリアの［反映］ボタンをクリックすることにより，現在ウォッチ パネル（ウォッチ 1）に登録されている上から 16 個のウォッチ式を，グラフ化対象として自動登録します。

なお，ウォッチ パネル（ウォッチ 1）に登録されているウォッチ式が 16 個未満の場合は，登録されている個数分のみが登録されます。

注意 ［反映］ボタンをクリックすると，それまで登録していたグラフ化対象の情報は破棄されます（表示中のグラフが消失します）。

備考 1. カテゴリ自体は登録されません。ただし，カテゴリ内の変数などは登録対象となります。

2. 子ノードを持つ変数（配列／構造体など）の場合，構造体のメンバや配列のインデクスなど，展開後の変数などについては登録対象となりません。

3. すでに登録済みのグラフ化対象と名前が一致する場合，重複して登録されます。

（b）Smart Analog 用に収集したデータをグラフ化する場合【E1/E20【RL78】】

この機能は，選択しているマイクロコントローラが Smart Analog IC 搭載品の場合のみ有効となります。

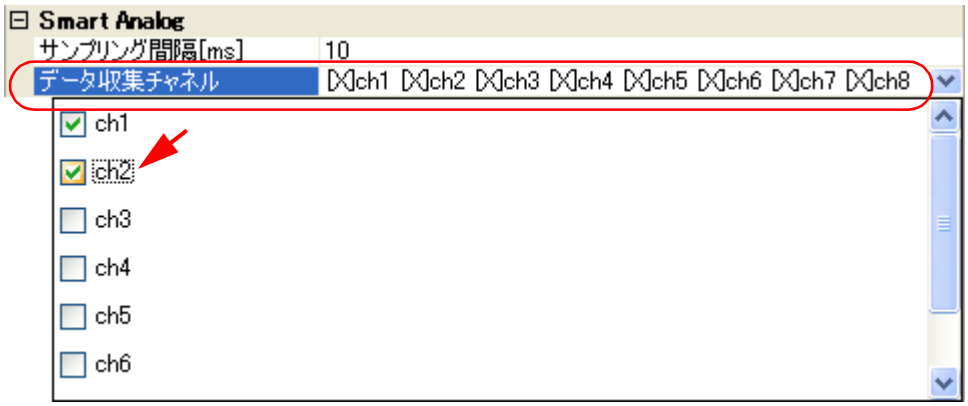
まず，使用しているデバッグ・ツールをデータ収集モードに設定します。

続いて，次のプロパティ パネルの［値の推移］タブ上の［Smart Analog］カテゴリ内［データ収集チャネル］プロパティにおいて，デバイスからデータを受信するチャネル番号を指定します。

ドロップダウン・リスト内において，使用するチャネル番号のチェック・ボックスをすべてチェックしてください（最大 8 チャネルまで複数選択可）。

注意 ［Smart Analog］カテゴリは，デバッグ・ツールがデータ収集モードに設定されている場合のみ表示されます。

図 2—28 ［Smart Analog］カテゴリ

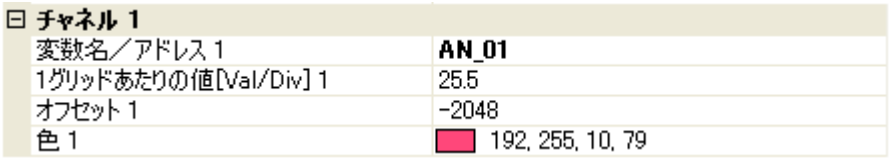


なお，プロパティ パネルの［値の推移］タブ上の［チャネル *1* ～ *16*］カテゴリでは，各チャネルごとに，次の詳細条件を指定することができます（「(4) グラフを表示する」参照）。

- ［変数名／アドレス *1* ～ *16*］ ： 変数名チェック・ボックスに表示する文字列
- ［1 グリッドあたりの値 [Val/Div] *1* ～ *16*］： グラフにおける単位グリッドあたりの数値

- ［オフセット *1* ～ *16*］ ： グラフにおけるオフセット値
- ［色 *1* ～ *16*］ ： グラフの描画色

図 2―29 ［チャネル *1* ～ *16*］カテゴリ【E1/E20【RL78】】



(2) グラフ・データの取得方法を選択する

グラフ化するためのデータの取得方法には次の 3 通りがあり，これらの指定は，プロパティ パネルの［値の推移］タブ上の［全般］カテゴリ内［解析方式］プロパティにより行います。

注意 1. プログラム実行中にグラフ・データの取得方法を変更することはできません。

2. グラフ・データの取得方法を変更した場合，それまで保持していたグラフ・データの情報は破棄されます（表示中のグラフが消失します）。

3. トレース・データ解析方式を指定する場合，「3. 4 トレース・データ解析方式について」も参照してください。

図 2―30 ［解析方式］プロパティ

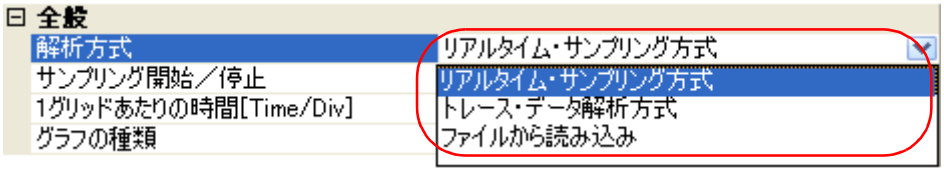


表 2―10 グラフ・データの取得方法の選択

グラフ・データの取得方法	説明
リアルタイム・サンプリング方式	デバッグ・ツールの RRM 機能／疑似 RRM（RAM モニタ）機能により取得したデータを基にグラフ表示を行います（デフォルト）。 【RL78】 デバッグ・ツールがデータ収集モードに設定されている場合，Smart Analog 用のサンプリング方式注によりデータ収集を行います。したがって，Smart Analog 用に収集したデータをグラフ化する場合【E1/E20【RL78】】は，この項目を選択してください。
トレース・データ解析方式	デバッグ・ツールのトレース機能により取得したトレース・データを基にグラフ表示を行います。 なお，この項目は，次のいずれかの場合は非表示となります。 - デバッグ・ツールがトレース機能をサポートしていない場合 - デバッグ・ツールがトレース・タイム・タグ機能をサポートしていない場合 - デバッグ・ツールがデータ収集モードに設定されている場合

グラフ・データの取得方法	説明
ファイルから読み込み	保存した解析グラフ・データ・ファイル（*.mtac）を読み込みグラフ表示の復帰を行います（「(6) グラフを復帰するためのグラフ・データを保存する」参照）。

注 【E1/E20【RL78】】

プロパティ パネルにおいて，次のプロパティ設定を行う必要があります。

- ［値の推移］タブ→［Smart Analog］カテゴリ→［サンプリング間隔 [ms]］プロパティ

グラフ・データの取得方法によるグラフ化の際の相違点は次のとおりです。

表 2—11 グラフ・データの取得方法によるグラフ表示の相違

相違点	リアルタイム・サンプリング方式	トレース・データ解析方式
グラフ表示の可否	デバッグ・ツールが次のいずれかの状態時のみ表示可 - RRM 機能／疑似 RRM（RAM モニタ）機能が有効 - データ収集モードに設定	デバッグ・ツールのトレース機能が有効な状態時のみ表示可
グラフ形式	次のいずれか**注1** - ステップ・プロット折れ線グラフ - 通常の折れ線グラフ	ステップ・プロット折れ線グラフ（固定）
グラフ化対象の登録	プログラム実行前	トレース・データによる解析のため，実行のタイミングに依存せず
グラフ化可能な対象の数	使用可能な対象領域のサイズに依存**注2**	ポイント・トレース・イベントを使用する場合，デバッグ・ツールの有効イベント数の制限に依存**注3**
グラフ化可能な対象サイズ	- 4 バイト以下	- 4 バイト以下【RH850】【RX】【V850】 - 2 バイト以下【RL78】【78K0R】【78K0】
時間表示範囲	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）	トレース・データとして記録されている時間
時間表示形式	*XXX*s*XXX*ms	*XXX*s*XXX*ms*XXX*µs*XXX*ns
トリガ機能	あり（「(3) トリガ機能を使用する」参照）	なし
プログラム実行中のグラフ更新	可	不可
値の遷移箇所	指定サンプリング間隔**注4**に依存するため，正確な時間／変異箇所の特定は不可	実際のタイミングと合致（ポップアップ表示から確認可能）
対象コア【RH850】	「(b) RRM 機能／疑似 RRM（RAM モニタ）機能」参照	「(a) トレース機能」参照

相違点	リアルタイム・サンプリング方式	トレース・データ解析方式
注意	- グラフ化対象の登録個数に依存して，サンプリング間隔が不定となる可能性あり - 値の取得を失敗する可能性あり 失敗した場合は，時間情報のみ表示し，遷移箇所と遷移箇所を結ぶ線は非表示 （「(a) グラフ」参照）	次の場合は値の推移の解析不可 - コンパイラの最適化により，変数がレジスタに割り当てられている区間 - 2 バイトの対象に 1 バイト単位で値を書き込んだ場合，または 4 バイトの対象に 1 バイト /2 バイト単位で値を書き込んだ場合 - アクセス系（リード／ライト）のトレース・データを解析することによりグラフを表示するため，デバッグ・ツールの外部トレース /OCD 内蔵トレースがアクセス系のトレース・データをサポートしていない場合

注 1. プロパティ パネルにおける次の設定に依存します。

- ［値の推移］タブ→［全般］カテゴリ→［グラフの種類］プロパティ

2. 【IECUBE【V850】】【MINICUBE2【78K0】】【E20(JTAG)【RX600 シリーズ】】【EZ Emulator【78K0】】
RRM 機能の対象領域に対して，サイズ／個数の制限があります。
“RRM 機能の対象領域”についての詳細は，使用するマイクロコントローラの「CubeSuite+ 統合開発環境 ユーザーズマニュアル デバッグ編」を参照してください。

3. 使用するマイクロコントローラ／デバッグ・ツールにより，有効イベント数の制限は異なります。
“有効イベント数の制限”についての詳細は，使用するマイクロコントローラの「CubeSuite+ 統合開発環境 ユーザーズマニュアル デバッグ編」を参照してください。

4. RRM 機能／疑似 RRM（RAM モニタ）機能を使用する場合
デバッグ・ツールのプロパティ パネル→［デバッグ・ツール設定］タブ→［実行中のメモリ・アクセス］カテゴリ→［表示更新間隔 [ms]］プロパティ
データ収集モードに設定している場合【E1/E20【RL78】】
プロパティ パネル→［値の推移］タブ→［Smart Analog］カテゴリ→［サンプリング間隔 [ms]］プロパティ

(3) トリガ機能を使用する

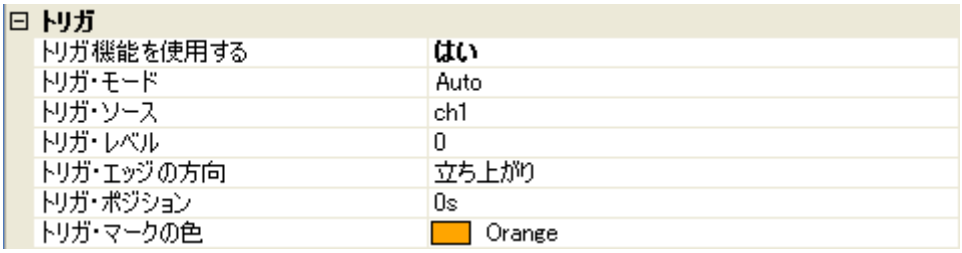
リアルタイム・サンプリング方式を指定した場合では，指定したトリガ信号によりグラフを表示するタイミングを制御することができます。このトリガ機能を使用することにより，オシロ・スコープと同じ感覚でグラフを表示することができます。

トリガ機能を使用するためには，プロパティ パネルの［値の推移］タブ上の［トリガ］カテゴリにおいて，次の設定を行います。

なお，ここで設定した内容は，このタブ上にトリガ情報として一覧表示されます。

注意 プログラム実行中に［トリガ］カテゴリ内のプロパティを変更することはできません。

図 2—31 トリガ機能の設定（[トリガ] カテゴリ）



(a) [トリガ機能を使用する]

トリガ機能を使用するか否かを選択します。

トリガ機能を使用するためには [はい] を選択してください（デフォルト：[いいえ]）。

(b) [トリガ・モード]

トリガ・モード（グラフの表示更新を行うタイミング）を選択します。

Auto	周期的にリアルタイムでグラフの表示更新を行います（デフォルト）。 トリガ信号が発生すると，トリガ信号直前までのデータをトリガ・ポジションの左側にグラフ化し，トリガ信号直後からのデータをトリガ・ポジションから右側にグラフ化します。 グラフがグラフ領域の最右端まで到達すると，左方向へスクロールを再開してグラフの表示更新を行います。
Single	サンプリング開始から最初のトリガ信号発生時のみグラフの表示更新を行います。 トリガ信号が発生すると，トリガ信号直前までのデータをトリガ・ポジションの左側にグラフ化し，トリガ信号直後からのデータをトリガ・ポジションから右側にグラフ化します。 グラフがグラフ領域の最右端まで到達すると，グラフの表示更新／サンプリングを停止[注]します。
Normal	トリガ信号発生時のみグラフの表示更新を行います。 トリガ信号が発生すると，トリガ信号直前までのデータをトリガ・ポジションの左側にグラフ化し，トリガ信号直後からのデータをトリガ・ポジションから右側にグラフ化します。 グラフがグラフ領域の最右端まで到達すると，グラフの表示更新を停止します。サンプリングは停止しないため，再びトリガ信号が発生するとグラフの表示更新を行います。

注 [連動] ／ [手動]（プロパティ パネルの [値の推移] タブ上の [全般] カテゴリ内 [サンプリング開始／停止] プロパティ）の指定に依存せず，サンプリングを停止します。

備考 トリガ信号発生によりグラフの表示更新を行っている間をトリガ保留期間とし，保留期間中のトリガ信号発生によるグラフの表示更新は行いません。保留期間中のトリガ信号は，[Auto] ／ [Single] では無視し，[Normal] では保留期間の解除直後に最新のトリガ信号に対応するグラフの表示更新を行います。

(c) [トリガ・ソース]

トリガ信号の入力として，どの変数（チャネル）を対象とするかを選択します。

ch1 ～ ch16 のうちいずれか 1 つのチャネルを選択します（デフォルト：[ch1]）。

(d)［トリガ・レベル］

トリガ信号の発生と判断するためのしきい値を指定します。

［トリガ・ソース］で指定した変数（チャネル）の数値が，ここで指定したしきい値を越えたか否かで，トリガ信号の発生とみなすか否かが決定されます。

“*トリガ・ソースの最小値*”～“*トリガ・ソースの最大値*”の範囲の 10 進数 /16 進数の数値（浮動小数指定可）を直接入力で指定します（デフォルト：[0]）。

備考 トリガ・レベルは，グラフの右側にトリガ・マーク（◀）として表示されます。
なお，このトリガ・マークをマウスによりドラッグすることで，トリガ・レベルの値を変更することができます（プログラム実行中を除く）。

(e)［トリガ・エッジの方向］

［トリガ・レベル］で指定したしきい値に対する方向を選択します。

［トリガ・ソース］で指定した変数（チャネル）の数値が，ここで指定した方向でしきい値を越えたか否かで，トリガ信号の発生とみなすか否かが決定されます。

立ち上がり	［トリガ・ソース］で選択した変数（チャネル）の数値が，［トリガ・レベル］で指定したしきい値未満からしきい値以上に変化した際にトリガ信号を発生します（デフォルト）。
立ち下がり	［トリガ・ソース］で選択した変数（チャネル）の数値が，［トリガ・レベル］で指定したしきい値より大きい値からしきい値以下に変化した際にトリガ信号を発生します。
両方	上記“立ち上がり”／“立ち下がり”のいずれかの条件でトリガ信号を発生します。

(f)［トリガ・ポジション］

トリガ信号が発生した箇所を描画する X 軸方向の位置（トリガ信号発生後のデータをグラフ化する位置）を指定します。

次の範囲の 10 進数値を直接入力により指定します（デフォルト：[0s]）。

なお，単位（s，ms，us，ns）を省略した場合は，“ms”として扱います（大文字／小文字不問）。

- 0 ～“(［1 グリッドあたりの時間 [Time/Div]］プロパティ値) × 10”

備考 トリガ・ポジションは，グラフの上側にトリガ・マーク（▼）として表示されます。
なお，このトリガ・マークをマウスによりドラッグすることで，トリガ・ポジションの値を変更することができます（プログラム実行中を除く）。

(g)［トリガ・マークの色］

トリガ・レベル，およびトリガ・ポジションを示すトリガ・マーク（◀／▼）の色を指定します。

色選択用コンボ・ボックスによる指定か，またはキーボードからの直接入力により 10 進数 /16 進数（0x 付き）の数値，または色名（「色の指定方法について」参照）を指定します（デフォルト：[Color Orange]）。

(4) グラフを表示する

プログラム実行[注]→停止を行ったのち，登録したグラフ化対象に関する最新のグラフが表示されます（デフォルト）。

ただし，該当するデータが取得できない場合は，グラフは表示されません。

なお，グラフが表示更新されるタイミングは次のとおりです。

- リアルタイム・サンプリング方式

  プログラム実行中においても，指定されているサンプリング間隔で表示内容が更新されます。

  なお，プロパティ パネルにおいて次の設定を行うことにより，このタブ上の［サンプリング］ボタンでサンプリングの開始／停止を手動で制御（トグル）することができます。

  - ［値の推移］タブ→［全般］カテゴリ→［サンプリング開始／停止］プロパティ→［手動］

- トレース・データ解析方式

  プログラムの実行が停止するごとに表示内容が更新されます（デフォルト）。

  ただし，プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［プログラム停止時に更新を行う］プロパティの指定を［はい］（デフォルト）以外に変更した場合，プロパティ パネルでの設定に従った表示内容の更新を行います。

注　【E1/E20【RL78】】

デバッグ・ツールをデータ収集モードで動作させる場合，デバッグ・ツールバーの  ボタンをクリックします（ボタン以外の実行系ボタンはすべて無効となります）。

詳細は，「CubeSuite+ 統合開発環境 ユーザーズマニュアル デバッグ編」を参照してください。

注意 1.　取得したグラフ・データがバッファ容量（10000 プロット分）を越えた場合，新しいグラフ・データを最も古いグラフ・データに上書きしていきます（リング・バッファ方式）。

この場合，グラフの描画が一部空白になります。

2.　リアルタイム・サンプリング方式を選択している場合，グラフ・データの取得に失敗すると，時間情報のみの表示となり，遷移箇所と遷移箇所を結ぶ線は表示されません（「(a) グラフ」参照）。

3.　【E1/E20【RL78】】

一度，デバッグ・ツールをデータ収集モードでプログラム実行させたのち，プロパティ パネルの［値の推移］タブ上の［Smart Analog］カテゴリ内［データ収集チャネル］プロパティを変更し，再度データ収集モードでプログラムを実行させた場合，グラフの最初の遷移箇所が不正になる場合があります。

備考　デバッグ・ツールのトレース・メモリ領域には限りがあります。したがって，トレース・データ解析方式によりグラフ・データを取得する場合，より広範囲での値の推移を表示するためには，ウォッチ パネルにおいてグラフ化対象にポイント・トレース・イベントを設定することをお勧めします。

このグラフに対して，次の表示設定を行うことができます。

(a) 表示グラフの限定

表示するグラフを限定することができます。

デフォルトでは，グラフ化対象が登録済みのチャネルのグラフはすべて表示されます。

グラフを非表示にする場合は，対応するチャネル番号の変数名チェック・ボックスのチェックを外します。

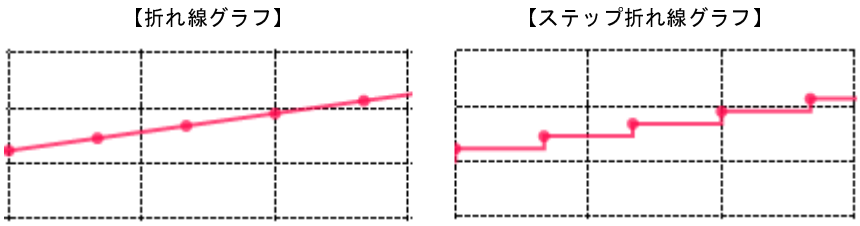
(b) グラフ形式の選択

リアルタイム・サンプリング方式を指定した場合，表示するグラフ形式を選択することができます。

グラフ形式の選択は，プロパティ パネルの［値の推移］タブ上の［全般］カテゴリ内［グラフの種類］プロパティにより行います（トレース・データ解析方式を指定している場合は，［ステップ折れ線グラフ］固定となります）。

折れ線グラフ	各プロットを直接線で結びグラフ化します。
ステップ折れ線グラフ	各プロットを垂直線で結びグラフ化します（ステップ・プロット形式）。

図 2—32 グラフ形式



備考 グラフの描画色は，チャネルごとに，同タブ上の［チャネル *1* ～ *16*］カテゴリ内［色 *1* ～ *16*］プロパティで変更することができます。

(c) 表示範囲の設定

グラフの表示は，X 軸 /Y 軸に対して 10 分割のグリッド線を表示して行われます。

- 自動調整機能を使用する（デフォルト）

取得したグラフ・データを基にして，X 軸に対する単位グリッドあたりの時間（Time/Div），および Y 軸に対する単位グリッドあたりの値（Val/Div）とオフセット値を，次のような最適な値に計算してグラフを表示します（デフォルト）。

- X 軸（時間）

  グラフの遷移箇所が描画領域（左端～右端）内で，指定された個数分[注]に収まるように自動調整します。

- Y 軸（値）

  グラフ・データの最大値／最小値が描画領域の上限／下限となるように自動調整します。

注 デフォルトで 20 個が指定されています。この値は，プロパティ パネルの［値の推移］タブ上の［全般］カテゴリ内［自動調整用の遷移箇所の数］プロパティにより変更することができます。

注意 1. **自動調整機能は，トリガ機能を使用する場合，またはデバッグ・ツールをデータ収集モードに設定している場合は無効となります。**

2. **次の操作を行うと，自動調整機能は無効となります（プロパティ パネルの［値の推移］タブ上の［全般］カテゴリ内［自動調整］プロパティが［行わない］に変更されます）。**

- 解析グラフ パネル上の［Time/Div］／［Val/Div］ラベルをダブルクリック

- マウス操作による表示範囲の変更（「手動で設定する」参照）

備考 自動調整を行うタイミングは，プロパティ パネルの［値の推移］タブ上の［全般］カテゴリ内［自動調整］プロパティにより指定することができます。

- 手動で設定する

プロパティ パネルの［値の推移］タブ上の［全般］カテゴリ内［自動調整］プロパティを［行わない］に設定したのち，次の値を設定します。

- X 軸（時間）

プロパティ パネルの［値の推移］タブ上の［全般］カテゴリ内［1 グリッドあたりの時間 [Time/Div]］プロパティにより，全チャネルを対象に，単位グリッドあたりの時間（Time/Div）を指定します。

- Y 軸（値）

プロパティ パネルの［値の推移］タブ上の［チャネル *1* ～ *16*］カテゴリ内［1 グリッドあたりの値 [Val/Div] *1* ～ *16*］プロパティ／［オフセット *1* ～ *16*］プロパティにより，各チャネルごとに，単位グリッドあたりの値（Val/Div）／オフセット値を指定します。

また，次のマウス操作によっても，上記の値を変更することができます。
ただし，プログラム実行中は，これらの機能は無効となります。

- 単位グリッドあたりの時間（Time/Div）

- 解析グラフ パネル上の［Time/Div］ラベルをダブルクリックすることにより，X 軸に対して自動調整機能による効果と同等の設定値となります。
- グラフ領域において，［Ctrl］キーを押下しながらマウス・ホイールを前後方に動かします。

- 単位グリッドあたりの値（Val/Div）／オフセット値

- 解析グラフ パネル上の［Val/Div］ラベルをダブルクリックすることにより，Y 軸に対して自動調整機能による効果と同等の設定値となります。
- 任意のグラフを選択した状態で（遷移箇所の形状が  に変化します），［Ctrl］キーを押下しながらマウス・ホイールを前後方に動かします。

- オフセット値

対象グラフの遷移箇所のいずれかにマウス・カーソルを重ねた状態で（マウス・カーソルの形状が  に変化します），［Shift］キーを押下しながらマウスを上下にクリック＆ドラッグします。
なお，ドラッグ中に［Esc］キーを押下した場合，オフセット値の変更はキャンセルされます。

（5）グラフ・データを検証する

表示されたグラフにおいて，必要に応じて次の操作を行うことができます。

注意 プログラム実行中は，この機能は無効となります。

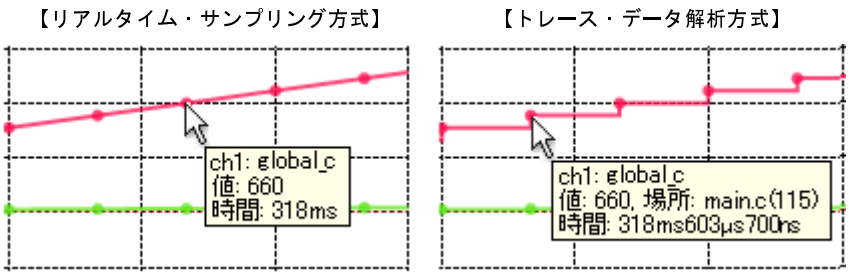
(a) 遷移箇所のポップアップ表示

グラフの遷移箇所の情報を確認します。

グラフの遷移箇所上にマウス・カーソルを重ねることにより，該当箇所の情報がポップアップ表示されます。

ただし，表示される内容は，グラフ・データの取得方法により異なります（「(i) ポップアップ表示」参照）。

図 2—33 遷移箇所のポップアップ表示例



備考 ポップアップ表示内に［場所］情報が表示されている場合，その箇所をダブルクリックすることにより，エディタ パネルで該当箇所を表示することができます（解析グラフ・データ・ファイル（*.mtac）の読み込みにより復帰したグラフの場合を除く）。

(b) カーソル計測

グラフ上のカーソル A/ カーソル B の位置に対する時間／数値を確認します。

X 軸（時間），または Y 軸 （値）を対象としてカーソル計測を行うことができます。

計測結果は，このタブ上のカーソル情報エリアに一覧表示されます（「(4) カーソル情報エリア」参照）。

操作方法は，まず，カーソル選択ボタン（［X 軸（Time 軸）］／［Y 軸 （Val 軸）］）により計測対象とする軸を選択します。次に，次の操作により，カーソル A/ カーソル B を任意の位置へ表示／設定します（カーソルはデフォルトで非表示です）。

カーソル	設定（表示）	設定解除（非表示）
カーソル A	［Ctrl］キー + クリック	［Ctrl］キー + ダブルクリック
カーソル B	［Ctrl］キー + 右クリック	［Ctrl］キー + 右ボタンのダブルクリック

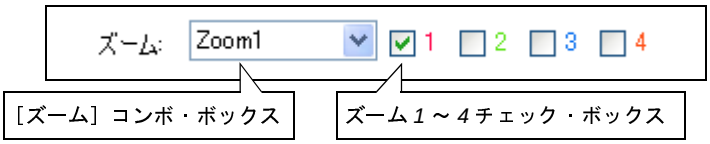
備考 プロパティ パネルの［値の推移］タブ上の［全般］カテゴリ内［カーソル A の色］／［カーソル B の色］プロパティにより，各カーソルの色を指定することができます。

(c) ズーム表示

グラフ上の任意の箇所をズーム表示します。

グラフ・コントロール・エリアのズーム *1* ～ *4* チェック・ボックスをチェックすることにより（複数選択可），チェックした番号に対応した値の推移（ズーム）パネルをオープンし，指定した範囲をズーム表示することができます（値の推移（ズーム）パネルは最大 4 個までオープンすることができます）。

図 2―34　ズーム表示

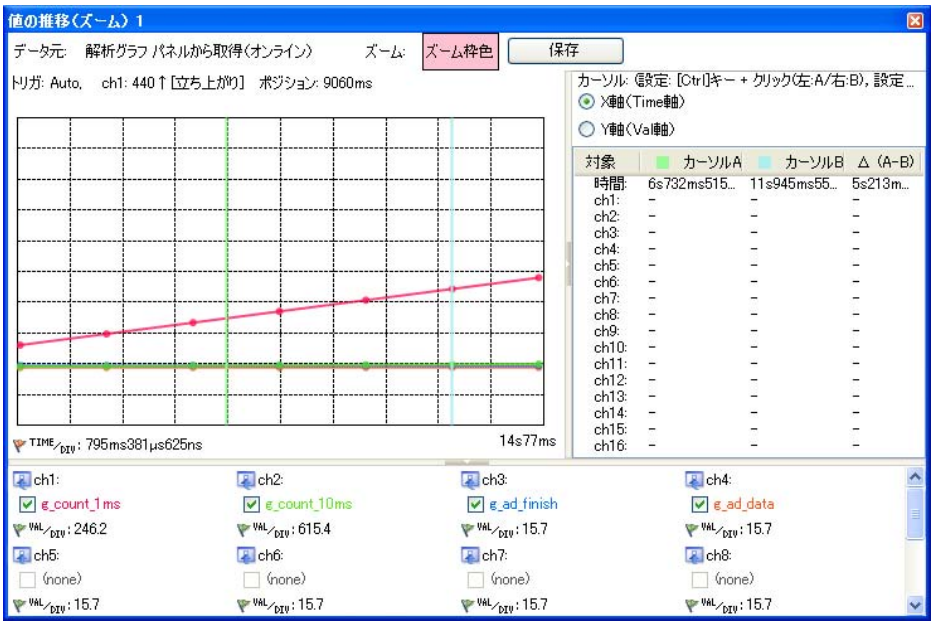


ズーム表示範囲の設定は，［ズーム］コンボ・ボックスにおいて表示対象となる値の推移（ズーム）パネルの番号を選択したのち，ズーム表示したい領域をマウスでクリック＆ドラッグすることで行います。この時グラフ上に表示されるドラッグ領域を示すズーム枠が表示されている限り，同動作でズーム表示範囲は再設定されます。

ドラッグ中に［Esc］キーを押下した場合，ズーム範囲の設定はキャンセルされます。また，設定した範囲を解除するには，グラフ領域内のいずれか（遷移箇所を除く）をダブルクリックします。

なお，ズーム表示の内容についての詳細は，値の推移（ズーム）パネルを参照してください。

図 2―35　ズーム表示（値の推移（ズーム）パネル）



備考 1.　ズーム表示範囲の設定は，4 個の値の推移（ズーム）パネルに対して，それぞれ個別に行うことができます。

2.　プロパティ パネルの［値の推移］タブ上の［全般］カテゴリ内［ズーム枠 *1* ～ *4* の色］プロパティにより，各ズーム枠の色を指定することができます。

(6) グラフを復帰するためのグラフ・データを保存する

現在表示しているグラフを解析グラフ・データ・ファイル（*.mtac）に保存したのち，そのファイルを読み込むことで，グラフを復帰（再表示）させることができます。

操作手順は次のとおりです。

(a) グラフ・データを保存する

- ［値の推移］タブ上のグラフの場合

保存したいグラフが表示されている状態で，［ファイル］メニュー→［名前を付けて 解析グラフ・データ を保存 ...］を選択し，名前を付けて保存 ダイアログをオープンします。

このダイアログにおいて，［ファイルの種類］エリアにおいて“解析グラフ・データ (*.mtac)”を選択，および［ファイル名］エリアにおいて任意のファイル名（拡張子は“mtac”に限ります）を指定したのち，［保存］ボタンをクリックします。

- 値の推移（ズーム）パネル上のグラフの場合

保存したいグラフが表示されている状態で，［保存］ボタンを選択し，名前を付けて保存 ダイアログをオープンします。

このダイアログにおいて，［ファイルの種類］エリアにおいて“解析グラフ・データ (*.mtac)”を選択，および［ファイル名］エリアにおいて任意のファイル名（*.mtac）を指定したのち，［保存］ボタンをクリックします。

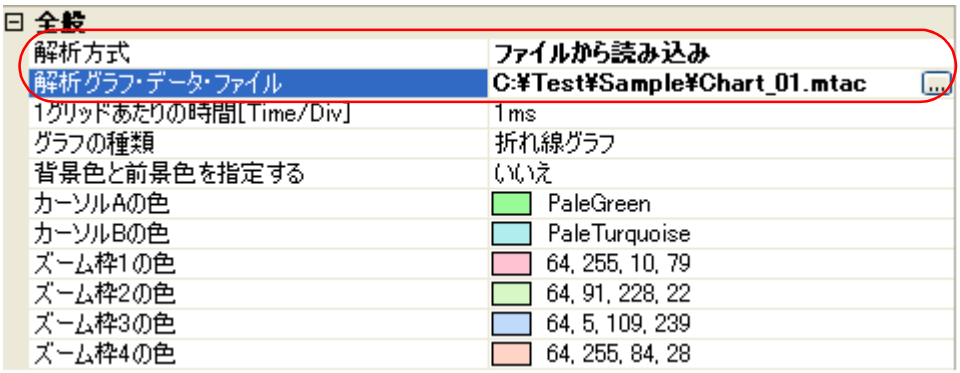
ただし，値の推移（ズーム）パネルで保存したグラフ・データは，ズーム表示した範囲のみに限られます。

(b) グラフ・データを読み込む

プロパティ パネルの［値の推移］タブ上の［全般］カテゴリ内［解析方式］プロパティにおいて，［ファイルから読み込み］を選択します（「(2) グラフ・データの取得方法を選択する」参照）。

続いて，同カテゴリ内［解析グラフ・データ・ファイル］プロパティにおいて，先に保存した解析グラフ・データ・ファイル（*.mtac）を指定します。相対パスによる指定の場合は，プロジェクト・フォルダを基点として指定してください。

図 2―36 グラフ・データの読み込み（［全般］カテゴリ）



なお，この際に保存／復帰の対象となるデータは，次のとおりです。

内容		復帰先
チャネルごとのグラフ・データ		
	数値	グラフ表示エリア，およびチャネル情報エリア
	時間	
	グラフの表示／非表示	
プロパティ パネルの［値の推移］タブ上のプロパティ値		
	単位グリッドあたりの時間	［全般］カテゴリ→［1 グリッドあたりの時間 [Time/Div]］
	各チャネルに登録された変数名	［チャネル *1* ～ *16*］カテゴリ→［変数名／アドレス *1* ～ *16*］
	チャネルごとの型／サイズ	［チャネル *1* ～ *16*］カテゴリ→［型／サイズ *1* ～ *16*］
	チャネルごとの単位グリッドあたりの数値	［チャネル *1* ～ *16*］カテゴリ→［1 グリッドあたりの値 [Val/Div]*1* ～ *16*］
	チャネルごとのオフセット値	［チャネル *1* ～ *16*］カテゴリ→［オフセット *1* ～ *16*］
	サンプリング間隔 **【E1/E20【RL78】】**	［Smart Analog］カテゴリ→［サンプリング間隔 [ms]］ ただし，選択しているマイクロコントローラが Smart Analog IC 搭載品の場合で，デバッグ・ツールをデータ収集モードでグラフ・データを取得した場合のみが保存の対象となります。

注意　グラフの表示／非表示にかかわらず，グラフ・データが存在しないチャネルについては保存の対象外となります。この場合，当該チャネルに対応するプロパティのデフォルト値が適用されます。

備考　グラフの復帰以外の目的のグラフ・データの保存についての詳細は，「2. 14　解析情報をファイルに保存する」を参照してください。

2. 13. 2　関数の実行時間率をグラフ化する

関数の実行時間の割合を円グラフで表示します。

グラフ表示は，現在取得している動的解析情報（関数一覧 パネルにおける［実行時間 (割合)[%]］と同等）を基に，解析グラフ パネルの［実行時間の割合］タブで行います。

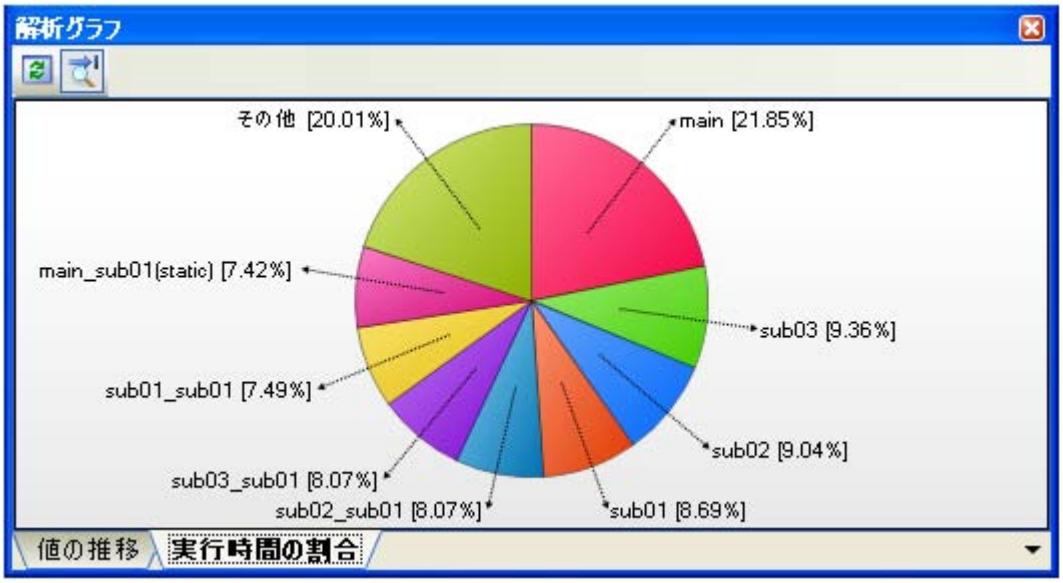
なお，表示される各エリアについての詳細は，［実行時間の割合］タブを参照してください。

注意 1.　デバッグ・ツールがトレース機能をサポートしていない場合，またはデバッグ・ツールのトレース機能を有効化していない場合，このグラフを表示することはできません。
また，トレース機能を有効化している状態であっても，トレース・メモリにトレース・データが存在しない場合は，グラフ表示は行わず，出力 パネルに次のメッセージを表示します。
“実行時間情報がありません。”

2.　グラフ表示を行うためには，「(a) トレース機能」の注意も参照してください。

3.　【IECUBE【78K0】】【E1/E20【RX】】【EZ Emulator【RX】】
トレース・タイム・タグ機能をサポートしていないため，このグラフを表示することはできません。

図 2―37 関数の実行時間率のグラフ化



このグラフに対して，次の操作を行うことができます。

(1) 表示する関数の数の設定

グラフに表示する関数の数を変更することができます。

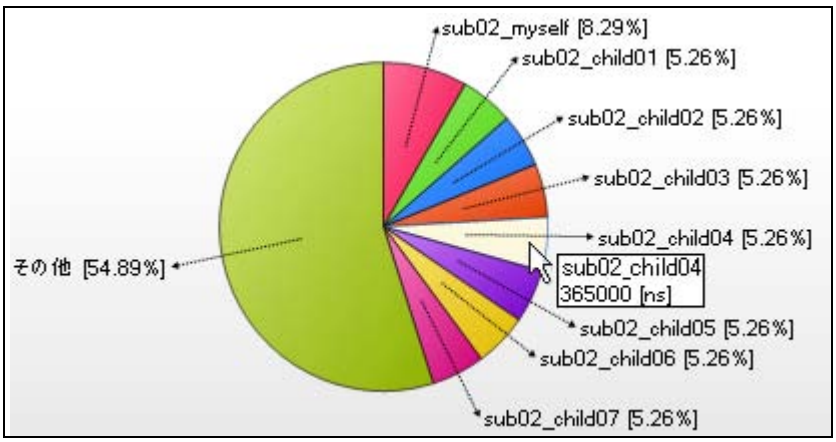
操作は，プロパティ パネルの［設定］タブ上の［解析グラフ］カテゴリ内［実行時間の割合グラフに表示する関数の数］プロパティの指定により行います（デフォルトでは［10］が指定されます）。

実行時間の割合の大きい順にグラフ化の対象となり，ここで指定した数を越える関数については，“その他”としてまとめて表示されます。

(2) 実行時間のポップアップ表示

グラフ上にマウス・カーソルを重ねることにより，該当関数の実行時間情報をポップアップ表示します。

図 2―38 実行時間のポップアップ表示例



備考 プログラムの実行が停止するごとに表示内容が更新されます（デフォルト）。
ただし，プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［プログラム停止時に更新を行う］プロパティの指定を［はい］（デフォルト）以外に変更した場合，プロパティ パネルでの設定に従った表示内容の更新を行います。

2. 14 解析情報をファイルに保存する

関数一覧 パネル／変数一覧 パネル／解析グラフ パネル／コール・グラフ パネル／値の推移（ズーム）パネルの内容は，ファイルに保存することができます。

（1）関数情報を保存する

操作は，関数一覧 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 関数一覧データを保存 ...］を選択するとオープンする名前を付けて保存 ダイアログにより行います。

保存の際は，次のファイル形式を指定することができます。

テキスト・ファイル (*.txt)	テキスト形式
CSV(カンマ区切り) (*.csv)	CSV 形式
Microsoft Office Excel ブック (*.xls)	Microsoft Office Excel ブック形式
関数一覧ファイル (*.mtfl)	関数情報をインポートするためのファイル形式（「2. 12 情報ファイルをインポート／エクスポートする」参照）

注意 現在パネル上に表示している項目／解析情報値のみが保存対象となります。

（2）変数情報を保存する

操作は，変数一覧 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 変数一覧データを保存 ...］を選択するとオープンする名前を付けて保存 ダイアログにより行います。

保存の際は，次のファイル形式を指定することができます。

テキスト・ファイル (*.txt)	テキスト形式
CSV(カンマ区切り) (*.csv)	CSV 形式
Microsoft Office Excel ブック (*.xls)	Microsoft Office Excel ブック形式
変数一覧ファイル (*.mtvl)	変数情報をインポートするためのファイル形式（「2. 12 情報ファイルをインポート／エクスポートする」参照）

注意 現在パネル上に表示している項目／解析情報値のみが保存対象となります。

（3）グラフ情報を保存する

操作は，解析グラフ 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 解析グラフ・データ を保存 ...］を選択，または値の推移（ズーム）パネル上の［保存］ボタンをクリックするとオープンする名前を付けて保存 ダイアログにより行います。

保存の際は，次のファイル形式を指定することができます。

テキスト・ファイル (*.txt)	テキスト形式
CSV(カンマ区切り) (*.csv)	CSV 形式
Microsoft Office Excel ブック (*.xls)	Microsoft Office Excel ブック形式
解析グラフ・データ (*.mtac) 注1	解析グラフ・データ・ファイル
ビットマップ (*.bmp)	ビットマップ形式（32 ビット）（画像形式）

JPEG ファイル (*.jpg)	JPEG 形式（画像形式）
PNG ファイル (*.png)	PNG 形式（画像形式）
EMF ファイル (*.emf) 注 2	EMF 形式（画像形式）

注 1. 解析グラフ パネルの［値の推移］タブのみサポートします。

2. グラフ描画領域のみが保存されます（トリガ・マーク／チャネル情報などは保存されません）。

注意 1. 解析グラフ パネルの場合，保存の対象は，現在表示しているタブの内容のみとなります。

2. 画像形式を選択した場合，保存の対象は，現在表示している部分のみとなります。

（4）コール・グラフ情報を保存する

操作は，コール・グラフ 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・グラフ・データ を保存 ...］を選択するとオープンする名前を付けて保存 ダイアログにより行います。

保存の際は，次のファイル形式を指定することができます。

なお，“（可視部のみ）”を選択すると，現在パネル上で表示されている部分のみをファイルに保存します。

ビットマップ（可視部のみ）(*.bmp)	ビットマップ形式（32 ビット）（画像形式）
JPEG ファイル（可視部のみ）(*.jpg)	JPEG 形式（画像形式）
PNG ファイル（可視部のみ）(*.png)	PNG 形式（画像形式）
ビットマップ (*.bmp)	ビットマップ形式（32 ビット）（画像形式）
JPEG ファイル (*.jpg)	JPEG 形式（画像形式）
PNG ファイル (*.png)	PNG 形式（画像形式）
EMF ファイル (*.emf)	EMF 形式（画像形式）

注意 プロジェクトが巨大な場合，コール・グラフ全域の画像ファイルを保存できない場合があります。

備考 ズーム機能を適用している場合，現在のズーム率で画像を保存します（EMF 形式を除く）。

第 3 章　注意事項

この章では，解析ツールを使用する際の注意事項を示します。

3.1　アクティブ・プロジェクトの変更について

アクティブ・プロジェクトを変更した場合，クロス・リファレンス・ファイルが生成されていないため，関数一覧 パネル／変数一覧 パネル／コール・グラフ パネル／クラス／メンバ パネルに何も表示されない場合があります。

この場合，プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［静的解析を有効にする］プロパティを［はい］に設定してリビルドを実行してください。パネルの内容が更新されます。

3.2　カバレッジ結果について

コード・カバレッジとデータ・カバレッジの結果は，プログラムの実行結果の蓄積を表示します。したがって，「プログラムをダウンロードして実行して停止」を繰り返した場合，繰り返した分の結果を蓄積して表示します。

カバレッジの結果をクリアするには，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［カバレッジ情報をクリア］を選択したのち，関数一覧 パネル／変数一覧 パネルの  ボタンをクリックしてください。

また，プログラムを修正してビルドを実行した結果，関数や変数の配置アドレスが前回のビルド時の配置アドレスと異なる場合があります。この場合，実行していない関数や，リード／ライトされていない変数のカバレッジ率が表示されます。

3.3　リアルタイム・サンプリング方式について

リアルタイム・サンプリング方式を指定してグラフ・データを取得する際において，グラフ化する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（「例　RL78 マイクロコントローラを使用した場合」参照）。

この 2 回の代入処理の間で変数の読み出しが行われると，変数へ値が代入される途中の値が読み出され，実際には代入していない値が表示されることがあるため注意が必要です。

例　RL78 マイクロコントローラを使用した場合

この例では，“命令 1”実行完了後から，“命令 2”実行完了前にサンプリングがあった場合，下位 2 バイトのみ代入が完了した変数“value_a”の値を読み出します。

【C 言語ソース】

```
long int    value_a = 0;    // 4 バイト変数定義

void func(void)
{
    value_a = 4000000000;    // 4 バイト変数への代入
}
```

【上記代入処理のアセンブラ命令】

```
MOVW    AX, #2800H
MOVW    !_value_a, AX       ; 命令 1：変数“value_a”の下位 2 バイトを代入
MOVW    AX, #0EE6BH
MOVW    !_value_a+2, AX     ; 命令 2：変数“value_a”の上位 2 バイトを代入
```

3.4 トレース・データ解析方式について

トレース・データ解析方式を指定してグラフ・データを取得した場合の注意事項を列挙します。

- トレースの種類として，ポイント・トレース・イベントと区間を指定したトレース・イベントを組み合わせて使用すると，区間を指定したトレース・イベントが終了したときの命令行が，以降のポイント・トレース・イベントに適用されることがあります。この場合，グラフにおけるポップアップ表示の［場所］に“-”が表示されず，不正なファイル名と行数が表示されます。

- bit 型変数 /boolean 型変数 /_Bool 型変数／構造体ビット・フィールドはバイト単位で解析されます。
したがって，同一アドレスに割り当てられている bit 型変数 /boolean 型変数 /_Bool 型変数／構造体ビット・フィールドへの書き込みが発生した箇所についても値の遷移箇所となります。
この遷移箇所をダブルクリックすると，登録した変数へのアクセスがないソース行（同一アドレスに割り当てられている変数のアクセスした箇所）にジャンプします。
また，構造体のビット・フィールドでバイトをまたがっているフィールドの場合，トレース・データに一部のバイト・アクセスのみが出力されます。この場合のグラフは，変数値が解析不能であるため，ロスト区間（「(a) グラフ」参照）として表示されます。

3.5 プログラム実行中にパネルをオープンした場合について

(1) 関数一覧 パネル／変数一覧 パネル

静的解析情報のみを更新して表示します。動的解析情報の更新は行いません。

(2) 解析グラフ パネル

- CubeSuite+ 起動後，一度もパネルをオープンしていない場合

［値の推移］タブ	何も表示しません。
［実行時間の割合］タブ	何も表示しません。

- CubeSuite+ 起動後，パネルをオープンしたあとの場合

［値の推移］タブ	- リアルタイム・サンプリング方式を選択している場合 指定したサンプリング間隔で，リアルタイムにグラフを更新して表示します。 - トレース・データ解析方式を選択している場合 前回の内容を表示します。
［実行時間の割合］タブ	前回の内容を表示します。

(3) コール・グラフ パネル

- CubeSuite+ 起動後，一度もパネルをオープンしていない場合

  静的解析情報のみを更新して表示します。動的解析情報の更新は行いません。

- CubeSuite+ 起動後，パネルをオープンしたあとの場合

  前回の内容を表示します。

(4) クラス／メンバ パネル

静的解析情報のみを更新して表示します。動的解析情報の更新は行いません。

(5) 値の推移（ズーム）パネル

前回の内容を表示します。

3.6 CC-RX（C++ ソース・ファイル）を使用する場合について

(1) 関数一覧 パネル

- テンプレート関数／テンプレート・クラス中に定義されているメンバ関数の注意事項は次のとおりです。
  - ［ファイル名］列には，“（定義箇所なし）”と表示されます。
  - ［引数］列には，引数の型のみが表示され，引数名は表示されません。
  - テンプレート・クラス中に定義されたメンバ関数の［開始アドレス］／［終了アドレス］列には，“-”が表示されます。

    また，［開始アドレス］列に“-”が表示されている場合は，エディタ パネル／逆アセンブル パネル／メモリ パネルへジャンプができません。
  - コンテキスト・メニューの［すべての参照を検索］では，定義箇所が表示されません。

    また，参照している関数情報／変数情報が表示されません。
  - テンプレート関数／テンプレート・クラス中に定義されているメンバ関数内で参照している関数の参照回数は計数されません。

    同様に，コンテキスト・メニューの［すべての参照を検索］による参照情報は表示されません。
  - テンプレート・クラス中に定義されているメンバ関数の場合，コンテキスト・メニューの［関数の先頭にブレークを設定］は無効となります。
- クラス宣言にて定義されているメンバ関数が，宣言のみで使用されている場合は，ファイル名が表示されません。定期箇所がない関数として扱われます。
- 関数の引数にクラス型を指定した場合，［開始アドレス］／［終了アドレス］／［コード・サイズ [バイト]］列には，“-”が表示されます。
- 関数の引数に signed char 型を指定している関数と char 型を指定しているオーバーロード関数が定義されている場合，［開始アドレス］／［終了アドレス］／［コード・サイズ [バイト]］列には，“-”が表示されます。

(2) 変数一覧 パネル

- テンプレート関数／テンプレート・クラス中に定義されているメンバ関数にて定義されている関数内のスタティック変数は表示されません。

- テンプレート関数／テンプレート・クラス中に定義されているメンバ関数内で参照している変数の参照回数は計数されません。
- extern/volatile 宣言されていない const 変数は，コンパイラによって定数値に置換されるため表示されません。
- ファイルが異なる無名名前空間にて定義されている同名のグローバル変数の型は同じ型として扱われます。
- 無名構造体／無名共用体の［アドレス］と［サイズ [バイト]］は表示できません。

(3) コール・グラフ パネル

- デフォルトの設定では，テンプレート関数／テンプレート・クラス中に定義されているメンバ関数は表示されません。プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［定義箇所がない関数／変数をコール・グラフの表示対象とする］プロパティの指定を［はい］にして表示させてください。
- テンプレート関数／テンプレート・クラス中に定義されているメンバ関数から呼び出している関数，または参照している変数は表示されません。

(4) クラス／メンバ パネル

- 名前空間の別名は表示されません。
- テンプレート関数／テンプレート・クラス中に定義されているメンバ関数に対して，コンテキスト・メニューの［ソースへジャンプ］／［ソースの宣言へジャンプ］は無効となります。

付録 A　ウインドウ・リファレンス

この付録では，解析ツールで使用するウインドウ／パネル／ダイアログについての詳細を説明します。

A. 1　説　　明

次に，解析ツールで使用するウインドウ／パネル／ダイアログの一覧を示します。

表 A—1　ウインドウ／パネル／ダイアログ一覧

ウインドウ／パネル／ダイアログ名	機能概要
メイン・ウインドウ	CubeSuite+ を起動した際，最初にオープンするウインドウ
プロジェクト・ツリー パネル	プロジェクトの構成要素のツリー表示
プロパティ パネル	解析ツールの詳細情報の表示，および設定の変更
関数一覧 パネル	取得した関数情報の表示
変数一覧 パネル	取得した変数情報の表示
解析グラフ パネル	取得した関数情報／変数情報のグラフ表示
コール・グラフ パネル	関数間の呼び出し関係（コール・グラフ）の表示
クラス／メンバ パネル	取得したクラス情報【CC-RX】注／関数情報／変数情報のツリー表示
値の推移（ズーム）パネル	グラフのズーム表示
出力 パネル	CubeSuite+ が提供している各種コンポーネントから出力されるメッセージの表示，および関数／変数の参照箇所一覧の表示
解析対象外ファイルを指定 ダイアログ	解析対象外とするファイルの選択
解析対象ファイルを指定 ダイアログ	解析対象とするファイルの選択
パス編集 ダイアログ	情報ファイル（関数一覧ファイル（*.mtfl）／変数一覧ファイル（*.mtvl））をインポートする際のファイルの指定
ファイルを開く ダイアログ	グラフを復帰するためのファイルの選択
列の選択 ダイアログ	関数一覧 パネル／変数一覧 パネルにおける表示項目の並べ替え，または表示／非表示の設定
コール・グラフ検索 ダイアログ	コール・グラフ パネルで表示しているコール・グラフ内に存在する関数／変数の検索
フィルタ設定 ダイアログ	関数一覧 パネル／変数一覧 パネルにおけるフィルタ表示の条件設定
名前を付けて保存 ダイアログ	関数一覧 パネル／変数一覧 パネル／解析グラフ パネル／コール・グラフ パネルの表示内容の新規保存，および情報ファイル（関数一覧ファイル（*.mtfl）／変数一覧ファイル（*.mtvl））の生成

注 【CC-RX】

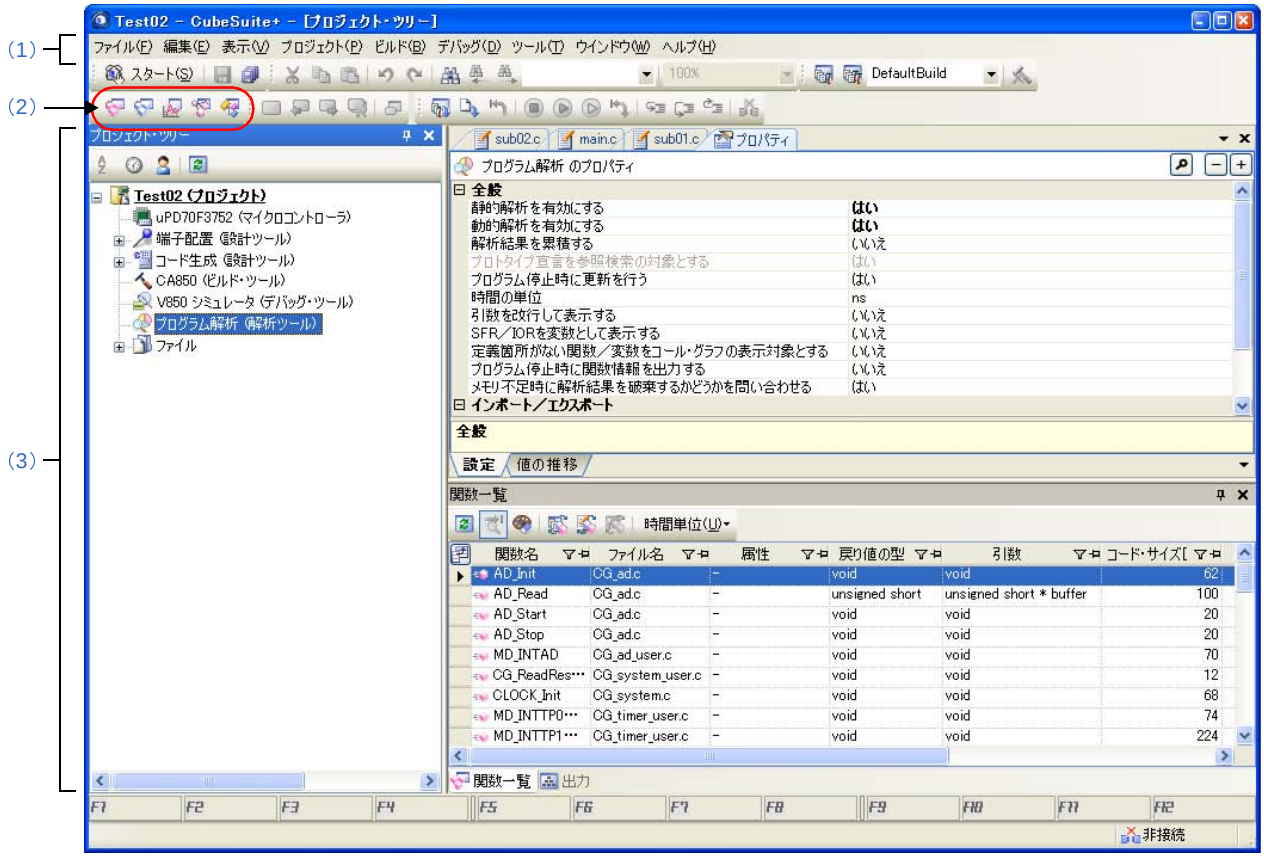
クラス情報は，C++ ソース・ファイルを対象とする場合のみ提供される情報です。

メイン・ウインドウ

CubeSuite+ を起動した際，最初にオープンするウインドウです。

解析ツールを使用する際は，このウインドウから各パネルのオープン操作を行います。

図 A—1　メイン・ウインドウ



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］

［オープン方法］

- Windows の［スタート］メニュー→［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

(a)［表示］

解析ツール専用の［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）。

出力	出力 パネルをオープンします。
プログラム解析	解析ツール用の各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。
関数一覧	関数一覧 パネルをオープンします。
変数一覧	変数一覧 パネルをオープンします。
解析グラフ	解析グラフ パネルをオープンします。
コール・グラフ	コール・グラフ パネルをオープンします。
クラス／メンバ	クラス／メンバ パネルをオープンします。

(2) ツールバー

解析ツール専用のツールバーの各ボタン，および機能は次のとおりです（デフォルト）。

	関数一覧 パネルをオープンします。 ［表示］メニュー→［関数一覧］の選択と同等です。
	変数一覧 パネルをオープンします。 ［表示］メニュー→［変数一覧］の選択と同等です。
	解析グラフ パネルをオープンします。 ［表示］メニュー→［解析グラフ］の選択と同等です。
	コール・グラフ パネルをオープンします。 ［表示］メニュー→［コール・グラフ］の選択と同等です。
	クラス／メンバ パネルをオープンします。 ［表示］メニュー→［クラス／メンバ］の選択と同等です。

(3) パネル表示エリア

CubeSuite+ が使用する各種パネルを表示するエリアです。

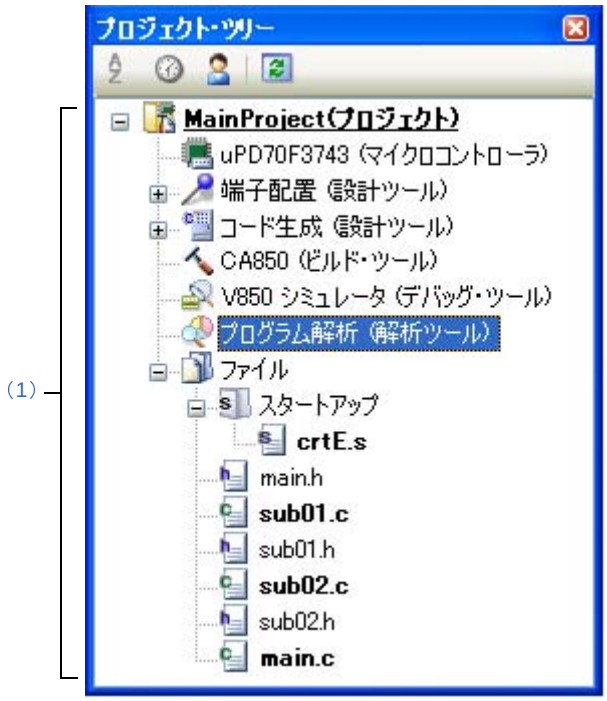
解析ツールが使用するパネルについての詳細は，次の各パネルの項を参照してください。

- プロジェクト・ツリー パネル
- プロパティ パネル
- 関数一覧 パネル
- 変数一覧 パネル
- 解析グラフ パネル
- コール・グラフ パネル
- クラス／メンバ パネル
- 値の推移（ズーム）パネル
- 出力 パネル

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，設計ツール，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。

図 A―2　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として，次の解析ツールのノードを表示します。

ノード名	説明
プログラム解析 (解析ツール)	使用する解析ツールです。

備考　ノードを選択すると，解析ツールの詳細情報（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]

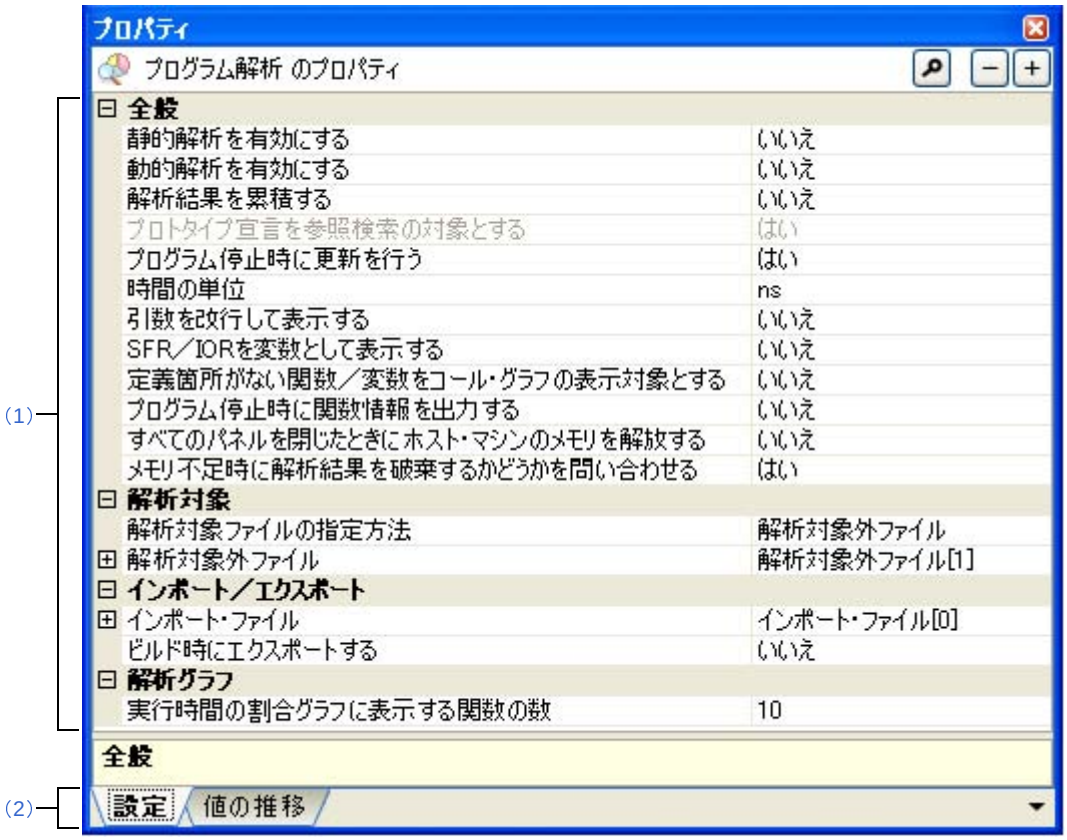
［プログラム解析（解析ツール）］ノードをマウスで右クリックすることにより表示されるコンテキスト・メニューの各項目，および機能は次のとおりです。

関数一覧	関数一覧 パネルをオープンします。
変数一覧	変数一覧 パネルをオープンします。
解析グラフ	解析グラフ パネルをオープンします。
コール・グラフ	コール・グラフ パネルをオープンします。
クラス／メンバ	クラス／メンバ パネルをオープンします。
プロパティ	解析ツールの詳細情報（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

解析ツールの詳細情報の表示，および設定の変更を行います。

図 A―3　プロパティ パネル（【V850】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，［プログラム解析（解析ツール）］ノードを選択したのち，［表示］メニュー→［プロパティ］を選択
- プロジェクト・ツリー パネルにおいて，［プログラム解析（解析ツール）］ノードを選択したのち，コンテキスト・メニュー→［プロパティ］を選択

備考　すでにプロパティ パネルが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，［プログラム解析（解析ツール）］ノードを選択することで，解析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

解析ツールの詳細情報を，カテゴリ別のリスト形式で表示し，設定の変更を直接行うことができるエリアです。

⊟マークは，そのカテゴリ内に含まれているすべてのプロパティ項目が展開表示されていることを示し，また，⊞マークは，カテゴリ内のプロパティ項目が折りたたみ表示されていることを示します。展開／折りたたみ表示の切り替えは，このマークのクリック，またはカテゴリ名のダブルクリックにより行うことができます。

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）。

- ［設定］タブ
- ［値の推移］タブ

[［編集］メニュー（プロパティ パネル専用部分）]

プロパティ パネル専用の［編集］メニューの各項目，および機能は次のとおりです。

元に戻す	直前に行ったプロパティの値の編集作業を取り消します。
切り取り	プロパティの値を編集中の場合，選択している文字列を切り取ってクリップ・ボードに移動します。
コピー	選択しているプロパティの値の文字列をクリップ・ボードにコピーします。
貼り付け	プロパティの値を編集中の場合，クリップ・ボードの内容を挿入します。
削除	プロパティの値を編集中の場合，選択している文字列を削除します。
すべて選択	プロパティの値を編集中の場合，選択しているプロパティの値文字列をすべて選択します。

[コンテキスト・メニュー]

このパネル上において，マウスを右クリックすることにより表示されるコンテキスト・メニューの各項目，および機能は次のとおりです。

(1) 文字列編集中以外の場合

デフォルトに戻す	選択しているプロパティ項目の設定値をデフォルトに戻します。
すべてデフォルトに戻す	現在選択しているタブ上の設定値をすべてデフォルトに戻します。

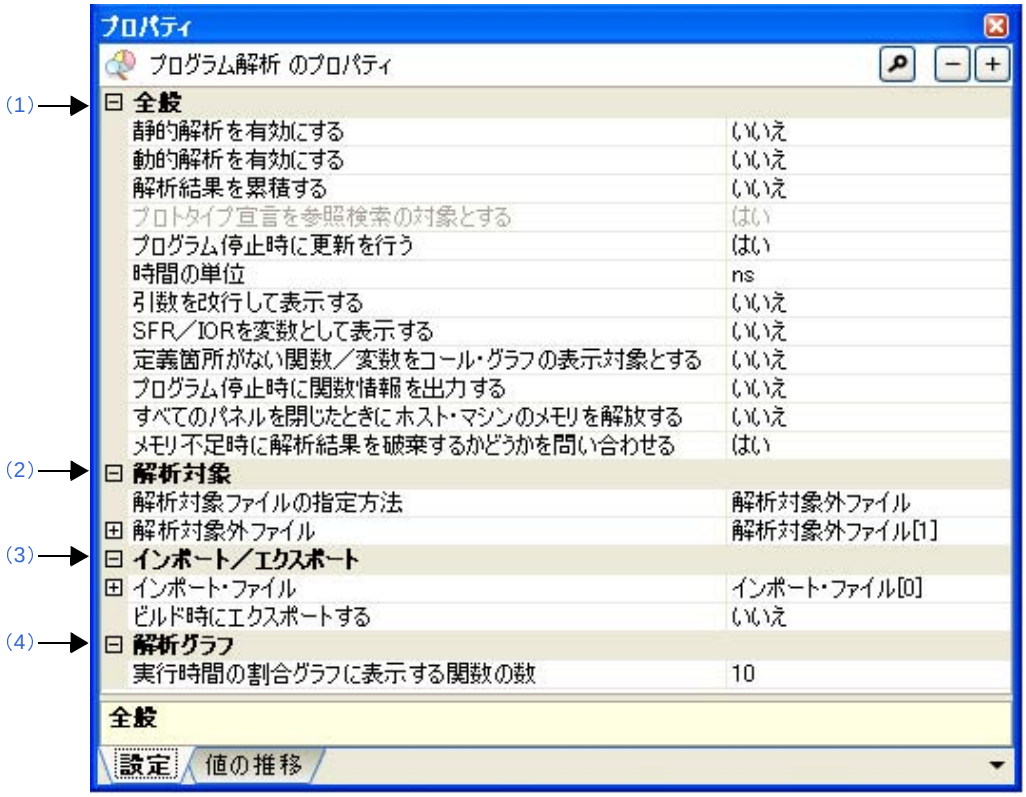
（2）文字列編集中の場合

元に戻す	直前に行ったプロパティの値の編集作業を取り消します。
切り取り	プロパティの値を編集中の場合，選択している文字列を切り取ってクリップ・ボードに移動します。
コピー	選択しているプロパティの値文字列をクリップ・ボードにコピーします。
貼り付け	プロパティの値を編集中の場合，クリップ・ボードの内容を挿入します。
削除	プロパティの値を編集中の場合，選択している文字列を削除します。
すべて選択	プロパティの値を編集中の場合，選択しているプロパティの値文字列をすべて選択します。

［設定］タブ

［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。

図 A―4　プロパティ パネル：［設定］タブ（【V850】の例）



［各カテゴリの説明］

（1）［全般］

解析ツールの全般に関する詳細情報の表示，および設定の変更を行います。

<table>
<tr><td rowspan="5">静的解析を有効にする</td><td colspan="3">解析ツールが静的解析情報を取得するために必要なクロス・リファレンス情報を得るために，ビルド・ツール上で指定しているクロス・リファレンス情報を出力するか否かのプロパティ設定^{注1}を無視し，ビルドの際に強制的にクロス・リファレンス情報を出力するか否かを選択します。</td></tr>
<tr><td>デフォルト</td><td colspan="2">いいえ</td></tr>
<tr><td>変更方法</td><td colspan="2">ドロップダウン・リストによる選択</td></tr>
<tr><td rowspan="2">指定可能値</td><td>はい</td><td>ビルド・ツールのプロパティ設定を無視し，強制的にクロス・リファレンス情報を出力します。</td></tr>
<tr><td>いいえ</td><td>ビルド・ツールのプロパティ設定を優先します。</td></tr>
<tr><td rowspan="5">動的解析を有効にする</td><td colspan="3">解析ツールが動的解析情報を取得するために必要なデバッグ・ツールの機能の有効／無効のプロパティ設定を無視し，強制的にデバッグ・ツールの機能^{注2}を有効化するか否かを選択します。</td></tr>
<tr><td>デフォルト</td><td colspan="2">いいえ</td></tr>
<tr><td>変更方法</td><td colspan="2">ドロップダウン・リストによる選択</td></tr>
<tr><td rowspan="2">指定可能値</td><td>はい</td><td>デバッグ・ツールのプロパティ設定を無視し，強制的にデバッグ・ツールの機能を有効化します。</td></tr>
<tr><td>いいえ</td><td>デバッグ・ツールのプロパティ設定を優先します。</td></tr>
<tr><td rowspan="5">解析結果を累積する</td><td colspan="3">解析情報として表示する実行回数／実行時間をプログラム実行ごとの累積による数値とするか否かを選択します。
このプロパティの対象となる項目は次のとおりです。
- 関数一覧 パネル
［実行回数］／［実行時間 [単位]］
- 変数一覧 パネル
［リード回数］／［ライト回数］／［リード／ライト回数］／［最大値］／［最小値］
- コール・グラフ パネル
実行回数／リード回数／ライト回数</td></tr>
<tr><td>デフォルト</td><td colspan="2">いいえ</td></tr>
<tr><td>変更方法</td><td colspan="2">ドロップダウン・リストによる選択</td></tr>
<tr><td rowspan="2">指定可能値</td><td>はい</td><td>前回のプログラム実行による計測値に今回のプログラム実行の計測値を加算して表示します。</td></tr>
<tr><td>いいえ</td><td>プログラム実行ごとの計測値を表示します。
なお，［はい］から［いいえ］に変更した場合，現在の解析結果をクリアしたのち，実行して計測した数値を表示します。</td></tr>
</table>

プロトタイプ宣言を参照検索の対象とする	関数の参照箇所の一覧表示を行う際に（「2. 11　参照箇所を一覧表示する」参照），プロトタイプ宣言を参照箇所として含めるか否かを選択します。		
	デフォルト	はい	
	変更方法	【RH850】【CC-RX】【CX】 　ドロップダウン・リストによる選択 【CA850】【CA78K0R】【CA78K0】 　変更不可	
	指定可能値	はい	プロトタイプ宣言を含めます。
		いいえ	プロトタイプ宣言を含めません。
プログラム停止時に更新を行う	プログラムの実行が停止した際に最新情報を取得し，関数一覧 パネル／変数一覧 パネル／解析グラフ パネル／コール・グラフ パネルの表示内容の更新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	プログラム実行の停止後，表示内容の更新を行います。
		いいえ	プログラム実行が停止しても表示内容の更新は行いません。
		個別に指定する	各パネルのツールバーの  ボタンが有効となります（プログラム実行後の表示内容の更新に関する設定は，このボタンにより行います）。
時間の単位	解析ツールで使用する時間の単位を選択します。		
	デフォルト	ns	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	ns	ナノ秒単位で表示します（整数表示）。
		μs	マイクロ秒単位で表示します（小数 3 桁表示）。
		ms	ミリ秒単位で表示します（小数 3 桁表示）。
		s	秒単位で表示します（小数 3 桁表示）。
		h:min:s	時間，分（0 ～ 59），秒（0 ～ 59）で表示します。
引数を改行して表示する	関数一覧 パネルにおける［引数］項目の値を改行して表示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	値ごとに改行し，複数行で表示します。
		いいえ	各値を “,” で区切り，改行せずに表示します。

SFR ／ IOR を変数として表示する	変数一覧 パネルにおいて，プログラム中で使用されている SFR/IOR を変数とみなして表示するか否かを選択します。		
	デフォルト	いいえ	
	変更方法	【CC-RH】【CC-RX】 変更不可 【CA850】【CX】【CA78KR0】【CA78K0】 ドロップダウン・リストによる選択	
	指定可能値	はい	SFR/IOR を変数とみなして表示します。
		いいえ	SFR/IOR を変数とみなしません。
定義箇所がない関数／変数をコール・グラフの表示対象とする	コール・グラフ パネルにおいて，定義箇所がない（ソース・ファイルが存在しない）関数／変数もコール・グラフに表示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	定義箇所がない関数／変数もコール・グラフに表示します。
		いいえ	定義箇所がない関数／変数はコール・グラフに表示しません。
プログラム停止時に関数情報を出力する 【RH850】 【V850】	STF 用の情報ファイル[注3]を，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→［出力ファイルの種類と場所］カテゴリ→［中間ファイル出力フォルダ］プロパティで指定しているフォルダに出力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	［はい］を選択したタイミングで，現在の関数一覧 パネルの内容をファイルに出力します（関数一覧 パネルが非表示の場合は，最新のトレース・データから取得します）。 以後，ブレークするたびに，最新のトレース・データを取得し，ファイルを出力します。 なお，ファイルは常に上書き保存で出力されます。
		いいえ	STF 用情報ファイルを出力しません。

すべてのパネルを閉じたときにホスト・マシンのメモリを解放する	解析ツールが提供するすべてのパネル（関数一覧 パネル／変数一覧 パネル／解析グラフ パネル／コール・グラフ パネル／クラス／メンバ パネル／値の推移（ズーム）パネル）をクローズした際に，現在解析ツールが使用しているホスト・マシン上のメモリを解放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	解析ツールが提供するすべてのパネルをクローズした際に，ホスト・マシン上のメモリを解放し，CubeSuite+ の他のプラグインの動作を安定させます。 ただし，次回解析ツールのパネルをオープンする際に時間がかかる場合があります。
		いいえ	解析ツールが提供するすべてのパネルをクローズしても，ホスト・マシン上のメモリを解放しません。 次回解析ツールのパネルをオープンする際に時間が短縮されます。
メモリ不足時に解析結果を破棄するかどうかを問い合わせる	解析処理中にメモリ不足エラーが発生した場合，解析結果を破棄してメモリ使用量を減らすか，破棄せずに途中までの解析結果を表示するか否かを選択します。 なお，破棄しない場合は，CubeSuite+ の動作が不安定になる可能性があります。		
	デフォルト	はい	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	メモリ不足エラーが発生した場合，解析結果を破棄するか否かを確認するメッセージ・ダイアログを表示します。
		いいえ	メモリ不足エラーが発生した場合，メッセージ・ダイアログを表示せず，解析結果を破棄します。

注 1. 使用するビルド・ツールのプロパティ パネルにおける次のプロパティ設定

- 【CC-RH】【CX】
  ［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
- 【CC-RX】
  ［コンパイル・オプション］タブ→［その他］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
- 【CA850】
  ［クロス・リファレンス・オプション］タブ→［クロス・リファレンス・ツール］カテゴリ→［クロス・リファレンス・ツールを使用する］プロパティ
- 【CA78K0R】【CA78K0】
  ［コンパイル・オプション］タブ→［リスト・ファイル］カテゴリ→［クロスリファレンス・リスト・ファイルを出力する］プロパティ

2. デバッグ・ツールの次の機能が対象となります（優先度高位順）。

- トレース機能
- リアルタイム表示更新機能（RRM 機能／疑似 RRM（RAM モニタ）機能）
- カバレッジ機能

なお，対応するプロパティの設定は，選択しているマイクロコントローラ，およびデバッグ・ツールにより異なります。詳細は，使用するマイクロコントローラの「CubeSuite+ 統合開発環境 ユーザーズマニュアル デバッグ編」を参照してください。

3. STF 用の情報ファイル（FuncInfo.csv）には，関数一覧 パネル上の内容と同等の情報を出力します（現在非表示に設定している項目の情報も含む）。

(2)［解析対象］

解析対象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，解析対象に関する詳細は，「1. 1. 1　解析対象」を参照してください。

解析対象ファイルの指定方法	解析ツールが解析対象とするファイルを指定する際のファイルの指定方法を選択します。		
	デフォルト	解析対象外ファイル	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	解析対象外ファイル	解析の対象外とするファイルを指定します。
		解析対象ファイル	解析の対象とするファイルを指定します。
解析対象外ファイル	解析ツールが解析の対象外とするファイルを指定します。 なお，このプロパティは，［解析対象ファイルの指定方法］プロパティにおいて［解析対象外ファイル］を指定した場合のみ表示されます。		
	デフォルト	解析対象外ファイル [0]	
	変更方法	解析対象外ファイルを指定 ダイアログによる指定 解析対象外ファイルを指定 ダイアログは，このプロパティを選択すると欄内右端に表示される［...］ボタンをクリックすることでオープンします（プロパティ パネル上で解析対象外ファイルを指定することはできません）。	
解析対象ファイル	解析ツールが解析の対象とするファイルを指定します。 なお，このプロパティは，［解析対象ファイルの指定方法］プロパティにおいて［解析対象ファイル］を指定した場合のみ表示されます。		
	デフォルト	解析対象ファイル [0]	
	変更方法	解析対象ファイルを指定 ダイアログによる指定 解析対象外ファイルを指定 ダイアログは，このプロパティを選択すると欄内右端に表示される［...］ボタンをクリックすることでオープンします（プロパティ パネル上で解析対象外ファイルを指定することはできません）。	

(3)［インポート／エクスポート］

インポート／エクスポート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。

なお，インポート／エクスポート機能に関する詳細は，「2. 12　情報ファイルをインポート／エクスポートする」を参照してください。

<table>
<tr><td>インポート・ファイル</td><td colspan="2">インポートするファイルを指定します。
次のプレースホルダに対応しています。
%ProjectName% ： プロジェクト名に置換します。
%MicomToolPath% ： CubeSuite+ のインストール・フォルダの絶対パスに置換します。
相対パスでの指定はプロジェクト・フォルダを基点とします。
なお，同一ファイルを指定した場合は，最初に指定したファイルのみをインポートします。
サブプロパティして，インポートするファイル名を下段に展開表示します。</td></tr>
<tr><td></td><td>デフォルト</td><td>インポート・ファイル [0]</td></tr>
<tr><td></td><td>変更方法</td><td>パス編集 ダイアログによる指定
パス編集 ダイアログは，このプロパティを選択すると欄内右端に表示される［...］ボタンをクリックすることでオープンします（プロパティ パネル上でインポート・ファイルを指定することはできません）。</td></tr>
<tr><td></td><td>指定可能値</td><td>64 個までのファイル</td></tr>
<tr><td>ビルド時にエクスポートする</td><td colspan="2">ビルド／リビルドを行う際に，関数一覧 パネル／変数一覧 パネルで表示される内容の情報ファイル（関数一覧ファイル（*.mtfl）／変数一覧ファイル（*.mtvl））を生成（エクスポート）するか否かを選択します。</td></tr>
<tr><td></td><td>デフォルト</td><td>いいえ</td></tr>
<tr><td></td><td>変更方法</td><td>ドロップダウン・リストによる選択</td></tr>
<tr><td></td><td rowspan="2">指定可能値</td><td>はい</td><td>情報ファイルを生成します。</td></tr>
<tr><td></td><td>いいえ</td><td>情報ファイルを生成しません。</td></tr>
<tr><td>関数用エクスポート・ファイル名</td><td colspan="2">生成する情報ファイル（関数一覧ファイル（*.mtfl））の名前を指定します。拡張子（*.mtfl）を変更することはできません（拡張子を省略した場合は，自動的に mtfl を付与します）。
このプロパティが空欄の場合，情報ファイルは生成しません。
次のプレースホルダに対応しています。
%ProjectName% ： プロジェクト名に置換します。
%ActiveProjectName% ： アクティブ・プロジェクト名に置換します。
相対パスでの指定はプロジェクト・フォルダを基点とします。
なお，このプロパティは，［ビルド時にエクスポートする］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。</td></tr>
<tr><td></td><td>デフォルト</td><td>%ProjectName%.mtfl</td></tr>
<tr><td></td><td>変更方法</td><td>キーボードからの直接入力</td></tr>
<tr><td></td><td>指定可能値</td><td>259 文字までの文字列</td></tr>
</table>

変数用エクスポート・ファイル名	生成する情報ファイル（変数一覧ファイル（*.mtvl））の名前を指定します。拡張子（*.mtvl）を変更することはできません（拡張子を省略した場合は，自動的に mtvl を付与します）。 このプロパティが空欄の場合，情報ファイルは生成しません。 次のプレースホルダに対応しています。 %ProjectName% ：プロジェクト名に置換します。 %ActiveProjectName% ：アクティブ・プロジェクト名に置換します。 相対パスでの指定はプロジェクト・フォルダを基点とします。 なお，このプロパティは，［ビルド時にエクスポートする］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	%ProjectName%.mtvl
	変更方法	キーボードからの直接入力
	指定可能値	259 文字までの文字列

（4）［解析グラフ］

解析グラフの表示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，解析グラフに関する詳細は，「2. 13 解析情報をグラフ化して表示する」を参照してください。

注意 このカテゴリ内のプロパティ値を変更すると，解析グラフ パネル上で現在表示しているグラフの内容を更新します。

実行時間の割合グラフに表示する関数の数	解析グラフ パネルの［実行時間の割合］タブで表示する円グラフにおいて，表示する関数の数を指定します。 実行時間の割合の大きい順にグラフ化の対象となり，ここで指定した数を越える関数については，“その他”としてまとめて表示します。	
	デフォルト	10
	変更方法	キーボードからの直接入力
	指定可能値	1 ～ 100（10 進数の整数のみ）

［値の推移］タブ

［値の推移］タブでは，解析グラフ パネルの［値の推移］タブで表示するグラフに対して，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）［全般］
（2）［トリガ］
（3）［Smart Analog］【E1/E20【RL78】】
（4）［チャネル 1 ～ 16］

図 A―5　プロパティ パネル：［値の推移］タブ（【E1【RL78】】の例）

[各カテゴリの説明]

(1) [全般]

表示するグラフ全般に関する詳細情報の表示，および設定の変更を行います。

解析方式	表示するグラフ・データの取得方法を選択します（「(2) グラフ・データの取得方法を選択する」参照）。		
	デフォルト	リアルタイム・サンプリング方式	
	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	リアルタイム・サンプリング方式	デバッグ・ツールのRRM機能／疑似RRM（RAMモニタ）機能により取得したデータを基にグラフ表示を行います。 ただし，デバッグ・ツールがデータ収集モードに設定されている場合注1は，Smart Analog用のサンプリング方式によりデータ収集を行います。
		トレース・データ解析方式	デバッグ・ツールのトレース機能により取得したトレース・データを基にグラフ表示を行います。 なお，この項目は，次のいずれかの場合は非表示となります。 - デバッグ・ツールがトレース機能をサポートしていない場合 - デバッグ・ツールがトレース・タイム・タグ機能をサポートしていない場合 - デバッグ・ツールがデータ収集モードに設定されている場合注1
		ファイルから読み込み	保存した解析グラフ・データ・ファイル（*.mtac）を読み込みグラフ表示の復帰を行います。
解析グラフ・データ・ファイル	グラフを復帰するための解析グラフ・データ・ファイル（*.mtac）を指定します。 次のプレースホルダに対応しています。 %ProjectName% ：プロジェクト名に置換します。 %MicomToolPath% ：CubeSuite+ のインストール・フォルダの絶対パスに置換します。 相対パスでの指定はプロジェクト・フォルダを基点とします。 なお，このプロパティは，［解析方式］プロパティにおいて［ファイルから読み込み］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	ファイルを開く ダイアログによる指定，またはキーボードからの直接入力 ファイルを開く 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。	
	指定可能値	1個のファイル（*.mtac）	

<table>
<tr><td rowspan="5">サンプリング開始／停止</td><td colspan="3">リアルタイム・サンプリングの開始／停止をプログラム実行の開始／停止と連動するか否かを選択します。
なお，このプロパティは，［解析方式］プロパティにおいて［リアルタイム・サンプリング方式］を指定した場合のみ表示されます。</td></tr>
<tr><td>デフォルト</td><td colspan="2">連動</td></tr>
<tr><td>変更方法</td><td colspan="2">ドロップダウン・リストによる選択</td></tr>
<tr><td rowspan="2">指定可能値</td><td>連動</td><td>プログラム実行の開始／停止と連動します。</td></tr>
<tr><td>手動</td><td>プログラム実行の開始／停止と連動せず，［値の推移］タブ上の［サンプリング］ボタンにより制御します。</td></tr>
<tr><td rowspan="5">自動調整</td><td colspan="3">グラフ表示の X 軸 /Y 軸において，取得したグラフ・データから最適な値を計算し，［1 グリッドあたりの時間 [Time/Div]］／［1 グリッドあたりの値 [Val/Div] 1 ～ 16］／［オフセット 1 ～ 16］プロパティに設定するか否かを選択します（「(c) 表示範囲の設定」参照）。
【E1/E20【RL78】】
このプロパティは，デバッグ・ツールをデータ収集モードに設定している場合注1は非表示となります。</td></tr>
<tr><td>デフォルト</td><td colspan="2">プログラム停止中のみ行う
ただし，［解析方式］プロパティにおいて［ファイルから読み込み］を指定した場合，または［トリガ機能を使用する］プロパティにおいて［はい］を指定した場合は［行わない］に固定</td></tr>
<tr><td>変更方法</td><td colspan="2">ドロップダウン・リストによる選択</td></tr>
<tr><td rowspan="3">指定可能値</td><td>常に行う</td><td>常にグラフ表示の自動調整を行います。</td></tr>
<tr><td>プログラム停止中のみ行う</td><td>プログラムの実行が停止した際にのみグラフ表示の自動調整を行います。</td></tr>
<tr><td>行わない</td><td>グラフ表示の自動調整を行いません。</td></tr>
<tr><td rowspan="4">自動調整用の遷移箇所の数</td><td colspan="3">グラフ表示を自動調整する際に，描画領域内に表示するグラフの遷移箇所の個数を指定します。
なお，このプロパティは，［自動調整］プロパティにおいて［行わない］を指定した場合は非表示となります。</td></tr>
<tr><td>デフォルト</td><td colspan="2">20</td></tr>
<tr><td>変更方法</td><td colspan="2">キーボードからの直接入力</td></tr>
<tr><td>指定可能値</td><td colspan="2">次の範囲の 10 進数
- 1 ～ 10000</td></tr>
</table>

1 グリッドあたりの時間 [Time/Div]	グラフにおける 10 分割された領域の 1 単位（グリッド）あたりの時間を指定します。 なお，このプロパティは，［自動調整］プロパティにおいて［行わない］を指定した場合のみ有効となります。		
	デフォルト	1ms ただし，［解析方式］プロパティにおいて［ファイルから読み込み］を指定した場合はファイルから読み込んだ値を設定	
	変更方法	キーボードからの直接入力	
	指定可能値	次のいずれか（1ns ～ 10s：10 進数）**注 2** - 1 ～ 10s - 1 ～ 10000ms - 1 ～ 10000000µs - 1 ～ 10000000000ns	
グラフの種類	グラフ形式（グラフの点と点を結ぶ線の形状）を選択します。		
	デフォルト	［解析方式］プロパティの指定に依存します。 -［リアルタイム・サンプリング方式］を指定した場合 折れ線グラフ -［トレース・データ解析方式］を指定した場合 ステップ折れ線グラフ（固定）	
	変更方法	［解析方式］プロパティの指定に依存します。 -［リアルタイム・サンプリング方式］を指定した場合 ドロップダウン・リストによる選択 -［トレース・データ解析方式］を指定した場合 変更不可	
	指定可能値	折れ線グラフ	折れ線グラフで表示します。
		ステップ折れ線グラフ	ステップ・プロット折れ線グラフで表示します。
背景色と前景色を指定する	グラフの背景色と前景色を指定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	背景色と前景色を指定します。
		いいえ	背景色と前景色を指定しません（オプション ダイアログで指定されている［標準］項目の色を使用します）。
背景色	グラフの背景色**注 3** を指定します。 なお，このプロパティは，［背景色と前景色を指定する］プロパティにおいて［はい］を指定した場合のみ表示されます。		
	デフォルト	オプション ダイアログで指定されている［標準］項目の背景色	
	変更方法	色選択用コンボ・ボックスによる指定，またはキーボードからの直接入力	
	指定可能値	キーボードからの直接入力の場合，10 進数 /16 進数（0x 付き）の数値，または色名（「色の指定方法について」参照）	

背景色 (ロスト)	ロスト区間（「(a) グラフ」参照）におけるグラフの背景色[注3] を指定します。 なお，このプロパティは，［背景色と前景色を指定する］プロパティにおいて［はい］を指定した場合のみ表示されます。	
	デフォルト	オプション ダイアログで指定されている［ロスト］の背景色
	変更方法	色選択用コンボ・ボックスによる指定，またはキーボードからの直接入力
	指定可能値	キーボードからの直接入力の場合，10 進数 /16 進数（0x 付き）の数値，または色名（「色の指定方法について」参照）
前景色	グラフの前景色[注3] を指定します。 なお，このプロパティは，［背景色と前景色を指定する］プロパティにおいて［はい］を指定した場合のみ表示されます。	
	デフォルト	オプション ダイアログで指定されている［標準］項目の文字色
	変更方法	色選択用コンボ・ボックスによる指定，またはキーボードからの直接入力
	指定可能値	キーボードからの直接入力の場合，10 進数 /16 進数（0x 付き）の数値，または色名（「色の指定方法について」参照）
カーソル *A* ～ *B* の色	カーソル A，およびカーソル B の色を指定します。	
	デフォルト	カーソル A：PaleGreen カーソル B：PaleTurquoise
	変更方法	色選択用コンボ・ボックスによる指定，またはキーボードからの直接入力
	指定可能値	キーボードからの直接入力の場合，10 進数 /16 進数（0x 付き）の数値，または色名（「色の指定方法について」参照）
ズーム枠 *1* ～ *4* の色	ズーム枠 *1* ～ *4* の色を指定します。	
	デフォルト	ズーム枠 1：64, 255,10,79 ズーム枠 2：64, 91, 228, 22 ズーム枠 3：64, 5, 109, 239 ズーム枠 4：64, 255, 84, 28
	変更方法	色選択用コンボ・ボックスによる指定，またはキーボードからの直接入力
	指定可能値	キーボードからの直接入力の場合，10 進数 /16 進数（0x 付き）の数値，または色名（「色の指定方法について」参照）

注 1. 【E1/E20【RL78】】

この機能は，選択しているマイクロコントローラが Smart Analog IC 搭載品の場合のみサポートしています。

2. 単位入力（大文字／小文字不問）が省略された場合，［解析方式］プロパティにおいて［リアルタイム・サンプリング方式］を選択している場合は“ms”として扱い，それ以外の場合は“ns”として扱います。

なお，このプロパティの値を変更した際に，［トリガ・ポジション］プロパティの値が“本プロパティの値× 10”の値を越えている場合，［トリガ・ポジション］プロパティの値は“本プロパティの値× 10”に自動的に設定されます。

3. ここで指定した色は，値の推移（ズーム）パネル上の背景色／前景色にも反映されます。

(2) [トリガ]

トリガ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（「(3) トリガ機能を使用する」参照）。

トリガ機能を使用する	トリガ機能を使用してグラフ表示を行うか否かを選択します。		
	デフォルト	いいえ	
	変更方法	[解析方式] プロパティの指定に依存します。 - [リアルタイム・サンプリング方式] を指定した場合 ドロップダウン・リストによる選択 - [トレース・データ解析方式] ／ [ファイルから読み込み] を指定した場合 変更不可 ただし，プログラム実行中は変更不可	
	指定可能値	はい	トリガ機能を使用します。
		いいえ	トリガ機能を使用しません。
トリガ・モード	トリガ・モード（グラフの表示更新を行うタイミング）を選択します。 なお，このプロパティは，[トリガ機能を使用する] 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	Auto	
	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	Auto	周期的にグラフを再描画します。トリガ信号発生時にもグラフをクリアして再描画します。
		Single	サンプリング開始から最初のトリガ信号発生時にのみグラフを描画します。
		Normal	トリガ信号発生時にのみグラフを再描画します。
トリガ・ソース	トリガ信号の基となる変数（チャネル）を選択します。 なお，このプロパティは，[トリガ機能を使用する] 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	ch1	
	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	ch1 ～ ch16 のいずれか 1 つ	
トリガ・レベル	トリガ・レベル（トリガ信号の発生と判断するためのしきい値）を指定します。 なお，このプロパティは，[トリガ機能を使用する] 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	0	
	変更方法	キーボードからの直接入力 ただし，プログラム実行中は変更不可	
	指定可能値	次の範囲の 10 進数 /16 進数（浮動小数による入力も可） *“トリガ信号の基となる変数（トリガ・ソース）の最小値”* ～ *“トリガ信号の基となる変数（トリガ・ソース）の最大値”*	

トリガ・エッジの方向	トリガ・エッジの方向を選択します。 なお，このプロパティは，［トリガ機能を使用する］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	立ち上がり	
	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	立ち上がり	トリガ・ソースの値が，トリガ・レベル未満からトリガ・レベル以上に変化した際にトリガ信号を発生します。
		立ち下がり	トリガ・ソースの値が，トリガ・レベルより大きい値からトリガ・レベル以下に変化した際にトリガ信号を発生します。
		両方	上記“立ち上がり”／“立ち下がり”のいずれかの条件でトリガ信号を発生します。
トリガ・ポジション	トリガ・ポジション（トリガ信号が発生した箇所を描画するX軸方向の位置）を指定します。 なお，このプロパティは，［トリガ機能を使用する］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	0s	
	変更方法	キーボードからの直接入力 ただし，プログラム実行中は変更不可	
	指定可能値	次の範囲の10進数値[注] - 0 ～“(［1グリッドあたりの時間 [Time/Div]］プロパティ値× 10”	
トリガ・マークの色	トリガ・マーク（トリガ・レベル，およびトリガ・ポジションを示すマーク）の色を指定します。 なお，このプロパティは，［トリガ機能を使用する］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	Orange	
	変更方法	色選択用コンボ・ボックスによる指定，またはキーボードからの直接入力 ただし，プログラム実行中は変更不可	
	指定可能値	キーボードからの直接入力の場合，10進数/16進数（0x付き）の数値，または色名（「色の指定方法について」参照）	

注 単位（s，ms，us/µs，ns）を省略した場合は，“ms”として扱います（大文字／小文字不問）。
なお，変更した値が，“(［1グリッドあたりの時間 [Time/Div]］プロパティの値) × 10”を越えた場合，このプロパティの値は“(［1グリッドあたりの時間 [Time/Div]］プロパティの値) × 10”に自動的に設定されます。

（3）［Smart Analog］【E1/E20【RL78】】

Smart Analog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 このカテゴリは，選択しているマイクロコントローラがSmart Analog IC搭載品で，デバッグ・ツールをデータ収集モードに設定している場合のみ表示されます。

サンプリング間隔 [ms]	データを収集するサンプリング間隔を ms 単位で指定します。 データ収集モードによるデータ収集開始直前に，特定のシンボル（r_dbg_graph.c#static_e1_waveout_rate）の値をこのプロパティで指定した値に更新してサンプリング間隔を変更します。	
	デフォルト	10
	変更方法	キーボードからの直接入力 ただし，プログラム実行中は変更不可
	指定可能値	1 ～ 1000 までの 10 進数
データ収集チャネル	データを収集するチャネルを選択します。 データ収集モードによるデータ収集開始直前に，特定のシンボル（r_dbg_graph.c#static_e1_waveout_flag）の値をこのプロパティで指定した値に更新してデータを収集するチャネルを指定します。	
	デフォルト	すべて非選択
	変更方法	ドロップダウン・リスト内のチェック・ボックスによる選択 ただし，プログラム実行中は変更不可
	指定可能値	ch1 ～ ch8 （複数選択可）

（4）［チャネル *1* ～ *16*］

各チャネルに登録するグラフ化対象に関する詳細情報の表示，および設定の変更を行います（「（1）グラフ化対象を登録する」参照）。

注意 ［解析方式］プロパティで［ファイルから読み込み］を指定している場合，ファイルから読み込んだ値がこのカテゴリ内のすべてのプロパティに反映され，各プロパティは変更不可となります。

変数名／アドレス *1* ～ *16*	グラフ化対象として登録する変数名／アドレス式を指定します（「（1）グラフ化対象を登録する」参照）。 なお，このプロパティの値を変更すると，このカテゴリ内のすべてのプロパティがデフォルト値に変更されます。 【E1/E20【RL78】】 デバッグ・ツールがデータ収集モードの場合，このプロパティで指定した文字列は，変数名チェック・ボックスにおけるラベルとして表示するのみです（グラフ化対象とはなりません）。	
	デフォルト	空欄
	変更方法	キーボードからの直接入力 ただし，プログラム実行中は変更不可
	指定可能値	2046 文字までの文字列**注 1**

型／サイズ *1* ～ *16*	登録する変数／アドレスの型／サイズを選択します。 【E1/E20【RL78】】 このプロパティは，デバッグ・ツールがデータ収集モードの場合は非表示となります。	
	デフォルト	自動
	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	次のいずれか 1 つ - 自動**注 2** - 符号付き 1 バイト (8 ビット) - 符号付き 2 バイト (16 ビット) - 符号付き 4 バイト (32 ビット) - 符号なし 1 バイト (8 ビット) - 符号なし 2 バイト (16 ビット) - 符号なし 4 バイト (32 ビット)
1 グリッドあたりの値 [Val/Div] *1* ～ *16*	グラフにおける 10 分割された領域の 1 単位（グリッド）あたりの値を指定します。 なお，このプロパティは，［自動調整］プロパティにおいて［行わない］を指定した場合のみ有効となります。	
	デフォルト	(“*変数の最大値*” － “*変数の最小値*”) ÷ 10 ただし，［解析方式］プロパティにおいて［ファイルから読み込み］を指定した場合は，ファイルから読み込んだ値を設定 【E1/E20【RL78】】 デバッグ・ツールがデータ収集モードの場合：409.5
	変更方法	キーボードからの直接入力 ただし，プログラム実行中は変更不可
	指定可能値	次の範囲の 10 進数 /16 進数（変数が浮動小数点の場合，小数の入力可） “*変数の正の最小値*” ～ (“*変数の最大値*” － “*変数の最小値*”)
オフセット *1* ～ *16*	グラフにおけるオフセット値を指定します。 変数値にオフセット値を加算してグラフを描画します。 なお，このプロパティは，［自動調整］プロパティにおいて［行わない］を指定した場合のみ有効となります。	
	デフォルト	0 【E1/E20【RL78】】 デバッグ・ツールがデータ収集モードの場合：-2048
	変更方法	キーボードからの直接入力
	指定可能値	次の範囲（小数の入力可） “Float 型の最小値（約 -3.4028235e+38）” ～ “Float 型の最大値（約 3.4028235e+38）”
色 *1* ～ *16*	グラフの描画色を指定します。	
	デフォルト	*1* ～ *16* 番号に依存**注 3**
	変更方法	色選択用コンボ・ボックスによる指定，またはキーボードからの直接入力
	指定可能値	キーボードからの直接入力の場合，10 進数 /16 進数（0x 付き）の数値，または色名（「色の指定方法について」参照）

注 1. グラフ化対象の入力形式は次のとおりです（ウォッチ パネルの入力形式と同等です）。
ただし，登録の際には，次の注意が必要となります。
- 構造体／共用体／配列の変数名を登録した場合は，グラフを表示できません。構造体／共用体／配列は，メンバ名／要素を指定する必要があります。
- 即値アドレスは，サイズを 1 バイトとみなして登録されます。

入力形式	取得値
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 【RH850】【RX】【V850】	I/O レジスタの値
SFR レジスタ名 【RL78】【78K0R】【78K0】	SFR 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

2. デバッグ・ツールが，アドレス式→レジスタ名→ IOR/SFR 名→変数名の順序で自動的に判断します。なお，アドレス式の指定と判断した場合，“符号付き 1 バイト (8 ビット)”が選択されているものとして扱います。

3. 各番号と色の関係は次のとおりです。

番号	色名	32 ビットの αRGB 値	番号	色名	32 ビットの αRGB 値
1	赤	0xC0FF0A4F	9	黄緑	0xC0BEE02F
2	緑	0xC05BE416	10	青紫	0xC05510FF
3	青	0xC0056DFF	11	桃色	0xC0FF97E4
4	橙	0xC0FF541C	12	茶	0xC0913A37
5	水色	0xC04FC1FF	13	黄土	0xC0C68E15
6	紫	0xC0A932FF	14	深緑	0xC0317F0C
7	黄	0xC0FFD91C	15	こげ茶	0xC060493E
8	赤紫	0xC0FF30A5	16	灰色	0xC072808E

備考 色の指定方法について

次のいずれかを指定することができます。

ただし，透明度を表現する“α”に相当する値を指定した場合は，常に“255(0xff)”（不透明）の指定として扱います。

- 32 ビットの値（上位ビットから 8 ビットずつ αRGB に割り当て）

  例：0xC0FF0A4F

- 24 ビットの値（上位ビットから 8 ビットずつ RGB に割り当て）

  例：0xFF0A4F

- カンマ区切りの 8 ビット× 4 個の値（左から αRGB に割り当て）

  例：192, 255, 10, 79

- カンマ区切りの 8 ビット× 3 個の値（左から RGB に割り当て）

  例：255, 10, 79

- 一般的な色の名前の英語表記（大文字／小文字不問）

  例：Blue

関数一覧 パネル

取得した関数情報を表示します。

このパネルで表示対象となる関数の種類は次のとおりです。

- グローバル関数
- スタティック関数
- メンバ関数（C++ ソース・ファイルを対象とする場合）

ただし，解析対象外に指定されているファイル内の関数情報は表示されません（「1. 1. 1　解析対象」参照）。

なお，関数情報を表示するための操作手順は，「2. 1　概　　要」を参照してください。

注意　【CA850】【CA78K0R】【CA78K0】

ビルド・ツールにおいてクリーンを実行すると，現在このパネルに表示している内容をすべてクリアします。

備考 1.　関数情報は，フィルタを設定して表示することができます（「2. 6. 5　解析情報をフィルタ表示する」参照）。

2.　ツールバーの 100% ，または［Ctrl］キーを押下しながらマウス・ホイールを前後方に動かすことにより，本パネルの表示を拡大／縮小することができます。

図 A—6　関数一覧 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（関数一覧 パネル専用部分）］
- ［［編集］メニュー（関数一覧 パネル専用部分）］
- ［コンテキスト・メニュー］

[オープン方法]

- メイン・ウインドウのツールバーの  ボタンのクリック
- ［表示］メニュー→［プログラム解析］→［関数一覧］の選択

[各エリアの説明]

(1) ヘッダ・エリア

(a) 項目名

取得した関数情報の項目名を表示します。

なお，各項目名に表示されるマーク（アイコン）の意味は次のとおりです。

マーク（アイコン）	意味
△／▽	ソート表示設定の有無を示します（「2.6.4　解析情報をソート表示する」参照）。
▽／▼	フィルタ表示設定の有無を示します（「2.6.5　解析情報をフィルタ表示する」参照）。
📌／📌	固定表示設定の有無を示します（「2.6.3　特定項目を固定表示に設定する」参照）。
	該当項目の情報に関するメッセージを出力 パネルに出力していることを示します。マウス・カーソルを重ねることにより，出力した最新のメッセージをポップアップ表示します。

備考　各項目は，このエリアをマウスで操作することによりカスタマイズすることができます。

- 表示項目を設定する
- 表示項目を並び替える
- 特定項目を固定表示に設定する

(b) ボタン

	このパネルに表示する項目（列）の並び替え，表示／非表示の設定，およびそれらのカスタマイズをデフォルトに戻す設定を行うための列の選択 ダイアログをオープンします（「2.6　表示方法をカスタマイズする」参照）。

(2) 情報表示エリア

取得した関数情報を表示します。

解析ツールが取得する関数情報には，静的解析情報と動的解析情報の２つの種別があり，それぞれ表示可能なタイミングは異なります（「1.1.2　解析情報の種別」参照）。

関数情報を表示した状態で，アクティブ・プロジェクトを変更した場合は，変更先プロジェクトの情報を表示します（ただし，変更先プロジェクトにおいて，クロス・リファレンス情報が生成されていない場合，または解析ツールをサポートしていない場合は，何も表示されません）。

なお，プログラムの実行により値が変化した情報は強調表示されます（強調表示の際の文字色／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）。強調表示を解除する場合は，ツールバーの  ボタンボタンをクリックしてください。

関数情報として取得する項目と内容は次のとおりです。

項目	種別	内容
関数名	静的解析	C/C++ ソース・ファイルで定義／参照されているグローバル関数／ファイル内スタティック関数／メンバ関数（クラス型で定義されている関数）の名前を表示します。 最大表示文字数は次のとおりです。 - 【CC-RH】【CC-RX】 : 1024 文字 - 【CA850】 : 1022 文字 - 【CX】 : 2046 文字 - 【CA78K0R】【CA78K0】 : 249 文字 なお，表示アイコンは次の意味を示します。 - : 関数 - : メンバ関数【CC-RX】 - : テンプレート関数【CC-RX】 【CC-RX】 const メンバ関数 /volatile メンバ関数の場合は，関数名の直後に“const”／“volatile”を表示します。
クラス名 【CC-RX】	静的解析	関数が属しているクラス名を表示します。 テンプレート・クラスの場合は，“*クラス名* <T: *テンプレート引数の数*>”の形式でテンプレート引数の数も表示します。 ただし，関数がメンバ関数以外の場合，または解析未完の場合は空欄となります。
名前空間 【CC-RX】	静的解析	関数が属している名前空間名を表示します。 名前空間がネストしている場合は，“*名前空間名* :: *名前空間名*”の形式で表示します。また，無名名前空間の場合は，“<unnamed>”を表示します。 ただし，解析未完の場合は空欄となります。
ファイル名	静的解析	関数が定義されている C/C++ ソース・ファイル名を表示します（パスを除く）。 ただし，プロジェクト内の C/C++ ソース・ファイルで定義されていない場合，または解析未完の場合は，“（定義箇所なし）”を表示します。 なお，ヘッダ・ファイルで定義されている場合は，該当ヘッダ・ファイル名を表示します。
ファイル・パス[注1]	静的解析	関数が定義されている C/C++ ソース・ファイルの絶対パスを表示します。 ただし，プロジェクト内の C/C++ ソース・ファイルで定義されていない場合，または解析未完の場合は空欄となります。 なお，ヘッダ・ファイルで定義されている場合は，該当ヘッダ・ファイルの絶対パスを表示します。
PM 情報【RH850】 PE 情報【V850E2】	静的解析	関数が実行される PE を次にように表示します。 - PE*n* で実行 : PM*n*/PE*n* - PE 共通で実行 : Common - 不明 : - ただし，この項目は，選択しているマイクロコントローラがマルチコア対応版の場合のみ表示します。

項目	種別	内容
インポート[注1]	静的 解析	関数情報の取得先を次のように表示します。 - アクティブ・プロジェクト内から取得した場合 “Original”を表示 - インポート・ファイルから取得した場合 インポート・ファイル名のすべてを表示 - アクティブ・プロジェクト内とインポート・ファイルから取得した場合 “Original”とインポート・ファイル名のすべてを表示 なお，インポート機能については，「2. 12 情報ファイルをインポート／エクスポートする」を参照してください。
アクセス指定子 【CC-RX】	静的 解析	メンバ関数に指定されているアクセス指定子を表示します。 ただし，解析未完の場合は“-”を表示します。 表示可能なアクセス指定子は次のとおりです。 public，private，protected
属性	静的 解析	関数のシンボル属性／シンボル修飾属性を表示します。 複数の属性が存在する場合は，“,”で区切り表示します。 ただし，解析未完の場合は“-”を表示します。 表示可能な属性は次のとおりです。 -【CC-RH】【CX】 static，interrupt，inline -【CC-RX】 static，interrupt，inline，template，virtual，abstract -【CA850】 static -【CA78K0R】 static，callt，interrupt，near，far，rtos task，rtos interrupt -【CA78K0】 static，callt，callf，noauto，norec，interrupt，bank，rtos task，rtos interrupt
戻り値の型	静的 解析	関数の戻り値の型を表示します[注2]。 ただし，解析未完の場合は“-”を表示します。 表示可能な最大ポインタ数は次のとおりです。 -【CC-RH】【CC-RX】 ：制限なし -【CA850】 ：6 個 -【CX】 ：8 個 -【CA78K0R】【CA78K0】 ：7 個
引数の数[注1]	静的 解析	関数の引数の数を 10 進数で表示します。 可変引数を持つ関数の場合は，関数の定義箇所に定義されている引数の数を表示します。 ただし，解析未完の場合は“-”を表示します。

項目	種別	内容
引数	静的 解析	関数の引数の型と仮引数名を表示します注2。 複数の引数が存在する場合は，“,”で区切り表示します。 可変引数を持つ関数の場合は，関数の定義箇所に定義されている引数の型と引数名を表示します。 引数が存在しない場合は“void”を表示します。 ただし，解析未完の場合は“-”を表示します。 表示可能な最大ポインタ数は次のとおりです（配列の 1 次元目はポインタとして扱います）。 -【CC-RH】【CC-RX】　：制限なし -【CA850】　：6 個 -【CX】　：8 個 -【CA78K0R】【CA78K0】：8 個
コード・サイズ[バイト]	静的 解析	関数のコード・サイズを 10 進数で表示します。 ただし，解析未完の場合は“-”を表示します。
スタック・サイズ[バイト] 【V850】 【RL78】 【78K0R】 【78K0】	静的 解析	関数のスタック・サイズを 10 進数で表示します。 ただし，解析未完の場合は“-”を表示します。 【CA78K0R】【CA78K0】 ここで表示する値は，コンパイラが関数の先頭，または基本ブロック先頭で確保するスタック・サイズを表示するため，スタック見積もりツールが表示するスタック・サイズの値とは異なります。また，関数内の CALL/PUSH/POP 命令が使用するスタック・サイズは含みません。
開始アドレス	静的 解析	関数の開始アドレスを 16 進数で表示します。 表示桁数は選択しているマイクロコントローラの最大アドレス値と同等です。 ただし，解析未完の場合は“-”を表示します。
終了アドレス注1	静的 解析	関数が配置されている ROM 上の終了アドレスを 16 進数で表示します。 表示桁数は選択しているマイクロコントローラの最大アドレス値と同等です。 ただし，解析未完の場合は“-”を表示します。
参照回数	静的 解析	プログラム中で関数が参照されている回数を 10 進数で表示します。 プロトタイプ宣言も参照として計数します。 なお，C/C++ ソース・ファイル中の“#if”／“#ifdef”などで，コンパイル時にプリプロセッサにより除外されるコードにおいて参照されている箇所は参照回数に含みません（コンテキスト・メニューの［すべての参照を検索］による検索結果においても出力しません）。 ただし，解析未完の場合は“-”を表示します。 【CC-RH】【CC-RX】【CX】 関数ポインタへの代入による関数の参照は参照回数に含みません。 【CA850】【CA78K0R】【CA78K0】 関数 A の定義と関数 B の定義の間において，関数 C のプロトタイプ宣言を記述している場合，または関数ポインタへの代入による関数 C の参照を行っている場合，関数 A で参照している関数として関数 C を計数します。 【CA78K0R】【CA78K0】 “#pragma 指令”内に関数名を含んでいる箇所も参照回数に計数します。

項目	種別	内容
実行回数 【Full-spec emulator】注6 【IECUBE】 【IECUBE2】 【E1/E20【RL78】】 【EZ Emulator【RL78】】 【シミュレータ】	動的解析	プログラムを実行した結果，関数が実行された（呼び出された）回数を 10 進数で表示します注3。 なお，関数のラベルが割り振られているアドレスに配置されている命令が実行された際に計数を行うため，関数の途中から測定を実施した場合は，不正な値を表示する場合があります。 ただし，デバッグ・ツールのトレース機能が無効，または解析未完の場合は“-”を表示します。 【E1/E20【RL78】】【EZ Emulator【RL78】】 トレース・データから分岐元アドレスのみ取得できるため，次の条件に従って実行回数を計算します。 このため，サイズが不明の関数の実行回数や割り込み処理からコールされている関数の実行回数などを計測できません。 - 分岐元の命令を逆アセンブルした結果が CALL/CALLT 命令である場合，次のトレース・フレームの分岐元アドレスを含む関数の実行回数を加算
実行時間 [*単位*] 【Full-spec emulator】注6 【IECUBE【V850】】 【IECUBE【RL78】】 【IECUBE【78K0R】】 【IECUBE2】 【シミュレータ】	動的解析	関数の実行時間（子関数を含めない関数本体のコード実行時間）を表示します注3，4。 なお，*単位*は［ツールバー］上の［時間の単位］による選択，またはプロパティパネルの［設定］タブ上の［全般］カテゴリ内［時間の単位］プロパティで変更可能です（時間表示のフォーマットについては，プロパティ パネルの［時間の単位］プロパティを参照してください）。 ただし，デバッグ・ツールのトレース機能が無効，または解析未完の場合は“-”を表示します。
実行時間 (割合)[%] 【Full-spec emulator】注6 【IECUBE【V850】】 【IECUBE【RL78】】 【IECUBE【78K0R】】 【IECUBE2】 【シミュレータ】	動的解析	全体の実行時間（トレース・データが取得できた範囲）に占める対象関数の実行時間の割合を，0.00 ～ 100.00 の範囲で表示します注3，4。 なお，セル内背景色の比率は，割合値を示します。 ただし，デバッグ・ツールのトレース機能が無効，または解析未完の場合は“-”を表示します。
平均実行時間 [*単位*] 【Full-spec emulator】注6 【IECUBE【V850】】 【IECUBE【RL78】】 【IECUBE【78K0R】】 【IECUBE2】 【シミュレータ】	動的解析	関数の平均実行時間（”*実行時間*”÷”*実行回数*”）を表示します注3，4。 計算の結果，ns 以下の数字は小数第 1 位を四捨五入して表示します。 なお，*単位*は［ツールバー］上の［時間の単位］による選択，またはプロパティパネルの［設定］タブ上の［全般］カテゴリ内［時間の単位］プロパティで変更可能です（時間表示のフォーマットについては，プロパティ パネルの［時間の単位］プロパティを参照してください）。 ただし，デバッグ・ツールのトレース機能が無効，または解析未完の場合は“-”を表示します。
コード・カバレッジ [%] 【IECUBE】 【IECUBE2】 【シミュレータ】注7	動的解析	関数のコード・カバレッジ率（C0：”*実行されたアドレス範囲のコードのバイト数*”÷”*関数のコード・サイズ*”× 100）を表示します注5。 なお，セル内背景色の比率は，コード・カバレッジ率を示します。 ただし，デバッグ・ツールのカバレッジ機能【IECUBE】【IECUBE2】【シミュレータ】が無効，または解析未完の場合は“-”を表示します。

注 1. この項目は，デフォルトでは表示されません。表示方法についての詳細は，「2. 6. 1 表示項目を設定する」を参照してください。

2. 【CC-RH】【CC-RX】【CA850】【CA78K0R】【CA78K0】
   “#define 文”または“typedef 文”により型に別名が付与されている場合，別名ではなく，基となる型を表示します。
   【CX】
   “typedef 文”により型に別名が付与されている場合，次の型を表示します。

typedef の基の型	表示する型
- unsigned long - unsigned int	unsigned long
- signed long - signed int - long - int	long
- unsigned short	unsigned short
- signed short - short	short
- unsigned char - char（CX のオプションに“-Xchar=unsigned”を指定している場合）	unsigned char
- signed char - char（CX のオプションに“-Xchar=unsigned”を指定していない場合）	char

3. プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［解析結果を累積する］プロパティにおいて［はい］を選択している場合，プログラム実行ごとの累積による数値を表示します。
4. 実行時間には，コンパイラが用意しているランタイム・ライブラリの実行時間は含まれません。
   また，使用しているデバッグ・ツールの設定として，トレース機能開始前にトレース・メモリをクリアしない設定としている場合，実行時間は不正な値となります。
5. コード・カバレッジ率は，プログラム実行ごとの累積による数値で計算します。
   したがって，コード・カバレッジ率をリセットしたい場合は，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［カバレッジ情報をクリア］を選択したのち，ツールバーの  ボタンをクリックしてください。また，プログラムを修正したのちビルドを実行した結果，各関数の配置アドレスが前回のビルド時の配置アドレスと異なる場合，未実行の関数のコード・カバレッジ率を表示します。
6. 【RH850】
   この測定値と現在デバッグ・マネージャ パネルで選択している PE*n* との関係についての詳細は，「(a) トレース機能」を参照してください。
7. 【RH850】
   この測定値と現在デバッグ・マネージャ パネルで選択している PE*n* との関係についての詳細は，「(c) カバレッジ機能【IECUBE】【IECUBE2】【シミュレータ】」を参照してください。

なお，［関数名］項目の最下段には，［合計値］として，各ファイル単位ごとの次の情報を表示します。
ただし，［ファイル名］項目に“(定義箇所なし)”を表示している場合は，これを 1 つのファイルとして扱います。

図 A―7　［合計値］の表示（関数一覧 パネル）

関数名	ファイル名	コード・サイズ	参照回数	実行回数	実行時間[ns]
合計値	CG_timer.c	412	12	2	10500
合計値	(定義箇所なし)	0	4	1	0
合計値	CG_main.c	114	8	0	0

表示	種別	項目	内容
合計値	―	ファイル名	対象ファイル名
		ファイル・パス[注]	対象ファイル・パス
		コード・サイズ[バイト]	対象ファイル内関数のコード・サイズの合計値
		参照回数	対象ファイル内関数の参照回数の合計値
		実行回数	対象ファイル内関数の実行回数の合計値
		実行時間 [*単位*]	対象ファイル内関数の実行時間の合計値
		実行時間 (割合)[%]	全体の実行時間（トレース・データが取得できた範囲）に占める対象ファイル内関数の実行時間の割合
		コード・カバレッジ [%]	対象ファイル内関数のファイル単位でのコード・カバレッジ率

注　この項目は，デフォルトでは表示されません。表示方法についての詳細は，「2. 6. 1　表示項目を設定する」を参照してください。

注意 1.　【IECUBE【V850】】【IECUBE2】

ステップ実行（ステップ・イン実行／ステップ・オーバー実行）を行った場合，トレース・データのタイム・タグに不正な値が出力されるため，［実行時間 [*単位*]］／［実行時間（割合）[%]］／［平均実行時間 [*単位*]］項目が不正な値となります。

2.　【IECUBE【RL78】】【IECUBE【78K0R】】

プログラム実行時の最初のトレース・データのタイム・タグには“0”が出力されます。

このため，実行→停止，またはステップ実行を繰り返し行った場合，［実行時間 [*単位*]］／［実行時間（割合）[%]］／［平均実行時間 [*単位*]］項目が不正な値となります。

3.　【IECUBE【78K0】】

次の項目は表示されません。

［実行時間 [*単位*]］／［実行時間（割合）[%]］／［平均実行時間 [*単位*]］

4.　【CC-RX】【CX】

最適化により削除された未使用スタティック関数は一覧に表示されません。

5.　【CA78K0R】【CA78K0】

同名のソース・ファイルがプロジェクトに登録されている場合，ビルド・ツールによりクロス・リファレンス情報が上書きされてしまうため，同名のソース・ファイルのうち最後にコンパイルされたソース・ファイルの情報以外を取得することができません。

6.　システム・ライブラリ関数については，次の項目の値を取得することができません。

［戻り値の型］／［引数の数］／［引数］／［コード・サイズ [バイト]］／［スタック・サイズ [

バイト]］／［終了アドレス］／［実行時間 [*単位*]］／［実行時間（割合）[%]］／［平均実行時間 [*単位*]］／［コード・カバレッジ [%]］

7. **デバッグ・ツールと切断時は，動的解析情報の項目は非表示となります（デフォルト）。**

備考 1. ヘッダ・ファイル内に“static”の関数定義を記述して，1 つ以上のソース・ファイルでインクルードしている場合は次のように表示します。

- 【CC-RH】【CC-RX】【CX】【CA78K0R】【CA78K0】

  ヘッダ・ファイルとすべてのソース・ファイルの情報を 1 行にまとめて表示します。

- 【CA850】

  ヘッダ・ファイルの情報を 1 行に，すべてのソース・ファイルの情報を 1 行にまとめて表示します。

2. 【CC-RX】

テンプレート関数の定義用の行と，テンプレート関数を使用している関数の行に表示する値は次のとおりです。

```
//
// テンプレート関数を使用する関数
//
int templatet_use(void)
{
    short   result = 0;
    short   s = 100;
    char    c = 200;

    result += template_func(s, c);

    return result;
}

//
// テンプレート関数の定義
//
template <typename T1, typename T2> T1 template_func(T1 t1, T2 t2)
{
    T1      result = 10;

    result  += t1 + t2;

    return  result;
}
```

例 1. テンプレート関数の定義用の行に表示する内容

［戻り値の型］ : -

［引数］ : -

［関数名］ : Template_func<T:2>

2. テンプレート関数を使用している関数の行に表示する内容

［戻り値の型］ : short

［引数］　：　short t1, char t2

［関数名］　：　Template_func<T:2>

3. 各情報の表示は，次のカスタマイズを行うことができます。
   - 解析情報をソート表示する
   - 解析情報をフィルタ表示する
4. このエリア左端のカレント行マーク（▶）は，該当行がカレント行であることを示します。
   カレント行に対しては，次の操作を行うことができます。
   - 定義箇所へジャンプする
   - ブレーク・イベントを設定する
   - 参照箇所を一覧表示する

［ツールバー］

ツールバー上の各ボタン，および機能は次のとおりです。

	最新情報を取得し，表示内容を更新します。
	プログラムの実行が停止するごとに最新情報を取得し，表示内容を更新します。 ただし，プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［プログラム停止時に更新を行う］プロパティにおいて［個別に指定する］以外を指定している場合，このボタンは無効となります（プロパティ パネルでの設定を反映した状態で固定されます）。
	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解除します。 ただし，プログラム実行中は無効となります。
	プロジェクト・ツリー パネルで現在選択しているファイル／カテゴリ内の関数の情報値のみを表示します（「(a) プロジェクト・ツリー パネルとの連携」参照）。
	エディタ パネル上のキャレット位置の単語で始まる関数の情報値のみを表示します（「(b) エディタ パネルとの連携」参照）。
 【RH850】 【V850E2】	デバッグ・マネージャ パネルで選択している PE*n*，および共通領域内の関数の情報値のみを表示します（「(c) デバッグ・マネージャ パネルとの連携【RH850】【V850E2】」参照）。 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対応版でない場合，またはデバッグ・ツールと切断時は無効となります。
時間の単位	時間表示の単位を設定するために，次のカスケード・メニューを表示します。 デフォルトでは，プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［時間の単位］プロパティにおける設定が選択されます。 なお，ここでの設定変更は，プロパティ パネルでの設定に反映されます。
時：分：秒表示	時間：分（0 ～ 59）：秒（0 ～ 59）で表示します。
秒表示	秒単位で表示します（小数 3 桁表示）。
ミリ秒表示	ミリ秒単位で表示します（小数 3 桁表示）。
マイクロ秒表示	マイクロ秒単位で表示します（小数 3 桁表示）。
ナノ秒表示	ナノ秒単位で表示します（整数表示）。

[[ファイル]メニュー（関数一覧 パネル専用部分）]

関数一覧 パネル専用の［ファイル］メニューの各項目，および機能は次のとおりです。

関数一覧データ を保存	このパネルの内容を前回保存したファイルに保存します（「2. 14 解析情報をファイルに保存する」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 関数一覧データ を保存 ...］の選択と同等の動作となります。
名前を付けて 関数一覧データ を保存 ...	このパネルの内容を指定したファイルに保存するために，名前を付けて保存 ダイアログをオープンします（「2. 14 解析情報をファイルに保存する」参照）。
印刷 ...	現在このパネルに表示している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。

[[編集]メニュー（関数一覧 パネル専用部分）]

関数一覧 パネル専用の［編集］メニューの各項目，および機能は次のとおりです。

コピー	選択している行の内容をタブ区切りの文字列としてクリップ・ボードにコピーします（複数行選択可）。 なお，コピーした内容をこのパネル上に貼り付け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行を選択状態にします。

[コンテキスト・メニュー]

このパネル上において，マウスを右クリックすることにより表示されるコンテキスト・メニューの各項目，および機能は次のとおりです。

すべての参照を検索	選択している行（複数行選択可）の関数を参照している箇所を検索し，出力 パネルの［参照の検索］タブ上に参照箇所一覧を表示します（「2. 11 参照箇所を一覧表示する」参照）。
関数の先頭にブレークを設定	カレント行の関数の先頭行（対象関数の先頭実行可能行）にブレークポイントを設定します（「2. 9. 1 関数にブレークポイントを設定する」参照）。 ただし，デバッグ・ツールと切断時は無効となります。
ソースへジャンプ	カレント行の関数が定義されているソース・ファイルをエディタ パネル上にオープンします（「2. 7 定義箇所へジャンプする」参照）。
逆アセンブルへジャンプ	カレント行の関数の開始アドレスに対応する逆アセンブル・データを逆アセンブル パネル（逆アセンブル 1）上にオープンします（「2. 7 定義箇所へジャンプする」参照）。 ただし，デバッグ・ツールと切断時は無効となります。
メモリへジャンプ	カレント行の関数の開始アドレスに対応するメモリ・リストをメモリ パネル（メモリ 1）上にオープンします（「2. 7 定義箇所へジャンプする」参照）。 ただし，デバッグ・ツールと切断時は無効となります。
コピー	選択している行（複数行選択可）の内容をタブ区切りの文字列としてクリップ・ボードにコピーします。 なお，コピーした内容をこのパネル上に貼り付けることはできません。

変数一覧 パネル

取得した変数情報を表示します。

このパネルで表示対象となる変数の種類は次のとおりです。

- グローバル変数
- ファイル内スタティック変数
- 関数内スタティック変数
- IOR【V850】
- SFR【RL78】【78K0R】【78K0】
- クラス変数（C++ ソース・ファイルを対象とする場合）

ただし，解析対象外に指定されているファイル内の変数情報は表示されません（「1. 1. 1　解析対象」参照）。

なお，変数情報を表示するための操作手順は，「2. 1　概　　要」を参照してください。

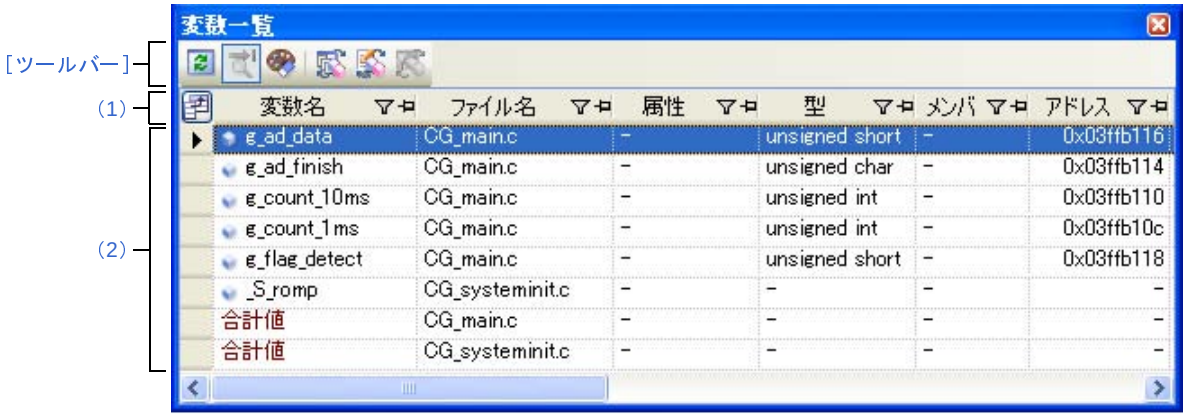
注意　【CA850】【CA78K0R】【CA78K0】

ビルド・ツールにおいてクリーンを実行すると，現在このパネルに表示している内容をすべてクリアします。

備考 1.　変数情報は，フィルタを設定して表示することができます（「2. 6. 5　解析情報をフィルタ表示する」参照）。

2.　ツールバーの 100% ，または［Ctrl］キーを押下しながらマウス・ホイールを前後方に動かすことにより，本パネルの表示を拡大／縮小することができます。

図 A―8　変数一覧 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（変数一覧 パネル専用部分）］
- ［［編集］メニュー（変数一覧 パネル専用部分）］
- ［コンテキスト・メニュー］

[オープン方法]

- メイン・ウインドウのツールバーの  ボタンのクリック
- ［表示］メニュー→［プログラム解析］→［変数一覧］の選択

[各エリアの説明]

(1) ヘッダ・エリア

(a) 項目名

取得した変数情報の項目名を表示します。

なお，各項目名に表示されるマーク（アイコン）の意味は次のとおりです。

マーク（アイコン）	意味
△／▽	ソート表示設定の有無を示します（「2. 6. 4　解析情報をソート表示する」参照）。
／	フィルタ表示設定の有無を示します（「2. 6. 5　解析情報をフィルタ表示する」参照）。
／	固定表示設定の有無を示します（「2. 6. 3　特定項目を固定表示に設定する」参照）。
	該当項目の情報に関するメッセージを出力 パネルに出力していることを示します。マウス・カーソルを重ねることにより，出力した最新のメッセージをポップアップ表示します。

備考　表示項目は，このエリアをマウスで操作することにより，次のカスタマイズを行うことができます。

- 表示項目を設定する
- 表示項目を並び替える
- 特定項目を固定表示に設定する

(b) ボタン

	このパネルに表示する項目（列）の並び替え，表示／非表示の設定，およびそれらのカスタマイズをデフォルトに戻す設定を行うための列の選択 ダイアログをオープンします（「2. 6　表示方法をカスタマイズする」参照）。

(2) 情報表示エリア

取得した変数情報を表示します。

解析ツールが取得する変数情報には，静的解析情報と動的解析情報の２つの種別があり，それぞれ表示可能なタイミングは異なります（「1. 1. 2　解析情報の種別」参照）。

変数情報を表示した状態で，アクティブ・プロジェクトを変更した場合は，変更先プロジェクトの情報を表示します（ただし，変更先プロジェクトにおいて，クロス・リファレンス情報が生成されていない場合，または解析ツールをサポートしていない場合は，何も表示されません）。

なお，プログラムの実行により値が変化した情報は強調表示されます（強調表示の際の文字色／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）。強調表示を解除する場合は，ツールバーの  ボタンボタンをクリックしてください。

変数情報として取得する項目と内容は次のとおりです。

項目	種別	内容
変数名	静的解析	C/C++ ソース・ファイルで定義／参照されているグローバル変数／ファイル内スタティック変数／関数内スタティック変数／クラス変数（クラス型の静的なメンバ変数）の名前を表示します。 また，プロパティ パネルの［設定］タブの［全般］カテゴリ内［SFR ／ IOR を変数として表示する］プロパティにおいて［はい］を選択している場合，プログラム中で使用している SFR/IOR 名を表示します（【CC-RH】【CC-RX】を除く）。 最大表示文字数は次のとおりです。 - 【CC-RH】【CC-RX】　：1024 文字 - 【CA850】　：1022 文字 - 【CX】　：2046 文字 - 【CA78K0R】【CA78K0】：249 文字 なお，表示アイコンは次の意味を示します。 -  ：変数 -  ：メンバ変数【CC-RX】 -  ：IOR【V850】/SFR【RL78】【78K0R】【78K0】 【CC-RX】 無名共用体を使用している場合，“<unnamed_*N*>”の形式で変数名を表示します（*N*：1 から出現順に自動的に付与される番号）。
クラス名 【CC-RX】	静的解析	変数が属しているクラス名を表示します。 なお，テンプレート・クラスの場合は，“*クラス名* <T: *テンプレート引数の数*>”の形式でテンプレート引数の数も表示します。 ただし，変数がメンバ変数以外の場合，または解析未完の場合は空欄となります。
名前空間 【CC-RX】	静的解析	変数が属している名前空間名を表示します。 名前空間がネストしている場合は，“*名前空間名* :: *名前空間名*”の形式で表示します。また，無名名前空間の場合は，“<unnamed>”を表示します。 ただし，解析未完の場合は空欄となります。
ファイル名	静的解析	変数が定義されている C/C++ ソース・ファイル名を表示します（パスを除く）。 ただし，プロジェクト内の C/C++ ソース・ファイルで定義されていない場合，または解析未完の場合は，“（定義箇所なし）”を表示します。 なお，ヘッダ・ファイルで定義されている場合は，該当ヘッダ・ファイル名を表示します。 【CA78K0R】【CA78K0】 同一ファイル内で，extern 宣言を記述している行と変数の定義行が存在する場合，変数の定義情報を取得することはできません。このため，この場合は“（定義箇所なし）”と表示します。

項目	種別	内容
関数名[注1]	静的 解析	変数が定義されている関数の関数名を表示します。 だたし，関数内スタティック変数以外の場合は空欄となります。 【CC-RX】 上記に加え，“()”内に関数の引数の型を列挙します。 また，const メンバ関数 /volatile メンバ関数の場合は，関数名の直後に“const”／“volatile”を表示します。
ファイル・パス[注1]	静的 解析	変数が定義されている C/C++ ソース・ファイルの絶対パスを表示します。 ただし，プロジェクト内の C/C++ ソース・ファイルで定義されていない場合，または解析未完の場合は空欄となります。 なお，ヘッダ・ファイルで定義されている場合は，該当ヘッダ・ファイルの絶対パスを表示します。
PM 情報【RH850】 PE 情報【V850E2】	静的 解析	変数に対してアクセス可能な PE を次にように表示します。 - PE*n* からアクセス : PM*n*/PE*n* - PE でアクセス : Common - 不明 : - ただし，この項目は，選択しているマイクロコントローラがマルチコア対応版の場合のみ表示します。
インポート[注1]	静的 解析	変数情報の取得先を次のように表示します。 - アクティブ・プロジェクト内から取得した場合 “Original”を表示 - インポート・ファイルから取得した場合 インポート・ファイル名のすべてを表示 - アクティブ・プロジェクト内とインポート・ファイルから取得した場合 “Original”とインポート・ファイル名のすべてを表示 なお，インポート機能については，「2. 12 情報ファイルをインポート／エクスポートする」を参照してください。
アクセス指定子 【CC-RX】	静的 解析	メンバ変数に指定されているアクセス指定子を表示します。 ただし，解析未完の場合は“-”を表示します。 表示可能なアクセス指定子は次のとおりです。 public，private，protected
属性	静的 解析	変数のシンボル属性／シンボル修飾属性を表示します。 複数の属性が存在する場合は，“,”で区切り表示します。 ただし，解析未完の場合は“-”を表示します。 表示可能な属性は次のとおりです。 - 【CC-RH】【CX】 static，ior，const，volatile - 【CC-RX】 static，const，volatile，restrict - 【CA850】 static - 【CA78K0R】 static，const，volatile，sreg，rwsfr，rosfr，wosfr，near，far - 【CA78K0】 static，const，volatile，sreg，rwsfr，rosfr，wosfr

項目	種別	内容
型	静的 解析	変数の型を表示します注2。 なお，“#define 文”／“typedef 文”により型に別名が付与されている場合，別名ではなく，基となる型を表示します。 ただし，または解析未完の場合は“-”を表示します。 表示可能な最大ポインタ数は次のとおりです（配列は最大 4 次元まで表示可）。 - 【CC-RH】【CC-RX】 ：制限なし - 【CA850】 ：6 個 - 【CX】 ：8 個 - 【CA78K0R】【CA78K0】 ：8 個
メンバ	静的 解析	構造体／共用体のメンバを表示します。 複数のメンバが存在する場合は，“,”で区切り表示します。 ただし，構造体／共用体以外の場合，または解析未完の場合は“-”を表示します。
アドレス	静的 解析	変数の配置アドレスを 16 進数で表示します。 表示桁数は選択しているマイクロコントローラの最大アドレス値と同等です。 ただし，解析未完の場合は“-”を表示します。
サイズ [バイト]	静的 解析	変数のサイズを 10 進数で表示します。 ただし，サイズがバイトで表示できないビット変数などの場合，または解析未完の場合は“-”を表示します。 【CC-RX(V1.xx.xx)】 定義のみで参照されていない変数は，コンパイラの最適化によりサイズ情報が削除されるため“0”が表示されます。
参照回数	静的 解析	プログラム中で変数が参照されている回数を 10 進数で表示します。 変数の定義箇所も参照として計数します。また，構造体／共有体／配列の場合は，変数単位で計数します（メンバごとや要素ごとの参照回数は非表示）。 なお，C/C++ ソース・ファイル中の“#if”／“#ifdef”などで，コンパイル時にプリプロセッサにより除外されるコードにおいて参照されている箇所は参照回数に含みません（コンテキスト・メニューの［すべての参照を検索］による検索結果においても出力しません）。 ただし，解析未完の場合は“-”を表示します。 【CC-RH】【CC-RX】【CX】 変数定義行において，代入文が記述されている行（“int variable = 10”など）も参照回数として計数します。 また，“variable++;”を記述している行は“variable = variable + 1”と解釈するため，参照回数は 2 回と計数します。
リード回数 【Full-spec emulator】注6 【IECUBE】 【IECUBE2】 【シミュレータ】	動的 解析	プログラムを実行した結果，変数がリードされた回数を 10 進数で表示します注3，4。 構造体／共用体の場合は，構造体／共用体の変数単位でリード回数を計数します（メンバごとや要素ごとのリード回数は非表示）。 ただし，デバッグ・ツールのトレース機能が無効，または解析未完の場合は“-”を表示します。

項目	種別	内容
ライト回数 【Full-spec emulator】注6 【IECUBE】 【IECUBE2】 【シミュレータ】	動的 解析	プログラムを実行した結果，変数がライトされた回数を10進数で表示します注3，4。 構造体／共用体の場合は，構造体／共用体の変数単位でライト回数を計数します（メンバごとや要素ごとのライト回数は非表示）。 ただし，デバッグ・ツールのトレース機能が無効，または解析未完の場合は“-”を表示します。
リード／ライト回数注1 【Full-spec emulator】注6 【IECUBE】 【IECUBE2】 【シミュレータ】	動的 解析	プログラムを実行した結果，変数がリード／ライトされた回数を10進数で表示します注3，4。 構造体／共用体の場合は，構造体／共用体の変数単位でリード／ライト回数を計数します（メンバごとや要素ごとのリード／ライト回数は非表示）。 なお，コンパイラの最適化により変数がレジスタに割り当てられている区間では，変数に対するリード／ライトが解析不能であるため，この区間のリード／ライトは計数されません。 ただし，デバッグ・ツールのトレース機能が無効，または解析未完の場合は“-”を表示します。
最小値 【Full-spec emulator】注6 【IECUBE】 【IECUBE2】 【シミュレータ】	動的 解析	プログラムを実行した結果，計測時間内での最小値を10進数で表示します注3。 ただし，bit型 /boolean型 /Bool型／構造体／共用体／配列／ポインタの場合，デバッグ・ツールのトレース機能が無効，または解析未完の場合は“-”を表示します。 【CC-RH】【CC-RX】【CA850】【CX】 4バイト以下の変数，またはIORのみ表示可能です。 【CA78K0R】【CA78K0】 2バイト以下の変数，またはSFRのみ表示可能です。
最大値 【Full-spec emulator】注6 【IECUBE】 【IECUBE2】 【シミュレータ】	動的 解析	プログラムを実行した結果，計測時間内での最大値を10進数で表示します注3。 ただし，bit型 /boolean型 /Bool型／構造体／共用体／配列／ポインタの場合，デバッグ・ツールのトレース機能が無効，または解析未完の場合は“-”を表示します。 【CC-RH】【CC-RX】【CA850】【CX】 4バイト以下の変数，またはIORのみ表示可能です。 【CA78K0R】【CA78K0】 2バイト以下の変数，またはSFRのみ表示可能です。
データ・カバレッジ [%] 【IECUBE【RL78】】 【IECUBE【78K0R】】 【IECUBE【78K0】】 【シミュレータ】注7	動的 解析	変数のデータ・カバレッジ率（”*アクセスされたアドレス範囲のバイト数*”÷”*変数のサイズ*”×100）を表示します注5。 なお，セル内背景色の比率は，データ・カバレッジ率を示します。 ただし，デバッグ・ツールのカバレッジ機能【IECUBE】【IECUBE2】【シミュレータ】が無効，または解析未完の場合は“-”を表示します。

注1. この項目は，デフォルトでは表示されません。表示方法についての詳細は，「2.6.1　表示項目を設定する」を参照してください。

2. 【CC-RH】【CC-RX】【CA850】【CA78K0R】【CA78K0】

“#define 文”または“typedef 文”により型に別名が付与されている場合，別名ではなく，基となる型を表示します。

【CX】

“typedef 文”により型に別名が付与されている場合，次の型を表示します。

typedef の基の型	表示する型
- unsigned long - unsigned int	unsigned long
- signed long - signed int - long - int	long
- unsigned short	unsigned short
- signed short - short	short
- unsigned char - char（CX のオプションに“-Xchar=unsigned”を指定している場合）	unsigned char
- signed char - char（CX のオプションに“-Xchar=unsigned”を指定していない場合）	signed char

3. プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［解析結果を累積する］プロパティにおいて［はい］を選択している場合，プログラム実行ごとの累積による数値を表示します。
4. 取得したトレース・データを基に計数するため，たとえば，4 バイト領域への 1 回のライトが，トレース・データ上では上位 2 バイトと下位 2 バイトで出力されている場合では，2 回と表示します。
5. データ・カバレッジ率は，プログラム実行ごとの累積値により計算します。
   データ・カバレッジ率をリセットする場合は，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［カバレッジ情報をクリア］を選択したのち，ツールバーの  ボタンをクリックしてください。
   また，プログラムを修正したのちビルドを実行した結果，各変数の配置アドレスが前回のビルド時の配置アドレスと異なる場合，アクセスされていない変数のデータ・カバレッジ率を表示します。
6. 【RH850】
   この測定値と現在デバッグ・マネージャ パネルで選択している PE*n* についての関係は，「(a) トレース機能」を参照してください。
7. 【RH850】
   この測定値と現在デバッグ・マネージャ パネルで選択している PE*n* についての関係は，「(c) カバレッジ機能【IECUBE】【IECUBE2】【シミュレータ】」を参照してください。

なお，［変数名］項目の最下段には，［合計値］として，各ファイル単位ごとの次の情報を表示します。

ただし，［ファイル名］項目に“(定義箇所なし)”を表示している場合は，これを 1 つのファイルとして扱います。

図 A—9　［合計値］の表示（変数一覧 パネル）

変数名	ファイル名	サイズ[バイト]	参照回数	リード回数	ライト回数
合計値	CG_main.c	13	9	0	4
合計値	CG_systeminit.c	0	1	0	0

表示	種別	項目	内容
合計値	―	ファイル名	対象ファイル名
		ファイル・パス[注]	対象ファイル・パス
		サイズ [バイト]	対象ファイル内変数のサイズの合計値
		参照回数	対象ファイル内変数の参照回数の合計値
		リード回数	対象ファイル内変数からのリード回数の合計値
		ライト回数	対象ファイル内変数へのライト回数の合計値
		リード／ライト回数	対象ファイル内変数へのリード／ライト回数の合計値
		データ・カバレッジ (%)	対象ファイル内変数のファイル単位でのデータ・カバレッジ率

注 この項目は，デフォルトでは表示されません。表示方法についての詳細は，「2. 6. 1 表示項目を設定する」を参照してください。

注意 1. bit 型変数 /boolean 型 /_Bool 型変数／構造体ビット・フィールドに対する［リード回数］／［ライト回数］／［リード／ライト回数］／［データ・カバレッジ [%]］項目の計測は，バイト単位で行います（変数が割り当てられているアドレスへのアクセスを計測）。このため，同一アドレスに割り当てられている bit 型変数 /boolean 型 /_Bool 型変数／構造体ビット・フィールドに対するこれらの項目の値は同じ値となります。

2. 【CC-RX】【CX】
最適化により削除された未使用スタティック変数は一覧に表示されません。

3. 【CA850】
“#pragma asm ～ #pragma endasm”を使用してアセンブラ命令を記述している場合，この箇所に記述されたレジスタ，および命令を変数として扱い表示します。

4. 【CA78K0R】【CA78K0】
同名のソース・ファイルがプロジェクトに登録されている場合，ビルド・ツールによりクロス・リファレンス情報が上書きされてしまうため，同名のソース・ファイルのうち最後にコンパイルされたソース・ファイルの情報以外を取得することができません。

5. デバッグ・ツールと切断時は，動的解析情報の項目は非表示となります（デフォルト）。

備考 1. 同一関数内に同名の関数内スタティック変数が宣言されている場合の扱いは次のとおりです。

【CC-RH】【CC-RX】

- ［メンバ］／［アドレス］／［サイズ [バイト]］／［リード回数］／［ライト回数］／［リード／ライト回数］／［最小値］／［最大値］／［データ・カバレッジ [%]］
  関数内で最初に宣言された変数の情報を表示します。
- 上記以外の項目
  関数内で最後に宣言された変数の情報を表示します。

【CX】

- ［型］／［メンバ］／［アドレス］／［サイズ [バイト]］／［リード回数］／［ライト回数］／［リード／ライト回数］／［最小値］／［最大値］／［データ・カバレッジ [%]］
  関数内で最初に宣言された変数の情報を表示します。

- 上記以外の項目

  関数内で最後に宣言された変数の情報を表示します。

【CA850】【CA78K0R】【CA78K0】

- ［変数名］／［ファイル名］／［関数名］／［ファイル・パス］／［インポート］／［属性］

  関数内で最初に宣言された変数の情報を表示します。

- ［参照回数］

  関数内で宣言された変数のすべての参照回数を表示します。

- 上記以外の項目

  関数内で最後に宣言された変数の情報を表示します。

2. ヘッダ・ファイル内に“static”の変数定義を記述して，1 つ以上のソース・ファイルでインクルードしている場合は次のように表示します。

   - 【CC-RH】【CC-RX】【CX】【CA78K0R】【CA78K0】

     ヘッダ・ファイルとすべてのソース・ファイルの情報を 1 行にまとめて表示します。

     また，型情報は“-”と表示します（【CC-RH】【CC-RX】を除く）。

   - 【CA850】

     ヘッダ・ファイルの情報を 1 行に，すべてのソース・ファイルの情報を 1 行にまとめて表示します。

3. 各情報の表示は，次のカスタマイズを行うことができます。

   - 解析情報をソート表示する
   - 解析情報をフィルタ表示する

4. このエリア左端のカレント行マーク（▶）は，該当行がカレント行であることを示します。

   カレント行に対しては，次の操作を行うことができます。

   - 定義箇所へジャンプする
   - ウォッチ式に登録する
   - 参照箇所を一覧表示する
   - 値の推移をグラフ化する

［ツールバー］

ツールバー上の各ボタン，および機能は次のとおりです。

	最新情報を取得し，表示内容を更新します。
	プログラムの実行が停止するごとに最新情報を取得し，表示内容を更新します。 ただし，プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［プログラム停止時に更新を行う］プロパティにおいて［個別に指定する］以外を指定している場合，このボタンは無効となります（プロパティ パネルでの設定を反映した状態で固定されます）。
	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解除します。 ただし，プログラム実行中は無効となります。
	プロジェクト・ツリー パネルで現在選択しているファイル／カテゴリ内の変数の情報値のみを表示します（「(a) プロジェクト・ツリー パネルとの連携」参照）。
	エディタ パネル上のキャレット位置の単語で始まる変数の情報値のみを表示します（「(b) エディタ パネルとの連携」参照）。

 【RH850】 【V850E2】	デバッグ・マネージャ パネルで選択している PE*n*，および共通領域内の変数の情報値のみを表示します（「(c) デバッグ・マネージャ パネルとの連携【RH850】【V850E2】」参照）。 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対応版でない場合，またはデバッグ・ツールと切断時は無効となります。

[[ファイル] メニュー（変数一覧 パネル専用部分）]

変数一覧 パネル専用の［ファイル］メニューの各項目，および機能は次のとおりです。

変数一覧データ を保存	このパネルの内容を前回保存したファイルに保存します（「2. 14　解析情報をファイルに保存する」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 変数一覧データ を保存 ...］の選択と同等の動作となります。
名前を付けて 変数一覧データ を保存 ...	このパネルの内容を指定したファイルに保存するために，名前を付けて保存 ダイアログをオープンします（「2. 14　解析情報をファイルに保存する」参照）。
印刷 ...	現在このパネルに表示している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。

[[編集] メニュー（変数一覧 パネル専用部分）]

変数一覧 パネル専用の［編集］メニューの各項目，および機能は次のとおりです。

コピー	選択している行の内容をタブ区切りの文字列としてクリップ・ボードにコピーします（複数行選択可）。 なお，コピーした内容をこのパネル上に貼り付け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行を選択状態にします。

[コンテキスト・メニュー]

このパネル上において，マウスを右クリックすることにより表示されるコンテキスト・メニューの各項目，および機能は次のとおりです。

すべての参照を検索	選択している行（複数行選択可）の変数を参照している箇所を検索し，出力 パネルの［参照の検索］タブ上に参照箇所一覧を表示します（「2. 11　参照箇所を一覧表示する」参照）。
解析グラフに登録	選択している行（複数行選択可）の変数を解析グラフ パネルに登録します（「2. 13. 1　値の推移をグラフ化する」参照）。 ただし，デバッグ・ツールと切断時は無効となります。
アクセス・ブレークの設定	ブレーク関連のイベント[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. 9. 2　変数にブレーク・イベントを設定する」参照）。 ただし，デバッグ・ツールと切断時は無効となります。
変数に読み込みブレークを設定	カレント行の変数に，リード・アクセスのブレーク・イベントを設定します。
変数に書き込みブレークを設定	カレント行の変数に，ライト・アクセスのブレーク・イベントを設定します。
変数に読み書きブレークを設定	カレント行の変数に，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。

ウォッチ 1 に登録	選択している行（複数行選択可）の変数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（「2. 10　ウォッチ式に登録する」参照）。 ただし，デバッグ・ツールと切断時は無効となります。
ソースへジャンプ	カレント行の変数が定義されているソース・ファイルをエディタ パネル上にオープンします（「2. 7　定義箇所へジャンプする」参照）。
メモリへジャンプ	カレント行の変数の開始アドレスに対応するメモリ・リストをメモリ パネル（メモリ 1）上にオープンします（「2. 7　定義箇所へジャンプする」参照）。 ただし，デバッグ・ツールと切断時は無効となります。
コピー	選択している行の内容をタブ区切りの文字列としてクリップ・ボードにコピーします（複数行選択可）。 なお，コピーした内容をこのパネル上に貼り付けることはできません。

注 【RX】

組み合わせブレークの場合，組み合わせ条件が“OR”の場合のみ有効となります。

解析グラフ パネル

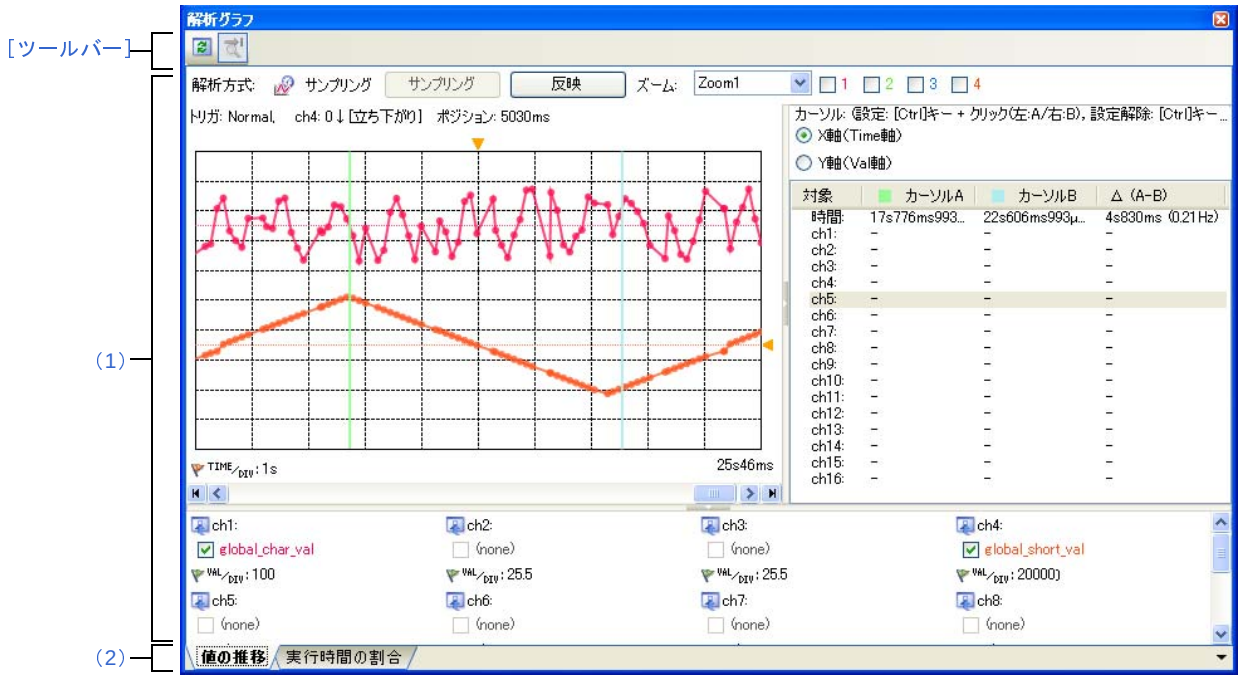
取得した関数情報／変数情報（Smart Analog 用のデータ情報[注]を含む）をグラフで表示します。

なお，グラフを表示するための操作手順は，「2. 13　解析情報をグラフ化して表示する」を参照してください。

注　【E1/E20【RL78】】

選択しているマイクロコントローラが Smart Analog IC 搭載品の場合のみサポートしている機能です。

図 A―10　解析グラフ 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（解析グラフ パネル専用部分）］

［オープン方法］

- メイン・ウインドウのツールバーの ボタンのクリック
- ［表示］メニュー→［プログラム解析］→［解析グラフ］の選択

[各エリアの説明]

(1) グラフ表示エリア

取得した関数情報／変数情報に関するグラフ，およびその情報を表示します。

(2) タブ選択エリア

タブを選択することにより，表示するグラフの種別が切り替わります。

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）。

- ［値の推移］タブ
- ［実行時間の割合］タブ

備考　グラフ情報に関するメッセージが出力 パネルに出力されている場合，該当タブに  マークを表示します。

[ツールバー]

ツールバー上の各ボタン，および機能は次のとおりです。

	最新情報を取得し，表示内容を更新します。 ただし，変数が１つも登録されていない場合は無効となります。
	プログラムの実行が停止するごとに最新情報を取得し，表示内容を更新します。 ただし，プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［プログラム停止時に更新を行う］プロパティにおいて［個別に指定する］以外を指定している場合は無効となります（プロパティ パネルでの設定を反映した状態で固定されます）。

[[ファイル］メニュー（解析グラフ パネル専用部分）]

解析グラフ パネル専用の［ファイル］メニューの各項目，および機能は次のとおりです。

解析グラフ・データ を保存	このパネルのタブの内容を前回保存したファイルに保存します（「2. 14　解析情報をファイルに保存する」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 解析グラフ・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 解析グラフ・データ を保存 ...	このパネルのタブの内容を指定したファイルに保存するために，名前を付けて保存 ダイアログをオープンします（「2. 14　解析情報をファイルに保存する」参照）。

［値の推移］タブ

グラフ化の対象として登録した変数／レジスタ／アドレス等の値と時間の関係を折れ線グラフで表示します。

また，選択しているマイクロコントローラが Smart Analog IC 搭載品の場合では，デバッグ・ツールをデータ収集モードに設定することにより，Smart Analog 用に収集したデータをグラフで表示することができます【E1/E20【RL78】】。

このパネルのグラフで指定した範囲を，値の推移（ズーム）パネル上でズーム表示することができます。

なお，このタブ上でグラフを表示するための操作手順は，「2. 13. 1　値の推移をグラフ化する」を参照してください。

注意 1. **デバッグ・ツールが取得したトレース・データ，またはリアルタイム RAM モニタ結果を基にグラフ表示を行います。グラフ表示を行うためには，「(a) トレース機能」／「(b) RRM 機能／疑似 RRM（RAM モニタ）機能」の注意も参照してください。**

2. **コンパイラの最適化により変数がレジスタに割り当てられている区間は，変数に対するリード／ライトがトレース・データとして取得できません。このため，トレース・データ解析方式を指定している場合，この区間における変数値の推移を表示することはできません。**

3. **【IECUBE【V850E1】【V850ES】】【E20【RX】】**
トレース・データ解析方式を選択している場合，プログラム実行中にトレース パネル上のコンテキスト・メニューの［トレース停止］／［トレース開始］を選択すると，不正なグラフを表示することがあります。

4. **【E1/E20【RL78】】**
Smart Analog 用に収集したデータをグラフ表示するためには，実行プログラムにデータ収集用のモニタ・プログラムをリンクする必要があります。

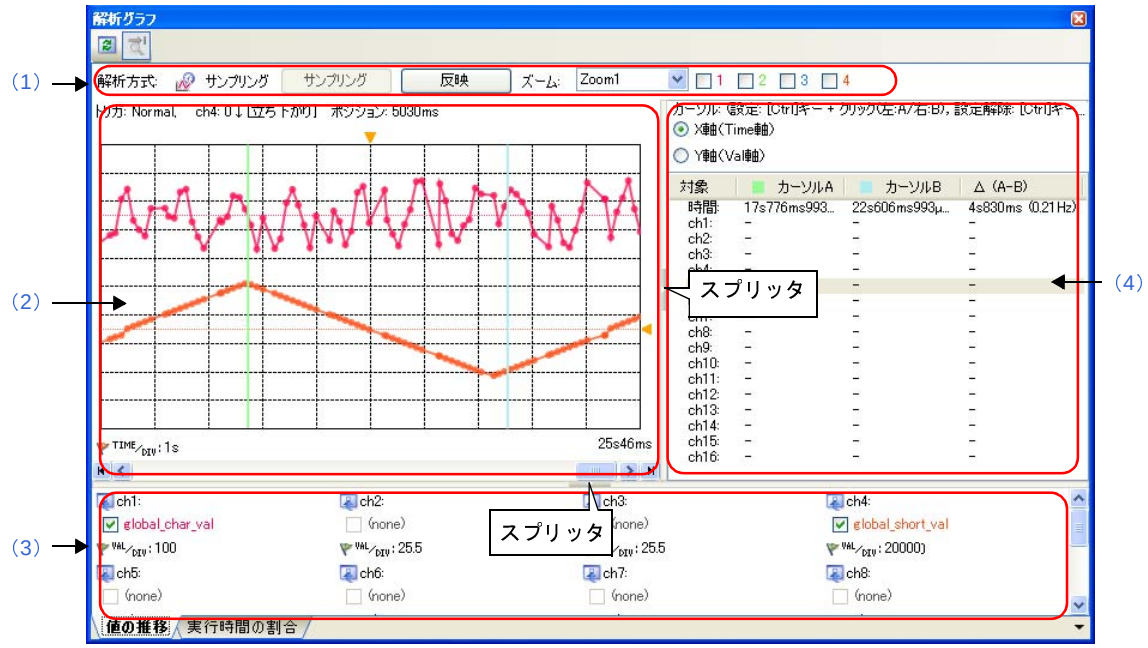
備考 1. 表示更新のタイミングについては，「(4) グラフを表示する」を参照してください。

2. 現在の表示内容を解析グラフ・データ・ファイル（*.mtac）として保存することにより，表示しているグラフを復帰させることができます（「(6) グラフを復帰するためのグラフ・データを保存する」参照）。

3. 取得したグラフ・データがバッファ容量（10000 プロット分）を越えた場合，新しいグラフ・データを最も古いグラフ・データに上書きしていきます（リング・バッファ方式）。
この場合，グラフの描画が一部空白になります。

4. チャネル情報エリア，およびカーソル情報エリアは，スプリッタ上の中央のマークをクリックすることにより，表示／非表示を切り替えることができます。

図 A―11　解析グラフ パネル：［値の推移］タブ



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］

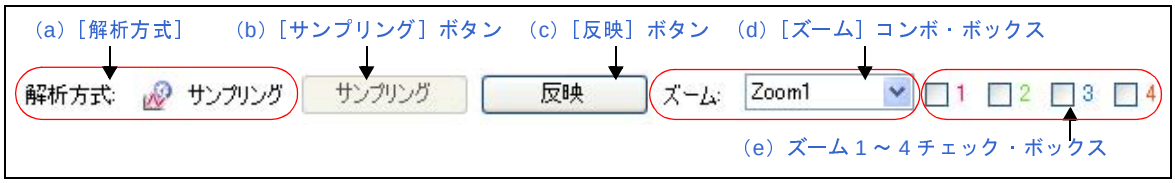
［オープン方法］

- メイン・ウインドウのツールバーの  ボタンのクリック→［値の推移］タブの選択
- ［表示］メニュー→［プログラム解析］→［解析グラフ］→［値の推移］タブの選択

［各エリアの説明］

(1) グラフ・コントロール・エリア

図 A―12　グラフ・コントロール・エリア



(a)［解析方式］

現在，プロパティ パネルの［値の推移］タブ上の［全般］カテゴリ内［解析方式］プロパティにおいて指定している，グラフの解析方式を表示します（「(2) グラフ・データの取得方法を選択する」参照）。

表示内容	説明
サンプリング	［リアルタイム・サンプリング方式］を選択していることを示します。
トレース	［トレース・データ解析方式］を選択していることを示します。
ファイル	［ファイルから読み込み］を選択していることを示します。

(b)［サンプリング］ボタン

リアルタイム・サンプリングの開始／停止の制御（トグル）を手動で行います。

ただし，このボタンは，次のすべての状態を満たしている場合のみ有効となります。

- デバッグ・ツールとの接続が完了している
- 解析方式としてリアルタイム・サンプリング方式を選択している
- プロパティ パネルの［値の推移］タブ上の［全般］カテゴリ内［サンプリング開始／停止］プロパティにおいて［手動］を選択している

注意 サンプリング停止から開始の際は，それまで保持していたグラフ・データの情報は破棄されます（表示中のグラフが消失します）。

(c)［反映］ボタン

現在ウォッチ パネル（ウォッチ 1）に登録されている上から 16 個のウォッチ式を，グラフ化対象として自動登録します（「ウォッチ パネルから反映する場合（自動登録）」参照）。

ただし，このボタンは，次のすべての状態を満たしている場合のみ有効となります。

- デバッグ・ツールとの接続が完了している
- プログラムが停止している
- 解析方式としてリアルタイム・サンプリング方式，またはトレース・データ解析方式を選択している
- デバッグ・ツールにおいて，データ収集モードを無効にしている【E1/E20【RL78】】

注意 このボタンをクリックすることにより，それまで登録していたグラフ化対象の情報は破棄されます（表示中のグラフが消失します）。

(d)［ズーム］コンボ・ボックス

ズーム表示する範囲設定を行う際に，ズーム表示の対象となる値の推移（ズーム）パネルの番号（Zoom*1* ～ *4*）を選択します（「(c) ズーム表示」参照）。

ただし，プログラム実行中は無効となります。

(e) ズーム *1* ～ *4* チェック・ボックス

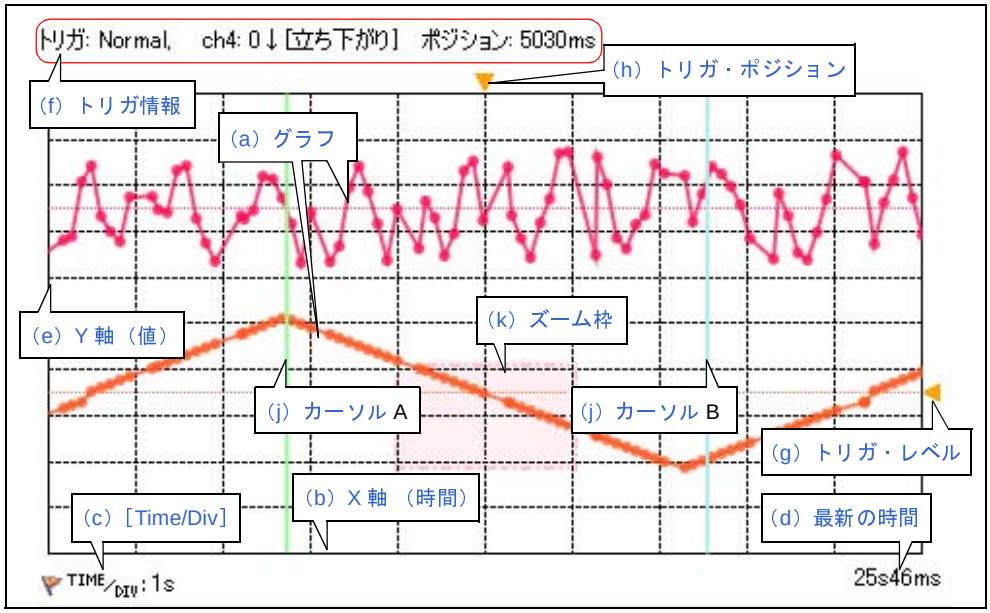
表示する値の推移（ズーム）パネルを指定します。

チェックすることにより，番号に対応する値の推移（ズーム）パネル（値の推移（ズーム）*1* ～ *4*）をオープンします（複数選択可）。

デフォルトでは，すべてのチェック・ボックスがチェックされていません。

(2) グラフ表示エリア

図 A―13　グラフ表示エリア



登録したグラフ化対象の値（Y 軸）と時間（X 軸）の関係を折れ線グラフで表示します。
各部の機能は，次のとおりです。

(a) グラフ

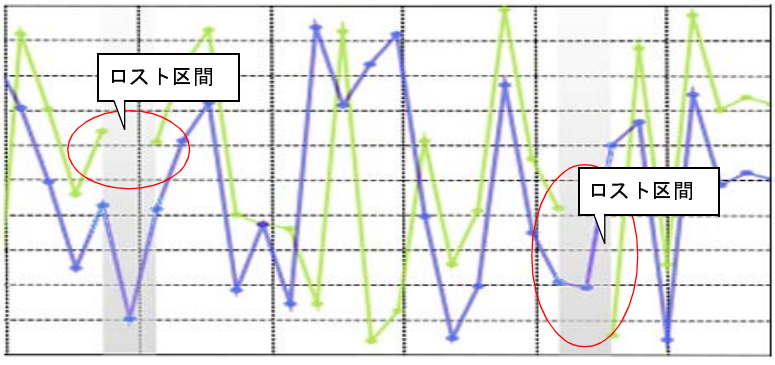
各チャネルに登録しているグラフ化対象における，プログラム実行による値の変化を示します。

ただし，リアルタイム・サンプリング方式でグラフ・データを取得した場合，グラフ・データの取得に失敗すると時間情報のみの表示となり，遷移箇所と遷移箇所を結ぶ線は表示されません。これに該当する区間は“ロスト区間”として，プログラム実行停止後，次のように表示されます（グラフの背景色は，プロパティ パネルの［値の推移］タブ上の［全般］カテゴリ内［背景色 (ロスト)］プロパティの設定に依存）。

なお，現在のグラフの表示範囲にかかわらず，グラフ・データの取得に 1 箇所でも失敗した場合は，出力 パネルに次のメッセージを表示します。

“データ取得中に，ロストまたはバッファのオーバーフローが発生しています。”

図 A―14　グラフ・データの取得に失敗した区間の表示



備考 1. グラフ・データの取得に失敗する原因には次の場合があります。

- グラフ化対象がファイル内スタティック変数，または関数内スタティック変数などで，スコープ指定のない当該変数が，サンプリング時の PC 位置によりスコープを外れた場合
- 【E1/E20【RL78】】
  デバッグ・ツールがデータ収集モードの際にデータの取りこぼしが発生した場合
- その他（「3. 4 トレース・データ解析方式について」参照）

2. プロパティ パネルの［値の推移］タブ上の［チャネル *1* ～ *16*］カテゴリ内［色 *1* ～ *16*］プロパティにより，各チャネルごとに，グラフの描画色を変更することができます。

3. リアルタイム・サンプリング方式を指定している場合，プロパティ パネルの［値の推移］タブ上の［全般］カテゴリ内［グラフの種類］プロパティにより，折れ線グラフの形式を変更することができます（「(b) グラフ形式の選択」参照）。

(b) X 軸 （時間）

経過時間を示します。

10 分割のグリッド線を表示します。単位グリッドあたりの時間（Time/Div）の設定は，全チャネルを対象に，プロパティ パネルの［値の推移］タブ上の［全般］カテゴリ内［1 グリッドあたりの時間 [Time/Div]］プロパティにより行います（「(c) 表示範囲の設定」参照）。

なお，X 軸（時間）の表示範囲は，グラフ・データの取得方法に依存します（「表 2—11 グラフ・データの取得方法によるグラフ表示の相違」参照）。

(c)［Time/Div］

現在，プロパティ パネルの［値の推移］タブ上の［全般］カテゴリ内［1 グリッドあたりの時間 [Time/Div]］プロパティで指定している値を表示します。

なお，プログラム停止中の状態に限り，このラベルをダブルクリックすることにより，グラフの遷移箇所が描画領域（左端～右端）内で，指定された個数分に収まるように［1 グリッドあたりの時間 [Time/Div]］プロパティを適切な数値に自動調整します（「(c) 表示範囲の設定」参照）。

(d) 最新の時間

グラフ・データに応じた最新の時間を表示します。

(e) Y 軸（値）

グラフ化対象の値を示します。

10 分割のグリッド線を表示します。単位グリッドあたりの値（Val/Div）の設定は，各チャネルごとに指定することができ，プロパティ パネルの［値の推移］タブ上の［チャネル *1* ～ *16*］カテゴリ内［1 グリッドあたりの値 [Val/Div] *1* ～ *16*］プロパティにより行います（「(c) 表示範囲の設定」参照）。

なお，プログラム停止時に限り，原点（数値 =0）を表す軸線を，チャネルごとの描画色を用いた点線で表示します。

注意 Y 軸 （値）の表示範囲は，［1 グリッドあたりの値 [Val/Div] *1* ～ *16*］プロパティ／［オフセット *1* ～ *16*］プロパティの設定に依存しますが，取得した値が表示範囲の上限／下限を越えている場合，該当区間のグラフ表示は行いません。

なお，現在のグラフの表示区間にかかわらず，取得した値が常に表示範囲外のチャネルが存在する場合は，出力 パネルにメッセージを表示します。

備考 プロパティ パネルの［値の推移］タブ上の［チャネル *1* ～ *16*］カテゴリ内［オフセット *1* ～ *16*］プロパティにより，各チャネルごとに，Y 軸のグリッド単位でオフセット値を指定することができます。

(f) トリガ情報

トリガ機能を使用している場合（「(3) トリガ機能を使用する」参照），プロパティ パネルの［値の推移］タブ上の［トリガ］カテゴリ内で指定されている各プロパティの内容を表示します。

トリガ機能を使用していない場合は，“-”を表示します。

(g) トリガ・レベル

トリガ機能を使用する場合のトリガ・レベルを示します（「(3) トリガ機能を使用する」参照）。

なお，このトリガ・マークをマウスでドラッグすることにより，トリガ・レベルの値を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。

(h) トリガ・ポジション

トリガ機能を使用する場合のトリガ・ポジションを示します（「(3) トリガ機能を使用する」参照）。

なお，このトリガ・マークをマウスでドラッグすることにより，トリガ・ポジションの値を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。

(i) ポップアップ表示

グラフ上の遷移箇所にマウス・カーソルを重ねることにより，その箇所の情報をポップアップ表示します。表示形式は次のとおりです。

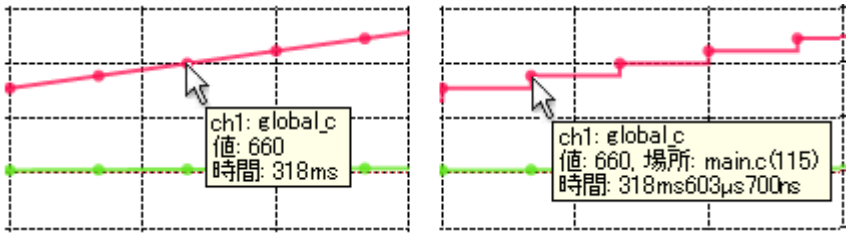
【リアルタイム・サンプリング方式の場合】

チャネル番号 : 変数名
値 : *データ値* ,
時間 : *データ値*

【トレース・データ解析方式の場合】

チャネル番号 : 変数名
値 : *データ値* , 場所 : *ファイル名* (*行数*)
時間 : *データ値*

【リアルタイム・サンプリング方式】　　【トレース・データ解析方式】



注意　プログラム実行中は，この機能は無効となります。

備考 1. ［場所］情報は，トレース・データ解析方式により取得したグラフ・データで，かつ遷移箇所の情報が存在する場合のみ表示します（存在しない場合は“-”を表示します）。
また，この場合，遷移箇所上をダブルクリックすることにより，エディタ パネルで該当箇所を表示することができます（解析グラフ・データ・ファイル（*.mtac）の読み込みにより復帰したグラフの場合を除く）。
ただし，デバッグ・ツールにおいて，トレース・イベントとポイント・トレース・イベントを組み合わせて取得したグラフ・データの場合，［場所］情報が不正な値となることがあります。

2. ［時間］情報の表示形式は次のとおりです。
- トレース・データ解析方式　　：　*XXX*s*XXX*ms*XXX*μs*XXX*ns
- リアルタイム・サンプリング方式：　*XXX*s*XXX*ms

(j) カーソル

X 軸 （時間），または Y 軸 （値）を対象として，時間／数値を確認するための 2 本のカーソル（カーソル A/ カーソル B）です。

カーソル選択ボタンの［X 軸 （Time 軸）］ボタンを選択することにより，X 軸 （時間）を対象としたカーソル計測を行い，また，［Y 軸 （Val 軸）］ボタンを選択することにより，Y 軸 （値）を対象としたカーソル計測を行います。

計測結果は，カーソル情報エリアに一覧表示されます。

カーソルの表示／設定方法は次のとおりです（カーソルはデフォルトで非表示です）。

カーソル	表示／設定	非表示／設定解除
カーソル A	［Ctrl］キー + クリック	［Ctrl］キー + ダブルクリック
カーソル B	［Ctrl］キー + 右クリック	［Ctrl］キー + 右ボタンのダブルクリック

注意 1.　プログラム実行中は，各カーソルは非表示となります。

2.　X 軸 （時間）を対象とした場合。グラフ・データの有無を問わず，0s より左側の領域にカーソルを設定することはできません。

備考　プロパティ パネルの［値の推移］タブ上の［全般］カテゴリ内［カーソル A の色］プロパティ／［カーソル B の色］プロパティにより，各カーソルの色を指定することができます。

（k）ズーム枠

値の推移（ズーム）パネルに表示するズーム表示範囲を示します（「（c）ズーム表示」参照）。

なお，ズーム表示範囲の設定は，［ズーム］コンボ・ボックスの選択により，4 個の値の推移（ズーム）パネルに対してそれぞれ個別に行うことができます。

備考　プロパティ パネルの［値の推移］タブ上の［全般］カテゴリ内［ズーム枠 *1* ～ *4* の色］プロパティにより，各ズーム枠の色を指定することができます。

（3）チャネル情報エリア

このエリアは，スプリッタを移動することにより表示領域を変更することができます。

また，スプリッタ上の中央のマークをクリックすることにより，このエリアの表示／非表示を切り替えることができます。

図 A―15　チャネル情報エリア

（a）チャネル番号

チャネル番号（ch1 ～ ch16）を表示します。

備考　【E1/E20【RL78】】

デバッグ・ツールがデータ収集モードの場合，表示されるアイコンが から に変化します。

（b）変数名チェック・ボックス

現在，各チャネルにグラフ化対象として登録されている変数名（レジスタ名／アドレス式などを含む）をチェック・ボックスとして表示します（未登録の場合は“(none)”を表示）。

なお，変数名の文字色は，グラフの描画色と同一です。

チェック・ボックスをチェックすることにより，対応するチャネルのグラフを表示します（複数選択可）。

デフォルトでは，登録済みのチェック・ボックスはすべてチェックされています。

注意　【E1/E20【RL78】】

デバッグ・ツールがデータ収集モードの場合，表示されている変数名はグラフ化対象とはなりません。

備考 1.　グラフ化対象の登録方法についての詳細は，「（1）グラフ化対象を登録する」を参照してください。

2.　グラフの描画色は，各チャネルごとに，プロパティ パネルの［値の推移］タブ上の［チャネル *1* ～ *16*］カテゴリ内［色 *1* ～ *16*］プロパティで変更することができます。

（c）［Val/Div］

現在，プロパティ パネルの［値の推移］タブ上の［チャネル *1* ～ *16*］カテゴリ内［1 グリッドあたりの値 [Val/Div] *1* ～ *16*］プロパティで指定している値を表示します。

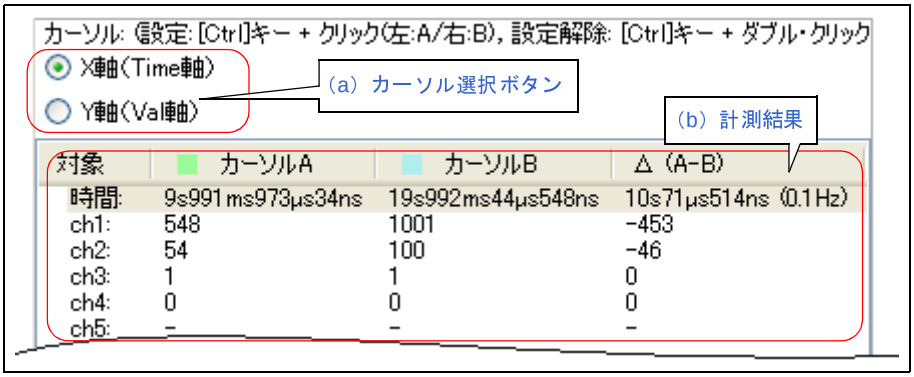
なお，プログラム停止中の状態に限り，このラベルをダブルクリックすることにより，そのチャネルのグラフが描画領域いっぱいに描画されるように［1 グリッドあたりの値 [Val/Div] *1* ～ *16*］プロパティと［オフセット *1* ～*16*］プロパティを適切な数値に自動調整します（「（c）表示範囲の設定」参照）。

（4）カーソル情報エリア

このエリアは，スプリッタを移動することにより表示領域を変更することができます。

また，スプリッタ上の中央のマークをクリックすることにより，このエリアの表示／非表示を切り替えることができます。

図 A―16　カーソル情報エリア



（a）カーソル選択ボタン

カーソル計測を行う際の対象軸を選択します。

ただし，プログラム実行中は無効となります。

X 軸 （Time 軸）	カーソル計測の対象を X 軸 （時間）に設定します（デフォルト）。
Y 軸 （Val 軸）	カーソル計測の対象を Y 軸 （値）に設定します。

（b）計測結果

現在のカーソル A/ カーソル B の位置から求まる次の計測結果を表示します。

ただし，プログラム実行中は非表示となります。

時間	- X 軸（時間）を対象とした場合 カーソル A が示す時間，カーソル B が示す時間，および両カーソル間の差分時間（差分値より求まる周期 [Hz]）を表示します。 - Y 軸 （値）を対象とした場合 “-” を表示します。

ch*1* ～ *16*	- X 軸（時間）を対象とした場合 カーソル A が示す時間時の値，カーソル B が示す時間時の値，および両カーソル間の差分値を表示します。 - Y 軸 （値）を対象とした場合 カーソル A が示す値，カーソル B が示す値，および両カーソル間の差分値を表示します。 ただし，値が取得できない場合は，“-”を表示します。

［実行時間の割合］タブ

取得した関数情報において，関数の実行時間の割合を円グラフで表示します。

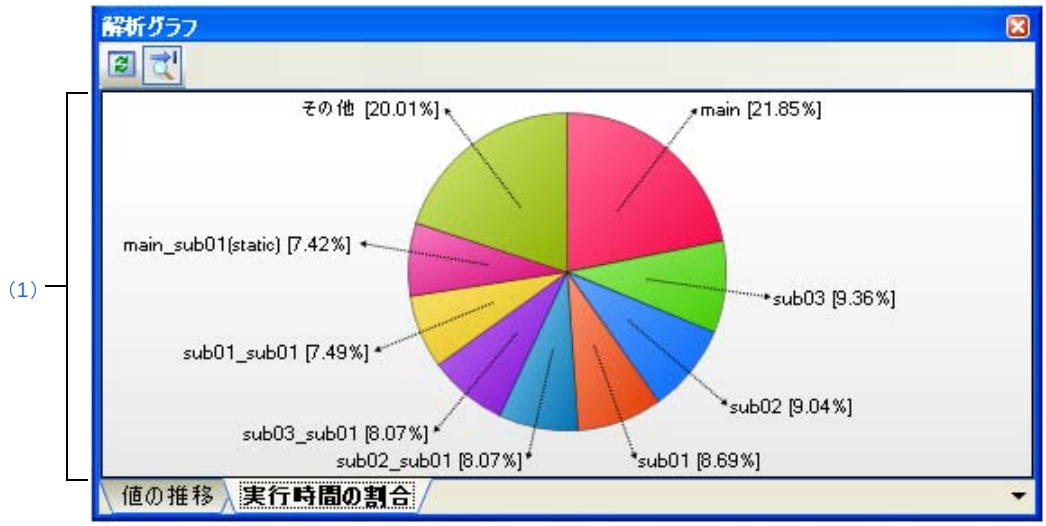
なお，このタブ上でグラフを表示するための操作手順は，「2. 13. 2　関数の実行時間率をグラフ化する」を参照してください。

注意 1.　デバッグ・ツールがトレース機能をサポートしていない場合，またはトレース機能を有効化していない場合，このグラフを表示することはできません。
また，トレース機能を有効化している状態であっても，トレース・メモリにトレース・データが存在しない場合は，グラフ表示は行わず，出力 パネルに次のメッセージを表示します。
“実行時間情報がありません。”

2.　グラフ表示を行うためには，「(a) トレース機能」の注意も参照してください。

3.　【IECUBE【78K0】】【E1/E20【RX】】【EZ Emulator【RX】】
トレース・タイム・タグ機能をサポートしていないため，このグラフを表示することはできません。

図 A―17　解析グラフ パネル：［実行時間の割合］タブ



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］

［オープン方法］

- メイン・ウインドウのツールバーの  ボタンのクリック→［実行時間の割合］タブの選択
- ［表示］メニュー→［プログラム解析］→［解析グラフ］→［実行時間の割合］タブの選択

［各エリアの説明］

（1）グラフ表示エリア

関数の実行時間の割合を示す円グラフを表示します。

表示する関数の数は，プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［実行時間の割合グラフに表示する関数の数］プロパティの指定により変更することができます（デフォルトでは［10］が指定されます）。実行時間の割合の大きい順にグラフ化の対象となり，ここで指定した数を越える関数については，“その他”としてまとめて表示します。

備考 1. ラベルの文字色とグラフの背景色は，オプション ダイアログにおける［全般 - フォントと色］カテゴリ内の［標準］項目の文字色／背景色の設定に依存します。

2. ［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，グラフの表示を 10 % ～ 100 % まで拡大／縮小することができます（ラベルのフォント・サイズは対象となりません）。

（a）ラベル表示

各関数のラベルとして，“*関数名* [*実行時間の割合* %]”を表示します。

なお，スタティック関数の場合は，“( )”内にファイル名を併記します。

備考　*実行時間の割合*は，関数一覧 パネルにおける［実行時間 (割合)[%]］項目の値と同値です。

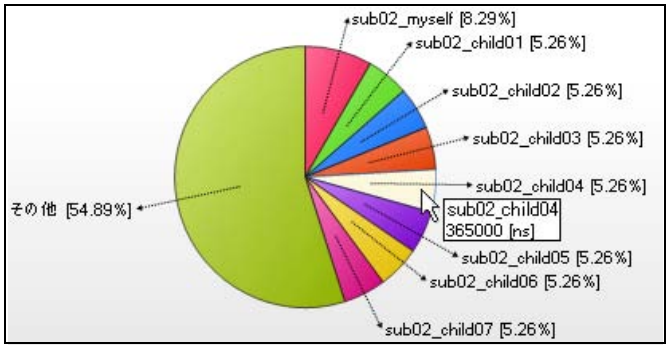
（b）ポップアップ表示

グラフ上にマウス・カーソルを重ねることにより，その関数の情報ををポップアップ表示します。

表示形式は次のとおりです。

関数名 *実行時間* [*単位*]

関数名	対象関数名を示します。 対象関数がメンバ関数の場合は，“*クラス名* :: *関数名*”の形式で表示します【CC-RX】。
実行時間 [*単位*]	対象関数の実行時間を示します（関数一覧 パネルにおける［実行時間 [単位]］項目の値と同値）。 なお，[*単位*] は，プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［時間の単位］プロパティの設定により変更可能です。



コール・グラフ パネル

関数間の呼び出し関係をツリー構造の図（コール・グラフ）で表示します。

このパネルで表示対象となる関数／変数の種類は次のとおりです。

- グローバル関数
- スタティック関数
- メンバ関数（C++ ソース・ファイルを対象とする場合）
- グローバル変数
- ファイル内スタティック変数
- 関数内スタティック変数
- IOR【V850】
- SFR【RL78】【78K0R】【78K0】
- クラス変数（C++ ソース・ファイルを対象とする場合）

ただし，解析対象外に指定されているファイル内の関数情報／変数情報は表示されません（「1.1.1　解析対象」参照）。

なお，コール・グラフを表示するための操作手順は，「2.4　関数間の呼び出し関係（コール・グラフ）を表示する」を参照してください。

注意 1.　デバッグ・ツールがトレース機能をサポートしていない場合，またはデバッグ・ツールのトレース機能を有効化していない場合，コール・グラフにおいて，動的解析情報（実行回数／リード回数／ライト回数）を表示することはできません。
また，動的解析情報については，「(a) トレース機能」の注意も参照してください。

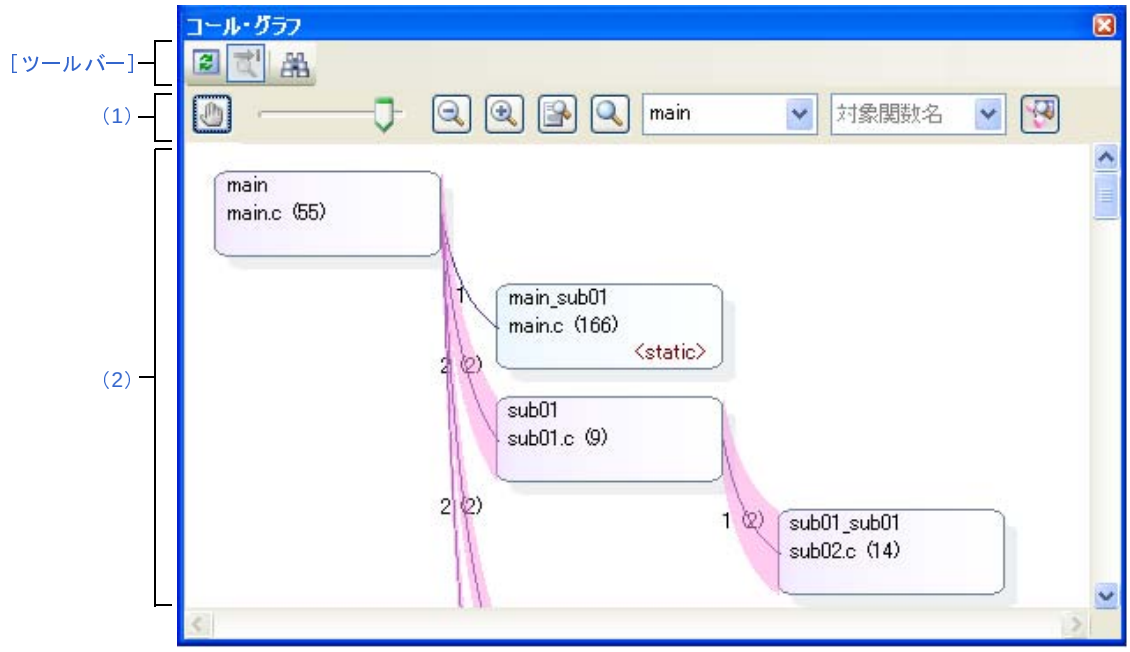
2.　【CC-RX】
C++ ソース・ファイルを対象としている場合，暗黙的に呼び出されるクラス型のコンストラクタ／デストラクタの呼び出しは，コール・グラフに表示されません。

3.　【CA850】【CA78K0R】【CA78K0】
ビルド・ツールにおいてクリーンを実行すると，現在表示しているコール・グラフは消失します。

備考 1.　プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［定義箇所がない関数／変数をコール・グラフの表示対象とする］プロパティの指定を［はい］に変更した場合，ソース・ファイルが存在しない関数／変数をコール・グラフに含めることができます。

2.　パネル・コントロール・エリアの  ，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（10 ～ 109 %）することができます。

図 A―18　コール・グラフ パネル（全体表示）



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・グラフ パネル専用部分）］
- ［［編集］メニュー（コール・グラフ パネル専用部分）］
- ［コンテキスト・メニュー］

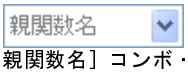
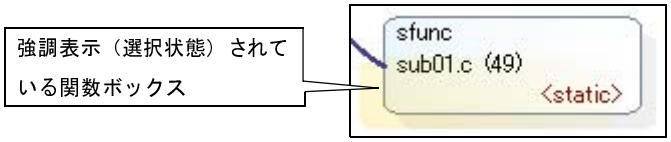
［オープン方法］

- メイン・ウインドウのツールバーの  ボタンのクリック
- ［表示］メニュー→［プログラム解析］→［コール・グラフ］の選択

［各エリアの説明］

（1）パネル・コントロール・エリア

	パネルを直接ドラッグすることにより，表示内容のスクロールを許可するか否かを設定します（トグル）。デフォルトではスクロールを許可しません。 なお，スクロールを許可した場合，マウス・カーソルの形状が変化し，コール・グラフ内の関数ボックス／変数ボックスをクリックしても，その関数／変数を強調表示（選択状態）することはできません（［対象関数名］コンボ・ボックスにも反映されません）。また，関数／変数の情報をポップアップ表示することもできません。

	コール・グラフのズーム率を変更します。 10 ～ 109 % の範囲の数値を選択することができます（デフォルトでは 100 % が指定されます）。 なお，［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ってもズーム率を変更することができます。
	コール・グラフのズーム率を変更します。 各ボタンを 1 回クリックすると，ズーム・スライダの値が 1 つ減少／増加します。
	コール・グラフのズーム率を自動で変更します。 パネルのサイズに合わせて，コール・グラフの全体サイズを縮小／拡大します。
	コール・グラフのズーム率をデフォルトの 100 % にリセットします。
 ［親関数名］コンボ・ボックス （コンボ・ボックス左）	コール・グラフの表示対象とする（コール・グラフの先頭となる）親関数をドロップダウン・リストにより選択します**注 1**。 デフォルトでは，“main”／“reset”**注 2**，またはそれを含む関数名のうち最初に出現した関数が指定されます（該当しない場合は“空欄”）。
 ［対象関数名］コンボ・ボックス （コンボ・ボックス右）	コール・グラフ上で強調表示（選択状態）する関数を次のドロップダウン・リストより選択します。 - ［親関数名］コンボ・ボックスが空欄の場合 全体表示 ：*プログラム中に存在するすべての関数名* 詳細表示 ：*現在表示対象となっているすべての関数名* - ［親関数名］コンボ・ボックスで親関数を指定している場合 全体表示 ：*対象親関数から呼び出されている関数名*（子関数／孫関数～を含む） 詳細表示 ：*現在表示対象となっているすべての関数名* なお，コール・グラフ内の任意の関数ボックスをクリックすることでも該当関数が強調表示（選択状態）され，該当関数名がこのコンボ・ボックスに反映されます（  ボタンによりマウスのドラッグによるスクロールを許可している場合を除く）。 
	現在選択している関数（［対象関数名］コンボ・ボックスで指定している関数）に対する親関数と子関数を表示する詳細表示にコール・グラフを切り替えます（トグル）。 ただし，プログラム実行中は無効となります。

注 1. 【RH850】【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では，“PM*n*”【RH850】／“PE*n*”【V850E2】を指定することができます。この場合，該当 PM*n*/PE*n* で実行される関数のみを対象としてコール・グラフを表示します。

2. 選択しているマイクロコントローラにより関数名は異なります。

- 【RH850】【V850】【RL78】【78K0R】【78K0】：main
- 【RX】　：reset

（2）コール・グラフ表示エリア

クロスリファレンス情報から取得した関数間の呼び出し関係を示すコール・グラフを表示します。

なお，コール・グラフは，  ボタンのクリック（トグル）により，次の 2 つの表示モードに切り替わります。

- 全体表示（デフォルト）

デフォルトでは，“main”／“reset”[注 1] またはそれを含む関数名のうち最初に出現した関数を親関数とみなし，その関数をコール・グラフ内の最左端に配置します（［親関数名］コンボ・ボックスに該当親関数名を表示します）。相当する関数名が存在しない場合は，プログラム中どの関数からも呼び出されていない関数（参照回数 =0）を親関数とみなし，それらの関数すべてを最左端に配置します（［親関数名］コンボ・ボックスは空欄となります）。

続いて，子関数→孫関数→ …，それぞれに相当する関数を左から右方向へ配置することでコール・グラフを表示します（上下の位置関係は，上から下方向へ関数の出現順を意味します）。

なお，［親関数名］コンボ・ボックスで任意の親関数を指定[注 2] した場合では，その関数から呼び出されている関数のみを対象としたコール・グラフを表示します。また，同コンボ・ボックスが空欄の場合は，プログラム中に存在するすべての関数を対象としてコール・グラフを表示します。

注 1. 選択しているマイクロコントローラにより関数名は異なります。

- 【RH850】【V850】【RL78】【78K0R】【78K0】: main
- 【RX】　: reset

2. 【RH850】【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で，［親関数名］コンボ・ボックスにおいて“PM*n*”【RH850】／“PE*n*”【V850E2】を指定した場合では，該当 PM*n*/PE*n* で実行される関数のみを対象としたコール・グラフを表示します。

- 詳細表示

現在，［対象関数名］コンボ・ボックスで指定している関数の親関数と子関数についてのコール・ペアを表示します。

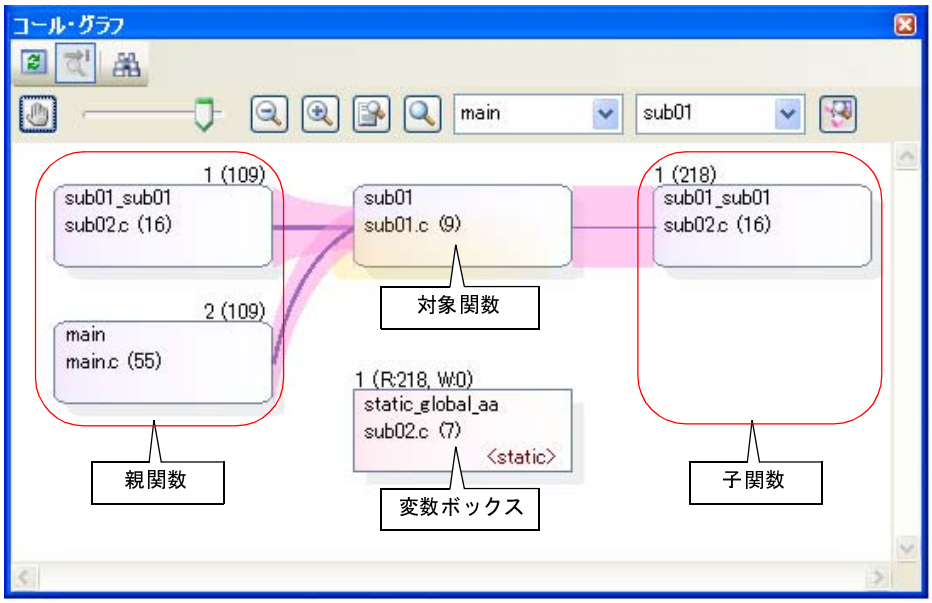
対象関数を中心に，左側に親関数を，右側に子関数を配置することでコール・グラフを表示します（上下の位置関係は，上から下方向へ関数の出現順を意味します）。

また，対象関数からアクセスしているグローバル変数／ファイル内スタティック変数／関数内スタティック変数が存在する場合，その変数を対象関数の直下に配置します（変数が複数存在する場合，上下の位置関係は，上から下方向へ変数の出現順を意味します）。

備考　次のいずれかの場合，コール・グラフは全体表示に切り替わります。

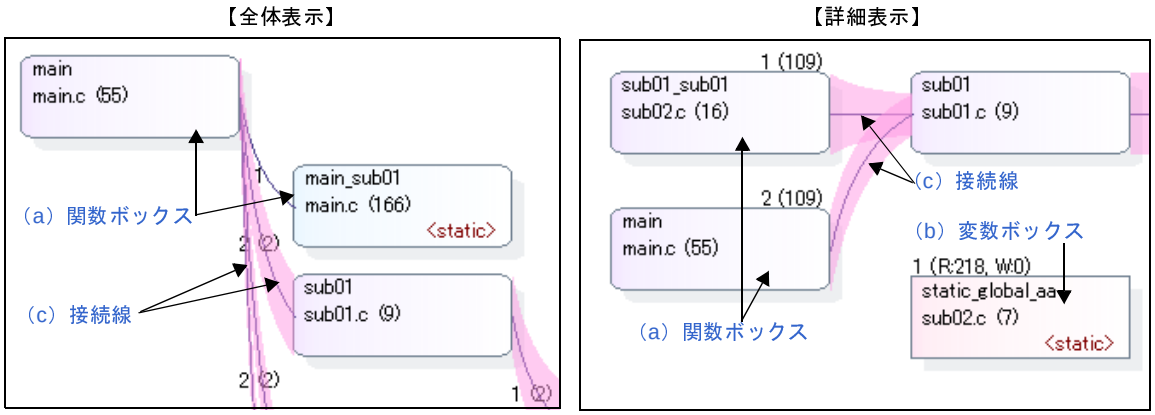
- ［親関数名］コンボ・ボックスの指定が変更された場合（ビルド・ツールにおけるビルド／リビルドの実行による結果を含む）
- ビルド・ツールにおいて，クリーンを実行した場合

図 A—19 コール・グラフ パネル（詳細表示）



コール・グラフの構成要素の詳細は次のとおりです。

図 A—20 コール・グラフの構成要素

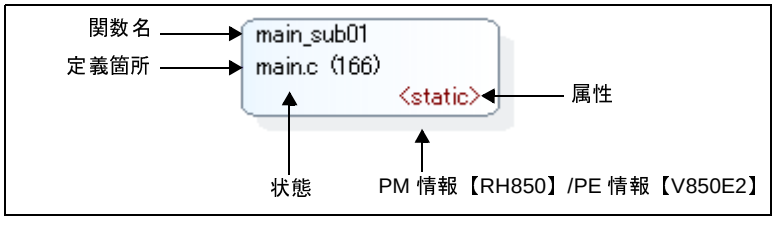


（a）関数ボックス

関数をボックス形式で表示します。

各関数ボックスに表示する情報は次のとおりです。

図 A—21 関数ボックスの情報



関数名	対象関数名を示します。 【CC-RX】 - 対象関数がオーバーロードされている場合，またはテンプレート関数の場合は，“()”内に引数の型を列挙します。 - 対象関数が const メンバ関数 /volatile メンバ関数の場合は，関数名の直後に“const”／“volatile”を表示します。
定義箇所	対象関数が定義されている箇所を“*ファイル名* (*行数*)”で示します。 ただし，定義箇所情報が存在しない場合は，“(定義箇所なし)”を表示します。
属性	対象関数の属性を示します。 - スタティック関数 ：<static> - テンプレート関数 ：<template>【CC-RX】 - 仮想関数 ：<virtual>【CC-RX】 - 純粋仮想関数 ：<abstract>【CC-RX】 上記以外の場合は空欄となります。
状態	対象関数の現在の実行状態を次の背景色で示します。 - 水色：未実行 - 紫色：実行済
PM 情報【RH850】 PE 情報【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，対象関数が割り付いている PE を次の影色で示します。 - 灰色 ：共通 - 赤色 ：PM1/PE1 - 緑色 ：PM2/PE2 - 青色 ：PM3/PE3 - 橙色 ：PM4/PE4

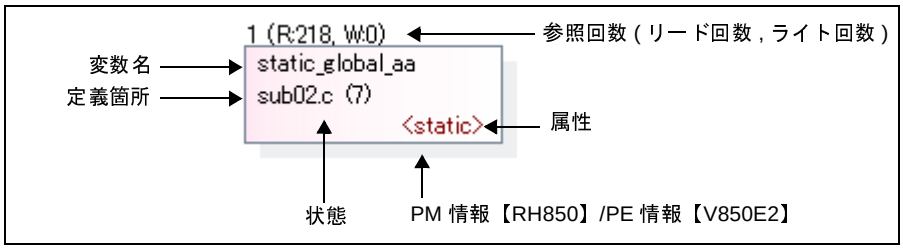
備考 関数ボックスをダブルクリックすることにより，該当関数が定義されているソース・テキスト箇所へジャンプすることができます（「2.7 定義箇所へジャンプする」参照）。
ただし，  ボタンの設定により，マウスのドラッグによるスクロールを許可している場合はこの機能を使用することはできません。この場合は，［対象関数名］コンボ・ボックスにより対象関数を選択したのち，コンテキスト・メニューの［ソースへジャンプ］を選択してください。

(b) 変数ボックス

詳細表示の際に，対象関数からアクセスしているグローバル変数／ファイル内スタティック変数／関数内スタティック変数をボックス形式で表示します。

各変数ボックスに表示する情報は次のとおりです。

図 A―22 変数ボックスの情報



変数名	対象変数名を示します。 なお，対象変数が関数内スタティック変数の場合は，“*変数名 # 関数名*”の形式で表示します。
定義箇所	対象変数が定義されている箇所を“*ファイル名 (行数)*”で示します。 ただし，定義箇所情報が存在しない場合は，“(定義箇所なし)”を表示します。
属性	対象変数の属性を示します。 - スタティック変数 : <static> - 関数内スタティック変数: <static local> 上記以外の場合は空欄となります。
参照回数	対象変数を静的に参照している回数を示します。
リード回数 , ライト回数	対象変数を動的に参照（R: リード , W: ライト）している回数を示します。 ただし，トレース・データが存在する場合のみ表示します。 なお，プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［解析結果を累積する］プロパティにおいて［はい］を選択している場合，プログラム実行ごとの累積による数値を表示します。
状態	対象変数の現在の使用状態を次の背景色で示します。 - 緑色 : 未使用 - 赤紫 : 使用済
PM 情報【RH850】 PE 情報【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，対象変数が割り付いている PE を次の影色で示します。 - 灰色 : 共通 - 赤色 : PM1/PE1 - 緑色 : PM2/PE2 - 青色 : PM3/PE3 - 橙色 : PM4/PE4

備考 変数ボックスをダブルクリックすることにより，該当変数が定義されているソース・テキスト箇所へジャンプすることができます（「2.7 定義箇所へジャンプする」参照）。
ただし，  ボタンの設定により，マウスのドラッグによるスクロールを許可している場合はこの機能を使用することはできません。この場合は，スクロールの許可をいったん解除してから操作を行ってください。

(c) 接続線

ある関数から別の関数を静的に呼び出している場合，双方の関数ボックス間に接続線を表示します。

なお，接続線の表示は，静的な関数呼び出しのみを対象とします。動的な関数呼び出しに対応する接続線は表示しません。

各接続線に表示する情報は次のとおりです。

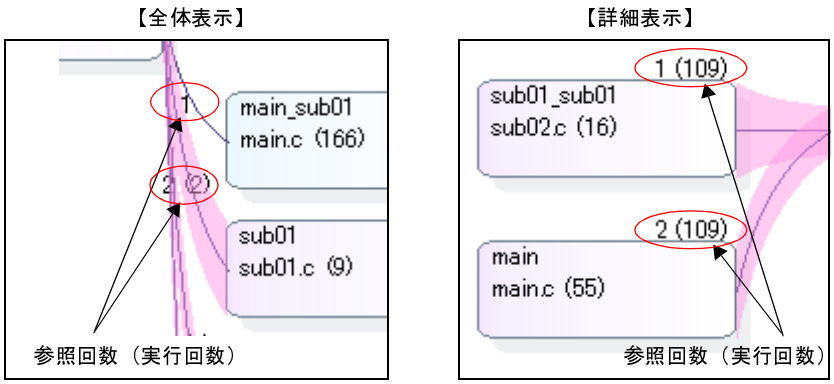
備考 1. 静的な関数呼び出しがなく動的な関数呼び出しがあった場合では（たとえば，関数ポインタを用いてしか関数呼び出しを行っていない場合など），その情報をポップアップ表示で確認することができます。

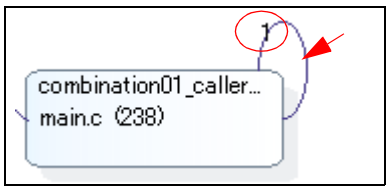
2. 【CA850】【CA78K0R】【CA78K0】

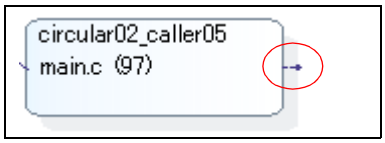
関数 A の定義と関数 B の定義の間において，関数 C のプロトタイプ宣言を記述している場合，関数 A が関数 C をコールしている関数として接続線を表示します。

また，関数のプロトタイプ宣言以外にも，変数の参照（変数ポインタへの代入），または関数の参照（関数ポインタ変数への代入）も同様の動作となります。

図 A―23　接続線の情報



参照回数	対象関数を静的に呼び出している回数を示します。
実行回数	対象関数を動的に呼び出している回数を“()”内に示します。 ただし，トレース・データが存在する場合のみ表示します。 なお，プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［解析結果を累積する］プロパティにおいて［はい］を選択している場合，プログラム実行ごとの累積による数値を表示します。
再帰呼び出し	自分自身を呼び出している関数の場合，それを示す次の接続線と参照回数を表示します。 

循環呼び出し	たとえば，3つの関数としてA，B，Cがあり，A→B→C→Aと呼び出している場合では，“A→B”と“B→C”についての接続線は表示しますが，“C→A”についての接続線は表示せず，循環していることを示す次の線分のみを表示します。 なお，循環呼び出しとなった関数についての情報は，ポップアップ表示で確認することができます。 

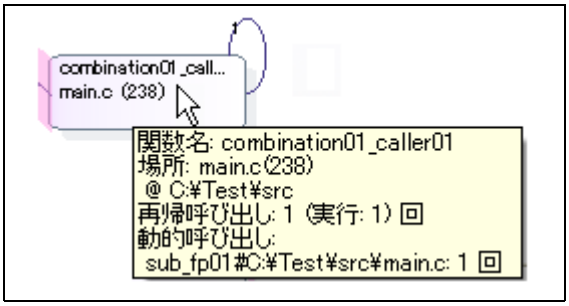
(d) ポップアップ表示

関数ボックス／変数ボックスにマウス・カーソルを重ねることにより，対象関数／変数の情報をポップアップ表示します。

表示形式は次のとおりです。

注意 **ボタンの設定により，マウスのドラッグによるスクロールを許可している場合はこの機能を使用することはできません。**

- 関数ボックス



関数名：*関数名*
場所：*ファイル名*(*行番号*)
　　@ *ファイルの絶対パス*
再帰呼び出し：*参照回数*(実行：*実行回数*)
循環呼び出し
　関数名：*参照回数*(実行：*実行回数*)
　関数名：*参照回数*(実行：*実行回数*)
　…
動的呼び出し
　関数名：*回数*
　関数名：*回数*
　…

関数名	対象関数名を示します。 【CC-RX】 - グローバル関数／スタティック関数の場合 名前空間（グローバル名前空間／無名名前空間を除く）に属する場合は，“*名前空間名* :: *関数名*”の形式で関数名を表示します。 また，対象関数がオーバーロードされている場合，またはテンプレート関数の場合は，関数名に続き“()”内に引数の型を列挙します。 - メンバ関数の場合 名前空間（グローバル名前空間／無名名前空間を除く）に属する場合は，“*名前空間名* :: *クラス名* :: *関数名*”の形式で，属さない場合は“*クラス名* :: *関数名*”の形式で関数名を表示します。 また，対象関数がオーバーロードされている場合，またはテンプレート関数の場合は，関数名に続き“()”内に引数の型を列挙します。 const メンバ関数 /volatile メンバ関数の場合は，関数名の直後に“const”／“volatile”を表示します。
場所	対象関数が定義されている箇所を示します。 ただし，定義箇所情報が存在しない場合は，“(定義箇所なし)”を表示します。
再帰呼び出し	対象関数が再帰呼び出しの場合に表示します。 参照回数 ： 再帰呼び出しとなる呼び出し回数 実行回数 ： 実行した回数（トレース・データが存在する場合のみ）
循環呼び出し	対象関数から循環呼び出ししている関数が存在する場合に表示します。 該当関数が複数存在する場合は，最大 4 個まで列挙します。 関数名 ： 対象関数から循環呼び出ししている関数名 参照回数 ： 循環呼び出しとなる呼び出し回数 実行回数 ： 実行した回数（トレース・データが存在する場合のみ）
動的呼び出し	対象関数から静的な関数呼び出しが一度もなく，かつ動的な関数呼び出しが存在する場合に表示します。 該当関数が複数存在する場合は，最大 4 個まで列挙します。 関数名 ： 対象関数から動的呼び出ししている関数名 回数 ： 実行した回数

備考 実行回数は，プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［解析結果を累積する］プロパティにおいて［はい］を選択している場合，プログラム実行ごとの累積による数値を表示します。

- 変数ボックス

変数名：*変数名* 場所：*ファイル名* (*行番号*) 　　@ *ファイルの絶対パス*

変数名	対象変数名を示します。 なお，対象変数が関数内スタティック変数の場合は，“*関数名* # *変数名*”の形式で表示します。 【CC-RX】 - グローバル変数／ファイル内スタティック変数の場合 名前空間（グローバル名前空間／無名名前空間を除く）に属する場合は，“*名前空間名* :: *変数名*”の形式で変数名を表示します。 - クラス変数の場合 名前空間（グローバル名前空間／無名名前空間を除く）に属する場合は，“*名前空間名* :: *クラス名* :: *変数名*”の形式で，属さない場合は“*クラス名* :: *変数名*”の形式で変数名を表示します。 - 関数内スタティック変数の場合 名前空間（グローバル名前空間／無名名前空間を除く）に属する場合は，“*名前空間名* :: *関数名* # *変数名*”の形式で，属さない場合は“*関数名* # *変数名*”の形式で変数名を表示します。また，クラス変数の場合は，“*クラス名* :: *関数名* # *変数名*”の形式で表示します。
場所	対象変数が定義されている箇所を示します。 ただし，定義箇所情報が存在しない場合は，“(定義箇所なし)”を表示します。

[ツールバー]

ツールバー上の各ボタン，および機能は次のとおりです。

	最新情報を取得し，表示内容を更新します。 ただし，プログラム実行中は無効となります。
	プログラムの実行が停止するごとに最新情報を取得し，表示内容を更新します。 ただし，プロパティ パネルの［設定］タブ上の［全般］カテゴリ内［プログラム停止時に更新を行う］プロパティにおいて［個別に指定する］以外を指定している場合は無効となります（プロパティ パネルでの設定を反映した状態で固定されます）。
	コール・グラフ検索 ダイアログをオープンし，現在このパネルに存在する関数／変数を検索します。

[[ファイル］メニュー（コール・グラフ パネル専用部分)]

コール・グラフ パネル専用の［ファイル］メニューの各項目，および機能は次のとおりです。

コール・グラフ・データ を保存	このパネルの内容を前回保存したファイルに保存します（「2. 14　解析情報をファイルに保存する」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・グラフ・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・グラフ・データ を保存 ...	このパネルの内容を指定したファイルに保存するために，名前を付けて保存 ダイアログをオープンします（「2. 14　解析情報をファイルに保存する」参照）。

[[編集]メニュー（コール・グラフ パネル専用部分）]

コール・グラフ パネル専用の［編集］メニューの各項目，および機能は次のとおりです。

検索 ...	コール・グラフ検索 ダイアログをオープンし，現在このパネルに存在する関数／変数を検索します。

[コンテキスト・メニュー]

このパネル上において，マウスを右クリックすることにより表示されるコンテキスト・メニューの各項目，および機能は次のとおりです。

ソースへジャンプ	選択している関数／変数が定義されているソース・ファイルをエディタ パネル上にオープンします（「2.7　定義箇所へジャンプする」参照）。
逆アセンブルへジャンプ	選択している関数／変数の開始アドレスに対応する逆アセンブル・データを逆アセンブル パネル（逆アセンブル 1）上にオープンします（「2.7　定義箇所へジャンプする」参照）。 ただし，デバッグ・ツールと切断時は無効となります。
メモリへジャンプ	選択している関数／変数関数の開始アドレスに対応するメモリ・リストをメモリ パネル（メモリ 1）上にオープンします（「2.7　定義箇所へジャンプする」参照）。 ただし，デバッグ・ツールと切断時は無効となります。
関数／変数一覧へジャンプ	関数一覧 パネル／変数一覧 パネルをオープンし，このパネルで選択している関数／変数を選択状態にします。
詳細表示	選択している関数（［対象関数名］コンボ・ボックスで指定している関数）に対する親関数と子関数を表示する詳細表示にコール・グラフを切り替えます。 ただし，プログラム実行中は無効となります。

クラス／メンバ パネル

クラス情報【CC-RX】／関数情報／変数情報をツリー形式で表示します。

このパネルで表示対象となる関数／変数の種類は次のとおりです。

- グローバル関数
- スタティック関数
- メンバ関数（C++ ソース・ファイルを対象とする場合）
- グローバル変数
- ファイル内スタティック変数
- クラス変数（C++ ソース・ファイルを対象とする場合）
- インスタンス変数（C++ ソース・ファイルを対象とする場合）

ただし，解析対象外に指定されているファイル内のクラス情報【CC-RX】／関数情報／変数情報は表示されません（「1. 1. 1　解析対象」参照）。

なお，クラス情報【CC-RX】／関数情報／変数情報を表示するための操作手順は，「2. 5　クラス／関数／変数の情報を表示する」を参照してください。

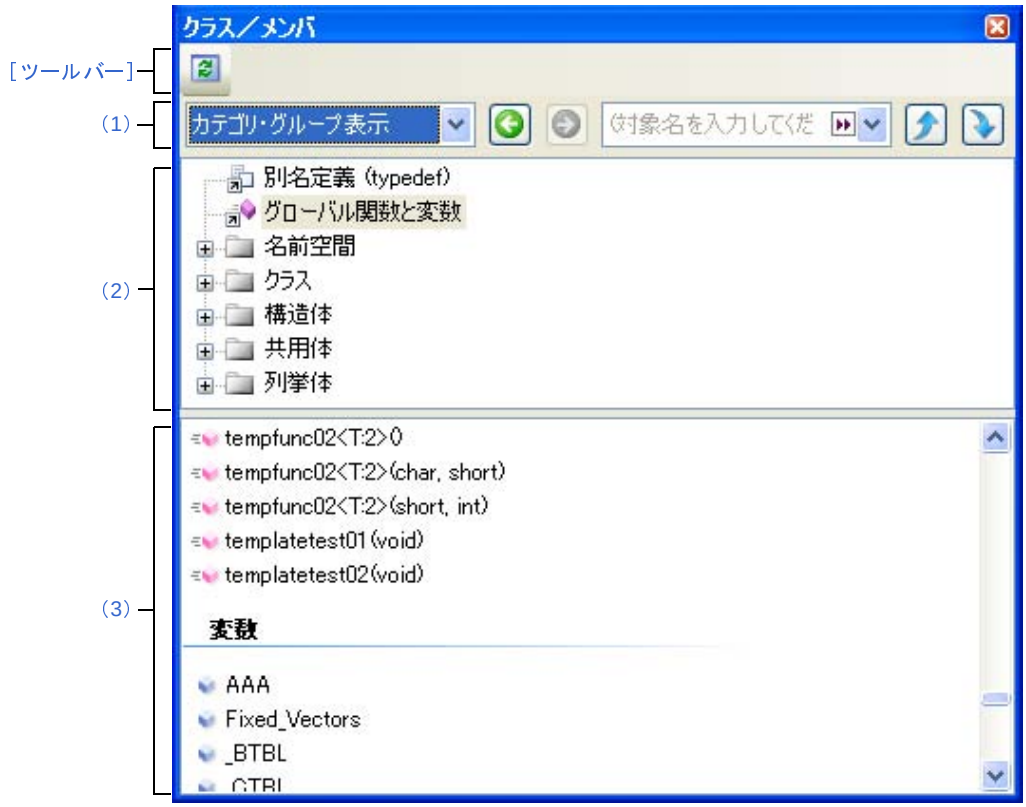
注意 1.　【CC-RX】

クラス情報は，C++ ソース・ファイルを対象とする場合のみ提供される情報です。

2.　【CA850】【CA78K0R】【CA78K0】

ビルド・ツールにおいてクリーンを実行すると，現在このパネルに表示している内容をすべてクリアします。

図 A―24　クラス／メンバ パネル



ここでは，次の項目について説明します。

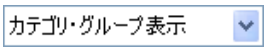
- ［オープン方法］
- ［各エリアの説明］
- ［ツールバー］
- ［［編集］メニュー（クラス／メンバ パネル専用部分）］
- ［コンテキスト・メニュー］

［オープン方法］

- メイン・ウインドウのツールバーの  ボタンのクリック
- ［表示］メニュー→［プログラム解析］→［クラス／メンバ］の選択

［各エリアの説明］

（1）パネル・コントロール・エリア

 ［ビュー設定］コンボ・ボックス （コンボ・ボックス左）	ツリーの分類方法を次のドロップダウン・リストより選択します。 - カテゴリ・グループ表示（デフォルト） - アクセス・グループ表示 - 名前空間グループ表示 - ファイル・グループ表示 - アルファベット・グループ表示
	1 つ前に選択したノードを選択状態にします。 ただし，以前に選択したノードの履歴が存在しない場合，またはプログラム実行中は無効となります。
	 ボタンをクリックする前に選択したノードを選択状態にします。 ただし，  ボタンにより選択したノードの履歴が存在しない場合，またはプログラム実行中は無効となります。
 ［対象名］コンボ・ボックス （コンボ・ボックス右）	 ／  ボタンによる検索を行う際の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）。 ただし，プログラム実行中は無効となります。
	直前にフォーカスのあったツリーに対して，［対象名］コンボ・ボックスで指定している文字列を含むノードを上方向に検索し，検索結果を選択状態にします。 ただし，［対象名］コンボ・ボックスが空欄の場合，またはプログラム実行中は無効となります。
	直前にフォーカスのあったツリーに対して，［対象名］コンボ・ボックスで指定している文字列を含むノードを下方向に検索し，検索結果を選択状態にします。 ただし，［対象名］コンボ・ボックスが空欄の場合，またはプログラム実行中は無効となります。

備考　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) クラス／カテゴリ用ツリー・エリア

プログラムで定義されているクラス情報などをツリー形式で表示します。

このツリー上で選択しているノードに関する情報を，関数／変数用リスト・エリアに表示します。

ツリーは，［ビュー設定］コンボ・ボックスにより，次の５つのグループに分類して表示することができます。

- カテゴリ・グループ表示（デフォルト）
- アクセス・グループ表示
- 名前空間グループ表示
- ファイル・グループ表示
- アルファベット・グループ表示

ツリーで表示するノード名とその順番は次のとおりです。

ただし，情報を取得できなかった場合，または表示する子ノードが存在しない場合は，そのノードは表示されません。

なお，子ノードを持つノードは，ダブルクリックすることにより，展開／折りたたみ表示が可能です。

(a) カテゴリ・グループ表示（デフォルト）

クラスやインタフェースなどの種類で分類して表示します。

ノード	備考
別名定義 (typedef)	
グローバル関数と変数	
マクロと定数	【CA78K0R】【CA78K0】
名前空間 {} *名前空間名*	【CC-RX】
クラス *クラス名*[注1] 基底型 *基底型 (クラス／インタフェース／構造体) 名* 派生型 *派生型 (クラス／インタフェース／構造体) 名* 内部型 *内部型 (クラス／インタフェース／構造体／共用体／列挙体) 名*	【CC-RX】
インタフェース *インタフェース名*[注1] 基底型 *基底型 (クラス／インタフェース／構造体) 名* 派生型 *派生型 (クラス／インタフェース／構造体) 名* 内部型 *内部型 (クラス／インタフェース／構造体／共用体／列挙体) 名*	【CC-RX】

ノード	備考
構造体 *構造体名*注 1，2	【CA850】以外
共用体 *共用体名*注 3	
列挙体 *列挙体名*	

注 1. 【CC-RX】

次のアクセス指定子順に表示を行い，アクセス指定子の種類を示すアイコンを重ねて表示します（同一のアクセス指定子の場合はアルファベット順で表示）。

アクセス指定子	アイコン
public	なし
protected	
private	

2. 【CC-RX】

必要に応じて，基底型ノード／派生型ノード／内部型ノードを表示します。

3. 【CC-RX】

必要に応じて，内部型ノードを表示します。

（b）アクセス・グループ表示

クラスなどに設定されたアクセス指定子で分類して表示します。

なお，同じ種類のノードはアルファベット順で表示します。

ノード	備考
別名定義 (typedef)	
グローバル関数と変数	
マクロと定数	【CA78K0R】【CA78K0】
public *クラス名*注 1 *インタフェース名*注 1 *構造体名*注 1 *共用体名*注 2 *列挙体名*	【CC-RX】
protected *クラス名*注 1 *インタフェース名*注 1 *構造体名*注 1 *共用体名*注 2 *列挙体名*	【CC-RX】

ノード	備考
private *クラス名*注 1 *インタフェース名*注 1 *構造体名*注 1 *共用体名*注 2 *列挙体名*	【CC-RX】

注 1. 【CC-RX】
必要に応じて，基底型ノード／派生型ノード／内部型ノードを表示します。

2. 【CC-RX】
必要に応じて，内部型ノードを表示します。

（c）名前空間グループ表示

クラスなどが定義されている名前空間で分類して表示します。

なお，同じ種類のノードはアクセス指定子順（「(a) カテゴリ・グループ表示（デフォルト）」参照）とし，同一のアクセス指定子の場合はアルファベット順で表示します。

ノード	備考
*名前空間名*注 1	
別名定義 (typedef)	
グローバル関数と変数	
マクロと定数	【CA78K0R】【CA78K0】
*クラス名*注 2	【CC-RX】
*インタフェース名*注 2	
*構造体名*注 2	【CA850】以外
*共用体名*注 3	
列挙体名	

注 1. 名前空間に属さない関数やクラスなどについては（C ソース・ファイルを対象としている場合も含む），“*名前空間名*”を“(グローバル)”として同等の表示を行います。
また，無名名前空間の場合は，“*名前空間名*”を“(無名 :< *ファイル名* >)”として同等の表示を行います。

2. 【CC-RX】
必要に応じて，基底型ノード／派生型ノード／内部型ノードを表示します。

3. 【CC-RX】
必要に応じて，内部型ノードを表示します。

（d）ファイル・グループ表示

クラスなどが定義されているファイルで分類して表示します。

なお，同じ種類のノードはアクセス指定子順（「(a) カテゴリ・グループ表示（デフォルト）」参照）とし，同一のアクセス指定子の場合はアルファベット順で表示します。

ノード	備考
ファイル名[注1]	
別名定義 (typedef)	
グローバル関数と変数	
マクロと定数	【CA78K0R】【CA78K0】
クラス名[注2]	【CC-RX】
インタフェース名[注2]	
構造体名[注2]	【CA850】以外
共用体名[注3]	
列挙体名	

注 1.　ファイルの種類により付与されるアイコンは異なります。

2.　【CC-RX】

必要に応じて，基底型ノード／派生型ノード／内部型ノードを表示します。

3.　【CC-RX】

必要に応じて，内部型ノードを表示します。

(e) アルファベット・グループ表示

クラスやアクセス指定子に関与せず，アルファベット順で表示します。

ノード	備考	
別名定義 (typedef)		
グローバル関数と変数		
マクロと定数	【CA78K0R】【CA78K0】	
クラス名[注1]	【CC-RX】	アルファベット順に表示
インタフェース名[注1]		
構造体名[注1]	【CA850】以外	
共用体名[注2]		
列挙体名		

注 1.　【CC-RX】

必要に応じて，基底型ノード／派生型ノード／内部型ノードを表示します。

2.　【CC-RX】

必要に応じて，内部型ノードを表示します。

注意 1.　別名定義 (typedef) ノード／グローバル関数と変数ノード／マクロと定数ノードは，関数／変数用リスト・エリアに表示する情報が存在しない場合は表示されません。

2. **【CC-RX】**
   C++ ソース・ファイルを対象とする場合，名前空間の別名は表示されません。
3. **【CX】**
   ビット・フィールド構造体の型と共用体の型は表示されません。
4. **【CA850】定数（const 宣言されている変数）を区別できないため，マクロと定数ノードは表示されません（変数として扱います）。**

備考 1. *xxx 名*ノード（*ファイル名*ノードを除く）にマウス・カーソルを重ねることにより，属している名前空間名をポップアップ表示します。
ただし，グローバル名前空間の場合は“-”を，無名名前空間の場合は“<unnamed>”を表示します。また，*名前空間名*ノードを選択している場合では，対象の名前空間が属している名前空間名（上位の名前空間名）を表示します。

2. *名前空間名*ノードについて
   内部に名前空間を含む場合，“*包括する名前空間名* :: *内部の名前空間名*”を表示します。
   例 1. namespace Name : Name
   2. 内部の名前空間の場合 : Name::SubName
3. *クラス名*ノードについて
   テンプレート・クラスの場合，クラス名に型情報を付与して表示します。
   また，内部クラスの場合は，“*包括するクラス名* :: *内部クラス名*”を表示します。
   例 1. class Sub : Sub
   2. template<class T> class List : List<T>
   3. 内部クラスの場合 : Main::SubInMain
4. *クラス／インタフェース／構造体*について
   基底型／派生型／内部型は，直系のクラス／インタフェースのみ表示します（複数存在する場合はすべてを表示）。
5. *クラス名／インタフェース名／構造体名／共用体名／列挙体名*が無名（タグ名がない）の場合は，これらを“<unnamed_*N*>”の形式で表示します（*N*：1 から出現順に自動的に付与される番号）。
6. 基底型ノード／派生型ノード／内部型ノード以下で表示されるノードをダブルクリックすることにより，対応するノード（同一ツリー内の*クラス名*ノード／*インタフェース名*ノードなど）にジャンプします。
7. 現在選択しているノードの定義箇所，または宣言箇所へジャンプすることができます（「2.7 定義箇所へジャンプする」／「2.8 宣言箇所へジャンプする」参照）。

（3）関数／変数用リスト・エリア

クラス／カテゴリ用ツリー・エリアで選択しているノードに関する情報（該当ノードで定義されている関数名／変数名など）を一覧表示します。

クラス／カテゴリ用ツリー・エリアで選択しているノードと，このエリアで表示する内容の関係は次のとおりです。

選択ノード	このエリアでの表示内容	備考
別名定義 (typedef)	*別名定義名*	

選択ノード	このエリアでの表示内容	備考
グローバル関数と変数	*グローバル関数名*	
	スタティック関数名	
	グローバル変数名	
	ファイル内スタティック変数名	
マクロと定数	*マクロ名*	【CA78K0R】【CA78K0】
	定数	
クラス名 *インタフェース名* *構造体名*（C++ ソース・ファイル）	*別名定義名*	【CC-RX】
	メンバ関数名	
	クラス変数名	
	インスタンス変数名	
	定数	
共用体名（C++ ソース・ファイル）	*別名定義名*	【CC-RX】
	メンバ関数名	
	インスタンス変数名	
	定数	
構造体名（C++ ソース・ファイル）	*メンバ変数名*	【CA850】以外
共用体名（C++ ソース・ファイル）	*メンバ変数名*	
列挙体名	*列挙体メンバ名*[注]	
名前空間名 *ファイル名* 上記以外	なし	

注　【CX】

列挙体のメンバは表示されません。

注意　**表示対象となる関数／変数が存在しない場合，またはクラス／カテゴリ用ツリー・エリアでノードを選択していない場合は，このエリアは何も表示されません。**

備考 1.　*xxx 名*ノード（*ファイル名*ノードを除く）にマウス・カーソルを重ねることにより，属している名前空間名をポップアップ表示します。

ただし，グローバル名前空間の場合は“-”を，無名名前空間の場合は“<unnamed>”を表示します。また，*名前空間名*ノードを選択している場合では，対象の名前空間が属している名前空間名（上位の名前空間名）を表示します。

2.　【CC-RH】【CC-RX】【CX】

関数名には，引数情報を付与して表示します。

例 1.　int main(void)　：　main(void)

2.　void main_sub01(int local_a, int local_b, int local_c)　：　main_sub01(int, int, int)

3.　int function01(int arg01) const　：　function01(int) const

また，テンプレート関数【CC-RX】については，テンプレート関数を定義している箇所と使用している箇所を個別に表示します。

例 1. template<class T> T max(T a, T b) （テンプレート定義行） : max<T:1>()

2. int max(int a, int b)（テンプレート使用行） : max<T:1>(int, int)

3. 【CC-RX】

次のアクセス指定子順に表示を行い，アクセス指定子の種類を示すアイコンを重ねて表示します（同一のアクセス指定子の場合はアルファベット順で表示）。

ただし，ツリーの分類方法をアルファベット・グループ表示としている場合は，すべてアルファベット順となります。

アクセス指定子	アイコン
public	なし
protected	
private	

4. 【CA78K0】【CA78K0R】

“型（構造体／共用体／列挙体）の定義のみ・変数宣言なし”と“無名の型（構造体／共用体／列挙体）”を続けて記述した場合，“無名の型のメンバ”を“型の定義のみ・変数宣言なしのメンバ”として扱います。次に示す例の場合，def_only_str 構造体のメンバとして mem01/mem02/num01/num02 を，このエリアに表示します。

```
struct def_only_str {
    char    mem01;
    short   mem02;
};
struct {
    short   num01;
    int     num02;
} anonymous_struct
```

5. 【CA78K0】【CA78K0R】

構造体／共用体／列挙体の中で構造体／共用体／列挙体を定義する記述をした場合，定義行以降のメンバを構造体／共用体／列挙体の中で定義した構造体／共用体／列挙体のメンバとして扱います。次に示す例の場合，SType 構造体のメンバとして mem01 を，InnerType 構造体のメンバとして inn01/mem02/mem03 をこのエリアに表示します。

```
struct SType {
    char    mem01;
    struct  InnerType {
        char    inn01;
    } mem02;
    int     mem03;
} struct01;
```

6. 現在選択しているノードの定義箇所，または宣言箇所へジャンプすることができます（「2.7 定義箇所へジャンプする」／「2.8 宣言箇所へジャンプする」参照）。

[ツールバー]

ツールバー上の各ボタン，および機能は次のとおりです。

	最新情報を取得し，表示内容を更新します。

[[編集] メニュー（クラス／メンバ パネル専用部分)]

クラス／メンバ パネル専用の［編集］メニューの各項目，および機能は次のとおりです。

コピー	選択しているノードの文字列をクリップ・ボードにコピーします。
すべて選択	関数／変数用リスト・エリアのノードをすべて選択状態にします。

[コンテキスト・メニュー]

このパネル上において，マウスを右クリックすることにより表示されるコンテキスト・メニューの各項目，および機能は次のとおりです。

ソースへジャンプ	選択しているノードが定義されているソース・ファイルをエディタ パネル上にオープンします（「2.7　定義箇所へジャンプする」参照)。
ソースの宣言へジャンプ	選択しているノードが宣言されているソース・ファイルをエディタ パネル上にオープンします（「2.8　宣言箇所へジャンプする」参照)。
逆アセンブルへジャンプ	関数／変数用リスト・エリアで選択しているノード注の開始アドレスに対応する逆アセンブル・データを逆アセンブル パネル（逆アセンブル 1）上にオープンします（「2.7　定義箇所へジャンプする」参照)。 ただし，デバッグ・ツールと切断時は無効となります。
メモリへジャンプ	関数／変数用リスト・エリアで選択しているノード注の開始アドレスに対応するメモリ・リストをメモリ パネル（メモリ 1）上にオープンします（「2.7　定義箇所へジャンプする」参照)。 ただし，デバッグ・ツールと切断時は無効となります。
関数／変数一覧へジャンプ	関数一覧 パネル／変数一覧 パネルをオープンし，このパネルで選択しているノード注の関数／変数を選択状態にします。
コピー	選択している項目の内容を文字列としてクリップ・ボードにコピーします。

注 【CC-RX】

インスタンス変数を示すノードを選択している場合は無効となります。

値の推移（ズーム）パネル

解析グラフ パネルの［値の推移］タブで表示しているグラフ上で範囲設定した箇所をズーム表示します。

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，“値の推移（ズーム）1”，“値の推移（ズーム）2”，“値の推移（ズーム）3”，“値の推移（ズーム）4”の名称で識別され（タイトルバーに表示），それぞれ個別に指定した範囲のズーム表示を行うことができます。

なお，このタブ上でグラフを表示するための操作手順は，「(c) ズーム表示」を参照してください。

注意 1.　プログラム実行中における，グラフのリアルタイム表示更新は行いません。

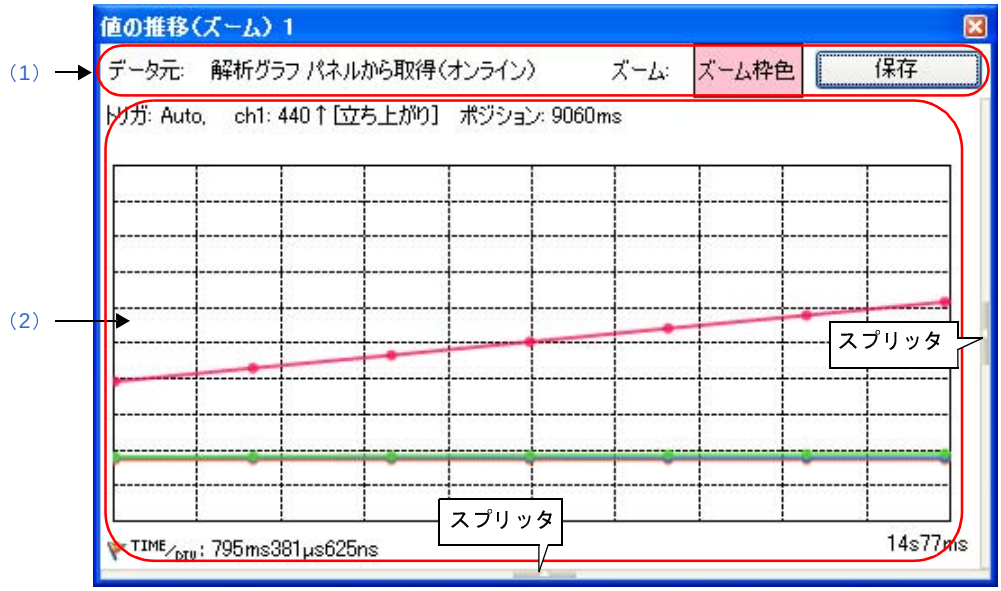
2.　このパネル上でグラフ化対象の登録／削除は行えません。

備考 1.　このパネルは次のタイミングで表示内容が更新されます。

- 解析グラフ パネルの［値の推移］タブのグラフ表示更新時（リアルタイム表示更新を除く）
- ズーム表示範囲の設定／解除時
- 解析グラフ・データ・ファイル（*.mtac）の読み込み時

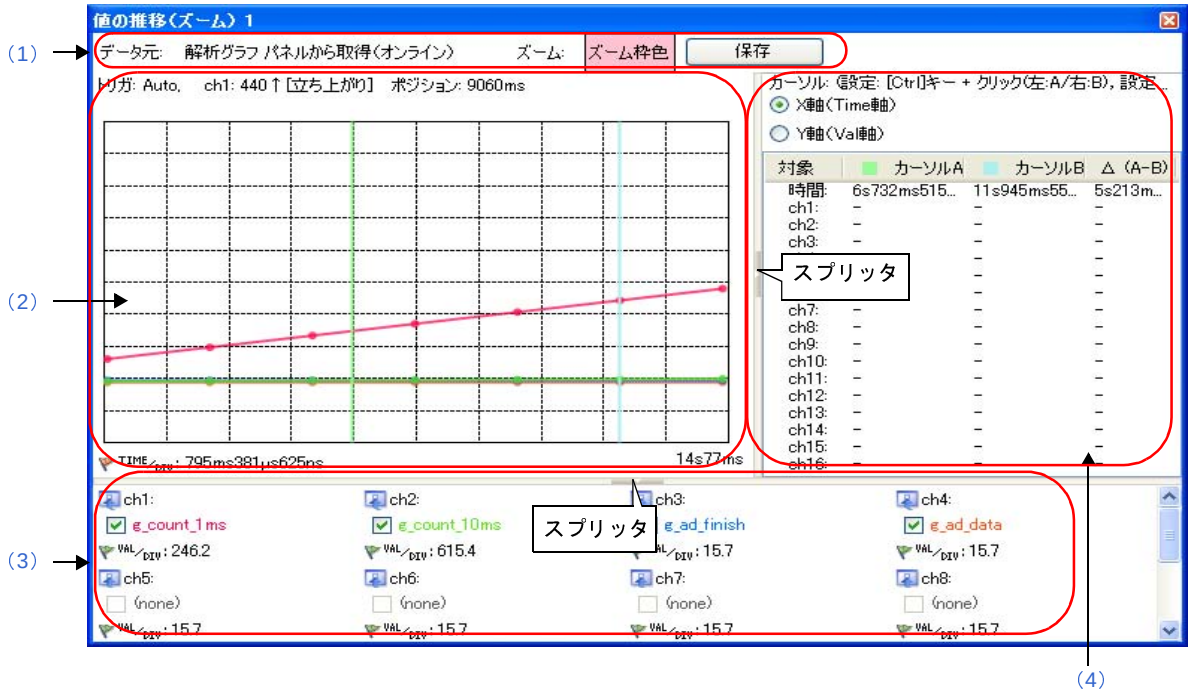
2.　現在の表示内容を解析グラフ・データ・ファイル（*.mtac）として保存することにより，表示しているグラフを復帰させることができます（「(6) グラフを復帰するためのグラフ・データを保存する」参照）。

図 A―25　値の推移（ズーム）パネル（デフォルト）



スプリッタ上の中央のマークをクリックすることにより，チャネル情報エリアとカーソル情報エリアを表示／非表示することができます。

図 A―26　値の推移（ズーム）パネル（全エリア表示）



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］

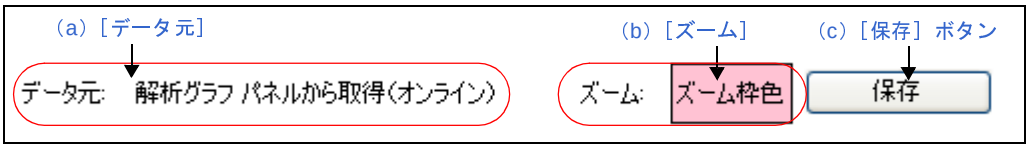
［オープン方法］

- 解析グラフ パネルの［値の推移］タブにおいて，ズーム *1* ～ *4* チェック・ボックスをチェック

［各エリアの説明］

(1) グラフ・コントロール・エリア

図 A―27　グラフ・コントロール・エリア



(a)［データ元］

現在表示しているグラフのデータ元を表示します。

解析グラフ パネルから取得（オンライン）	現在表示している［値の推移］タブ上のグラフをデータ元としています。
ファイルから復元（オフライン）	解析グラフ・データ・ファイル（*.mtac）から読み込んだ情報をデータ元としています。

（b）［ズーム］

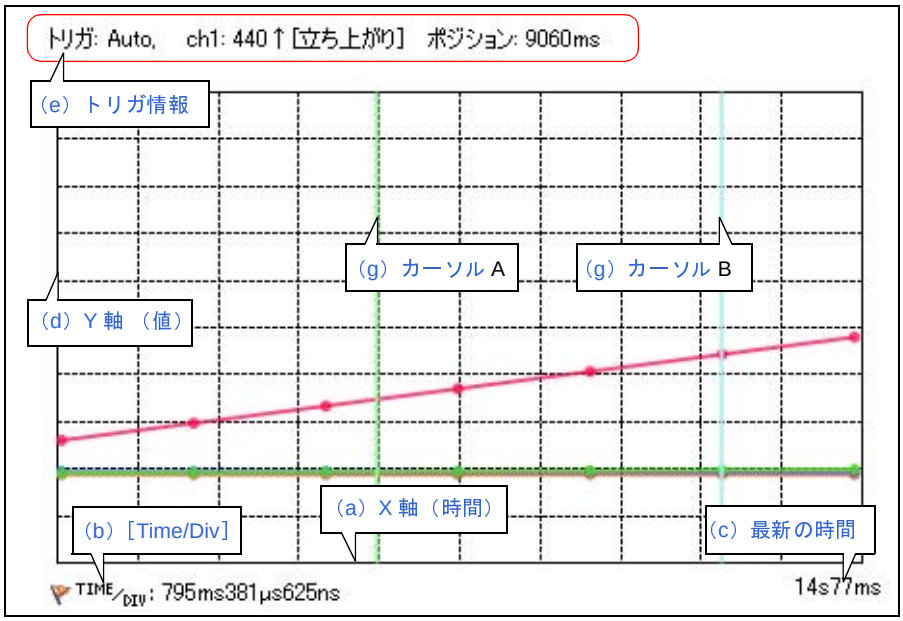
現在，プロパティ パネルの［値の推移］タブ上の［全般］カテゴリ内［ズーム枠 *1* ～ *4* の色］プロパティにおいて指定している，ズーム枠の色を示します。

（c）［保存］ボタン

現在表示している内容を指定したファイルに保存するために，名前を付けて保存 ダイアログをオープンします（「2. 14　解析情報をファイルに保存する」参照）。

（2）グラフ表示エリア

図 A―28　グラフ表示エリア



解析グラフ パネルの［値の推移］タブで表示しているグラフ上で範囲設定した箇所をズーム表示します。各部の機能は，次のとおりです。

（a）X 軸（時間）

経過時間を示します。

（b）［Time/Div］

単位グリッドあたりの時間（解析グラフ パネルの［値の推移］タブで設定したズーム表示範囲の総時間の 10%分）を表示します（変更不可）。

(c) 最新の時間

解析グラフ パネルの［値の推移］タブで設定したズーム表示範囲に応じた最新の時間を表示します。

(d) Y 軸 （値）

グラフ化対象の値を示します。

単位グリッドあたりの値（Val/Div）は，各チャネルごとに，解析グラフ パネルの［値の推移］タブで設定したズーム表示範囲の数値（最大値と最小値の差分）の 10%分となります（変更不可）。

なお，ズーム表示範囲内に原点（数値 =0）が存在する場合に限り，それを表す軸線をチャネルごとの描画色を用いた点線で表示します。

注意 データ元から取得したグラフ・データが，上記の表示範囲の上限／下限を越えた場合，該当区間のグラフ表示は行いません。

(e) トリガ情報

トリガ機能を使用している場合（「(3) トリガ機能を使用する」参照），プロパティ パネルの［値の推移］タブ上の［トリガ］カテゴリ内で指定されている各プロパティの内容を表示します。

トリガ機能を使用していない場合は，“-”を表示します。

(f) ポップアップ表示

グラフ上の遷移箇所にマウス・カーソルを重ねることにより，その箇所の情報をポップアップ表示します。表示形式についての詳細は，「(i) ポップアップ表示」を参照してください。

注意 プログラム実行中は，この機能は無効となります。

(g) カーソル

X 軸（時間），または Y 軸 （値）を対象として，時間／数値を確認するための 2 本のカーソル（カーソル A/ カーソル B）です。

カーソル選択ボタンの［X 軸（Time 軸）］ボタンを選択することにより，X 軸（時間）を対象としたカーソル計測を行い，また，［Y 軸 （Val 軸）］ボタンを選択することにより，Y 軸 （値）を対象としたカーソル計測を行います。

計測結果は，カーソル情報エリアに一覧表示されます。

カーソルの表示／設定方法は次のとおりです（カーソルはデフォルトで非表示です）。

カーソル	表示／設定	非表示／設定解除
カーソル A	［Ctrl］キー + クリック	［Ctrl］キー + ダブルクリック
カーソル B	［Ctrl］キー + 右クリック	［Ctrl］キー + 右ボタンのダブルクリック

注意 プログラム実行中は，各カーソルは非表示となります。

(3) チャネル情報エリア

このエリアは，スプリッタを移動することにより表示領域を変更することができます。

また，スプリッタ上の中央のマークをクリックすることにより，このエリアの表示／非表示を切り替えることができます（デフォルトでは表示されません）。

図 A―29　チャネル情報エリア

（a）変数名チェック・ボックス

現在，各チャネルにグラフ化対象として登録されている変数名（レジスタ名／アドレス式などを含む）をチェック・ボックスとして表示します（未登録の場合は“(none)”を表示）。

なお，変数名の文字色は，グラフの描画色と同一です。

チェック・ボックスをチェックすることにより，その変数のグラフを表示します（複数選択可）。

デフォルトではデータ元の設定値が反映されます。

ただし，プログラム実行中は無効となります。

（b）［Val/Div］

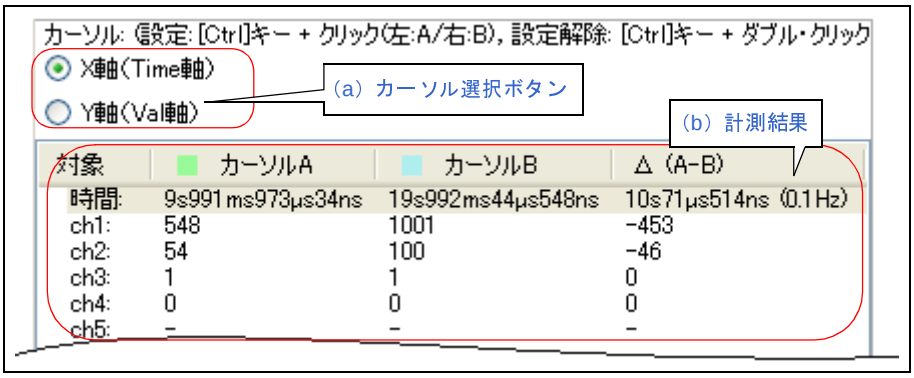
現在，プロパティ パネルの［値の推移］タブ上の［チャネル *1* ～ *16*］カテゴリ内［1 グリッドあたりの値 [Val/Div] *1* ～ *16*］プロパティで指定している“*単位グリッドあたりの値*”を表示します。

（4）カーソル情報エリア

このエリアは，スプリッタを移動することにより表示領域を変更することができます。

また，スプリッタ上の中央のマークをクリックすることにより，このエリアの表示／非表示を切り替えることができます（デフォルトでは表示されません）。

図 A―30　カーソル情報エリア



（a）カーソル選択ボタン

カーソル計測を行う際の対象軸を選択します。

ただし，プログラム実行中は無効となります。

X 軸（Time 軸）	カーソルの対象を X 軸（時間）に設定します（デフォルト）。
Y 軸 （Val 軸）	カーソルの対象を Y 軸 （値）に設定します。

（b）計測結果

現在のカーソル A/ カーソル B の位置から求まる次の計測結果を表示します。

ただし，プログラム実行中は非表示となります。

時間	- X 軸（時間）を対象とした場合 カーソル A が示す時間，カーソル B が示す時間，および両カーソル間の差分時間（差分値より求まる周期 [Hz]）を表示します。 - Y 軸 （値）を対象とした場合 “-”を表示します。
ch*1* ～ *16*	カーソル A が示す時間の値，カーソル B が示す時間の値，および両カーソル間の差分値を表示します。 ただし，値が取得できない場合は，“-”を表示します。

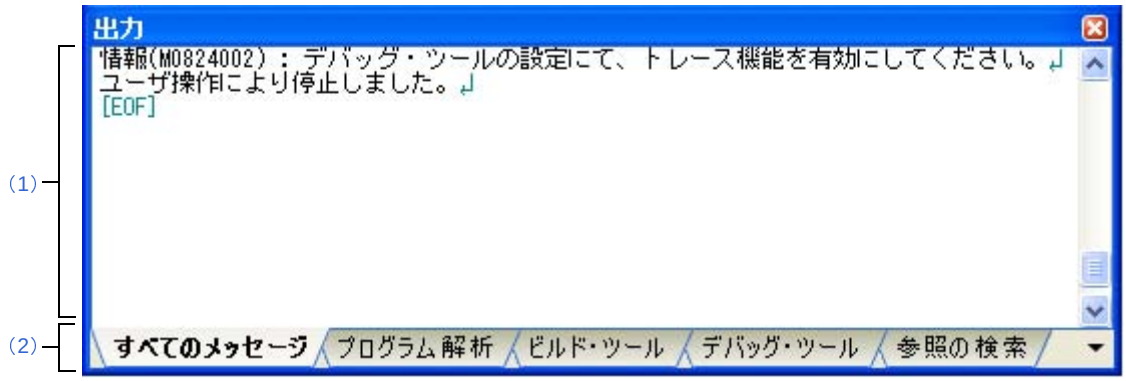
出力 パネル

CubeSuite+ が提供している各種コンポーネント（解析ツールを含む，設計ツール／ビルド・ツール／デバッグ・ツールなど）から出力されるメッセージ，または指定した関数／変数の参照箇所一覧を表示します。

なお，関数／変数の参照箇所一覧の出力方法についての詳細は，「2. 11　参照箇所を一覧表示する」を参照してください。

備考　ツールバーの 100% ，または［Ctrl］キーを押下しながらマウス・ホイールを前後方に動かすことにより，本パネルの表示を拡大／縮小することができます。

図 A―31　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］の選択

［各エリアの説明］

（1）メッセージ・エリア

CubeSuite+ が提供している各種コンポーネント（解析ツールを含む，設計ツール／ビルド・ツール／デバッグ・ツールなど）から出力されたメッセージ，または指定した関数／変数の参照箇所一覧を表示します。

表示内容についての詳細は，各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，メッセージの出力元が切り替わります。

解析ツールでは，次のタブを使用します。

- ［すべてのメッセージ］タブ
- ［プログラム解析］タブ
- ［参照の検索］タブ

備考 新しいメッセージ，または参照箇所一覧が出力された場合，タブ名の直前に“*”マークが表示されます。

[[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］メニューの各項目，および機能は次のとおりです。

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。 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 出力 - *タブ名* を保存 ...］の選択と同等の動作となります。
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。

[[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。

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 ダイアログをオープンします。
置換 ...	検索・置換 ダイアログをオープンします。

[コンテキスト・メニュー]

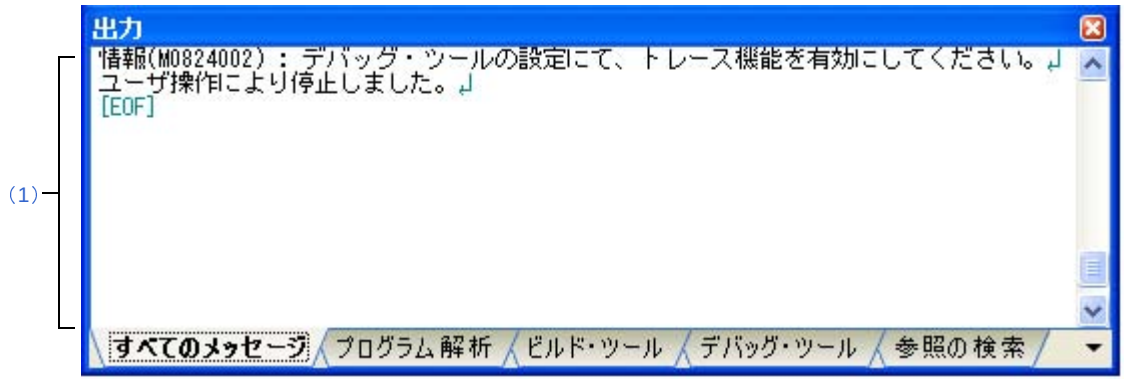
このパネル上において，マウスを右クリックすることにより表示されるコンテキスト・メニューの各項目，および機能は次のとおりです。

コピー	選択している文字列をクリップ・ボードにコピーします。
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非表示となります。
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。

[すべてのメッセージ]タブ

CubeSuite+ が提供している全コンポーネント（解析ツールを含む，設計ツール／ビルド・ツール／デバッグ・ツールなど）から出力されるメッセージを表示します。

図 A―32 出力 パネル：[すべてのメッセージ]タブ



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]

[オープン方法]

- [表示]メニュー→[出力]の選択

[各エリアの説明]

(1) メッセージ・エリア

CubeSuite+ が提供している全コンポーネント（解析ツール含む，設計ツール／ビルド・ツール／デバッグ・ツールなど）から出力されたメッセージを表示します。

ただし，解析ツールが解析中に出力するメッセージの表示は行いません（[プログラム解析]タブ上でのみ表示）。

なお，メッセージの表示色は，出力メッセージの種別により，次のように異なります（表示の際の文字色／背景色は，オプション ダイアログにおける[全般 - フォントと色]カテゴリの設定に依存）。

メッセージ種別	表示例（デフォルト）			説明
通常メッセージ	AaBbCc	文字色	黒	何らかの情報を通知する際に表示されます。
		背景色	白	
警告メッセージ	AaBbCc	文字色	青	操作に対して，何らかの警告を通知する際に表示されます。
		背景色	標準色	

メッセージ種別	表示例（デフォルト）			説明
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	

備考 1. 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，該当ファイルの該当行番号を表示します。

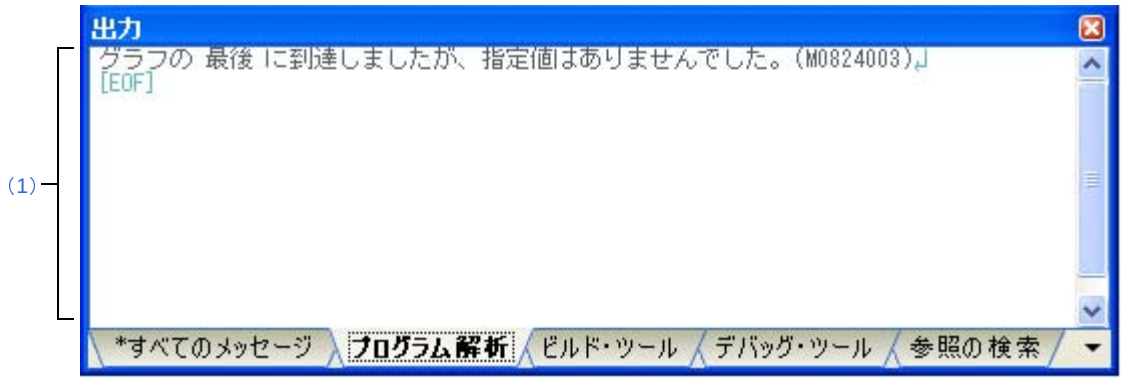
2. 警告メッセージ，またはエラー・メッセージを表示している行にキャレットがある状態で，コンテキスト・メニューの［メッセージに関するヘルプ］を選択，または［F1］キーを押下することにより，該当行のメッセージに関するヘルプを表示します。

3. ［ファイル］メニュー→［名前を付けて 出力 - すべてのメッセージ を保存 ...］を選択することにより，出力内容をテキスト・ファイル（*.txt）に保存することができます。

［プログラム解析］タブ

CubeSuite+ が提供している各種コンポーネント（解析ツールを含む，設計ツール／ビルド・ツール／デバッグ・ツールなど）から出力されるメッセージのうち，解析ツールが出力するメッセージを表示します。

図 A―33　出力 パネル：［プログラム解析］タブ



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］

［オープン方法］

- ［表示］メニュー→［出力］の選択

［各エリアの説明］

（1）メッセージ・エリア

CubeSuite+ が提供している各種コンポーネント（解析ツールを含む，設計ツール／ビルド・ツール／デバッグ・ツールなど）から出力されるメッセージのうち，解析ツールが出力したメッセージを表示します。

なお，メッセージの表示色は，出力メッセージの種別により，次のように異なります（表示の際の文字色／背景色は，オプション ダイアログにおける［全般 - フォントと色］カテゴリの設定に依存）。

メッセージ種別	表示例（デフォルト）			説明
通常メッセージ	AaBbCc	文字色	黒	何らかの情報を通知する際に表示されます。
		背景色	白	
警告メッセージ	AaBbCc	文字色	青	操作に対して，何らかの警告を通知する際に表示されます。
		背景色	標準色	
エラー・メッセージ	AaBbCc	文字色	赤	致命的なエラー，または操作ミスにより実行が不可能な場合に表示されます。
		背景色	薄グレー	

備考 1. 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，該当ファイルの該当行番号を表示します。

2. 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，または［F1］キーを押下することにより，該当行のメッセージに関するヘルプを表示します。

3. ［ファイル］メニュー→［名前を付けて 出力 - プログラム解析 を保存 ...］を選択することにより，出力内容をテキスト・ファイル（*.txt）に保存することができます。

［参照の検索］タブ

指定した関数／変数の参照箇所一覧を表示します。

なお，関数／変数の参照箇所一覧の出力方法についての詳細は，「2. 11　参照箇所を一覧表示する」を参照してください。

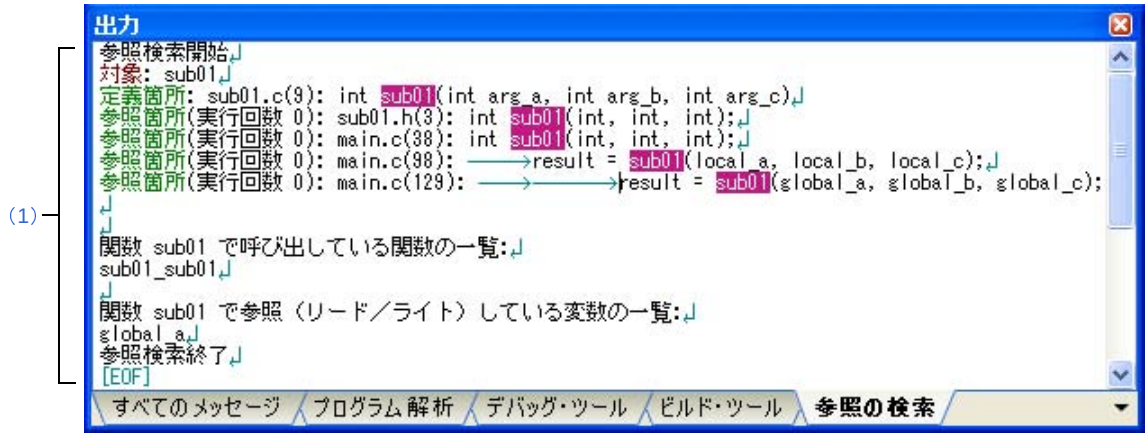
注意 1.　このタブは，参照箇所を一覧表示する操作を一度も実行していない場合は出現しません。

2.　C/C++ ソース・ファイル中の“#if”／“#ifdef”などで，コンパイル時にプリプロセッサにより除外されるコードにおいて参照されている箇所は参照箇所として出力されません。

3.　【CC-RH】【CC-RX】【CX】
関数ポインタへの代入による関数の参照は参照箇所として出力されません。

4.　参照箇所の検索を行うごとに，出力された参照箇所一覧はクリアされます。

図 A―34　出力 パネル：［参照の検索］タブ



ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］

［オープン方法］

- ［表示］メニュー→［出力］の選択

［各エリアの説明］

（1）メッセージ・エリア

関数一覧 パネル／変数一覧 パネルにおいて，関数／変数を参照している箇所を検索した際に，その結果として，次の参照箇所一覧を表示します（「2. 11　参照箇所を一覧表示する」参照）。

備考 1. 出力結果上の関数／変数をダブルクリックすることにより，エディタ パネルをオープンし，その関数／変数が定義されているソース・テキスト箇所へジャンプします。

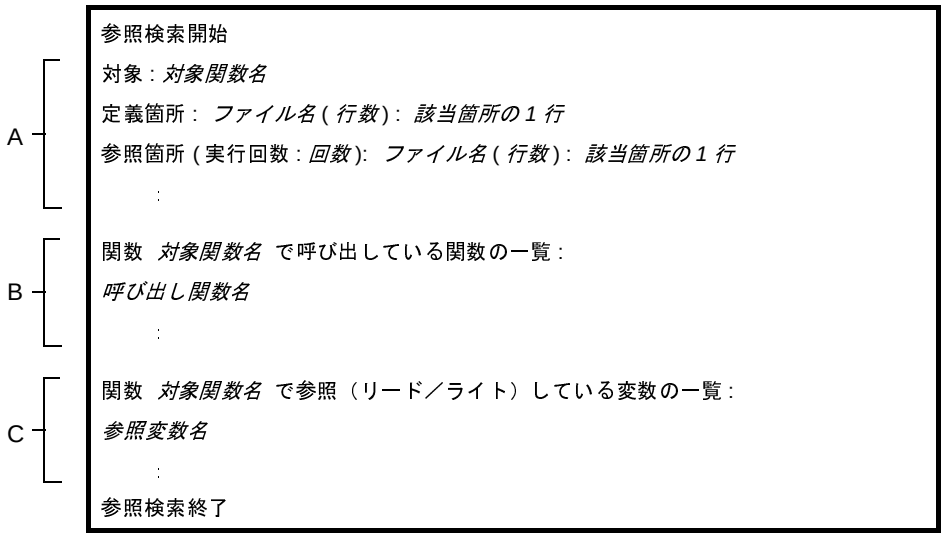
2. 対象関数名／対象変数名の強調表示色は，オプション ダイアログにおける［全般 - フォントと色］カテゴリ内［強調］項目の設定に依存します。

3. ［ファイル］メニュー→［名前を付けて 出力 - 参照の検索 を保存 ...］を選択することにより，参照箇所一覧をテキスト・ファイル（*.txt）に保存することができます。

（a）関数の参照箇所一覧の出力フォーマット

検索結果として出力される内容は，次の各部で構成されます。

図 A—35 関数の参照箇所一覧の出力フォーマット



A	対象関数の定義箇所と対象関数を参照している箇所の一覧		
	対象	検索の対象となった関数名を示します。	
	定義箇所	対象関数の定義箇所を示します。 ただし，クロス・リファレンス・ファイルから情報が取得できない場合は“なし”を表示します。	
		ファイル名 (行数)	該当箇所が存在するファイル名[注1]を示します。 （）内には，ファイル内における行数（行番号）を示します。
		該当箇所の 1 行	該当箇所の 1 行をファイルから抜き出し表示します。 この際，対象関数名が強調表示されます。
	参照箇所	対象関数を参照している箇所を列挙します。動的解析情報が存在する場合は，（）内に対象関数の実行回数を示します。 ただし，クロス・リファレンス・ファイルから情報が取得できない場合は“なし”を表示します。	
		ファイル名 (行数)	該当箇所が存在するファイル名[注1]を示します。 （）内には，ファイル内における行数（行番号）を示します。
		該当箇所の 1 行	該当箇所の 1 行をファイルから抜き出し表示します。 この際，対象関数名が強調表示されます。

B	対象関数内で呼び出している関数の一覧	
	呼び出し関数名	対象関数内で呼び出している関数名を列挙します。 ただし，呼び出している関数が存在しない場合は“なし”を表示します。
C	対象関数内で参照（リード／ライト）している変数の一覧	
	参照変数名	対象関数内で参照（リード／ライト）している変数名を列挙します[注2]。 ただし，参照している変数が存在しない場合は“なし”を表示します。

注 1. 関数一覧 パネルにおいて，［ファイル・パス］項目を表示している場合（デフォルトでは表示されません），ファイルの絶対パスを表示します。

2. 【CA850】【CA78K0R】【CA78K0】

関数 A の定義と関数 B の定義の間に変数 C の参照を記述している場合，関数 A で参照している変数として変数 C が出力されます。

また，変数の参照以外にも，関数プロトタイプ宣言，または関数の参照 (関数ポインタ変数への代入) も同様の動作となります。

図 A―36 関数の参照箇所一覧の出力例（動的解析情報が存在しない場合）

```
対象: sub01
定義箇所: sub01.c(9): int sub01(int arg_a, int arg_b, int arg_c)
参照箇所: sub01.h(3): int sub01(int, int, int);
参照箇所: main.c(38): int sub01(int, int, int);
参照箇所: main.c(98): →result = sub01(local_a, local_b, local_c);
参照箇所: main.c(129): →――――→result = sub01(global_a, global_b, global_c);

関数 sub01 で呼び出している関数の一覧:
sub01_sub01

関数 sub01 で参照（リード／ライト）している変数の一覧:
global a
```

図 A―37 関数の参照箇所一覧の出力例（動的解析が存在する場合）

```
対象: sub01
定義箇所: sub01.c(9): int sub01(int arg_a, int arg_b, int arg_c)
参照箇所(実行回数 0): sub01.h(3): int sub01(int, int, int);
参照箇所(実行回数 0): main.c(38): int sub01(int, int, int);
参照箇所(実行回数 0): main.c(98): ――→result = sub01(local_a, local_b, local_c);
参照箇所(実行回数 14): main.c(129): ――→――――→result = sub01(global_a, global_b, global_c);

関数 sub01 で呼び出している関数の一覧:
sub01_sub01

関数 sub01 で参照（リード／ライト）している変数の一覧:
global a
```

(b) 変数の参照箇所一覧の出力フォーマット

検索結果として出力される内容は，対象変数を定義している箇所，および対象変数を参照している箇所の一覧で構成されます。

図 A―38 変数の参照箇所一覧の出力フォーマット

```
参照検索開始
対象：対象変数名
定義箇所： ファイル名 ( 行数 ) : 該当箇所の 1 行
参照箇所 ( リード／ライト回数 : 回数 ): ファイル名 ( 行数 ) : 該当箇所の 1 行
      :
参照検索終了
```

対象	検索の対象となった変数名を示します。	
定義箇所	対象変数の定義箇所を示します。 ただし，クロス・リファレンス・ファイルから情報が取得できない場合は“なし”を表示します。	
	ファイル名 (行数)	該当箇所が存在するファイル名[注]を示します。 （）内には，ファイル内における行数（行番号）を示します。
	該当箇所の 1 行	該当箇所の 1 行をファイルから抜き出し表示します。 この際，対象変数名が強調表示されます。
参照箇所	対象変数を参照している箇所を列挙します。動的解析情報が存在する場合は，（）内に対象変数のリード／ライト回数を示します。 ただし，クロス・リファレンス・ファイルから情報が取得できない場合は“なし”を表示します。	
	ファイル名 (行数)	該当箇所が存在するファイル名[注]を示します。 （）内には，ファイル内における行数（行番号）を示します。
	該当箇所の 1 行	該当箇所の 1 行をファイルから抜き出し表示します。 この際，対象変数名が強調表示されます。

注 変数一覧 パネルにおいて，［ファイル・パス］項目を表示している場合（デフォルトでは表示されません），ファイルの絶対パスを表示します。

備考 1. extern 宣言が記述されている変数の行の扱いは，使用するコンパイラにより次のように異なります。

- 【CC-RH】【CC-RX】【CX】
  extern 宣言が記述されている変数の行を参照箇所として扱います。
- 【CA850】
  extern 宣言が記述されている変数の行を定義箇所として扱います。
- 【CA78K0R】【CA78K0】
  extern 宣言が記述されている変数の行を参照箇所として扱います。
  また，同一ファイル内で extern 宣言が記述されている行と変数の定義行が存在する場合は，定義箇所は“なし”と表示します。

2. 【CX】
変数定義行において，代入文が記述されている行（“int variable = 10”など）も参照箇所として扱います。

図 A―39　変数の参照箇所一覧の出力例（動的解析情報が存在しない場合）

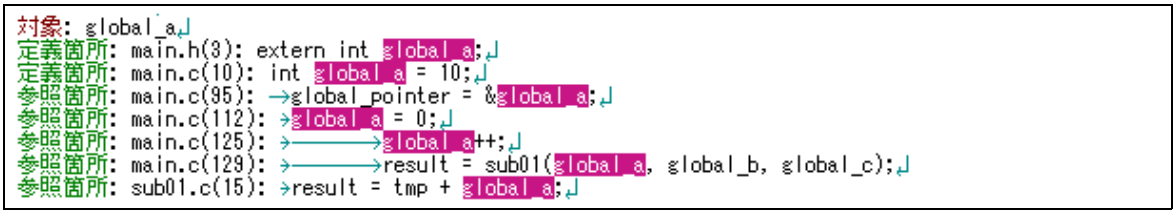
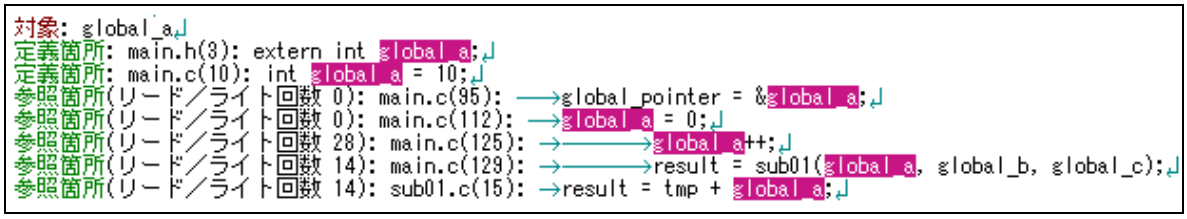


図 A―40　変数の参照箇所一覧の出力例（動的解析が存在する場合）



解析対象外ファイルを指定 ダイアログ

解析ツールが解析の対象外とするファイルを指定します（「1. 1. 1　解析対象」参照）。

このダイアログで解析対象外と指定されたファイル内の関数情報／変数情報は，次のパネル上では表示されません。

- 関数一覧 パネル
- 変数一覧 パネル
- コール・グラフ パネル
- クラス／メンバ パネル

図 A―41　解析対象外ファイルを指定 ダイアログ

ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］
- ［機能ボタン］

［オープン方法］

- プロパティ パネルの［設定］タブにおいて，［解析対象］カテゴリ内［解析対象外ファイル］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) ヘッダ・エリア

ファイル情報の項目名（ファイル名／カテゴリ／定義シンボル数）を表示します。

各項目名を，マウスにより任意の列へ直接ドラッグ・アンド・ドロップすることにより，表示項目（列）の順序を並び替えることができます。

また，各項目名をクリックすることにより，ファイル情報表示エリアの内容をソート表示することができます（クリックの繰り返しにより，昇順表示／降順表示／デフォルト表示（プロジェクト・ツリー上の順番）が切り替わります）。この際のソートは，ソート対象の項目の情報値が数値（10 進数 /16 進数）の場合は数値の大小により行い，それ以外の場合（文字列など）は，文字コード順に行います。

注意　表示項目（列）の並び替え，およびソート表示は保持されません。
このダイアログのオープン時は，常にデフォルトの状態で表示を行います。

(2) ファイル情報表示エリア

プロジェクトに現在登録されている C/C++ ソース・ファイル名，およびそのファイル情報をリスト表示します。

(a) [ファイル名]

C/C++ ソース・ファイル名，またはカテゴリ名（“()”内に表示）を表示します。
各項目のチェック・ボックスの指定により，解析対象外とするファイルを設定することができます。

☑	このファイルを解析の対象から外します。
☐	このファイルを解析の対象とします。

備考 1. カテゴリ名のチェック・ボックスをチェックした場合は，そのカテゴリに属するすべてのファイル名のチェック・ボックスがチェックされます。

2. ファイル名にマウス・カーソルを重ねると，そのファイルの絶対パス，およびプロジェクト・ツリーに表示されているツリー・ノードのファイル名を含めたパスを表示します。

(b) [カテゴリ]

プロジェクト・ツリーにおいて，対応する C/C++ ソース・ファイルが登録されているカテゴリ名を表示します。

なお，[ファイル名] においてカテゴリ名を表示している場合は，“ファイル”を表示します（ルート・カテゴリの場合は“-”を表示）。

(c) [定義シンボル数]

対応する C/C++ ソース・ファイル／カテゴリの中で定義されている関数／変数の合計数（10 進数）を表示します。

ただし，定義シンボル数を取得できない場合は“-”を表示します。

(3) [全選択／全解除] チェック・ボックス

このチェック・ボックスをチェックすることにより，[ファイル] 列のすべてのチェック・ボックスがチェック状態になります。

また，チェック・ボックスのチェックを外すことにより，[ファイル] 列のすべてのチェック・ボックスのチェックが外れます。

	［ファイル］列のすべてのチェック・ボックスがチェック状態にあることを示します。
	［ファイル］列のチェック・ボックスが，すべてチェックされている状態ではない，またはすべてチェックが外れている状態ではないことを示します。
	［ファイル］列のすべてのチェック・ボックスのチェックが外れている状態にあることを示します（デフォルト）。

［機能ボタン］

ボタン	機能
OK	指定したファイルを解析対象外ファイルに設定し，このダイアログをクローズします。
キャンセル	ファイルの指定を中止し，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。

解析対象ファイルを指定 ダイアログ

解析ツールが解析の対象とするファイルを指定します（「1. 1. 1　解析対象」参照）。

このダイアログで解析対象と指定されたファイル内の関数情報／変数情報のみを，次のパネル上で表示します。

- 関数一覧 パネル
- 変数一覧 パネル
- コール・グラフ パネル
- クラス／メンバ パネル

図 A―42　解析対象ファイルを指定 ダイアログ

ここでは，次の項目について説明します。

- ［オープン方法］
- ［各エリアの説明］
- ［機能ボタン］

［オープン方法］

- プロパティ パネルの［設定］タブにおいて，［解析対象］カテゴリ内［解析対象ファイル］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) ヘッダ・エリア

ファイル情報の項目名（ファイル名／カテゴリ／定義シンボル数）を表示します。

各項目名を，マウスにより任意の列へ直接ドラッグ・アンド・ドロップすることにより，表示項目（列）の順序を並び替えることができます。

また，各項目名をクリックすることにより，ファイル情報表示エリアの内容をソート表示することができます（クリックの繰り返しにより，昇順表示／降順表示／デフォルト表示（プロジェクト・ツリー上の順番）が切り替わります）。この際のソートは，ソート対象の項目の情報値が数値（10 進数 /16 進数）の場合は数値の大小により行い，それ以外の場合（文字列など）は，文字コード順に行います。

注意　表示項目（列）の並び替え，およびソート表示は保持されません。
このダイアログのオープン時は，常にデフォルトの状態で表示を行います。

(2) ファイル情報表示エリア

プロジェクトに現在登録されている C/C++ ソース・ファイル名，およびそのファイル情報をリスト表示します。

(a) [ファイル名]

C/C++ ソース・ファイル名，またはカテゴリ名（“()”内に表示）を表示します。
各項目のチェック・ボックスの指定により，解析対象とするファイルを設定することができます。

☑	このファイルを解析の対象とします。
☐	このファイルを解析の対象から外します。

備考 1. カテゴリ名のチェック・ボックスをチェックした場合は，そのカテゴリに属するすべてのファイル名のチェック・ボックスがチェックされます。

2. ファイル名にマウス・カーソルを重ねると，そのファイルの絶対パス，およびプロジェクト・ツリーに表示されているツリー・ノードのファイル名を含めたパスを表示します。

(b) [カテゴリ]

プロジェクト・ツリーにおいて，対応する C/C++ ソース・ファイルが登録されているカテゴリ名を表示します。

なお，[ファイル名] においてカテゴリ名を表示している場合は，“ファイル”を表示します（ルート・カテゴリの場合は“-”を表示）。

(c) [定義シンボル数]

対応する C/C++ ソース・ファイル／カテゴリの中で定義されている関数／変数の合計数（10 進数）を表示します。

ただし，定義シンボル数を取得できない場合は“-”を表示します。

(3) [全選択／全解除] チェック・ボックス

このチェック・ボックスをチェックすることにより，[ファイル] 列のすべてのチェック・ボックスがチェック状態になります。

また，チェック・ボックスのチェックを外すことにより，[ファイル] 列のすべてのチェック・ボックスのチェックが外れます。

	［ファイル］列のすべてのチェック・ボックスがチェック状態にあることを示します。
	［ファイル］列のチェック・ボックスが，すべてチェックされている状態ではない，またはすべてチェックが外れている状態ではないことを示します。
	［ファイル］列のすべてのチェック・ボックスのチェックが外れている状態にあることを示します（デフォルト）。

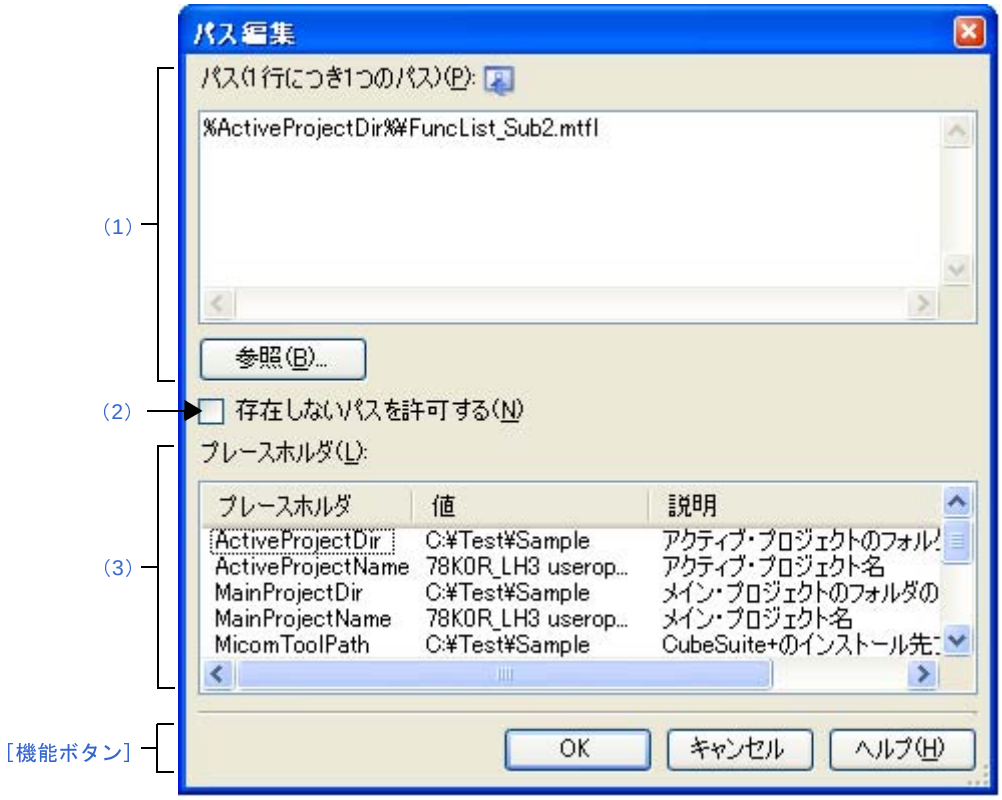
［機能ボタン］

ボタン	機能
OK	指定したファイルを解析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。
キャンセル	ファイルの指定を中止し，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。

パス編集 ダイアログ

情報ファイル（関数一覧ファイル（*.mtfl）／変数一覧ファイル（*.mtvl））をインポートする際の，ファイルを指定します（「2. 12　情報ファイルをインポート／エクスポートする」参照）。

図 A―43　パス編集 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［設定］タブにおいて，［インポート／エクスポート］カテゴリ内［インポート・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）パス編集エリア

パスを含むファイル名の編集，追加を行います。

(a)［パス (1 行につき 1 つのパス)］エリア

インポートする関数一覧ファイル（*.mtfl）／変数一覧ファイル（*.mtvl）のすべてを，1 行に 1 つずつ，パスを含めて指定します（1 行に 259 文字 /64 行まで指定可）。相対パスによる指定の場合は，プロジェクト・フォルダを基点として指定します。

なお，パスを含むファイル名の追加は，次の方法でも行うことができます。

- ［参照 ...］ボタンをクリックしたファイルの選択
- エクスプローラなどからファイルをドラッグ・アンド・ドロップ

(b)［参照 ...］ボタン

インポートするファイルを指定するためのダイアログをオープンします。

ファイルを選択すると，［パス (1 行につき 1 つのパス)］にファイル名を追加します。

注意 絶対パスで非常に長いパスを相対パスで指定すると，［OK］ボタンのクリック時にエラーになる場合があります。その場合は，絶対パスで指定してください。

(2)［存在しないパスを許可する］チェック・ボックス

☑	［OK］ボタンをクリックした際に，指定したパスやファイル名のチェックを行います。
☐	［OK］ボタンをクリックした際に，指定したパスやファイル名のチェックを行いません（デフォルト）。

(3)［プレースホルダ］エリア

このダイアログの呼び出し元に指定可能なプレースホルダの一覧を表示します（昇順）。

行をダブルクリックすると，プレースホルダの前後に“%”を付加してパス編集エリアに表示します。

(a)［プレースホルダ］

プレースホルダを表示します。

(b)［値］

プレースホルダの置換後の文字列を表示します。

(c)［説明］

プレースホルダの説明を表示します。

注意 このエリアは，このダイアログの呼び出し元がプレースホルダに対応している場合のみ表示されます。

備考 指定可能なプレースホルダは，このダイアログの呼び出し元によって異なります。

具体的なプレースホルダについては，呼び出し元の説明を参照してください。

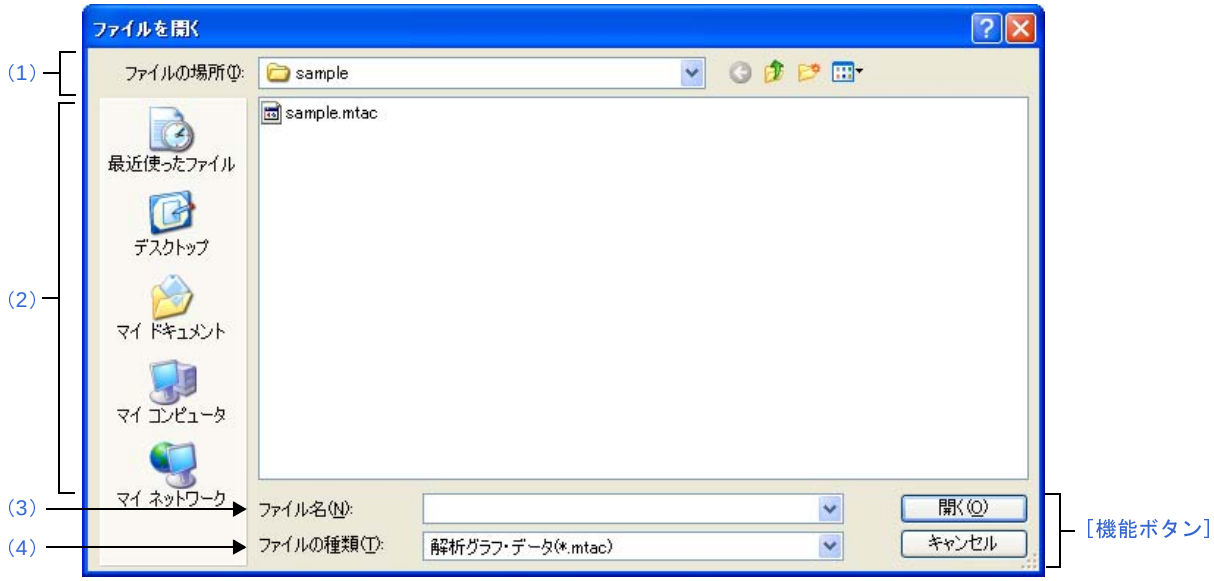
［機能ボタン］

ボタン	機能
OK	指定したファイルをインポート・ファイルに設定し，このダイアログをクローズします。
キャンセル	ファイルの指定を中止し，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。

ファイルを開く ダイアログ

グラフを復帰するための解析グラフ・データ・ファイル（*.mtac）の選択を行います。

図 A—44　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］

- プロパティ パネルの［値の推移］タブにおいて，［全般］カテゴリ内［解析グラフ・データ・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

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。

(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。

解析グラフ・データ (*.mtac)	解析グラフ・データ・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読み込み，グラフを復帰します。
キャンセル	このダイアログをクローズします。

列の選択 ダイアログ

関数一覧 パネル／変数一覧 パネルに表示する項目（列）の並び替え，または表示／非表示を設定します。

また，各パネルにおいて表示方法をカスタマイズする操作を行っている場合では，このダイアログより，それらのカスタマイズをすべてデフォルトの状態に戻すことができます。

図 A―45　列の選択 ダイアログ（関数一覧 パネル用）

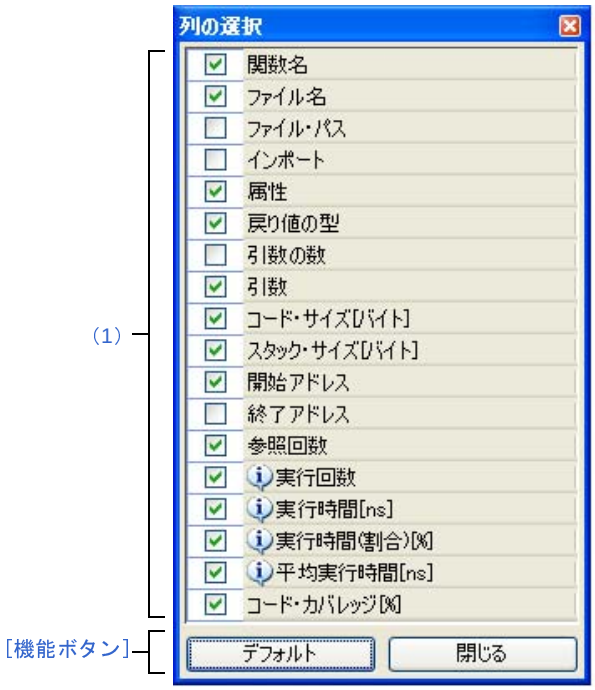
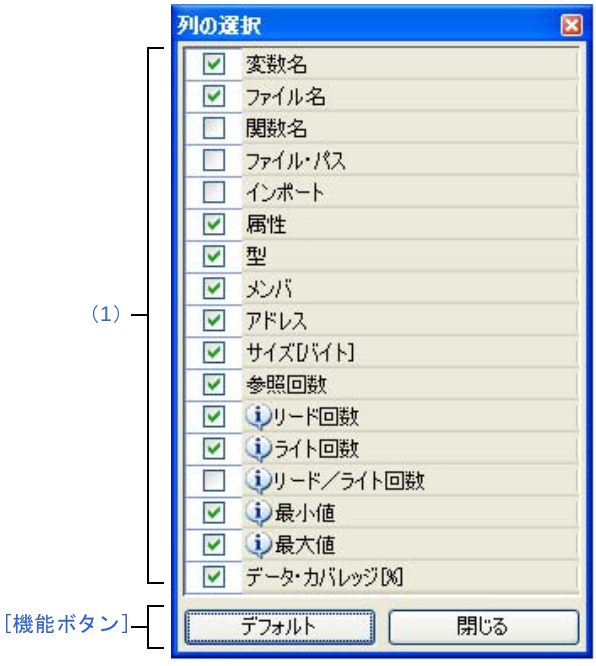


図 A―46　列の選択 ダイアログ（変数一覧 パネル用）



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］

- 関数一覧 パネルにおいて，ヘッダ・エリア上の  ボタンをクリック
- 変数一覧 パネルにおいて，ヘッダ・エリア上の  ボタンをクリック

［各エリアの説明］

（1）項目名リスト・エリア

関数一覧 パネル／変数一覧 パネルで表示可能な項目（列）のすべてをリスト表示します（表示可能な項目（列）は，使用するマイクロコントローラ／デバッグ・ツールにより異なります）。

なお，リスト内の各項目の表示順，およびチェック・ボックスの状態は，対応するパネルにおける現在の表示順，および表示／非表示の状態と同等です。

各項目のチェック・ボックスの指定により，パネル上での表示／非表示を設定することができます。

	この項目をパネル上に表示します。
	この項目をパネル上に表示しません。
	該当項目の情報に関するメッセージを出力 パネルに出力していることを示します。マウス・カーソルを重ねることにより，出力した最新のメッセージをポップアップ表示します。

［機能ボタン］

ボタン	機能
デフォルト	関数一覧 パネル／変数一覧 パネル上の各項目の表示順，および表示／非表示の設定をデフォルトの状態に戻します。
閉じる	このダイアログをクローズします。

備考　各項目のデフォルトの表示状態は次のとおりです。

表内［項目］の表記順は，各パネル上の項目（列）の並び順に相当します。

なお，固定表示／フィルタ表示などのカスタマイズ設定はすべて解除されます。

また，使用するマイクロコントローラ／デバッグ・ツールにより，表示項目が限定されます。

表 A—2　関数一覧 パネルのデフォルト状態

項目	表示状態	
	デバッグ・ツールと切断時	デバッグ・ツールと接続時
関数名	表示	表示

項目	表示状態	
	デバッグ・ツールと切断時	デバッグ・ツールと接続時
クラス名 【CC-RX】	表示	表示
名前空間 【CC-RX】	非表示	非表示
ファイル名	表示	表示
ファイル・パス	非表示	非表示
PM 情報【RH850】注 PE 情報【V850E2】注	表示	表示
インポート	非表示	非表示
アクセス指定子 【CC-RX】	表示	表示
属性	表示	表示
戻り値の型	表示	表示
引数の数	非表示	非表示
引数	表示	表示
コード・サイズ [バイト]	表示	表示
スタック・サイズ [バイト] 【V850】 【RL78】 【78K0R】 【78K0】	表示	表示
開始アドレス	表示	表示
終了アドレス	非表示	非表示
参照回数	表示	表示
実行回数 【Full-spec emulator】 【IECUBE】 【IECUBE2】 【シミュレータ】	非表示	表示
実行時間 [*単位*] 【Full-spec emulator】 【IECUBE【V850】】 【IECUBE【RL78】】 【IECUBE【78K0R】】 【IECUBE2】 【シミュレータ】	非表示	表示

項目	表示状態	
	デバッグ・ツールと切断時	デバッグ・ツールと接続時
実行時間 (割合)[%] 【Full-spec emulator】 【IECUBE【V850】】 【IECUBE【RL78】】 【IECUBE【78K0R】】 【IECUBE2】 【シミュレータ】	非表示	表示
平均実行時間 [*単位*] 【Full-spec emulator】 【IECUBE【V850】】 【IECUBE【RL78】】 【IECUBE【78K0R】】 【IECUBE2】 【シミュレータ】	非表示	表示
コード・カバレッジ [%] 【IECUBE】 【IECUBE2】 【シミュレータ】	非表示	表示

注 【RH850】【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のみ表示される項目です。

表 A—3 変数一覧 パネルのデフォルト状態

項目	表示状態	
	デバッグ・ツールと切断時	デバッグ・ツールと接続時
変数名	表示	表示
クラス名 【CC-RX】	表示	表示
名前空間 【CC-RX】	非表示	非表示
ファイル名	表示	表示
関数名	非表示	非表示
ファイル・パス	非表示	非表示
PM 情報【RH850】**注** PE 情報【V850E2】**注**	非表示	表示
インポート	非表示	非表示
アクセス指定子 【CC-RX】	表示	表示
属性	表示	表示

項目	表示状態	
	デバッグ・ツールと切断時	デバッグ・ツールと接続時
型	表示	表示
メンバ	表示	表示
アドレス	表示	表示
サイズ [バイト]	表示	表示
参照回数	表示	表示
リード回数 【Full-spec emulator】 【IECUBE】 【IECUBE2】 【シミュレータ】	非表示	表示
ライト回数 【Full-spec emulator】 【IECUBE】 【IECUBE2】 【シミュレータ】	非表示	表示
リード／ライト回数 【Full-spec emulator】 【IECUBE】 【IECUBE2】 【シミュレータ】	非表示	非表示
最小値 【Full-spec emulator】 【IECUBE】 【IECUBE2】 【シミュレータ】	非表示	表示
最大値 【Full-spec emulator】 【IECUBE】 【IECUBE2】 【シミュレータ】	非表示	表示
データ・カバレッジ [%] 【IECUBE【RL78】】 【IECUBE【78K0R】】 【IECUBE【78K0】】 【IECUBE2】 【シミュレータ】	非表示	表示

注 【RH850】【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のみ表示される項目です。

コール・グラフ検索 ダイアログ

コール・グラフ パネルで表示しているコール・グラフ内に存在する関数／変数を検索します。

注意　コール・グラフとして表示している関数／変数（関数ボックス／変数ボックス）のみが検索の対象となります。

図 A―47　コール・グラフ検索 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］

- コール・グラフ パネルのツールバーの ボタンのクリック
- コール・グラフ パネルにフォーカスがある状態で，［編集］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

検索条件を指定します。

なお，複数の検索条件を指定した場合，すべての条件を満たす関数／変数のみを検索します。

(a)［関数／変数名］

検索対象となる関数名／変数名を指定します。

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：2046 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）。

デフォルトでは，前回指定した関数名／変数名を表示します（存在しない場合は“空欄”）。

- ［条件］

［関数／変数名］で指定した文字列に対する条件を次のドロップダウン・リストより選択します。

と等しい	指定した文字列と完全に一致する関数名／変数名を検索します。
で始まる	指定した文字列で始まる関数名／変数名を検索します。
で終わる	指定した文字列で終わる関数名／変数名を検索します。
を含む	指定した文字列を含む関数名／変数名を検索します（デフォルト）。

備考 関数名／変数名は，次の操作によっても指定することができます。

- 関数一覧 パネル／変数一覧 パネルの任意の行をこのエリアにドラッグ・アンド・ドロップ
- 任意の文字列をこのエリアにドラッグ・アンド・ドロップ

(b)［クラス名］【CC-RX】

関数／変数の検索条件の 1 つとして必要な場合，検索対象のメンバ関数／メンバ変数が属しているクラス名を指定します。

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：2046 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）。

デフォルトでは，前回指定したクラス名を表示します（存在しない場合は“空欄”）。

- ［条件］

［クラス名］で指定した文字列に対する条件を次のドロップダウン・リストより選択します。

と等しい	指定した文字列と完全に一致するクラス名に属する関数名／変数名を検索します。
で始まる	指定した文字列で始まるクラス名に属する関数名／変数名を検索します。
で終わる	指定した文字列で終わるクラス名に属する関数名／変数名を検索します。
を含む	指定した文字列を含むクラス名に属する関数名／変数名を検索します（デフォルト）。

備考 クラス名は，次の操作によっても指定することができます。

- 任意の文字列をこのエリアにドラッグ・アンド・ドロップ

(c)［大文字と小文字を区別する］

［関数／変数名］／［クラス名］【CC-RX】で指定した文字列に対して，大文字と小文字を区別する否かを指定します。

☑	大文字と小文字を区別して検索します。
☐	大文字と小文字を区別せず検索します（デフォルト）。

(d)［親関数の個数］

関数の検索条件の 1 つとして必要な場合，検索対象の関数の親関数の個数を指定します。

キーボードより数値を直接入力するか（指定可能範囲：0 ～ 65535），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）。

デフォルトでは，前回指定した個数を表示します（存在しない場合は“空欄”）。

- ［条件］

［親関数の個数］で指定した数値に対する条件を次のドロップダウン・リストより選択します。

より大きい	指定した数値より大きい（指定値を含まない）個数の親関数を持つ関数名を検索します。
以上	指定した数値以上（指定値を含む）の個数の親関数を持つ関数名を検索します。
と等しい	指定した数値と等しい個数の親関数を持つ関数名を検索します（デフォルト）。
以下	指定した数値以下（指定値を含む）の個数の親関数を持つ関数名を検索します。
より小さい	指定した数値より小さい（指定値を含まない）個数の親関数を持つ関数名を検索します。

(e)［子関数の個数］

関数の検索条件の 1 つとして必要な場合，検索対象の関数の子関数の個数を指定します。

キーボードより数値を直接入力するか（指定可能範囲：0 ～ 65535），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）。

デフォルトでは，前回指定した個数を表示します（存在しない場合は“空欄”）。

- ［条件］

［子関数の個数］で指定した数値に対する条件を次のドロップダウン・リストより選択します。

より大きい	指定した数値より大きい（指定値を含まない）個数の子関数を持つ関数名を検索します。
以上	指定した数値以上（指定値を含む）の個数の子関数を持つ関数名を検索します。
と等しい	指定した数値と等しい個数の子関数を持つ関数名を検索します（デフォルト）。
以下	指定した数値以下（指定値を含む）の個数の子関数を持つ関数名を検索します。
より小さい	指定した数値より小さい（指定値を含まない）個数の子関数を持つ関数名を検索します。

(2)［プレビュー］エリア

コール・グラフ全体のプレビューを表示します。

現在コール・グラフ パネルで表示している領域が，コール・グラフ全体の一部である場合は，その領域を赤枠で示します。

[機能ボタン]

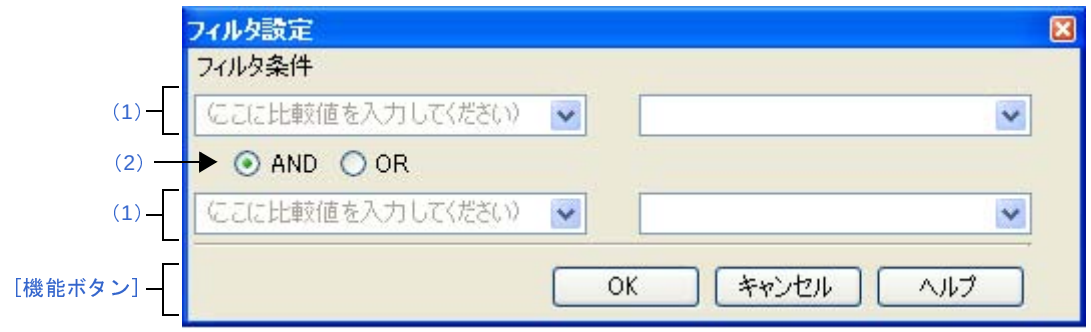
ボタン	機能
前を検索	指定した条件でコール・グラフ内の最下段から上段方向へ関数名／変数名の検索を行い，該当関数ボックス／変数ボックスを強調表示します。 なお，コール・グラフ パネル上で関数ボックス／変数ボックスを選択している場合は，対象関数／変数から上段方向へ検索を開始します。 ただし，不正な値を指定している場合，またはプログラム実行中は，メッセージを表示し，関数名／変数名の検索は行いません。 また，コール・グラフ パネルが非表示の場合，または他のパネルにフォーカスがある状態からこのダイアログへフォーカスを移動した場合，このボタンは無効となります。
次を検索	指定した条件でコール・グラフ内の最上段から下段方向へ関数名／変数名の検索を行い，該当関数ボックス／変数ボックスを強調表示します。 なお，コール・グラフ パネル上で関数ボックス／変数ボックスを選択している場合は，対象関数から下段方向へ検索を開始します。 ただし，不正な値を指定している場合，またはプログラム実行中は，メッセージを表示し，関数名／変数名の検索は行いません。 また，コール・グラフ 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	関数の検索を中止し，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。

フィルタ設定 ダイアログ

関数一覧 パネル／変数一覧 パネル上の解析情報を表示する際のフィルタ条件を設定します。

注意　このダイアログにより設定するフィルタ表示とパネルと連携したフィルタ表示は排他使用の機能です。
このため，これら 2 つのフィルタ表示機能を同時に有効化することはできません（どちらか一方のフィルタ表示を行っている際に，もう一方のフィルタ表示の設定を行った場合，それまで行っていたフィルタ表示はすべて解除されます）。
フィルタ表示についての詳細は，「2. 6. 5　解析情報をフィルタ表示する」を参照してください。

図 A―48　フィルタ設定 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］

- 関数一覧 パネル／変数一覧 パネルにおいて，ヘッダ・エリア上のフィルタ・アイコン（／）をクリックすることで表示されるメニュー項目より［(カスタム)］を選択

［各エリアの説明］

フィルタ条件を設定します。

第 1 条件設定エリア（上段）／第 2 条件設定エリア（下段）において，2 つまでの条件を指定することができ，論理条件指定ボタン（［AND］／［OR］）の選択により，両条件を 1 つのフィルタ条件として設定することができます。

(1) 第 1 条件設定エリア（上段）／第 2 条件設定エリア（下段）

(a) 比較値（コンボ・ボックス左）

フィルタの対象となる比較値（数値／文字列）を指定します。

キーボードより直接入力するか（最大指定文字数：2048 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）。

デフォルトでは，現在対象の項目に設定されている比較値を表示します。

（b）条件（コンボ・ボックス右）

上記（a）で指定した比較値に対する条件を次のドロップダウン・リストより選択します。

デフォルトでは，現在対象の項目に設定されている条件が選択状態となります。

項目	比較値が数値の場合	比較値が文字列の場合
条件なし注	数値として比較	文字列として比較
と等しい		
と等しくない		
より大きい		
以上		
より小さい		
以下		
で始まる	文字列として比較	
で始まらない		
で終わる		
で終わらない		
を含む		
を含まない		

注 ［条件なし］を選択した場合，比較値は無視されます（条件として設定されません）。

注意 1つの条件のみでフィルタ条件を設定する場合は，第1条件指定エリア（上段）において条件の指定を行ってください。

（2）論理条件指定ボタン

第1条件設定エリア（上段）／第2条件設定エリア（下段）で指定した条件に適用する論理条件を次のオプション・ボタンにより選択します。

AND	第1条件と第2条件の両方を満たす情報値のみ表示します。 ［OR］ボタンとは排他使用となります。
OR	第1条件と第2条件のどちらかを満たす情報値のみ表示します。 ［AND］ボタンとは排他使用となります。

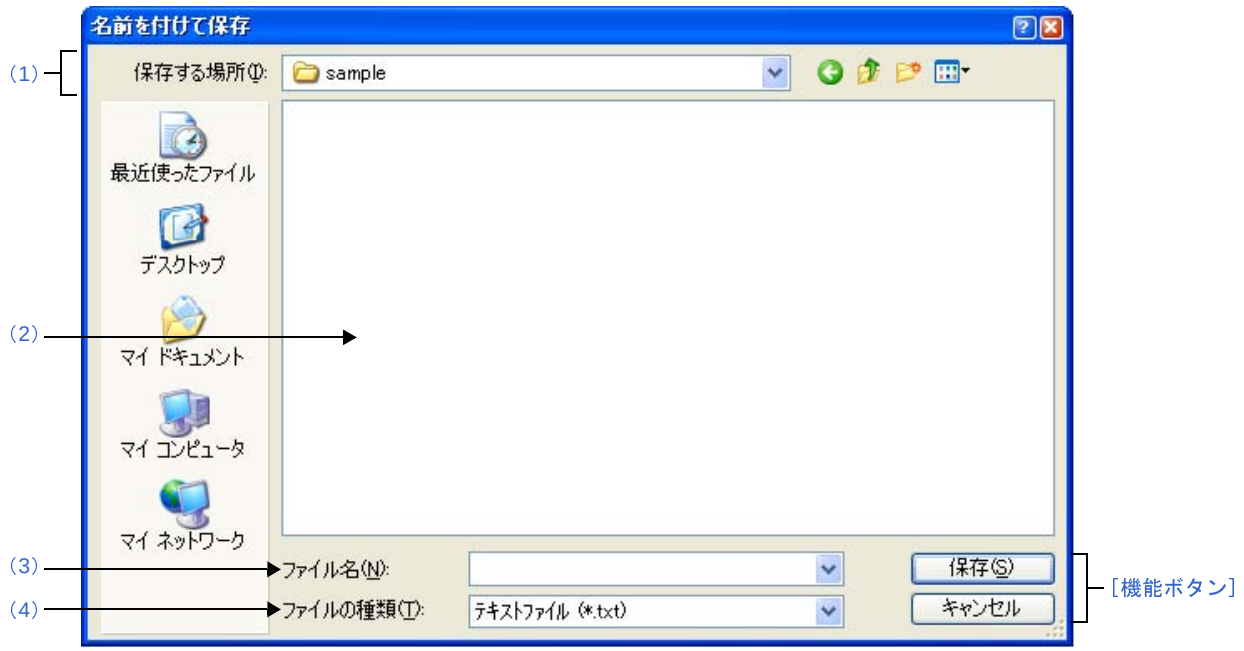
[機能ボタン]

ボタン	機能
OK	指定したフィルタ条件で，関数一覧 パネル／変数一覧 パネル上の解析情報を表示します。 なお，パネルと連携したフィルタ表示を行っている場合では，それまで行っていたパネルと連携したフィルタ表示をすべて解除します。
キャンセル	フィルタ条件の設定を無効とし，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。

名前を付けて保存 ダイアログ

関数一覧 パネル／変数一覧 パネル／解析グラフ パネル／コール・グラフ パネル／値の推移（ズーム）パネル／出力 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―49　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］

- 関数一覧 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 関数一覧データ を保存 ...］を選択
- 変数一覧 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 変数一覧データ を保存 ...］を選択
- 解析グラフ 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 解析グラフ・データ を保存 ...］を選択
- コール・グラフ 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・グラフ・データ を保存 ...］を選択
- 値の推移（ズーム）パネルにフォーカスがある状態で，［保存］ボタンをクリック
- 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 出力 - *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

ドロップダウン・リストより選択した次のファイル形式で，パネルの内容をファイルに保存します。

なお，CSV 形式で保存する場合，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。

（a）関数一覧 パネルの場合

テキスト・ファイル (*.txt)	テキスト形式
CSV(カンマ区切り) (*.csv)	CSV 形式
Microsoft Office Excel ブック (*.xls)	Microsoft Office Excel ブック形式
関数一覧ファイル (*.mtfl)	関数情報をインポートするためのファイル形式（「2. 12　情報ファイルをインポート／エクスポートする」参照）

注意　保存の対象は，現在表示している項目（列）／解析情報（行）の内容のみとなります。

（b）変数一覧 パネルの場合

テキスト・ファイル (*.txt)	テキスト形式
CSV(カンマ区切り) (*.csv)	CSV 形式
Microsoft Office Excel ブック (*.xls)	Microsoft Office Excel ブック形式
変数一覧ファイル (*.mtvl)	変数情報をインポートするためのファイル形式（「2. 12　情報ファイルをインポート／エクスポートする」参照）

注意　保存の対象は，現在表示している項目（列）／解析情報（行）の内容のみとなります。

（c）解析グラフ パネル／値の推移（ズーム）パネルの場合

テキスト・ファイル (*.txt)	テキスト形式
CSV(カンマ区切り) (*.csv)	CSV 形式

Microsoft Office Excel ブック (*.xls)	Microsoft Office Excel ブック形式
解析グラフ・データ (*.mtac) 注 1	解析グラフ・データ・ファイル
ビットマップ (*.bmp)	ビットマップ形式（32 ビット）（画像形式）
JPEG ファイル (*.jpg)	JPEG 形式（画像形式）
PNG ファイル (*.png)	PNG 形式（画像形式）
EMF ファイル (*.emf) 注 2	EMF 形式（画像形式）

注 1. 解析グラフ パネルの［値の推移］タブのみサポートします。

2. グラフ描画領域のみを保存します（トリガ・マーク／チャネル情報などは保存されません）。

注意 1. 解析グラフ パネルの場合，保存の対象は，現在表示しているタブの内容のみとなります。

2. 画像形式を選択した場合，保存の対象は，現在表示している部分のみとなります。

なお，画像形式以外のファイル形式で保存する場合の保存項目は次のとおりです。

ただし，解析グラフ・データ・ファイル（*.mtac）を除きます（「(6) グラフを復帰するためのグラフ・データを保存する」参照）。

- ［値の推移］タブ

【リアルタイム・サンプリング方式の場合】

時間　値 (変数名)　値 (変数名) …

値 (変数名) ：各変数の値を常に出力（ただし，値が不明の場合は空欄）

【トレース・データ解析方式の場合】

時間　値 (変数名)　値 (変数名) …場所

値 (変数名) ：値が変化した変数のみ出力（ただし，値が不明の場合は空欄）

場所 ：情報が存在しない場合は空欄

- ［実行時間の割合］タブ

関数名　割合 [%]　時間

（d）コール・グラフ パネルの場合

ビットマップ（可視部のみ）(*.bmp)	ビットマップ形式（32 ビット）（画像形式）
JPEG ファイル（可視部のみ）(*.jpg)	JPEG 形式（画像形式）
PNG ファイル（可視部のみ）(*.png)	PNG 形式（画像形式）
ビットマップ (*.bmp)	ビットマップ形式（32 ビット）（画像形式）
JPEG ファイル (*.jpg)	JPEG 形式（画像形式）
PNG ファイル (*.png)	PNG 形式（画像形式）
EMF ファイル (*.emf)	EMF 形式（画像形式）

注意 プロジェクトが巨大な場合，コール・グラフ全域の画像ファイルを保存できない場合があります。

備考 1. “（可視部のみ）”を選択した場合，保存の対象は，現在表示している部分のみとなります。

2. ズーム機能を適用している場合，現在のズーム率で画像を保存します（EMF 形式を除く）。

（e）出力パネルの場合

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
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。

付録B 索 引

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2013.09.01	―	初版発行

CubeSuite+ V2.01.00　ユーザーズマニュアル
解析編

発行年月日　2013年 9月 1日　　Rev.1.00

発行　　ルネサス エレクトロニクス株式会社
〒211-8668 神奈川県川崎市中原区下沼部 1753

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+ V2.01.00

RENESAS

ルネサス エレクトロニクス株式会社

R20UT2686JJ0100