

CubeSuite Ver.1.40

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

78K0R マイクロコントローラ

V850 マイクロコントローラ

本資料に記載の全ての情報は本資料発行時点のものであり、ルネサス エレクトロニクスは、予告なしに、本資料に記載した製品または仕様を変更することがあります。
ルネサス エレクトロニクスのホームページなどにより公開される最新情報をご確認ください。

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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

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

このマニュアルの使い方

このマニュアルは、78K0 マイクロコントローラ、78K0R マイクロコントローラ、および V850 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、78K0 マイクロコントローラ、78K0R マイクロコントローラ、および V850 マイクロコントローラの統合開発環境（ソフトウェア開発における、設計、実装、デバッグなどの各開発フェーズに必要なツールをプラットフォームである IDE に統合）です。統合することで、さまざまなツールを使い分ける必要がなく、本製品のみを使用して開発のすべてを行うことができます。

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

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

構成 このマニュアルは、大きく分けて次の内容で構成しています。

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

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

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

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 ユーザズ・マニュアル	起動編	R20UT0256J	R20UT0256E
	解析編	このマニュアル	R20UT0265E
	プログラミング編	R20UT0266J	R20UT0266E
	メッセージ編	R20UT0267J	R20UT0267E
	コーディング編 (CX コンパイラ)	R20UT0259J	R20UT0259E
	ビルド編 (CX コンパイラ)	R20UT0261J	R20UT0261E
	78K0 コーディング編	R20UT0004J	R20UT0004E
	78K0 ビルド編	R20UT0005J	R20UT0005E
	78K0 デバッグ編	R20UT0262J	R20UT0262E
	78K0 設計編	R20UT0006J	R20UT0006E
	78K0R コーディング編	U19382J	U19382E
	78K0R ビルド編	U19385J	U19385E
	78K0R デバッグ編	R20UT0263J	R20UT0263E
	78K0R 設計編	R20UT0007J	R20UT0007E
	V850 コーディング編	U19383J	U19383E
	V850 ビルド編	U19386J	U19386E
	V850 デバッグ編	R20UT0264J	R20UT0264E
V850 設計編	R20UT0257J	R20UT0257E	

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。

〔メ モ〕

〔メ モ〕

〔メ モ〕

目 次

第1章 概 説 … 9

- 1.1 概 要 … 9
 - 1.1.1 解析対象 … 9
 - 1.1.2 解析情報の種別 … 9
 - 1.1.3 デバッグ・ツールの設定 … 10
- 1.2 特 長 … 14

第2章 機 能 … 17

- 2.1 概 要 … 17
- 2.2 関数情報を表示する … 21
- 2.3 変数情報を表示する … 22
- 2.4 関数間の呼び出し関係（コール・グラフ）を表示する … 23
- 2.5 表示方法をカスタマイズする … 27
 - 2.5.1 表示項目を設定する … 27
 - 2.5.2 表示項目を並び替える … 29
 - 2.5.3 特定項目を固定表示に設定する … 30
 - 2.5.4 解析情報をソート表示する … 31
 - 2.5.5 解析情報をフィルタ表示する … 31
- 2.6 定義箇所へジャンプする … 35
- 2.7 ブレーク・イベントを設定する … 36
 - 2.7.1 関数にブレークポイントを設定する … 36
 - 2.7.2 変数にブレーク・イベントを設定する … 36
- 2.8 ウォッチ式に登録する … 37
- 2.9 参照箇所を一覧表示する … 38
- 2.10 情報ファイルをインポート／エクスポートする … 39
- 2.11 解析情報をグラフ化して表示する … 43
 - 2.11.1 変数値の推移をグラフ化する … 43
 - 2.11.2 関数の実行時間率をグラフ化する … 50
- 2.12 解析情報をファイルに保存する … 52

付録 A ウィンドウ・リファレンス … 54

- A.1 説 明 … 54

付録 B 索 引 … 135

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、78K0 マイクロコントローラ、78K0R マイクロコントローラ、V850 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 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 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. デバッグ専用プロジェクト（「[CubeSuite 起動編](#)」参照）を対象に、解析ツールを使用することはできません。

2. **[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（*.csfl）／変数一覧ファイル（*.csvl））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 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（[2.10 情報ファイルをインポート／エクスポートする](#)参照）。

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・リファレンス情報から取得／解析した関数情報／変数情報です。

注意 [CX]

-Xno_cube_suite_inf オプションを指定している場合、クロス・リファレンス情報が生成されないため、本ツールが提供する各パネル（[解析グラフパネル](#)の [\[値の推移\]](#) タブを除く）において解析情報は何も表示されません。

備考 【CA78K0】【CA78K0R】【CA850】

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プロパティ（**プロパティパネル**の**〔設定〕**タブ上の**〔全般〕**カテゴリ内**〔強制的にクロス・リファレンス・ファイルを出力する〕**プロパティ）を用意しています。

このプロパティ設定はデフォルトで有効化（**〔はい〕**）されており、この設定に変更を加えない限り、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（この設定を**〔いいえ〕**にした場合、ビルド・ツールでの設定が優先されます）。

(2) 動的解析情報

プログラムの実行が停止した時点で表示可能となり、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結果、カバレッジ結果、またはリアルタイム RAM モニタ結果から取得／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、カバレッジ機能、またはRRM機能が有効化（**「1.1.3 デバッグ・ツールの設定」**参照）されている必要があります。

なお、取得する動的解析情報の項目により、必要となるデバッグ・ツールの機能は異なります。また、デバッグ・ツールの種類によりサポートしている機能が異なるため、使用するデバッグ・ツールにより、取得可能な動的解析情報の項目は異なります。

動的解析情報の各項目とそれを取得するために必要なデバッグ・ツールの機能についての関係は、各パネルの項を参照してください。

注意 リアルタイム OS「RX シリーズ」が提供している解析ツール（「AZ シリーズ（AZ78K0R/AZ850/AZ850V4）」など）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。

1.1.3 デバッグ・ツールの設定

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・ツールの設定は次のとおりです。

なお、デバッグ・ツールの各設定は、使用するデバッグ・ツールの**プロパティパネル**の**〔デバッグ・ツール設定〕**タブで行ってください。

- (1) **トレース機能**
- (2) **カバレッジ機能**
- (3) **RRM 機能**

(1) トレース機能

デバッグ・ツール		デバッグ・ツールの設定
IECUBE	78K0 78K0R	なし（常に有効） ^{注1}
	V850E1 V850ES	<p>【関数情報】（関数一覧パネル／コール・グラフパネル）</p> <p>[トレース] カテゴリ内 [トレース・データの用途] プロパティにおいて [トレース] を指定している場合に限り、次の設定が必要^{注2}</p> <p>[トレース] カテゴリ</p> <p>[トレース・データの選択] プロパティ→次のいずれかの指定</p> <ul style="list-style-type: none"> - [分岐命令] - [全命令] - [分岐命令とアクセス・アドレス／データ値] - [全命令とアクセス・アドレス／データ値] - [分岐／アクセス命令とアクセス・アドレス／データ値] <p>【変数情報】（変数一覧パネル／解析グラフパネル（トレース・データ解析方式））</p> <p>[トレース] カテゴリ</p> <p>[トレース・データの用途] プロパティ→ [トレース]</p> <p>[トレース・データの選択] プロパティ→次のいずれかの指定</p> <ul style="list-style-type: none"> - [アクセス・アドレス／データ値] - [分岐命令とアクセス・アドレス／データ値] - [全命令とアクセス・アドレス／データ値] - [アクセス命令とアクセス・アドレス／データ値] - [分岐／アクセス命令とアクセス・アドレス／データ値]
IECUBE2	V850E2M シングルコア	<p>【関数情報】（関数一覧パネル／コール・グラフパネル）</p> <p>[トレース] カテゴリ内 [トレース・データの用途] プロパティにおいて [トレース] を指定している場合に限り、次の設定が必要^{注3}</p> <p>[トレース] カテゴリ</p> <p>[トレース・データの選択] プロパティ→次のいずれかの指定</p> <ul style="list-style-type: none"> - [分岐命令] - [分岐命令とアクセス・アドレス／データ値] - [分岐／アクセス命令とアクセス・アドレス／データ値] <p>【変数情報】（変数一覧パネル／解析グラフパネル（トレース・データ解析方式））</p> <p>[トレース] カテゴリ</p> <p>[トレース・データの用途] プロパティ→ [トレース]</p> <p>[トレース・データの選択] プロパティ→次のいずれかの指定</p> <ul style="list-style-type: none"> - [分岐命令とアクセス・アドレス／データ値] - [分岐／アクセス命令とアクセス・アドレス／データ値]
	V850E2M マルチコア	サポートしていません。
MINICUBE		サポートしていません。
MINICUBE2		サポートしていません。
E1		サポートしていません。

デバッグ・ツール		デバッグ・ツールの設定
シミュレータ	78K0	[トレース] カテゴリ
	78K0R	[トレース機能を使用する] プロパティ → [はい]
	V850	

注1. 【IEUCBE 【78K0】】

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。

2. 【IECUBE 【V850】】

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。

3. 【IECUBE2】

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。

(2) カバレッジ機能

デバッグ・ツール		デバッグ・ツールの設定
IECUBE	78K0	なし（常に有効）
	78K0R	
	V850E1	[トレース] カテゴリ
	V850ES	[トレース・データの用途] プロパティ → [カバレッジ] ただし、コード・カバレッジ（ 関数一覧 パネル ）のみのサポートとなります。 また、使用する IECUBE にカバレッジ・ボードが搭載されていない場合、カバレッジ機能を有効化することはできません。
IECUBE2	V850E2M	[トレース] カテゴリ
	シングルコア	[トレース・データの用途] プロパティ → [カバレッジ] ただし、コード・カバレッジ（ 関数一覧 パネル ）のみのサポートとなります。
	V850E2M	サポートしていません。
	マルチコア	
MINICUBE		サポートしていません。
MINICUBE2		サポートしていません。
E1		サポートしていません。
シミュレータ	78K0	[カバレッジ] カテゴリ
	78K0R	[カバレッジ機能を使用する] プロパティ → [はい]
	V850	

(3) RRM 機能

デバッグ・ツール		デバッグ・ツールの設定
IECUBE	78K0 78K0R	[実行中のメモリ・アクセス] カテゴリ [実行を一瞬停止してアクセスする] プロパティ→ [はい] [実行中に表示更新を行う] プロパティ→ [はい] [表示更新間隔 [ms]] プロパティ→ 有効な数値
	V850E1 V850ES	[トレース] カテゴリ [トレース・データの用途] プロパティ→ [RRM] [実行中のメモリ・アクセス] カテゴリ [実行を一瞬停止してアクセスする] プロパティ→ [はい] [実行中に表示更新を行う] プロパティ→ [はい] [表示更新間隔 [ms]] プロパティ→ 有効な数値 [リアルタイム表示更新を自動設定する] プロパティ→ [はい]
IECUBE2	V850E2M シングルコア	[実行中のメモリ・アクセス] カテゴリ [実行を一瞬停止してアクセスする] プロパティ→ [はい] [実行中に表示更新を行う] プロパティ→ [はい] [表示更新間隔 [ms]] プロパティ→ 有効な数値
	V850E2M マルチコア	サポートしていません。
MINICUBE	V850	[実行中のメモリ・アクセス] カテゴリ [実行を一瞬停止してアクセスする] プロパティ→ [はい] [実行中に表示更新を行う] プロパティ→ [はい] [表示更新間隔 [ms]] プロパティ→ 有効な数値
MINICUBE2	78K0	[実行中のメモリ・アクセス] カテゴリ [実行を一瞬停止してアクセスする] プロパティ→ [はい] [実行中に表示更新を行う] プロパティ→ [はい] [表示更新間隔 [ms]] プロパティ→ 有効な数値 [リアルタイム表示更新を自動設定する] プロパティ→ [はい]
	78K0R V850	[実行中のメモリ・アクセス] カテゴリ [実行を一瞬停止してアクセスする] プロパティ→ [はい] [実行中に表示更新を行う] プロパティ→ [はい] [表示更新間隔 [ms]] プロパティ→ 有効な数値
E1		[実行中のメモリ・アクセス] カテゴリ [実行を一瞬停止してアクセスする] プロパティ→ [はい] [実行中に表示更新を行う] プロパティ→ [はい] [表示更新間隔 [ms]] プロパティ→ 有効な数値
シミュレータ	78K0 78K0R V850	[実行中のメモリ・アクセス] カテゴリ [実行中に表示更新を行う] プロパティ→ [はい] [表示更新間隔 [ms]] プロパティ→ 有効な数値

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／戻り値の型など）、および動的解析情報（参照回数／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。

関数名	ファイル名	属性	戻り値の型	引数
AD_Init	CG_adc	-	void	void
AD_Read	CG_adc	-	unsigned short	unsigned...
AD_Start	CG_adc	-	void	void
AD_Stop	CG_adc	-	void	void
MD_INTAD	CG_ad_user.c	-	void	void
TMP0_Start	CG_timer.c	-	void	void
TMP1_Start	CG_timer.c	-	void	void
ad_receive	(定義箇所なし)	-	-	-
func1	CG_main.c	-	void	void
func1a	CG_main.c	-	void	void
func2	CG_main.c	-	void	void

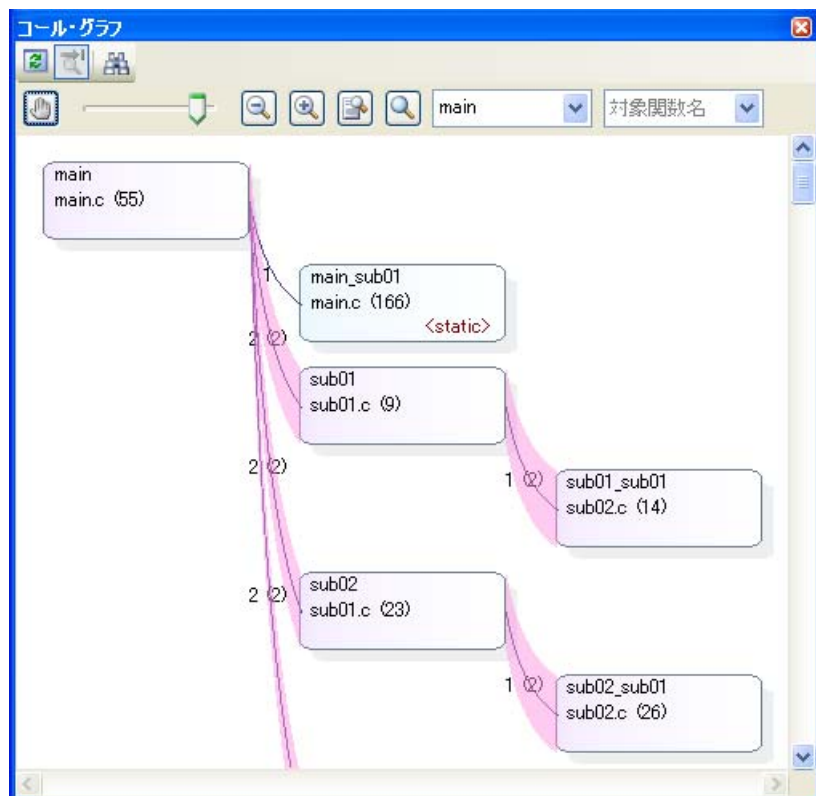
- 変数情報の表示

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。

変数名	ファイル名	属性	型	メンバ	アドレス
g_ad_data	CG_main.c	-	unsigned short	-	0x03ffb116
g_ad_finish	CG_main.c	-	unsigned char	-	0x03ffb114
g_count_10ms	CG_main.c	-	unsigned int	-	0x03ffb110
g_count_1ms	CG_main.c	-	unsigned int	-	0x03ffb10c
g_flag_detect	CG_main.c	-	unsigned short	-	0x03ffb118
_S_rom	CG_systeminit.c	-	-	-	-
合計値	CG_main.c	-	-	-	-
合計値	CG_systeminit.c	-	-	-	-

- 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つリー構造の図（コール・グラフ）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（関数一覧パネル／変数一覧パネル）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ソース・テキスト箇所へのジャンプ
- 指定した関数／変数に対するブレーク・イベントの設定
- 指定した変数をウォッチ式に登録

- 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。

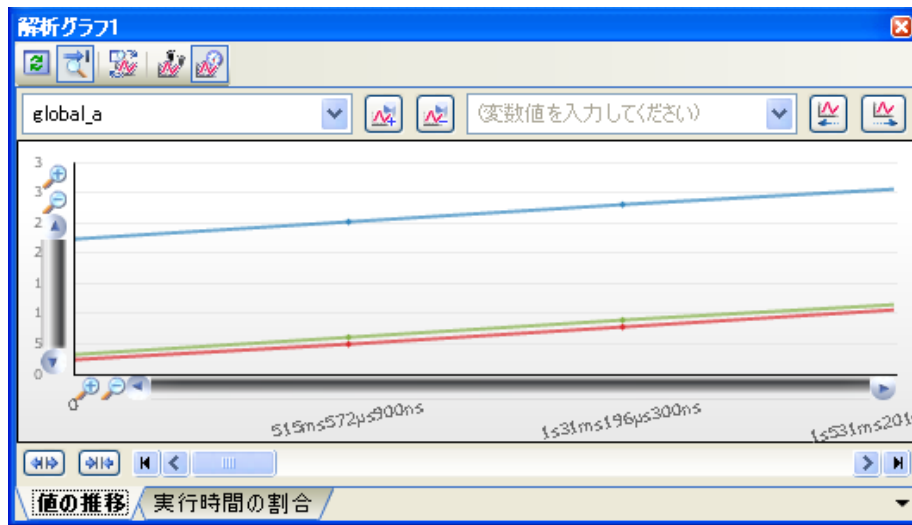
```

出力
参照検索開始
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 0): 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
参照検索終了
[EOF]

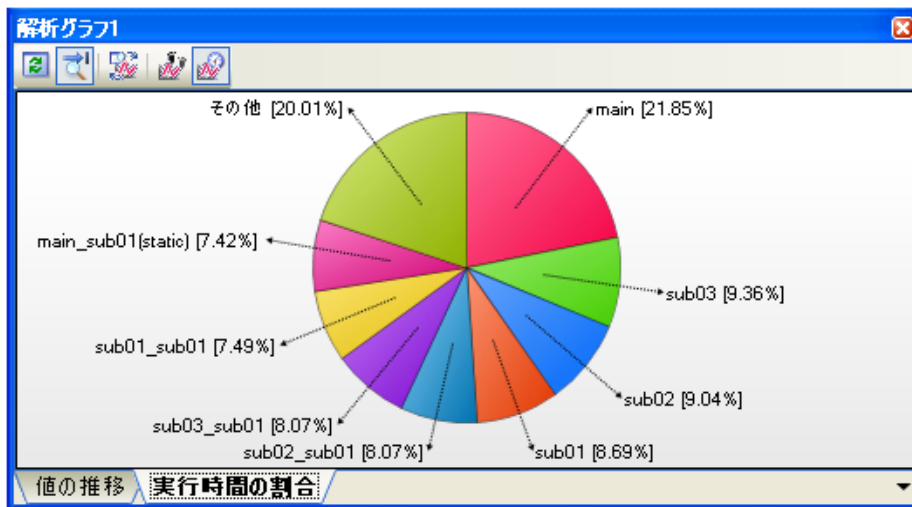
```

- 解析情報のグラフ化

- 変数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実践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、次のパネル上に表示されます。

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	関数情報を表示	
変数一覧パネル	変数情報を表示	
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）を表示	
解析グラフパネル	関数情報／変数情報のグラフを表示	
	[値の推移] タブ	変数値の推移を表示（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を表示（円グラフ）

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 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 起動編」を参照してください。

(3) ロード・モジュールを作成する

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

備考 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 ビルド編」を参照してください。

- 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](#)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。

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 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 1. “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 デバッグ編」を参照してください。

- プログラムの実行が停止した時点で、各パネルにおいて、[動的解析情報](#)が表示可能となります。なお、[動的解析情報](#)を取得するためには、デバッグ・ツールの各機能が有効化（「[1.1.3 デバッグ・ツールの設定](#)」参照）されている必要があります。

(6) 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ソース・テキスト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、「[2.10 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。

(7) 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ソース・テキスト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、「[2.10 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) 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。

(b) 関数を検索する

コール・グラフ内に存在する関数を検索します。

(9) 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) 変数値の推移をグラフ化する

変数／レジスタ等の値と時間の関係を折れ線グラフで表示します。

(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。

(10) 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。

(11) 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 起動編」を参照してください。

2.2 関数情報を表示する

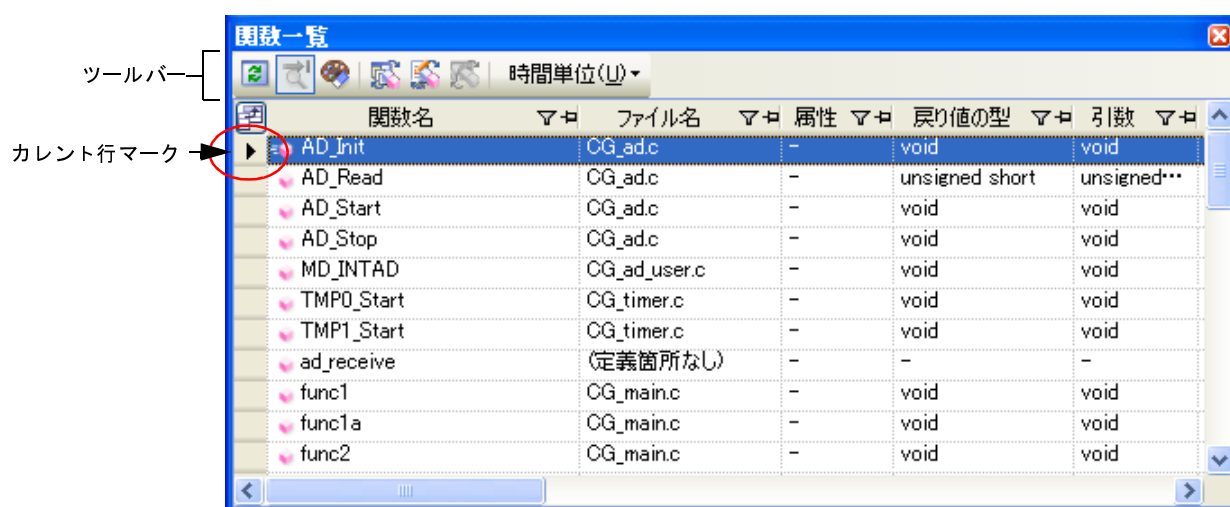
プログラム中の関数情報（関数名／ファイル名／属性／参照回数／実行回数／コード・カバレッジ率など）を表示します。

関数情報は、**メイン・ウィンドウ**のツールバー上の  ボタンをクリックすることでオープンする **関数一覧** パネルに表示されます。

このパネルのツールバー上の  ボタンをクリックすることにより、現在表示可能（「1.1.2 解析情報の種別」参照）な最新情報が表示されます。


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。

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、ツールバー上の  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、「2.10 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

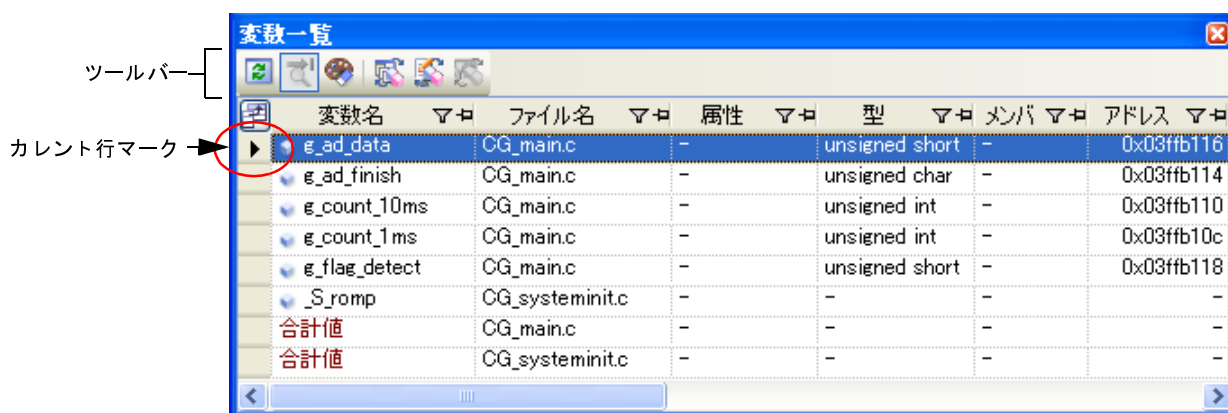
プログラム中の変数情報（変数名／ファイル名／属性／リード回数／ライト回数／データ・カバレッジ率など）を表示します。

変数情報は、**メイン・ウインドウ**のツールバー上の  ボタンをクリックすることでオープンする **変数一覧 パネル** に表示されます。

このパネルのツールバー上の  ボタンをクリックすることにより、現在表示可能（「1.1.2 解析情報の種別」参照）な最新情報が表示されます。


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。

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、ツールバー上の  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

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、「2.10 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。

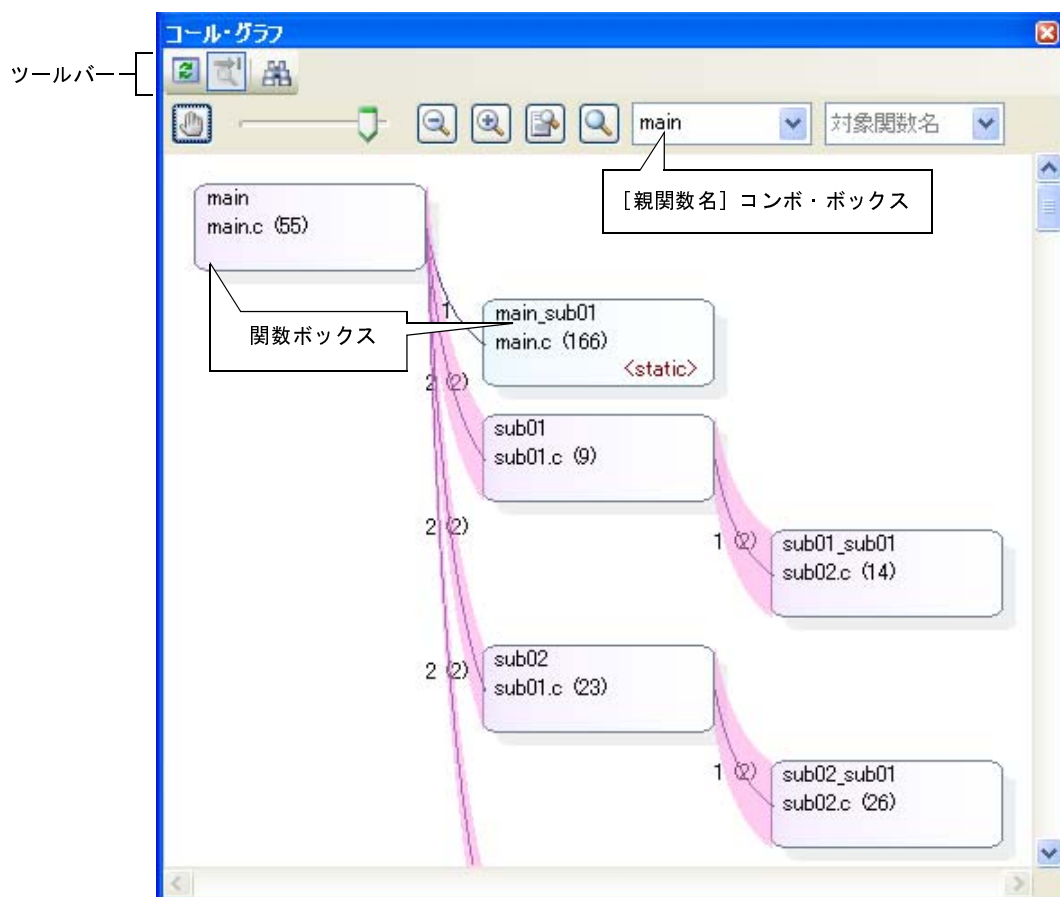
コール・グラフは、**メイン・ウィンドウ**のツールバー上の  ボタンをクリックすることでオープンする**コール・グラフパネル**に表示されます。

このパネルのツールバー上の  ボタンをクリックすることにより、現在表示可能（「1.1.2 解析情報の種別」参照）な最新情報が表示されます。

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）を表示することはできません。
2. 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）



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” 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。

備考 【V850E2M】

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“PE n ”を指定することができます。この場合は、該当 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) 定義箇所へジャンプする

任意の関数ボックスをダブルクリックすることにより、その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（「[2.6 定義箇所へジャンプする](#)」参照）。

(3) 関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にマウス・カーソルを重ねることにより、該当関数の情報がポップアップ表示されます（表示形式についての詳細は、「[\(c\) ポップアップ表示](#)」を参照してください）。


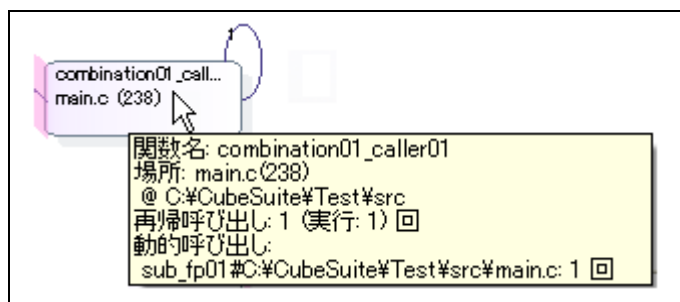
注意 コール・グラフパネルの表示設定エリア上の  ボタンの設定により、コール・グラフのスクロールを許可している場合、この機能を使用することはできません。

図 2—4 関数情報のポップアップ表示例

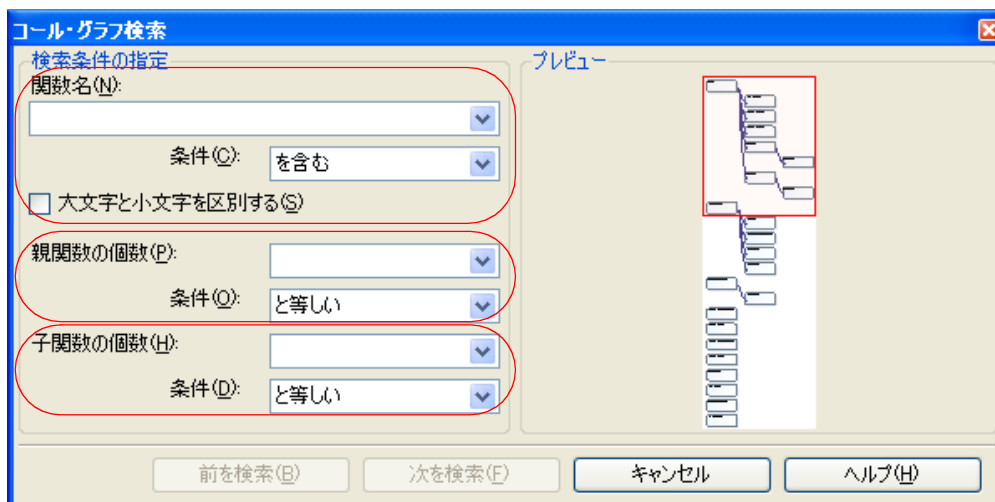
**(4) 関数を検索する**

コール・グラフ内に存在する関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**のツールバー上の  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 2-5 関数の検索 (コール・グラフ検索 ダイアログ)

**(a) [関数名] の指定**

検索対象となる関数名を指定します。

キーボードより文字列を直接入力するか、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。

- 関数一覧パネルの任意の行をこのダイアログ上にドラッグ・アンド・ドロップ
- 任意の文字列をこのコンボ・ボックスに直接ドラッグ・アンド・ドロップ

(b) [親関数の個数] の指定

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）。

(c) [子関数の個数] の指定

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）。

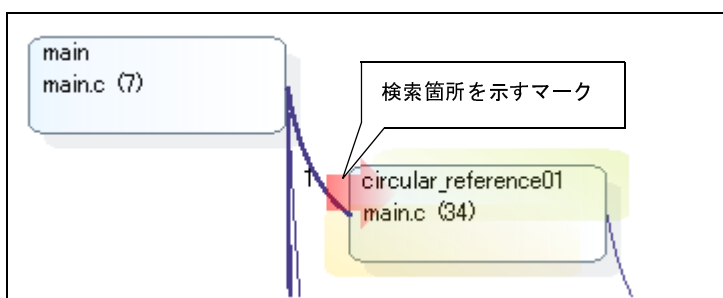
(d) [前を検索] / [次を検索] ボタンのクリック

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

なお、**コール・グラフパネル**上の [対象関数名] コンボ・ボックスにおいて関数を指定している場合は、対象関数から上段方向 / 下段方向へ検索を開始します。

図 2—6 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をコール・グラフに含めることができます。

2.5 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。

2.5.1 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

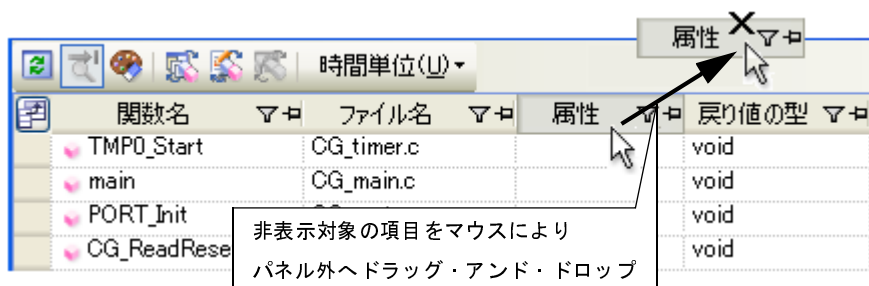
(1) 表示項目を限定する場合

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—7 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


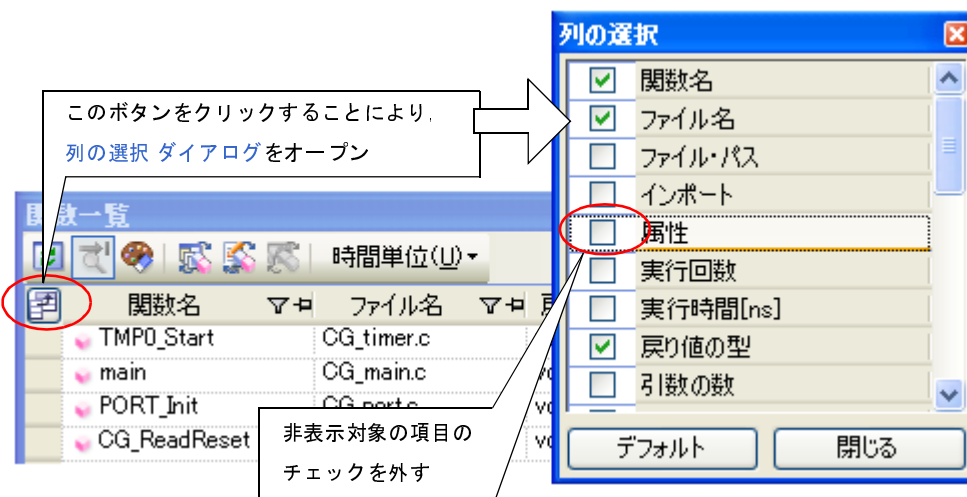
パネルのヘッダ・エリア左端の  ボタンをクリックすることによりオープンする列の選択ダイアログにおいて、非表示とする対象項目名のチェック・ボックスのチェックを外します。

図 2—8 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—9 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）

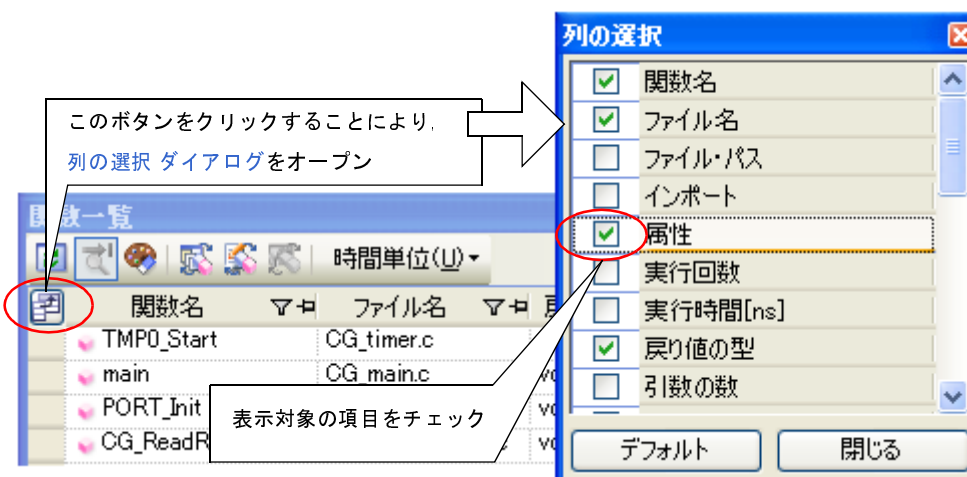
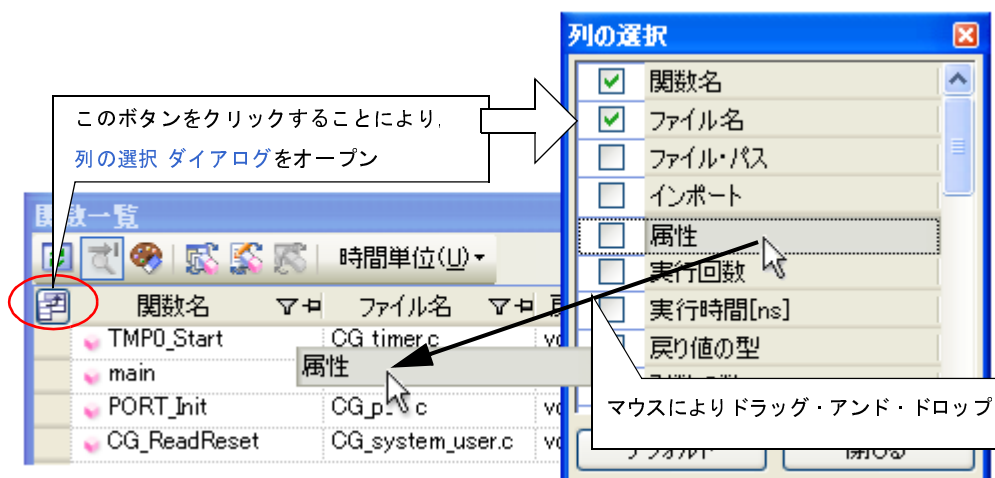


図 2—10 表示項目の追加 (列の選択ダイアログからのドラッグ・アンド・ドロップ操作)



2.5.2 表示項目を並び替える

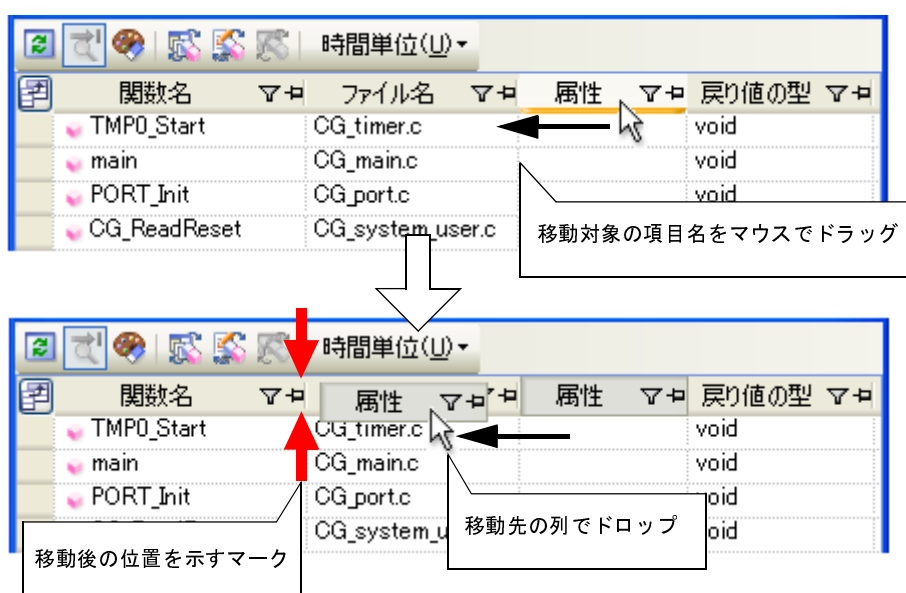
関数一覧 パネル／変数一覧 パネル上の表示項目 (列) の順序を並び替えることができます。操作方法は、次の2通りです。

- (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—11 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


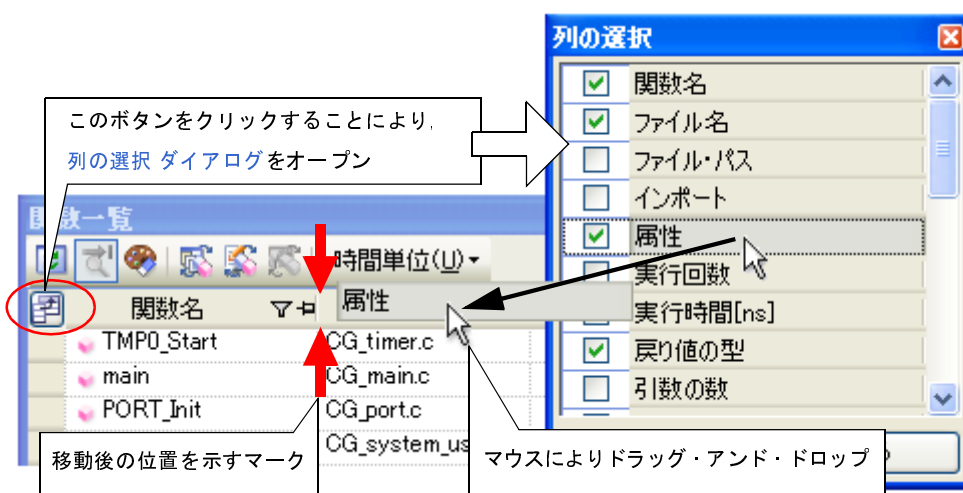


パネルのヘッダ・エリア左端の  ボタンをクリックすることによりオープンする列の選択ダイアログにおいて、移動対象の項目名を、マウスによりパネル上の任意の列（ヘッダ・エリア）へ直接ドラッグ・アンド・ドロップします。

図 2—12 表示項目の並び替え（列の選択ダイアログからの操作）



2.5.3 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—2 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上にドラッグ・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。

2.5.4 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—3 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。

2.5.5 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。

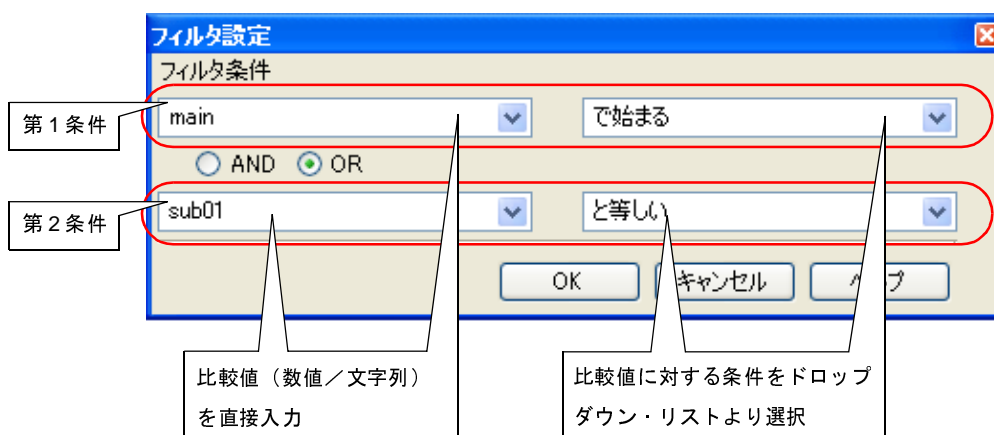
表 2—4 フィルタの設定項目

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—13 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—5 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。


備考 この操作により設定したフィルタ表示を行っている際に、「(2) パネルと連携したフィルタ表示」の設定を行うと、それまで行っていた項目ごとのカスタム設定によるフィルタ表示はすべて解除されます。

(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【V850E2M】

(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上の  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—6 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 1.** 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。
- 2.** この操作により設定した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。
- 3.** このフィルタ機能は、「(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【V850E2M】」と併用することができます。

(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](#)のツールバー上の  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。

- 備考 1.** この操作により設定した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。
- 2.** このフィルタ機能は、「[\(a\) プロジェクト・ツリー パネルとの連携](#)」／「[\(c\) デバッグ・マネージャ パネルとの連携【V850E2M】](#)」と併用することができます。

(c) デバッグ・マネージャ パネルとの連携【V850E2M】

デバッグ・マネージャ パネルで指定している PE 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](#)のツールバー上の  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。


- 備考 1.** この操作により設定した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。
- 2.** このフィルタ機能は、「[\(a\) 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](#)」と併用することができます。

2.6 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・グラフパネル上の関数／変数が定義されているソース・テキスト箇所へジャンプすることができます。

操作は、関数一覧パネル／変数一覧パネルでは対象の関数／変数の行を、コール・グラフパネルでは対象の関数ボックスをダブルクリックすることで行います。

ジャンプを実行すると、対象の関数／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、定義記述のある行にcaretが移動します。

注意 1. コール・グラフパネルにおいて、表示設定エリア上の  ボタンの設定によりコール・グラフの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。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] を選択することにより同等の動作となります。

2. 【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。

1. #pragma task TASK_A
2. #pragma interrupt INTP0 functionA

なお、次の方法により、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／メモリ・リストへジャンプすることも可能です（ただし、デバッグ・ツールと接続中の場合のみ）。

ジャンプを実行すると、対象関数／変数の開始アドレスに対応する逆アセンブル・データ／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caretが移動します。

- 関数一覧パネル／変数一覧パネルの場合

対象となる関数／変数の表示行を選択したのち（選択行左端にcaret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 [逆アセンブルへジャンプ]、または [メモリへジャンプ] を選択します。

- コール・グラフパネルの場合

対象となる関数ボックスを選択したのち（[対象関数名] コンボ・ボックスに対象関数名が表示されていることを確認）、コンテキスト・メニューの [逆アセンブルへジャンプ]、または [メモリへジャンプ] を選択します。

2.7 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

2.7.1 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。

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。

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。

備考 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

2.7.2 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。

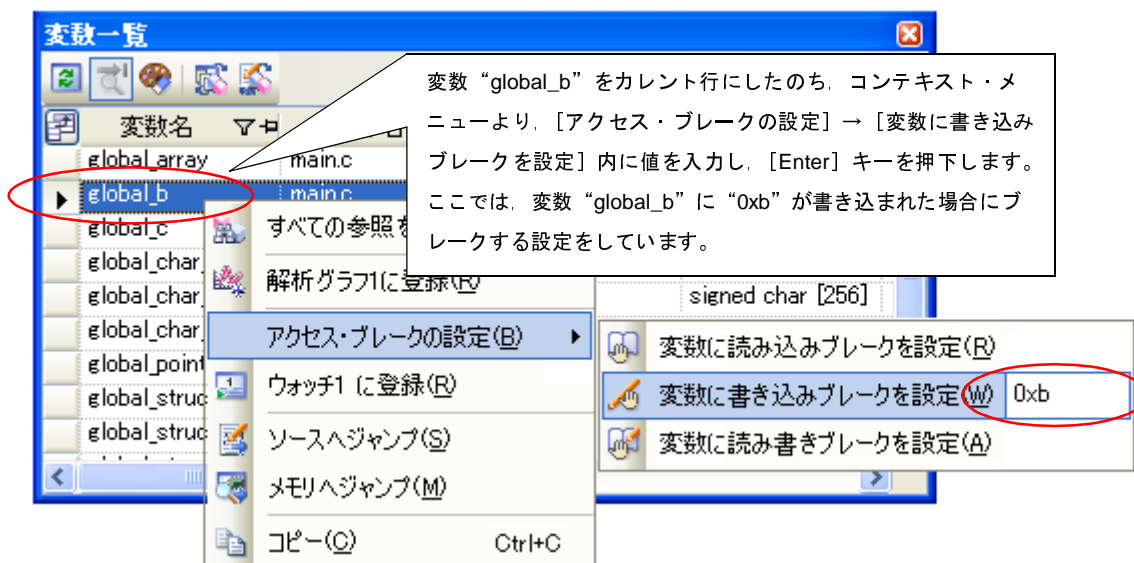
- [アクセス・ブレークの設定] → [変数に読み込みブレークを設定] :リード・アクセスによるブレーク
- [アクセス・ブレークの設定] → [変数に書き込みブレークを設定] :ライト・アクセスによるブレーク
- [アクセス・ブレークの設定] → [変数に読み書きブレークを設定] :リード／ライト・アクセスによるブレーク

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。

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 2-14 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 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。

備考 該当箇所すでにブレーク・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

2.8 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”

2.9 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（複数選択可能）、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—15 関数 / 変数の参照箇所一覧の出力例（出力パネル）

```

出力
参照検索開始↓
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 0): 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↓
参照検索終了↓
[EOF]
すべてのメッセージ / プログラム解析 / デバッグ・ツール / ビルド・ツール / 参照の検索

```

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 ソース・ファイル中の “#if” / “#ifdef” など、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.** [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。

2.10 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、情報ファイル（関数一覧ファイル（*.csfl）／変数一覧ファイル（*.csvl））をインポートすることにより、強制的にこれらの関数／変数の情報を表示することができます。

次の手順により操作を行ってください。

注意 情報ファイルによるインポート／エクスポートの対象は、[関数一覧パネル](#)／[変数一覧パネル](#)上の次の項目に限定されます。

関数一覧パネル	[関数名] / [ファイル名] / [ファイル・パス] / [PE 情報] ^注 / [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]]
変数一覧パネル	[変数名] / [ファイル名] / [関数名] / [ファイル・パス] / [PE 情報] ^注 / [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表示される項目です。

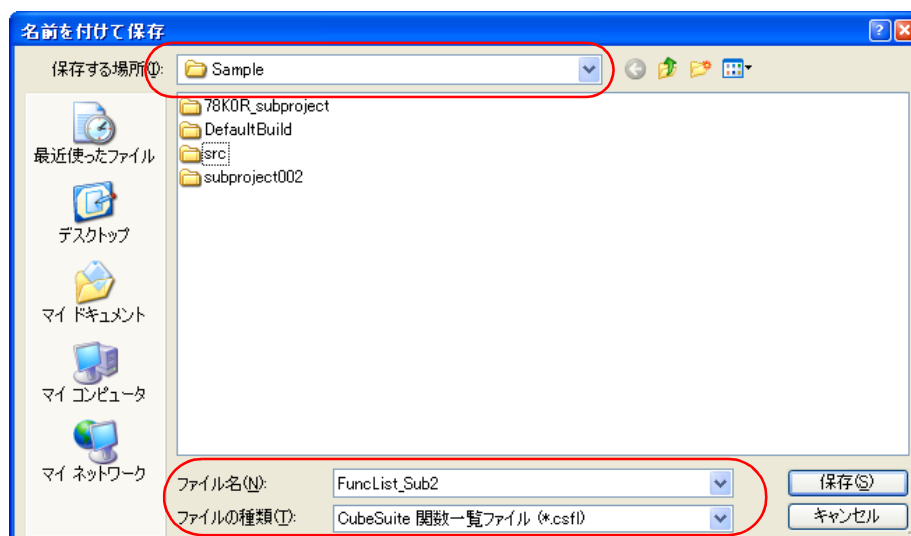
(1) 情報ファイルを生成（エクスポート）する

インポートする必要がある関数／変数の情報ファイル（関数一覧ファイル（*.csfl）／変数一覧ファイル（*.csvl））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—16 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、次のファイル形式を選択します。

呼び出しパネル	ファイル形式	説明
関数一覧 パネル	CubeSuite 関数一覧ファイル (*.csfl)	関数一覧ファイル (*.csfl) を生成します。
変数一覧 パネル	CubeSuite 変数一覧ファイル (*.csvl)	変数一覧ファイル (*.csvl) を生成します。

[ファイル名] エリアにおいて、生成する任意の情報ファイル名を入力します（ただし、拡張子は、関数一覧ファイルの場合は“csfl” / 変数一覧ファイルの場合は“csvl”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（関数一覧ファイル (*.csfl) / 変数一覧ファイル (*.csvl)）が生成されます。

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 生成される情報ファイルには、[関数一覧 パネル](#) / [変数一覧 パネル](#)において選択した関数 / 変数のみの情報が含まれます。

なお、ここで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 / 変数についても情報が必要な場合は、同様の操作により、別途情報ファイルを生成してください。

(2) 情報ファイルをインポートする

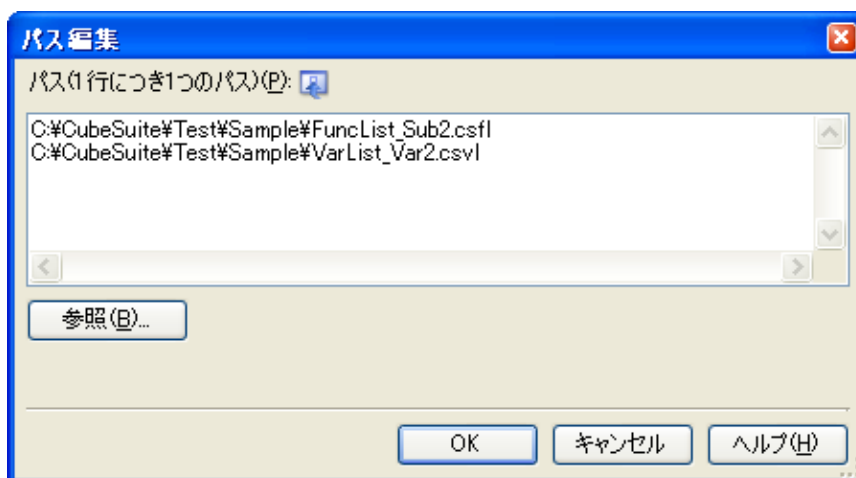
生成（エクスポート）した情報ファイル（関数一覧ファイル (*.csfl) / 変数一覧ファイル (*.csvl)）をインポートする設定を行います。

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—17 [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—18 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。

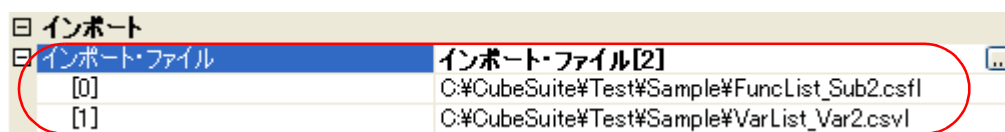
なお、この際に、次の埋め込みマクロ名を使用することができます。

マクロ名	機能
%ProjectName%	プロジェクト名に置換します。
%CubeSuitePath%	CubeSuite のインストール・フォルダの絶対パスに置換します。

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—19 [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 (関数一覧ファイル (*.csfl) / 変数一覧ファイル (*.csvl)) のインポートの設定は完了です。

- 備考 1. 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. 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。

- ファイル名が異なり、かつインポート・ファイル内の属性がスタティック関数／変数である場合、スタティック関数／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・ファイル内の属性がスタティック変数である場合、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／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。



なお、この際におけるマージの決定規則は次のとおりです。

項目	規則
【関数情報】 [ファイル名] / [ファイル・パス] / [PE 情報] ^注 / [スタック・サイズ[バイト]] / [引数の数] / [引数] / [戻り値の型] / [属性] 【変数情報】 [ファイル名] / [関数名] / [ファイル・パス] / [PE 情報] ^注 / [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。
【関数情報】 [コード・サイズ[バイト]] 【変数情報】 [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。

2.11 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**のツールバー上の  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（ ボタンは、デバッグ・ツールと接続時のみ選択可能です）。

なお、**解析グラフパネル**は、最大4個までオープンすることができ、各パネルは“解析グラフ1”、“解析グラフ2”、“解析グラフ3”、“解析グラフ4”の名称で識別されます（タイトルバーに表示）。

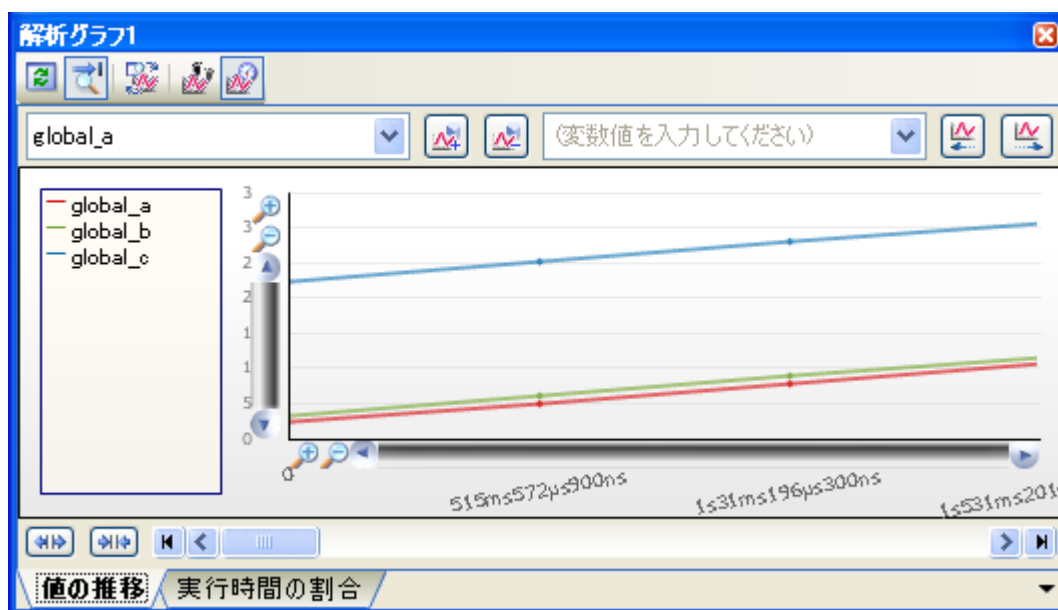
解析グラフパネルでは、次の内容をグラフ化することができます。

- 変数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

注意 **解析グラフパネル**は、デバッグ・ツールと接続している場合のみ表示可能です。

デバッグ・ツールと切断すると、このパネルはすべてクローズします。

図2—20 解析情報のグラフ化（解析グラフパネル）



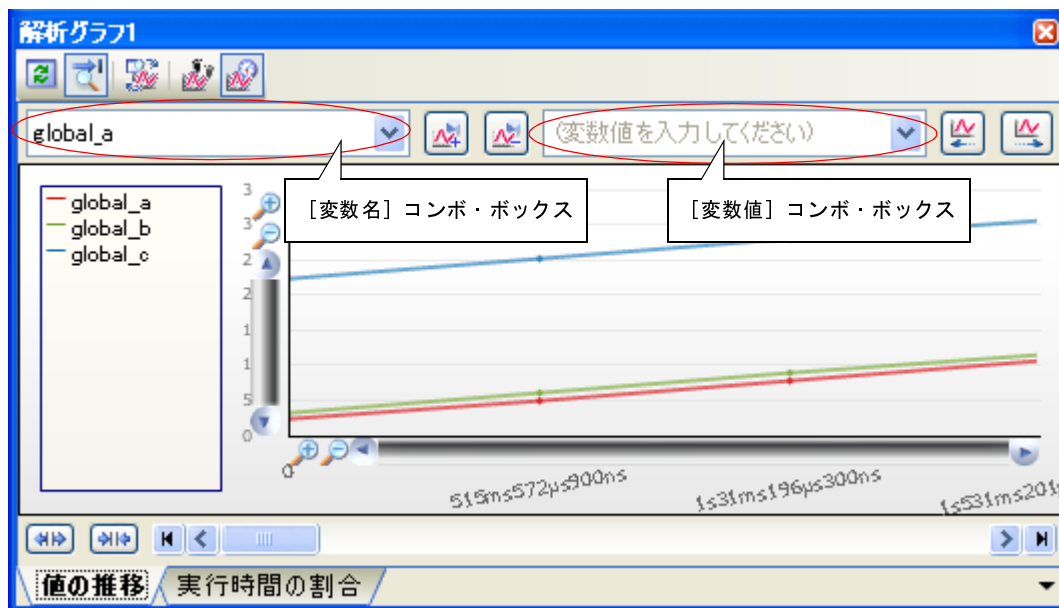
2.11.1 変数値の推移をグラフ化する

登録した変数／レジスタ等の値と時間の関係を折れ線グラフで表示します。

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、**解析グラフパネル**を参照してください。

図 2—21 変数値の推移のグラフ化



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。

登録可能な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
- CPU レジスタ
- SFR 【78K0】 【78K0R】
- IOR 【V850】
- アドレス

グラフ化対象の登録方法には、次の2通りがあります。

- (a) 対象を個別に登録する場合
- (b) ウォッチパネルと連動する場合（自動登録）

注意 プログラム実行中は、グラフ化対象の登録を変更することはできません。

(a) 対象を個別に登録する場合

次のパネルから、対象をこのタブ上へ直接ドラッグ・アンド・ドロップすることにより行います。


- 変数一覧パネル^注
- エディタ パネル
- CPU レジスタ パネル
- SFR パネル【78K0】【78K0R】
- IOR パネル【V850】
- 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登録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”

なお、“解析グラフ 1～4”の名称で識別される解析グラフパネルのこのタブでは、それぞれのパネルのタブごとに対象を個別に登録することができます。また、1つのタブにおいて、最大4個までの対象を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）。

備考 1. グラフ化対象の登録は、このタブ上の「変数名」コンボ・ボックスにおいて、対象名をキーボードより直接入力したのち、 ボタンをクリックすることによっても可能です。

なお、この際の対象名の入力形式については、「表 A—2 グラフ化対象の入力形式」を参照してください。


2. 登録したグラフ化対象を削除する場合は、「変数名」コンボ・ボックスのドロップダウン・リストより削除対象名を選択したのち、 ボタンをクリックします。
3. グラフ化対象の登録完了後、このパネルをクローズしても、登録されているグラフ化対象の情報は失われません。

(b) ウォッチ パネルと連動する場合（自動登録）

このタブのツールバー上の ボタン（トグル）による設定で、現在ウォッチ パネルに登録されているウォッチ式をグラフ化対象にすることができます。

この際のウォッチ パネル（ウォッチ 1～4）と解析グラフ パネル（解析グラフ 1～4）の連動関係は次のとおりです。

- 解析グラフ 1： ウォッチ 1 に登録されている上から 1～4 番目のウォッチ式
- 解析グラフ 2： ウォッチ 2 に登録されている上から 1～4 番目のウォッチ式
- 解析グラフ 3： ウォッチ 3 に登録されている上から 1～4 番目のウォッチ式
- 解析グラフ 4： ウォッチ 4 に登録されている上から 1～4 番目のウォッチ式

なお、「(a) 対象を個別に登録する場合」の操作により、すでに対象を登録しているタブ上でこの機能を有効にした場合は、いったんその登録が解除されウォッチ式との連動を行います（再度  ボタンをクリックすることでこの機能を無効とした際に、解除されていた対象が再登録されます）。

(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2通りがあり、必要に応じてどちらかをツールバー上のボタンにより指定します。

表 2—7 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	ボタン	説明
トレース・データ解析方式 【IECUBE 【78K0R】】 【IECUBE 【V850】】 【IECUBE2】注 【シミュレータ】		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。
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 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）。

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を除く

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—8 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	トレース・データ解析方式 【IECUBE】【シミュレータ】	リアルタイム・サンプリング方式
グラフ表示の可否	デバッグ・ツールの トレース機能 が有効 ^{注1} の場合のみ表示可	デバッグ・ツールの RRM機能 が有効 ^{注1} の場合のみ表示可
グラフ形式	ステップ・プロット折れ線グラフ	通常の折れ線グラフ
グラフ化可能な対象の数	ポイント・トレース・イベントによりトレース・データを取得する場合、デバッグ・ツールの有効イベント数の制限に依存 ^{注2}	【IECUBE 【V850】】 【MINICUBE2 【78K0】】 デバッグ・ツールの RRM 機能が使用可能な領域に依存 ^{注3}
グラフ化可能な対象のサイズ	- 2 バイト以下 【78K0】 【78K0R】 - 4 バイト以下 【V850】	4 バイト以下
時間表示範囲	トレース・データとして記録されている時間	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）
値の遷移箇所	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）	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

相違点	トレース・データ解析方式 【IECUBE】【シミュレータ】	リアルタイム・サンプリング方式
注意	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	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

- 注1. 各機能を有効化するための設定方法についての詳細は、「[1.1.3 デバッグ・ツールの設定](#)」を参照してください。
2. 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 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、RRM機能の対象領域は異なります。
“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 デバッグ編」を参照してください。
4. デバッグ・ツールのプロパティパネルにおける次の設定
- [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]]

- 注意1. プログラム実行中は、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. すでにグラフ・データを取得している場合、この設定を変更すると、保持しているグラフ・データが削除されます（表示中のグラフが消失します）。

(3) グラフを表示する

プログラム実行→停止を行ったのち、ツールバー上の  ボタンをクリックすることにより、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。

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、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。

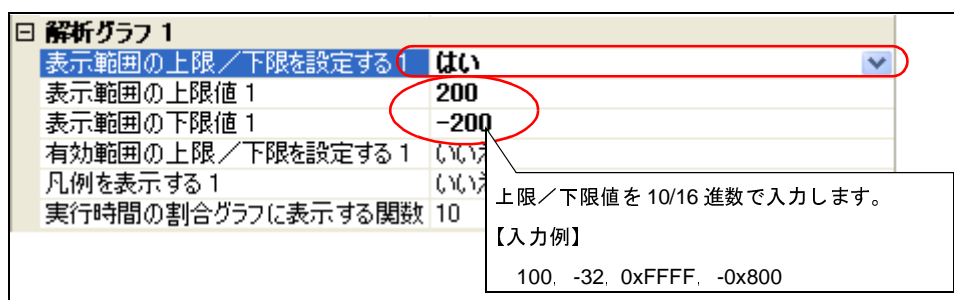
(a) 表示範囲の設定

グラフのY軸に対して、表示する値の範囲を指定することができます。この場合、表示範囲が固定されるため、取得したデータが指定した範囲を越えている場合は該当区間のグラフは表示されません。

設定は、[プロパティパネル](#)の [設定] タブ上の [解析グラフ 1～4] カテゴリ内 [表示範囲の上限／下限を設定する 1～4] プロパティの設定を [はい] にしたのち、下段に表示されるプロパティ項目に上限値／下限値を指定することで行います。

注意 上限値と下限値の差が値に対して非常に小さい場合、指定した範囲を表示できない場合があります。

図 2—22 グラフ表示における表示範囲の設定



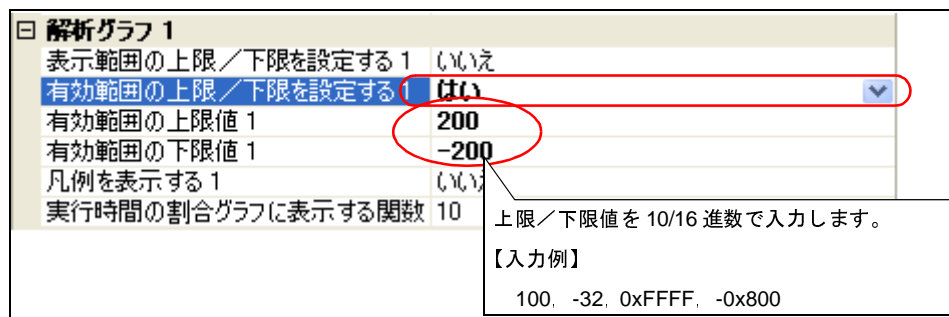
(b) 有効範囲の設定 (異常値の切り捨て)


グラフ化対象の値の有効範囲を指定することができます。この場合、取得した値が指定した有効範囲を越えている場合は異常値として該当値の遷移を表示せず、グラフ上では値が変化しなかったものとして表示されます。

設定は、プロパティパネルの [設定] タブ上の [解析グラフ 1~4] カテゴリ内 [有効範囲の上限/下限を設定する 1~4] プロパティの設定を [はい] にしたのち、下段に表示されるプロパティ項目に上限値/下限値を指定することで行います。

注意 この機能は、リアルタイム・サンプリング方式により取得したグラフ・データの場合のみが対象となります。

図 2—23 グラフ表示における有効範囲の設定



備考 1. ツールバー上の  ボタン (トグル) による設定で、プログラムの実行が停止するごとに表示内容を更新させることができます。

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] (デフォルト) 以外に指定を変更している場合、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)。

2. プロパティパネルの [設定] タブ上の [解析グラフ 1~4] カテゴリ内 [凡例を表示する] の設定により、グラフ上の凡例を非表示にすることができます。

3. 取得したグラフ・データがバッファ容量を越えた場合、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. 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**によりグラフ表示を行う場合、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。

(4) 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。

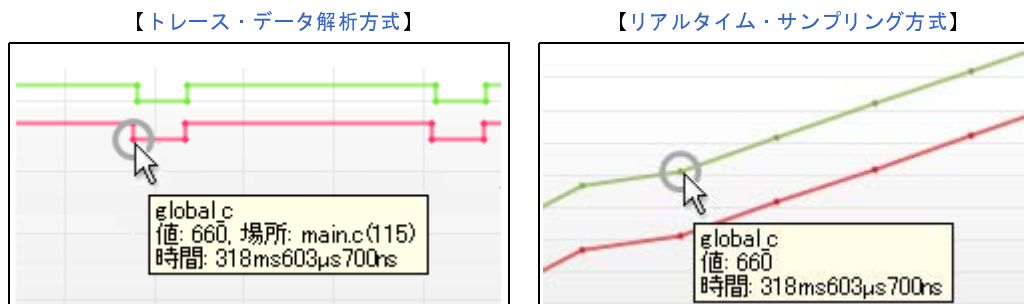
(a) 遷移箇所のポップアップ表示

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。

図 2—24 遷移箇所のポップアップ表示例



備考 [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。

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。

(b) 値の検索

特定のグラフにおいて、指定した値に遷移した箇所を検索します。

操作は、まず、[変数名] コンボ・ボックス内のドロップダウン・リストにより検索対象のグラフを指定したのち、[変数値] コンボ・ボックスに検索する遷移値を入力します。



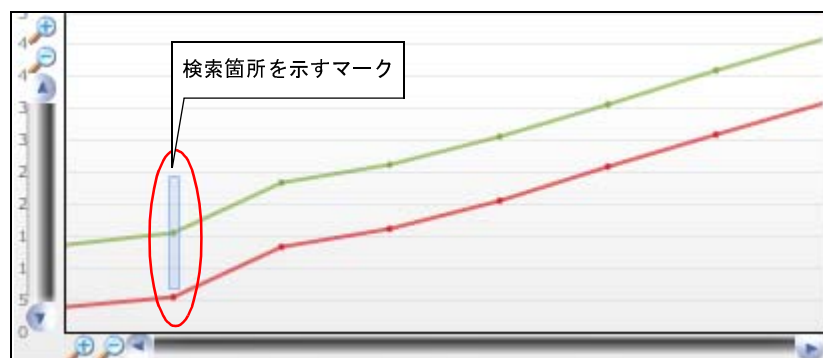
次に、このエリアの  /  ボタンをクリックすることにより、指定したグラフにおいて、指定した値に遷移した箇所を左/右方向に検索し、検索箇所に次のマークを表示します。


図 2—25 グラフ・データの検索箇所



2.11.2 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、[解析グラフ パネル](#)の[\[実行時間の割合\]](#)タブで行います。

ツールバー上の ボタンをクリックすることにより、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）を基にグラフが表示されます（トレース・メモリが空の場合、“実行時間情報がありません”とメッセージを表示します）。

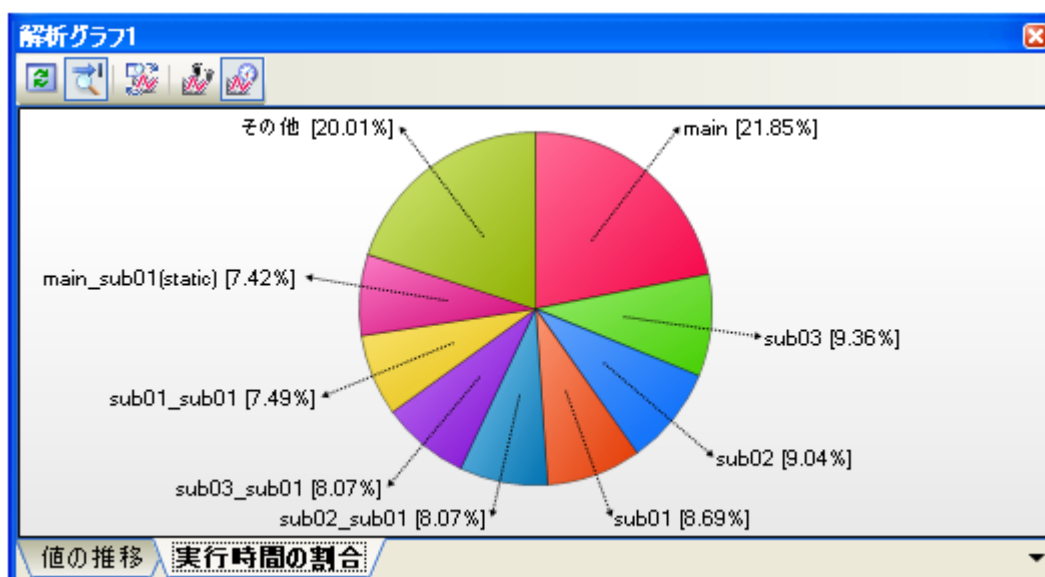
なお、表示される各エリアについての詳細は、[解析グラフ パネル](#)を参照してください。

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、このグラフを表示することはできません。

2. [\[IECUBE \[78K0\]\]](#)

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—26 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。

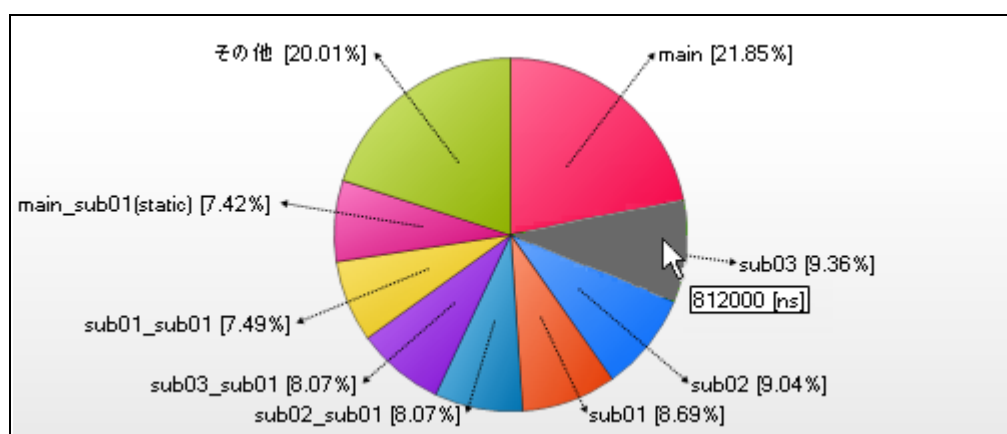
操作は、プロパティパネルの [設定] タブ上の [解析グラフ 1～4] カテゴリ内 [実行時間の割合グラフに表示する関数の数 1～4] プロパティの指定により行います（デフォルトでは [10] が指定されます）。


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—27 実行時間のポップアップ表示例



備考 ツールバー上の  ボタン（トグル）による設定で、プログラムの実行が停止するごとに表示内容を更新させることができます。

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]（デフォルト）以外に指定を変更している場合、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）。

2.12 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／コール・グラフパネル／解析グラフパネルの内容は、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 Excel 97 - Excel 2003、および 5.0/95 ブック (*.xls) 形式
CubeSuite 関数一覧ファイル (*.csfl)	関数情報をインポートするためのファイル形式（ 「2.10 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 Excel 97 - Excel 2003、および 5.0/95 ブック (*.xls) 形式
CubeSuite 変数一覧ファイル (*.csvl)	変数情報をインポートするためのファイル形式（ 「2.10 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) 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)。

(4) 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 Excel 97 - Excel 2003、および 5.0/95 ブック (*.xls) 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)

注意 現在表示しているタブの内容のみが保存対象となります。

付録A ウィンドウ・リファレンス

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

A.1 説明

次に、解析ツールで使用するウィンドウ／パネル／ダイアログの一覧を示します。

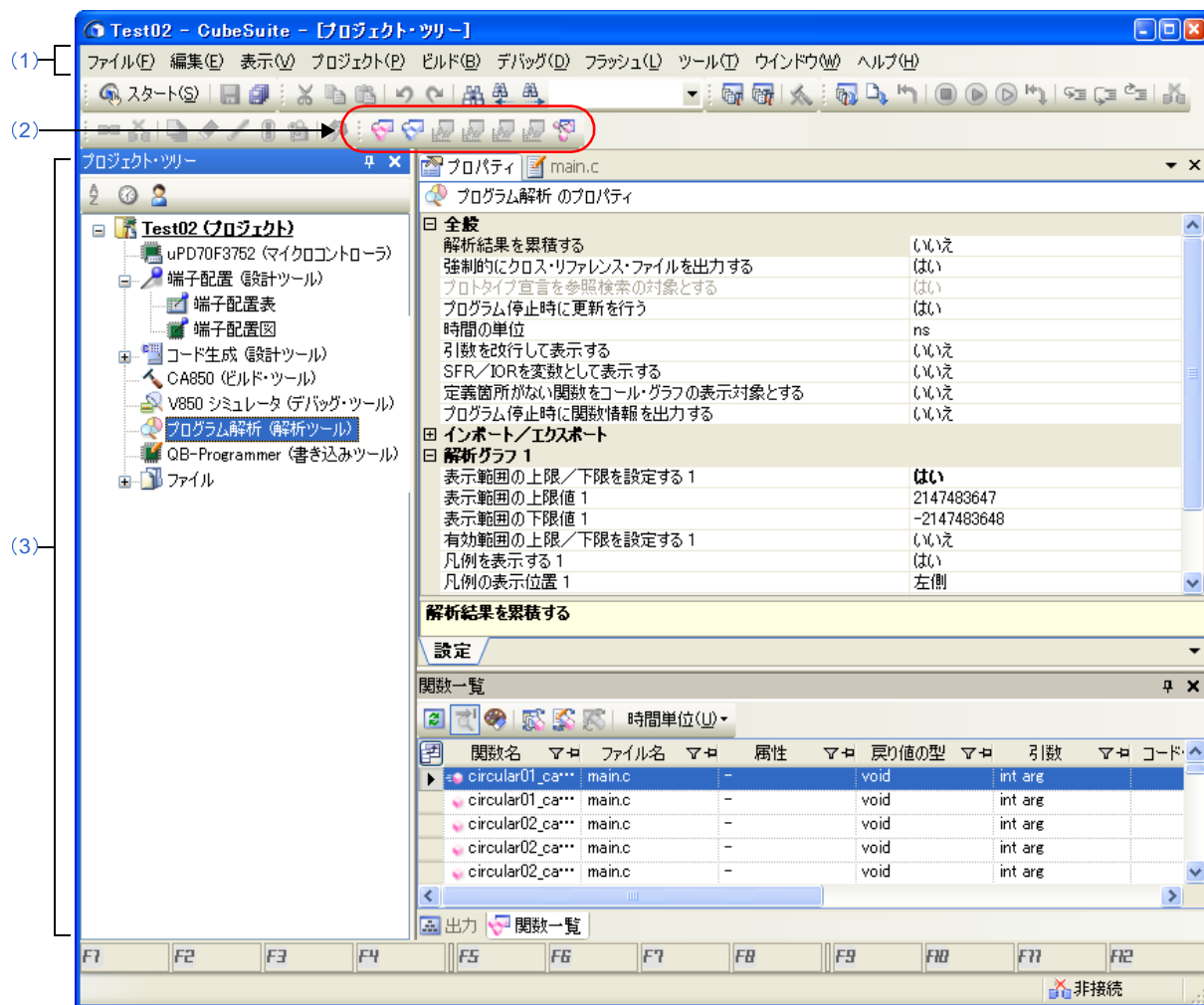
表 A-1 ウィンドウ／パネル／ダイアログ一覧

ウィンドウ／パネル／ダイアログ名	機能概要
メイン・ウィンドウ	CubeSuite を起動した際、最初にオープンするウィンドウ
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
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
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
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
パス編集 ダイアログ	情報ファイル（関数一覧ファイル（*.csfl）／変数一覧ファイル（*.csvl））をインポートする際のファイルの指定
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／コール・グラフ パネル／解析グラフ 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（*.csfl）／変数一覧ファイル（*.csvl））の生成

メイン・ウィンドウ

CubeSuite を起動した際、最初にオープンするウィンドウです。
 解析ツールを使用する際は、このウィ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・ウィンドウ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]

[オープン方法]

- Windows の [スタート] メニュー → [プログラム] → [NEC 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)。

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 1	解析グラフ パネル (解析グラフ 1) をオープンします。 ただし、デバッグ・ツールと切断時は無効となります。
解析グラフ 2	解析グラフ パネル (解析グラフ 2) をオープンします。 ただし、デバッグ・ツールと切断時は無効となります。
解析グラフ 3	解析グラフ パネル (解析グラフ 3) をオープンします。 ただし、デバッグ・ツールと切断時は無効となります。
解析グラフ 4	解析グラフ パネル (解析グラフ 4) をオープンします。 ただし、デバッグ・ツールと切断時は無効となります。
コール・グラフ	コール・グラフ 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。
 (1 番目)	解析グラフ パネル (解析グラフ 1) をオープンします。 ただし、デバッグ・ツールと切断時は無効となります。 [表示] メニュー → [解析グラフ 1] の選択と同等です。
 (2 番目)	解析グラフ パネル (解析グラフ 2) をオープンします。 ただし、デバッグ・ツールと切断時は無効となります。 [表示] メニュー → [解析グラフ 2] の選択と同等です。
 (3 番目)	解析グラフ パネル (解析グラフ 3) をオープンします。 ただし、デバッグ・ツールと切断時は無効となります。 [表示] メニュー → [解析グラフ 3] の選択と同等です。
 (4 番目)	解析グラフ パネル (解析グラフ 4) をオープンします。 ただし、デバッグ・ツールと切断時は無効となります。 [表示] メニュー → [解析グラフ 4] 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。

(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。

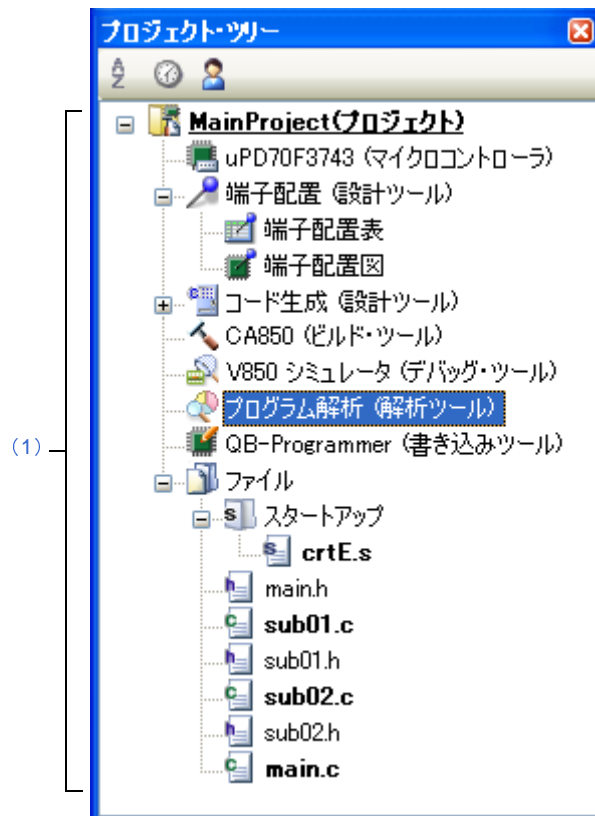
解析ツールが使用するパネルについての詳細は、次の各パネルの項を参照してください。

- 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
- 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]

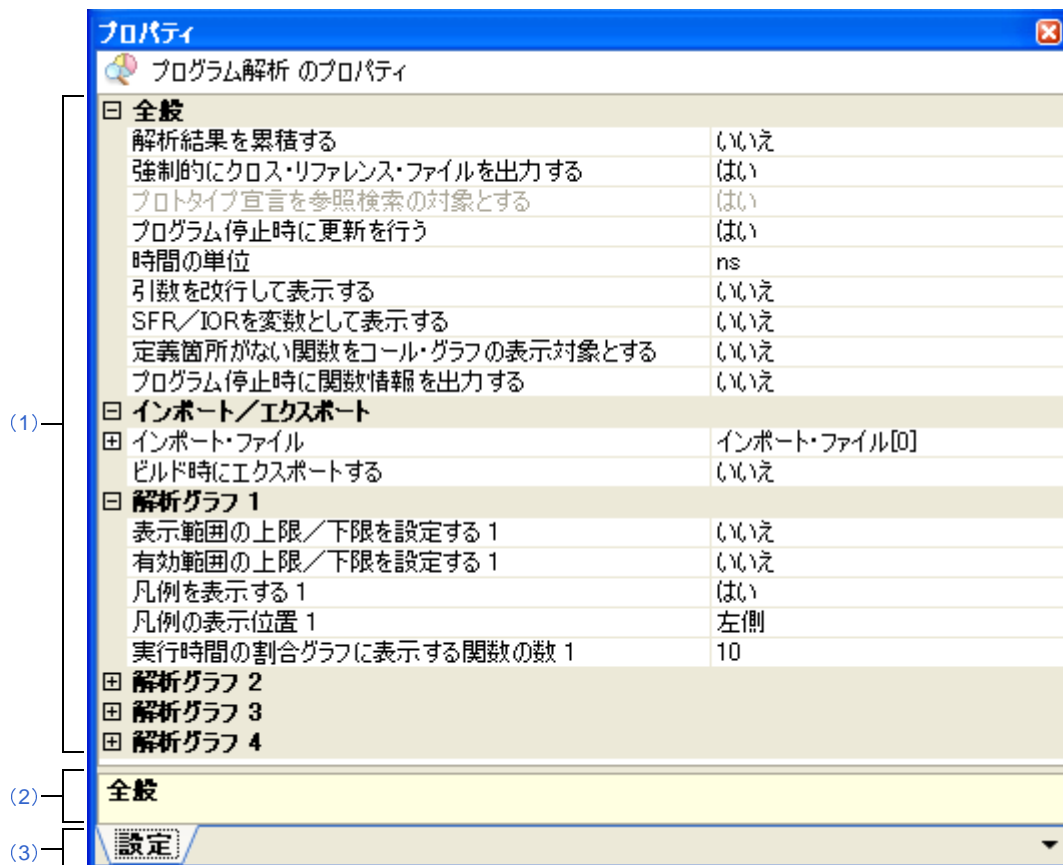
[プログラム解析 (解析ツール)] ノードをマウスで右クリックすることにより表示されるコンテキスト・メニューの各項目、および機能は次のとおりです。

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するために、次のカスケード・メニューを表示します。 ただし、デバッグ・ツールと切断時は無効となります。
解析グラフ 1	解析グラフ 1 をオープンします。
解析グラフ 2	解析グラフ 2 をオープンします。
解析グラフ 3	解析グラフ 3 をオープンします。
解析グラフ 4	解析グラフ 4 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。
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



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

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) プロパティの説明エリア

詳細情報表示／変更エリアで選択したカテゴリやプロパティの簡単な説明を表示します。

(3) タブ選択エリア

タブを選択することにより、詳細情報を表示するカテゴリが切り替わります。

このパネルには、次のタブが存在します（各タブ上における表示内容／設定方法についての詳細は、該当するタブの項を参照してください）。

- [設定] 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の場合、選択している文字列を切り取ってクリップ・ボードに移動します。
コピー	選択しているプロパティの値の文字列をクリップ・ボードにコピーします。
貼り付け	プロパティの値を編集の場合、クリップ・ボードの内容を挿入します。
削除	プロパティの値を編集の場合、選択している文字列を削除します。
すべて選択	プロパティの値を編集の場合、選択しているプロパティの値文字列をすべて選択します。

[コンテキスト・メニュー]

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

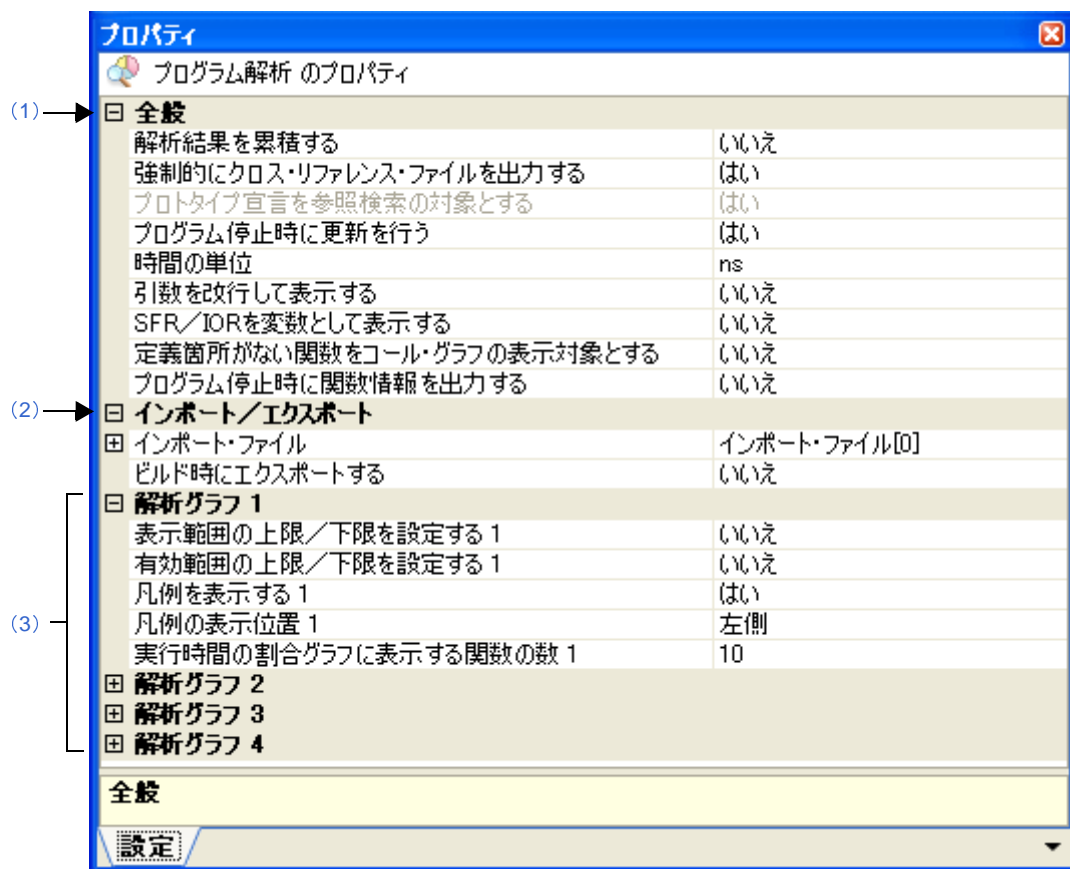
元に戻す	直前に行ったプロパティの値の編集作業を取り消します。
切り取り	プロパティの値を編集中の場合、選択している文字列を切り取ってクリップ・ボードに移動します。
コピー	選択しているプロパティの値文字列をクリップ・ボードにコピーします。
貼り付け	プロパティの値を編集中の場合、クリップ・ボードの内容を挿入します。
削除	プロパティの値を編集中の場合、選択している文字列を削除します。
すべて選択	プロパティの値を編集中の場合、選択しているプロパティの値文字列をすべて選択します。

[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。

- (1) [全般]
- (2) [インポート/エクスポート]
- (3) [解析グラフ 1 ~ 4]

図 A—4 プロパティ パネル : [設定] タブ





[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。

解析結果を累積する	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指定します。 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]		
	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	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。
		いいえ	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。
強制的にクロス・リファレンス・ファイルを出力する	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指定します。		
	デフォルト	はい	
	変更方法	【CA78K0】 【CA78K0R】 【CA850】 ドロップダウン・リストによる選択 【CX】 変更不可	
	指定可能値	はい	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。
		いいえ	ビルド・ツールのプロパティ設定を優先します。
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（「2.9 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指定します。		
	デフォルト	はい	
	変更方法	【CA78K0】 【CA78K0R】 【CA850】 変更不可 【CX】 ドロップダウン・リストによる選択	
	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 / コール・グラフパネル / 解析グラフパネル の表示内容の更新を行うか否かを指定します。						
	デフォルト	はい					
	変更方法	ドロップダウン・リストによる選択					
	指定可能値	<table border="1"> <tr> <td>はい</td> <td>プログラム実行の停止後、表示内容の更新を行います。</td> </tr> <tr> <td>いいえ</td> <td>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。</td> </tr> <tr> <td>個別に指定する</td> <td>各パネルのツールバー上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）。</td> </tr> </table>	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
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上の  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指定します。						
	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指定します。						
	デフォルト	いいえ					
	変更方法	ドロップダウン・リストによる選択					
	指定可能値	<table border="1"> <tr> <td>はい</td> <td>値ごとに改行し、複数行で表示します。</td> </tr> <tr> <td>いいえ</td> <td>各値を“,”で区切り、改行せずに表示します。</td> </tr> </table>	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。	
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指定します。						
	デフォルト	いいえ					
	変更方法	ドロップダウン・リストによる選択					
	指定可能値	<table border="1"> <tr> <td>はい</td> <td>SFR/IOR を変数とみなして表示します。</td> </tr> <tr> <td>いいえ</td> <td>SFR/IOR を変数とみなしません。</td> </tr> </table>	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。	
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）関数もコール・グラフに表示するか否かを指定します。						
	デフォルト	いいえ					
	変更方法	ドロップダウン・リストによる選択					
	指定可能値	<table border="1"> <tr> <td>はい</td> <td>定義箇所がない関数もコール・グラフに表示します。</td> </tr> <tr> <td>いいえ</td> <td>定義箇所がない関数はコール・グラフに表示しません。</td> </tr> </table>	はい	定義箇所がない関数もコール・グラフに表示します。	いいえ	定義箇所がない関数はコール・グラフに表示しません。	
はい	定義箇所がない関数もコール・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 【V850】	STF 用の情報ファイル ^{注2} 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指定します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい [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。

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

- 【CA78K0】【CA78K0R】

[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

- 【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

2. 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) [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、「2.10 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。

インポート・ファイル	インポートするファイルを指定します。 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。	
	デフォルト	インポート・ファイル [0]
	変更方法	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クルード・ファイルを指定することはできません）。
ビルド時にエクスポートする	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 (*.csfl) /変数一覧ファイル (*.csvl)）を生成（エクスポート）するか否かを指定します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい 情報ファイルを生成します。
	いいえ	情報ファイルを生成しません。

関数用エクスポート・ファイル名	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。 生成する情報ファイル（関数一覧ファイル (*.csfl)）の名前を指定します。	
	デフォルト	%ProjectName%.csfl
	変更方法	キーボードからの直接入力 ただし、拡張子 (*.csfl) を変更することはできません。 なお、“%ProjectName%” は、プロジェクト名に置換するためのマクロ名です。
変数用エクスポート・ファイル名	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。 生成する情報ファイル（変数一覧ファイル (*.csvl)）の名前を指定します。	
	デフォルト	%ProjectName%.csvl
	変更方法	キーボードからの直接入力 ただし、拡張子 (*.csvl) を変更することはできません。 なお、“%ProjectName%” は、プロジェクト名に置換するためのマクロ名です。

(3) [解析グラフ 1 ~ 4]

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、「2.11 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。

表示範囲の上限/下限を設定する 1 ~ 4	解析グラフパネルの [値の推移] タブで表示するグラフにおいて、Y軸の上限/下限を設定するか否かを指定します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい Y軸の上限/下限を設定します。 いいえ Y軸の上限/下限の設定を行いません。
表示範囲の上限値 1 ~ 4	このプロパティは、[表示範囲の上限/下限を設定する 1 ~ 4] 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。 表示するグラフにおいて、Y軸の上限値を指定します。	
	デフォルト	2147483647
	変更方法	キーボードからの直接入力
	指定可能値	- 10進数指定の場合 -2147483648 ~ 4294967295 - 16進数指定の場合 (“0x” の付与) 0x80000000 ~ 0xFFFFFFFF

表示範囲の下限値 1～4	このプロパティは、[表示範囲の上限/下限を設定する 1～4] 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。 表示するグラフにおいて、Y 軸の下限値を指定します。	
	デフォルト	-2147483648
	変更方法	キーボードからの直接入力
	指定可能値	- 10 進数指定の場合 -2147483648 ～ 4294967295 - 16 進数指定の場合 (“0x” の付与) 0x80000000 ～ 0xFFFFFFFF
有効範囲の上限/下限を設定する 1～4	解析グラフ パネルの [値の推移] タブで表示するグラフにおいて、値の有効範囲を設定するか否かを指定します注。	
	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 値の有効範囲を設定します。 いいえ 値の有効範囲の設定を行いません。
有効範囲の上限値 1～4	このプロパティは、[有効範囲の上限/下限を設定する 1～4] 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。 表示するグラフにおいて、値の有効範囲の上限値を指定します。 ここで指定した数値より大きい値への推移はグラフに表示されません。	
	デフォルト	2147483647
	変更方法	キーボードからの直接入力
	指定可能値	- 10 進数指定の場合 -2147483648 ～ 4294967295 - 16 進数指定の場合 (“0x” の付与) 0x80000000 ～ 0xFFFFFFFF
有効範囲の下限値 1～4	このプロパティは、[有効範囲の上限/下限を設定する 1～4] 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。 表示するグラフにおいて、値の有効範囲の下限値を指定します。 ここで指定した数値より小さい値への推移はグラフに表示されません。	
	デフォルト	-2147483648
	変更方法	キーボードからの直接入力
	指定可能値	- 10 進数指定の場合 -2147483648 ～ 4294967295 - 16 進数指定の場合 (“0x” の付与) 0x80000000 ～ 0xFFFFFFFF
凡例を表示する 1～4	解析グラフ パネルの [値の推移] タブで表示するグラフにおいて、凡例を表示するか否かを指定します。	
	デフォルト	はい
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい 凡例を表示します。 いいえ 凡例を表示しません。

凡例の表示位置 1～4	このプロパティは、[凡例を表示する 1～4] 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。 表示するグラフにおいて、凡例を表示する際の位置を指定します。		
	デフォルト	左側	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	上側	凡例をグラフの上部に表示します。
		下側	凡例をグラフの下部に表示します。
左側		凡例をグラフの左に表示します。	
右側		凡例をグラフの右に表示します。	

実行時間の割合グラフに表示する関数の数 1～4	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)

注 有効範囲の設定は、[リアルタイム・サンプリング方式](#)でグラフ・データを取得したグラフを表示しているパネルのみが対象となります。

関数一覧 パネル

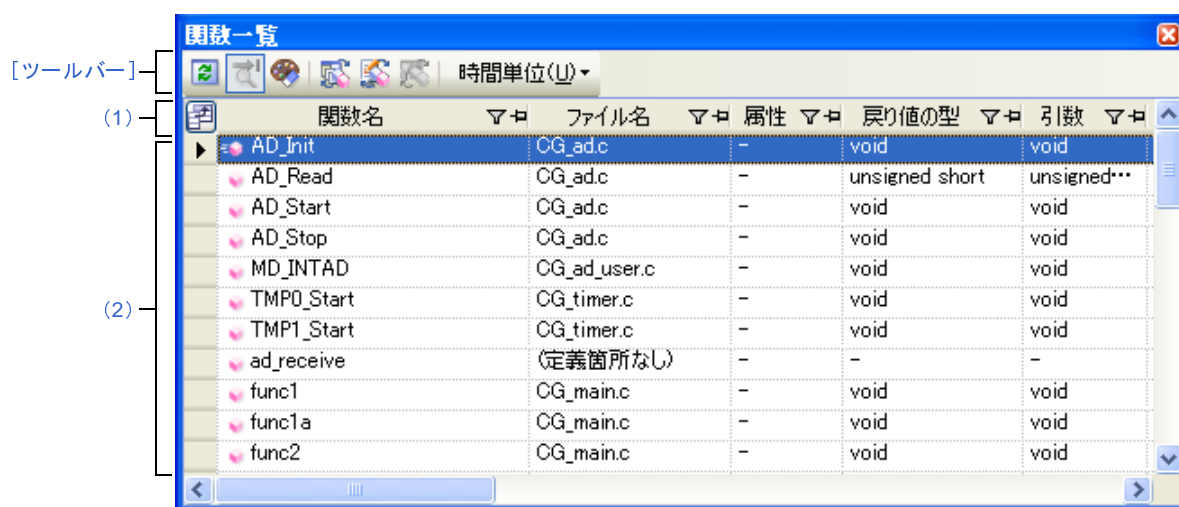
取得した関数情報を表示します。

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 【CA78K0】 【CA78K0R】 【CA850】

【ビルド】メニュー→【クリーン・プロジェクト】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—5 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・ウィンドウのツールバー上の  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
	固定表示設定の有無を示します（「2.5.3 特定項目を固定表示に設定する」参照）。
	ソート表示設定の有無を示します（「2.5.4 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（「2.5.5 解析情報をフィルタ表示する」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（「2.5 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、ツールバー上の  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の名前を表示します。 最大表示文字数は次のとおりです。 - 【CA78K0】【CA78K0R】: 249文字 - 【CA850】: 1022文字 - 【CX】: 2046文字 なお、表示アイコンは次の意味を示します。 -  : 関数
ファイル名	静的解析	関数が定義されている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。
PE情報 【V850E2M】	静的解析	関数が実行されるPEを次のように表示します。 - PE <i>n</i> で実行: PE <i>n</i> - 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。
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、「 2.10 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。

項目	種別	内容
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。 - 【CA78K0】 static, callt, callf, noauto, norec, interrupt, bank, rtos task, rtos interrupt - 【CA78K0R】 static, callt, interrupt, near, far, rtos task, rtos interrupt - 【CA850】 static - 【CX】 static, interrupt, inline, delete
戻り値の型	静的 解析	関数の戻り値の型 ^{注2} を表示します。 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、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。 - 【CA78K0】【CA78K0R】: 7 個 - 【CA850】: 6 個 - 【CX】: 8 個
引数の数 ^{注1}	静的 解析	関数の引数の値を 10 進数で表示します。 引数に可変引数が含まれる場合は、“可変”文字列が付与されます。 ただし、解析未完の場合は“-”を表示します。
引数	静的 解析	関数の引数の型 ^{注2} ／仮引数名を表示します。 複数の引数が存在する場合は、“;”で区切り表示します。 可変引数の場合は、すでに定義されている引数の型のみを表示します。 引数が存在しない場合は“void”を表示します。 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、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）。 - 【CA78K0】【CA78K0R】: 8 個 - 【CA850】: 6 個 - 【CX】: 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]	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。 【CA78K0】【CA78K0R】 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。

項目	種別	内容
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 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。 【CA78K0】 【CA78K0R】 “#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。 【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。
実行回数 【IECUBE】 【IECUBE2】 【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。
実行時間 [単位] 【IECUBE 【78K0R】】 【IECUBE 【V850】】 【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) [%] 【IECUBE 【78K0R】】 【IECUBE 【V850】】 【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。


項目	種別	内容
平均実行時間 [単位] 【IECUBE 【78K0R】】 【IECUBE 【V850】】 【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】	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、「2.5.1 表示項目を設定する」を参照してください。

2. 【CX】

“typedef” を使用して基本形に別名を付与している場合、次の型を表示します。

typedef の元の型	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

- 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。
- 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。
- 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 [カバレッジ情報のクリア] を選択したのち、ツールバー上の  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。

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—6 [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、「2.5.1 表示項目を設定する」を参照してください。

注意 1. 【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]]

2. 【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 【V850】】

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。

4. 【CA78K0】【CA78K0R】

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. 【CX】

最適化により削除された未使用 `static` 関数は一覧に表示されません。

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. 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](#)

2. 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。

ただし、プログラム実行中はすべて無効となります。

	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。
	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)。
 【V850E2M】	デバッグ・マネージャパネルで現在選択している PE、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【V850E2M】」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、このボタンは無効となります。

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（ 「2.12 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（ 「2.12 解析情報をファイルに保存する」 参照）。

[[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（複数選択可能）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（ 「2.9 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（ 「2.7.1 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。
ソースヘジャンプ	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（ 「2.6 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（「2.6 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（「2.6 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（複数選択可能）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

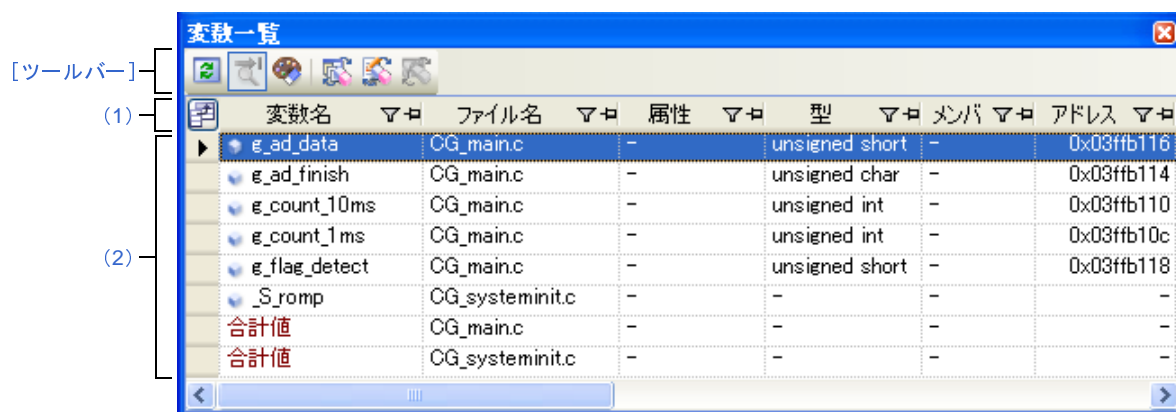
取得した変数情報を表示します。

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 【CA78K0】 【CA78K0R】 【CA850】

【ビルド】メニュー→【クリーン・プロジェクト】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—7 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・ウィンドウのツールバー上の  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
	固定表示設定の有無を示します（「 2.5.3 特定項目を固定表示に設定する 」参照）。
	ソート表示設定の有無を示します（「 2.5.4 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（「 2.5.5 解析情報をフィルタ表示する 」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（「 2.5 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、ツールバー上の 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	<p>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の名前を表示します。</p> <p>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。</p> <p>最大表示文字数は次のとおりです。</p> <ul style="list-style-type: none"> - 【CA78K0】【CA78K0R】: 249 文字 - 【CA850】: 1022 文字 - 【CX】: 2046 文字 <p>なお、表示アイコンは次の意味を示します。</p> <ul style="list-style-type: none"> - : 変数 - : SFR 【78K0】【78K0R】/IOR 【V850】
ファイル名	静的解析	<p>変数が定義されているCソース・ファイル名を表示します（パスを除く）。</p> <p>ただし、プロジェクト内の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。</p> <p>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。</p> <p>【CA78K0】【CA78K0R】</p> <p>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。</p>
関数名 ^{注1}	静的解析	<p>変数が定義されている関数の関数名を表示します。</p> <p>ただし、関数内スタティック変数以外の場合は空欄となります。</p>
ファイル・パス ^{注1}	静的解析	<p>変数が定義されているCソース・ファイルの絶対パスを表示します。</p> <p>ただし、プロジェクト内の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。</p> <p>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。</p>
PE 情報 【V850E2M】	静的解析	<p>変数に対してアクセス可能なPEを次のように表示します。</p> <ul style="list-style-type: none"> - PE_nからアクセス: PE_n - PEでアクセス: Common - 不明: - <p>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。</p>
インポート ^{注1}	静的解析	<p>変数情報の取得先を次のように表示します。</p> <ul style="list-style-type: none"> - 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 <p>なお、インポート機能については、「2.10 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。</p>

項目	種別	内容
属性	静的解析	<p>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。</p> <ul style="list-style-type: none"> - 【CA78K0】 static, const, volatile, sreg, rwsfr, rosfr, wosfr - 【CA78K0R】 static, const, volatile, sreg, rwsfr, rosfr, wosfr/near, far - 【CA850】 static - 【CX】 static, ior, const, volatile, delete
型	静的解析	<p>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）。</p> <ul style="list-style-type: none"> - 【CA78K0】【CA78K0R】: 8個 - 【CA850】: 6個 - 【CX】: 8個
メンバ	静的解析	<p>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。</p>
アドレス	静的解析	<p>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。</p>
サイズ[バイト]	静的解析	<p>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。</p>
参照回数	静的解析	<p>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。</p> <p>【CX】</p> <p>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。</p>


項目	種別	内容
リード回数 【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。
ライト回数 【IECUBE】 【シミュレータ】 【IECUBE2】	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} 【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 【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/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。 【CA78K0】【CA78K0R】 2バイト以下の変数、またはSFRのみ表示可能です。 【CA850】【CX】 4バイト以下の変数、またはIORのみ表示可能です。
最大値 【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/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。 【CA78K0】【CA78K0R】 2バイト以下の変数、またはSFRのみ表示可能です。 【CA850】【CX】 4バイト以下の変数、またはIORのみ表示可能です。
データ・カバレッジ[%] 【IECUBE 【78K0】】 【IECUBE 【78K0R】】 【シミュレータ】	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 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、「[2.5.1 表示項目を設定する](#)」を参照してください。

2. 【CX】

“typedef” を使用して基本形に別名を付与している場合、次の型を表示します。

typedef の元の型	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 [カバレッジ情報のクリア] を選択したのち、ツールバー上の  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。

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—8 [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
合計値	CG_systeminit.c		0	1	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、「[2.5.1 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 型変数**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 型変数**に対するこれらの項目の値は同じ値となります。

2. 【CA78K0】【CA78K0R】

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。

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. 【CX】

最適化により削除された未使用 **static** 変数は一覧に表示されません。

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。

【CA78K0】【CA78K0R】【V850】

- [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。

【CX】

- [型] / [メンバ]







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。
- 2. 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](#)
- 3. 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](#)

[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。

ただし、プログラム実行中はすべて無効となります。

	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。
	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)。
 [V850E2M]	デバッグ・マネージャパネル で現在選択しているPE、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【V850E2M】 」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、このボタンは無効となります。

[[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(「 2.12 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(「 2.12 解析情報をファイルに保存する 」参照)。

[[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 (複数選択可能) 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 (「2.9 参照箇所を一覧表示する」参照)。
解析グラフ 1 に登録	選択している行 (複数選択可能) の変数を解析グラフ パネル (解析グラフ 1) に登録します (「2.11.1 変数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・ブレイクの設定	ブレイク関連のイベントを設定するために、次のカスケード・メニューを表示します (「2.7.2 変数にブレイク・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イクを設定	カレント行の変数に、リード・アクセスのブレイク・イベントを設定します。
変数に書き込みブレイクを設定	カレント行の変数に、ライト・アクセスのブレイク・イベントを設定します。
変数に読み書きブレイクを設定	カレント行の変数に、リード/ライト・アクセスのブレイク・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 (複数選択可能) 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 (「2.8 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。
ソースヘジャンプ	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 (「2.6 定義箇所へジャンプする」参照)。
メモリヘジャンプ	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 (「2.6 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。

コール・グラフパネル

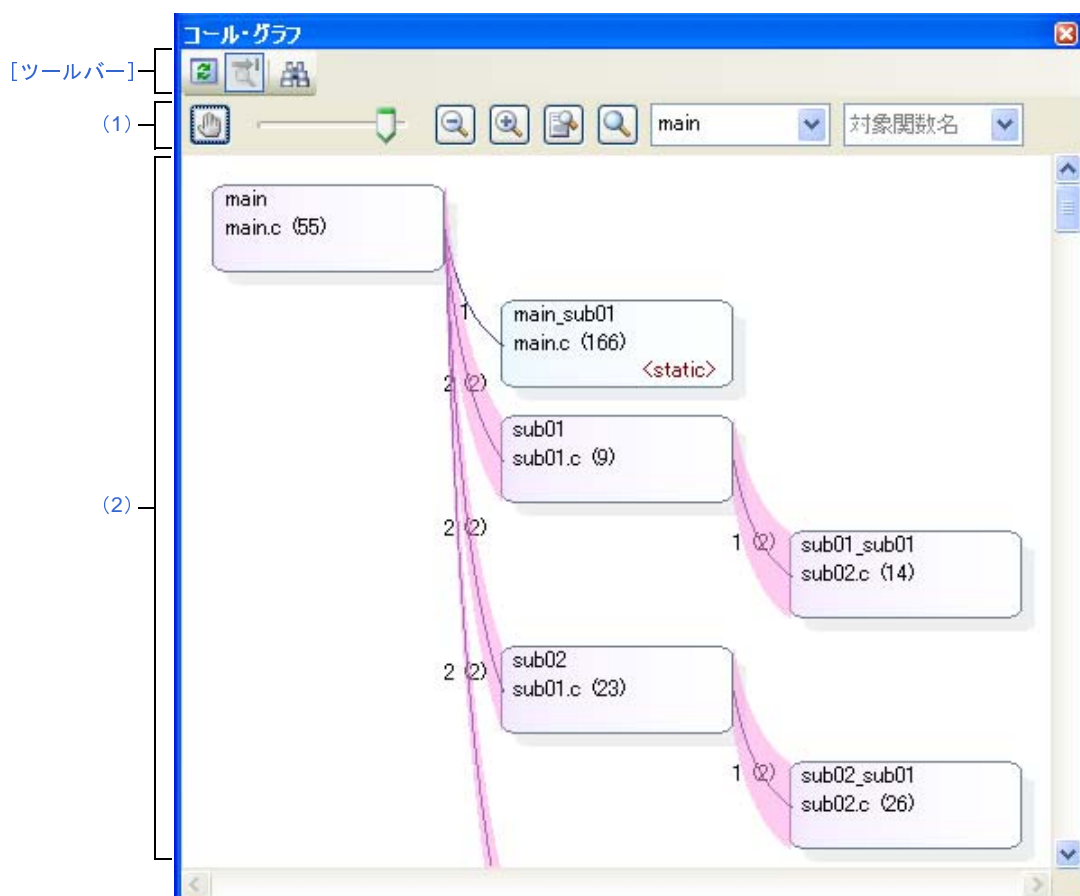
関数間の呼び出し関係をツリー構造の図（コール・グラフ）で表示します。

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）を表示することはできません。
2. 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 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をコール・グラフに含めることができます。


図 A—9 コール・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 (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・ウィンドウのツールバー上の  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) 表示設定エリア

 スクロール・ボタン	<p>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。</p> <p>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をクリックしても、その関数を強調表示することはできません ([対象関数名] コンボ・ボックスに反映されません)。また、関数の情報をポップアップ表示することもできません。</p>
 ズーム・スライダ	<p>コール・グラフのズーム率を変更します。</p> <p>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)。</p>
 縮小/拡大ボタン	<p>コール・グラフのズーム率を変更します。</p> <p>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。</p>
 パネル幅ボタン	<p>コール・グラフのズーム率を自動で変更します。</p> <p>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。</p>
 標準ボタン	<p>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。</p>
 [親関数名] コンボ・ボックス (コンボ・ボックス左)	<p>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。</p> <p>デフォルトでは、“main”、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”)。</p>

<p>対象関数名 </p> <p>[対象関数名] コンボ・ボックス (コンボ・ボックス右)</p>	<p>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。</p> <ul style="list-style-type: none"> - [親関数名] コンボ・ボックスが空欄の場合 プログラム中に存在するすべての関数名 - [親関数名] コンボ・ボックスで親関数を指定している場合 該当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） <p>なお、スクロール・ボタンによりスクロールを禁止している場合は、コール・グラフ内の関数ボックスをクリックすることで、該当関数名がコンボ・ボックスに反映されます。</p>
--	--

注 【V850E2M】

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、“PE n ”を指定することができます。この場合、該当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) 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。

デフォルトでは、“main”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～と、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）。

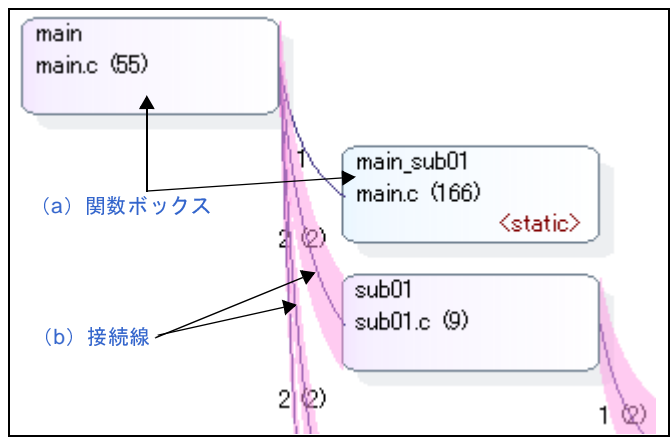
なお、[親関数名] コンボ・ボックスで任意の親関数を指定^注した場合は、その関数から呼び出されている関数のみを対象としたコール・グラフを表示します。また、同コンボ・ボックスが空欄の場合は、プログラム中に存在するすべての関数を対象としてコール・グラフを表示します。

注 【V850E2M】

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“PE n ”を指定した場合は、該当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。

コール・グラフの構成は次のとおりです。

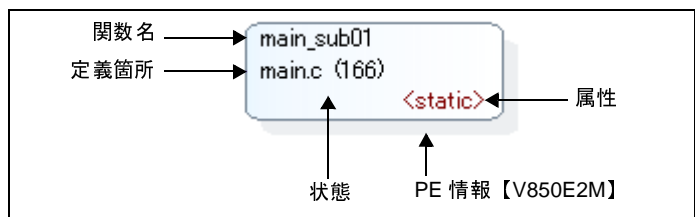
図 A—10 コール・グラフの構成



(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—11 関数ボックスの情報



関数名	対象関数名を示します。
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	対象関数がスタティック関数の場合のみ“<static>”を表示します。
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
PE 情報 【V850E2M】	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：PE1 - 緑色：PE2 - 青色：PE3 - 橙色：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（「2.6 定義箇所へジャンプする」参照）。

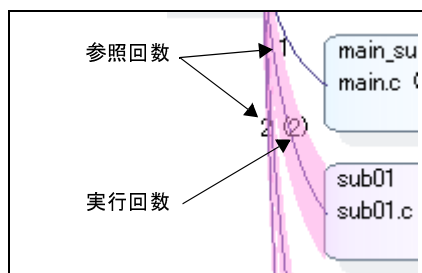
ただし、スクロール・ボタン ( ボタン) の設定により、コール・グラフのスクロールを許可している場合、この機能を使用することはできません。この場合は、[対象関数名] コンボ・ボックスにより対象関数を選択したのち、コンテキスト・メニューの [ソースヘジャンプ] を選択することにより同等の動作となります。

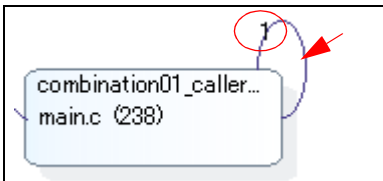
(b) 接続線

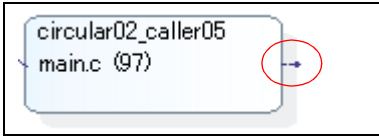
ある関数から別の関数を静的に呼び出している場合、双方の関数ボックス間に接続線を表示します。
 なお、接続線の表示は、静的な関数呼び出しのみを対象とします。動的な関数呼び出しに対応する接続線は表示しません。
 各接続線に表示する情報は次のとおりです。

備考 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。

図 A—12 接続線の情報




参照回数	静的に呼び出している回数を示します。
実行回数	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。 <div style="text-align: center;">  </div>

<p>循環呼び出し</p>	<p>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。</p> <div style="text-align: center;">  </div> <p>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。</p>
---------------	---

(c) ポップアップ表示

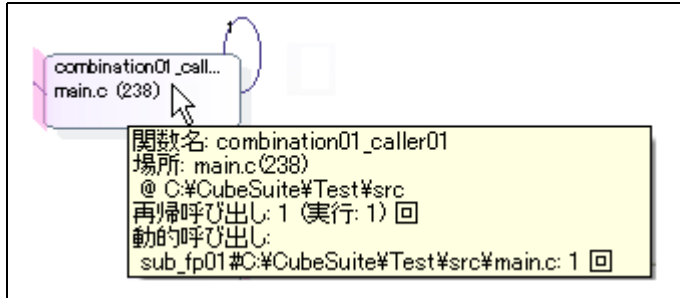
関数ボックスにマウス・カーソルを重ねることにより、該当関数の情報をポップアップ表示します。表示形式は次のとおりです。

注意 [スクロール・ボタン](#) ( ボタン) の設定により、コール・グラフのスクロールを許可している場合、この機能は無効となります。

<p>関数名: <i>関数名</i></p> <p>場所: <i>ファイル名 (行番号)</i> @ <i>ファイルの絶対パス</i></p> <p>再帰呼び出し: <i>参照回数 (実行: 実行回数)</i></p> <p>循環呼び出し</p> <p> <i>関数名: 参照回数 (実行: 実行回数)</i></p> <p> <i>関数名: 参照回数 (実行: 実行回数)</i></p> <p> ...</p> <p>動的呼び出し</p> <p> <i>関数名: 回数</i></p> <p> <i>関数名: 回数</i></p> <p> ...</p>
--

関数名	対象関数名を示します。
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	<p>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は表示します。</p> <p>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。</p> <p>関数名： 対象関数から動的呼び出ししている関数名</p> <p>回数： 実行した回数</p>
--------	--



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。

[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。
ただし、プログラム実行中はすべて無効となります。

	<p>最新情報を取得し、表示内容を更新します。</p> <p>ただし、プログラム実行中はこのボタンは無効となります。</p>
	<p>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。</p> <p>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)。</p>
	<p>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を検索します。</p>

[[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	<p>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 (「2.12 解析情報をファイルに保存する」参照)。</p> <p>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。</p>
名前を付けてコール・グラフ・データを保存...	<p>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 (「2.12 解析情報をファイルに保存する」参照)。</p>

[[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。

ソースヘジャンプ	[対象関数名] コンボ・ボックスで選択している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 (「2.6 定義箇所へジャンプする」参照)。
逆アセンブルヘジャンプ	[対象関数名] コンボ・ボックスで選択している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 (「2.6 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。
メモリヘジャンプ	[対象関数名] コンボ・ボックスで選択している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 (「2.6 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。
関数一覧ヘジャンプ	関数一覧 パネルをオープンし、[対象関数名] コンボ・ボックスで選択している関数を選択状態にします。

解析グラフ パネル

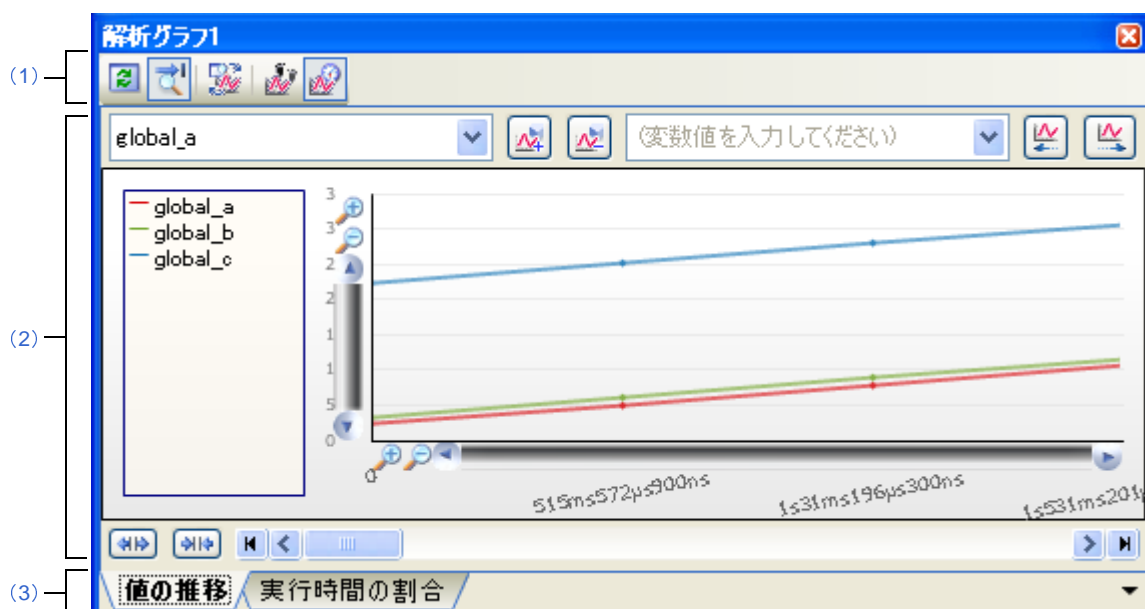
取得した関数情報／変数情報をグラフで表示します。

このパネルは、最大4個までオープンすることができ、各パネルは“解析グラフ1”、“解析グラフ2”、“解析グラフ3”、“解析グラフ4”の名称で識別されます（タイトルバーに表示）。

なお、グラフを表示するための操作手順は、「2.11 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 このパネルは、デバッグ・ツールと接続している場合のみ表示可能です（デバッグ・ツールと切断すると、このパネルはクローズします）。


図 A—13 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]
- [[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・ウィンドウのツールバー上の  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 1～4] の選択

[各エリアの説明]

(1) ツールバー

ツールバー上で使用できるボタンは、現在選択しているタブにより異なります。

詳細は、該当するタブの項を参照してください。

(2) 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を表示します。

(3) 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。

[[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（「 2.12 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（「 2.12 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／レジスタ等の値と時間の関係を折れ線グラフで表示します。

“解析グラフ 1”，“解析グラフ 2”，“解析グラフ 3”，“解析グラフ 4” の名称で識別される解析グラフパネルのこのタブでは、グラフ化する対象をそれぞれ個別に設定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、「2.11.1 変数値の推移をグラフ化する」を参照してください。

図 A—14 解析グラフパネル：[値の推移] タブ

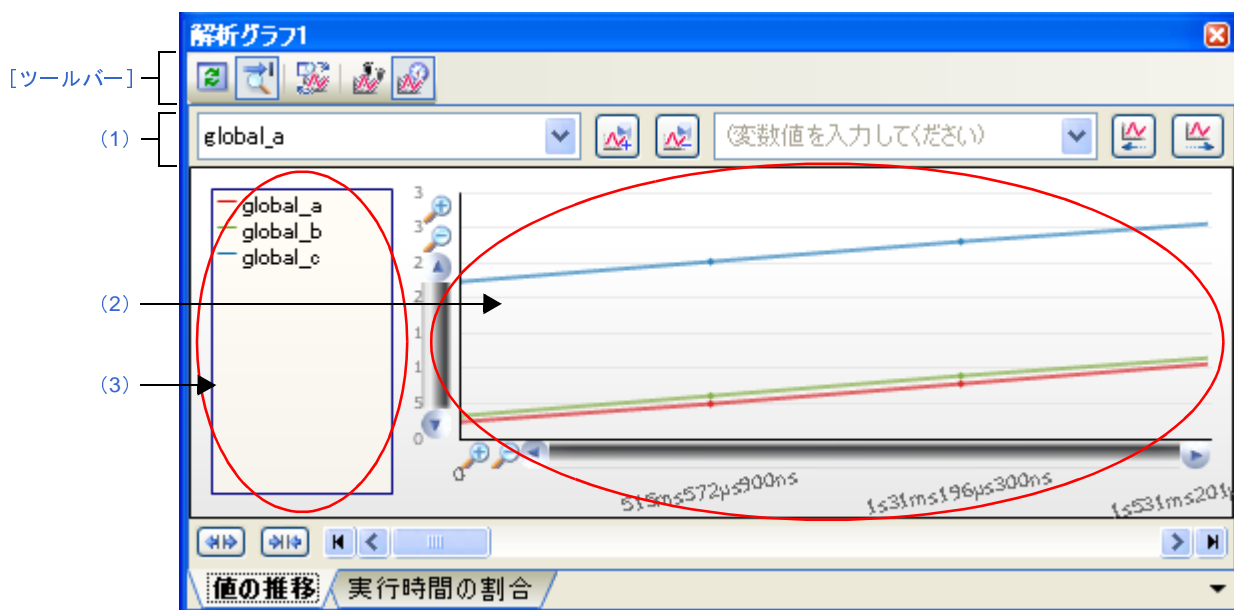
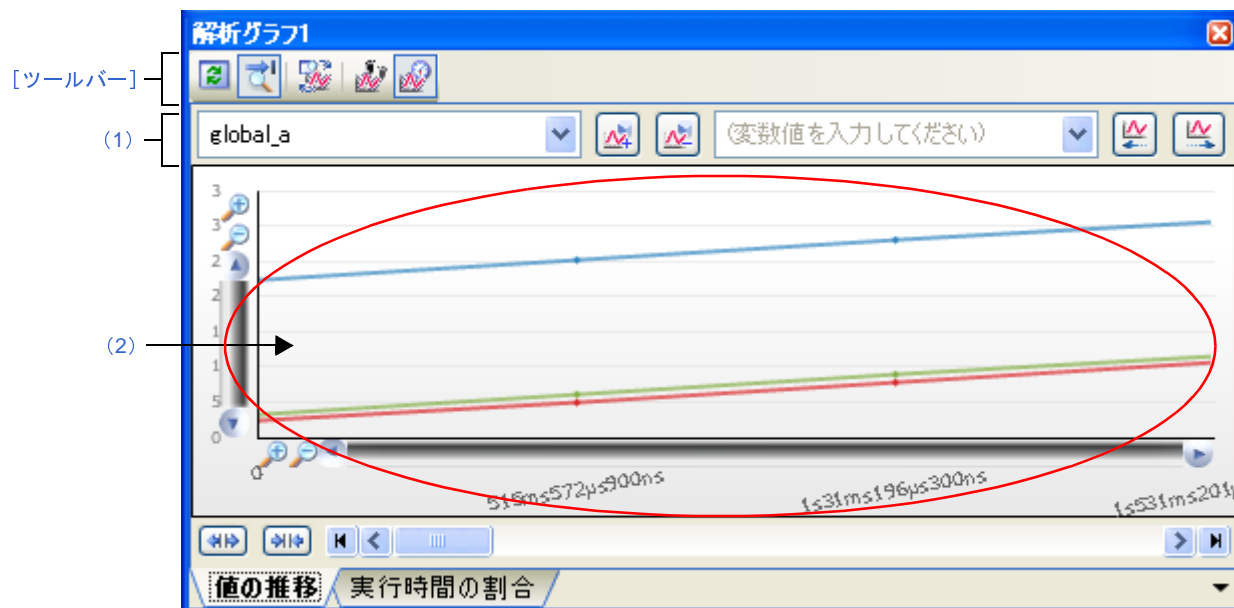



図 A—15 解析グラフパネル：[値の推移] タブ（凡例表示をしない場合）



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]

[オープン方法]

- メイン・ウィンドウのツールバー上の  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 1～4] → [値の推移] タブの選択

[各エリアの説明]

(1) 変数名／変数値設定エリア




このエリアは、次の機能を備えています。

ただし、プログラム実行中はすべて無効となります。

(a) グラフ化対象の登録／削除

このタブ上でグラフ化する対象の登録／削除を行います。



<p> (変数名を入力してください) ▼</p> <p>[変数名] コンボ・ボックス</p>	<ul style="list-style-type: none"> - グラフ化対象を新規に登録する場合 新規に登録するグラフ化対象名をキーボードより直接入力します（最大指定文字数：2046文字）注。 なお、グラフ化対象として登録可能な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 - CPU レジスタ - SFR 【78K0】 【78K0R】 - IOR 【V850】 - アドレス - 登録済みのグラフ化対象を削除する場合 削除するグラフ化対象名をドロップダウン・リストより選択します。 ドロップダウン・リストには、現在登録されているグラフ化対象名のすべてが表示されます。 なお、グラフ化対象の選択は、グラフのクリック、または凡例表示を行っている場合は対象の凡例をクリックすることでも可能です。
<p> 登録ボタン</p>	<p>[変数名] コンボ・ボックスで指定した対象名をグラフ化対象として登録します。</p>
<p> 削除ボタン</p>	<p>[変数名] コンボ・ボックスで指定した対象名をグラフ化対象から削除します。</p>

- 注 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。
 - 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）。

表 A—2 グラフ化対象の入力形式

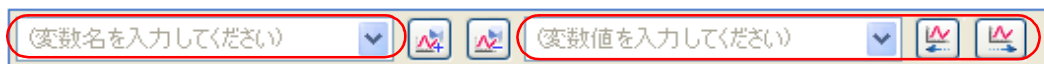
入力形式	取得値
C 言語変数名	C 言語の変数の値
変数式 [変数式]	配列の要素値
変数式 . メンバ名	構造体／共用体のメンバ値
変数式 -> メンバ名	ポインタの指し示す構造体／共用体のメンバ値
* 変数式	ポインタの変数の値
CPU レジスタ名	CPU レジスタの値
SFR レジスタ名 [78K0] [78K0R]	SFR レジスタの値
I/O レジスタ名 [V850]	I/O レジスタの値
ラベル, EQU シンボル, 即値アドレス	ラベルの値, EQU シンボルの値, 即値アドレスの値
ビット・シンボル	ビット・シンボルの値

注意 同一アドレスでサイズの異なる即値アドレスを登録することはできません。
 この場合、最初に登録された即値アドレスが優先されます。



備考 グラフ化対象のその他の登録方法については、「(1) グラフ化対象を登録する」を参照してください。

(b) 値の検索

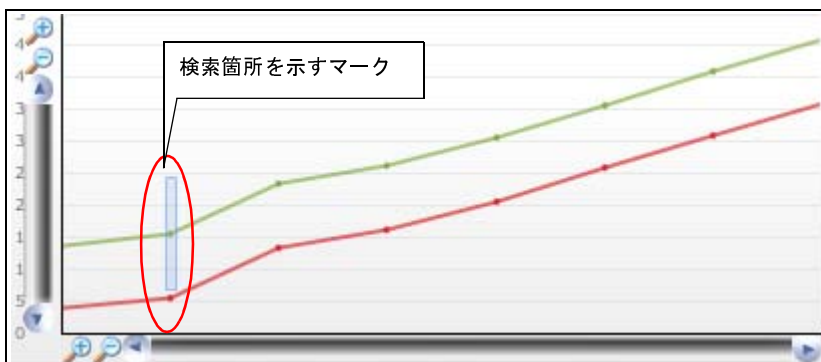
現在表示しているグラフにおいて、指定した値に遷移した箇所を検索します（「(4) グラフ・データを検証する」参照）。



<p>(変数名を入力してください) ▼</p> <p>[変数名] コンボ・ボックス</p>	<p>値の検索を行うグラフ化対象をドロップダウン・リストより選択します。 ドロップダウン・リストには、現在登録されているグラフ化対象名のすべてが表示されます。 なお、グラフ化対象の選択は、グラフのクリック、または凡例表示を行っている場合は対象の凡例をクリックすることでも可能です。</p>
---	--

[変数値を入力してください] [変数値] コンボ・ボックス	検索する値をキーボードより直接入力で指定します（最大履歴数：10 個）。 なお、値は 10 進数でのみ指定可能です。
 前へ移動ボタン	[変数名] コンボ・ボックスで選択しているグラフ化対象が、[変数値] コンボ・ボックスで指定した値に遷移した箇所を左方向に検索します。
 次へ移動ボタン	[変数名] コンボ・ボックスで選択しているグラフ化対象が、[変数値] コンボ・ボックスで指定した値に遷移した箇所を右方向に検索します。

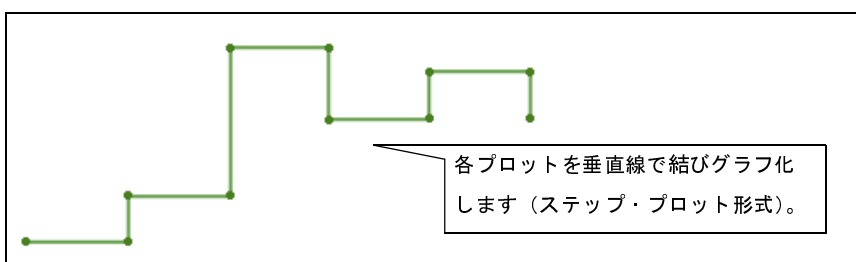
なお、検索した結果、合致した箇所がある場合は該当箇所に次のマークを表示します（合致する箇所がない場合は、出力パネルにメッセージを表示します）。



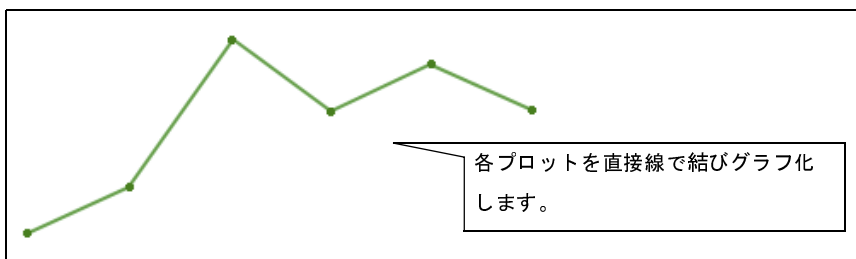
(2) グラフ表示エリア

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。
 なお、折れ線グラフの形式は、グラフ・データの取得方法により次のように異なります。

- トレース・データ解析方式を指定している場合



- リアルタイム・サンプリング方式を指定している場合




注意 トレース・データ解析方式を指定している場合、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、変数値の推移を表示できません。

備考 取得したグラフ・データがバッファ容量を越えた場合、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。

このエリアは、次の機能を備えています。

(a) 時間（X 軸）

経過時間を示します。取得したデータに応じて単位表示を行います。

 マークをクリックすることにより、グラフを X 軸方向に拡大表示します。

なお、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—8 グラフ・データの取得方法によるグラフ表示の相違](#)」参照）。

(b) 値（Y 軸）

登録したグラフ化対象の値を示します。

デフォルトでは、取得したデータ中の最大値／最小値が軸の上限／下限となりますが、表示する値の範囲を指定することができます（「[\(a\) 表示範囲の設定](#)」参照）。

また、異常値の表示を避けるため、値の有効範囲を指定することもできます（「[\(b\) 有効範囲の設定（異常値の切り捨て）](#)」参照）。

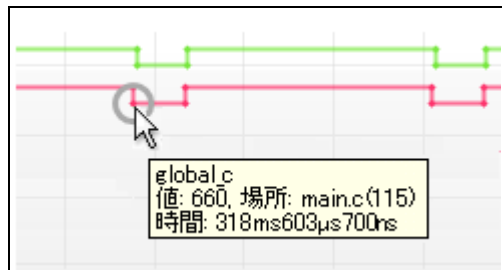
(c) ポップアップ表示

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。

なお、表示形式は、グラフ・データの取得方法により次のように異なります。

- [トレース・データ解析方式](#)を指定している場合

<p>変数名</p> <p>値：データ値，場所：ファイル名(行数)</p> <p>時間：データ値</p>
--



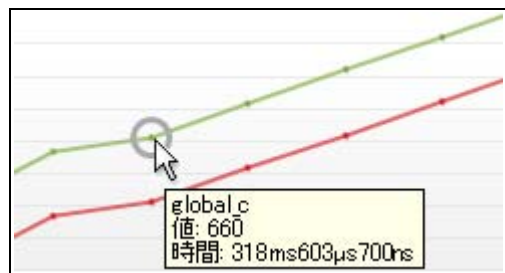
備考 [場所] 情報は、遷移箇所の情報が存在する場合のみ表示します（存在しない場合は“-”を表示します）。

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。

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ることで取得したグラフ・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。

- リアルタイム・サンプリング方式を指定している場合

変数名
 値: データ値
 時間: データ値



(3) 凡例表示エリア

登録したグラフ化対象の凡例を表示します（デフォルト）。






なお、凡例表示の必要がない場合、プロパティパネルの [設定] タブ上の [解析グラフ 1~4] カテゴリ内 [凡例を表示する] を [いいえ] に設定してください。

また、同カテゴリ内 [凡例の表示位置] プロパティにより、凡例の表示位置を指定することができます。

[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。

ただし、プログラム実行中はすべて無効となります。

	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] (デフォルト) 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)。
	ウォッチパネル上のウォッチ式をグラフ化の対象とします ((1) グラフ化対象を登録する 参照)。
	グラフ・データの取得方法を トレース・データ解析方式 とします。
	グラフ・データの取得方法を リアルタイム・サンプリング方式 とします (デフォルト)。

[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。

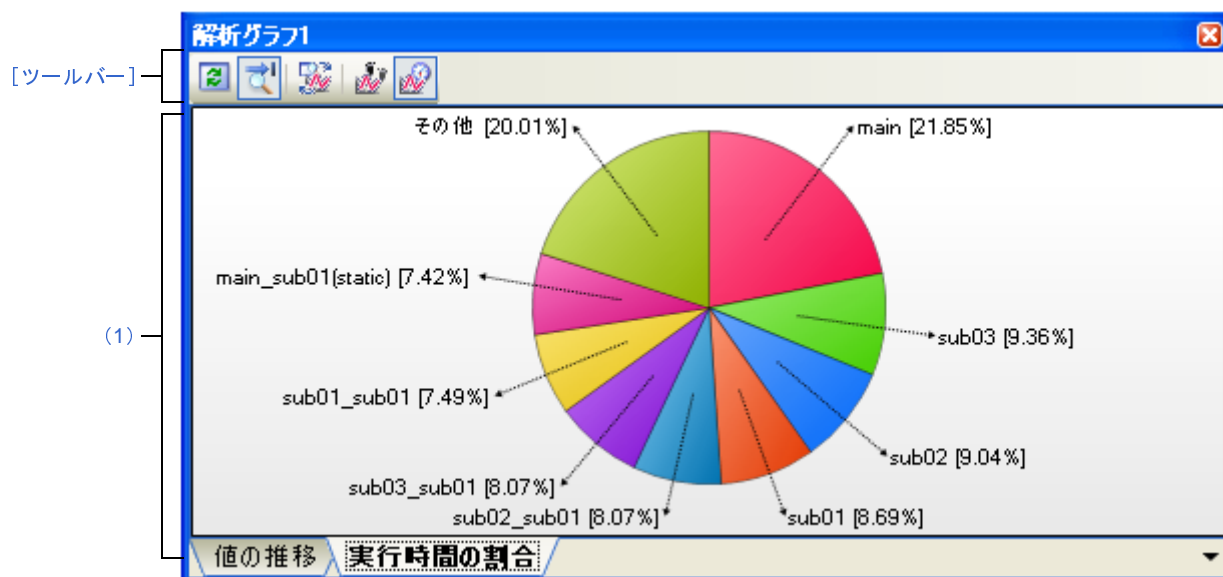
なお、このタブ上でグラフを表示するための操作手順は、「[2.11.2 関数の実行時間率をグラフ化する](#)」を参照してください。

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。

2. [IECUBE [78K0]]

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—16 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。

- [\[オープン方法\]](#)
- [\[各エリアの説明\]](#)
- [\[ツールバー\]](#)

[オープン方法]

- [メイン・ウィンドウ](#)のツールバー上の  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 → [\[プログラム解析\]](#) → [\[解析グラフ 1～4\]](#) → [\[実行時間の割合\]](#) 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。

(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。

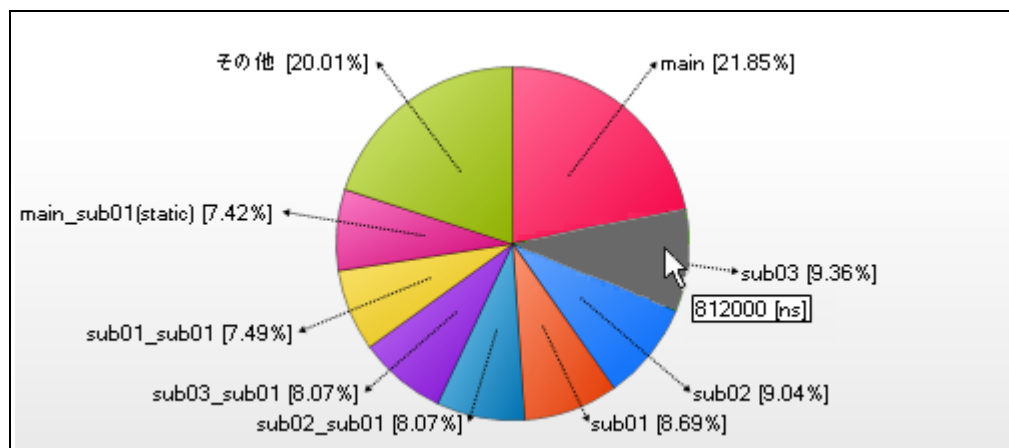
なお、スタティック関数の場合は、“()” 内にファイル名を併記します。

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、その関数の実行時間情報として、“実行時間 [単位]” をポップアップ表示します。

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。








備考 実行時間は、関数一覧パネルにおける [実行時間 [単位]] 項目の値と同値です。

[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。

ただし、プログラム実行中はすべて無効となります。

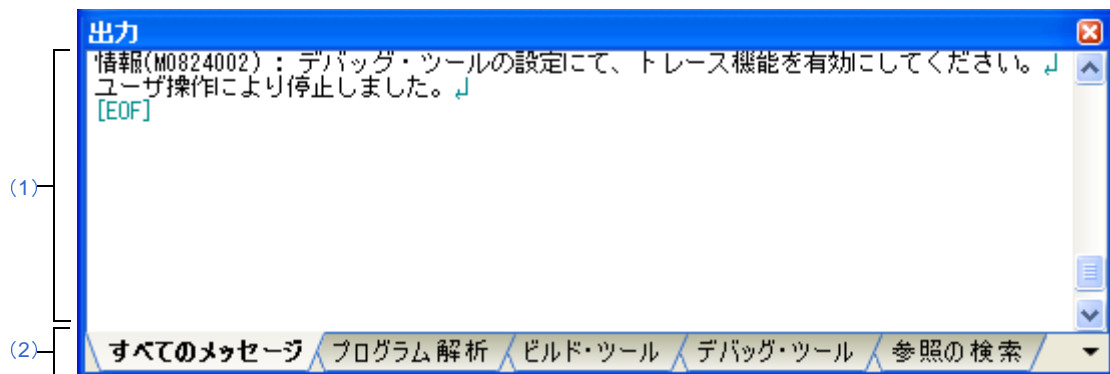
	最新情報を取得し、表示内容を更新します。
	プログラムの実行が停止するごとに最新情報を取得し、表示内容を更新します。 ただし、プロパティパネルの [設定] タブ上の [全般] カテゴリ内 [プログラム停止時に更新を行う] プロパティにおいて [個別に指定する] (デフォルト) 以外を指定している場合、このボタンは無効となります (プロパティパネルでの設定を反映した状態で固定されます)。
	このタブ上で、このボタンは機能しません。
	このタブ上で、このボタンは機能しません。
	このタブ上で、このボタンは機能しません。

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、「[2.9 参照箇所を一覧表示する](#)」を参照してください。

図 A—17 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]

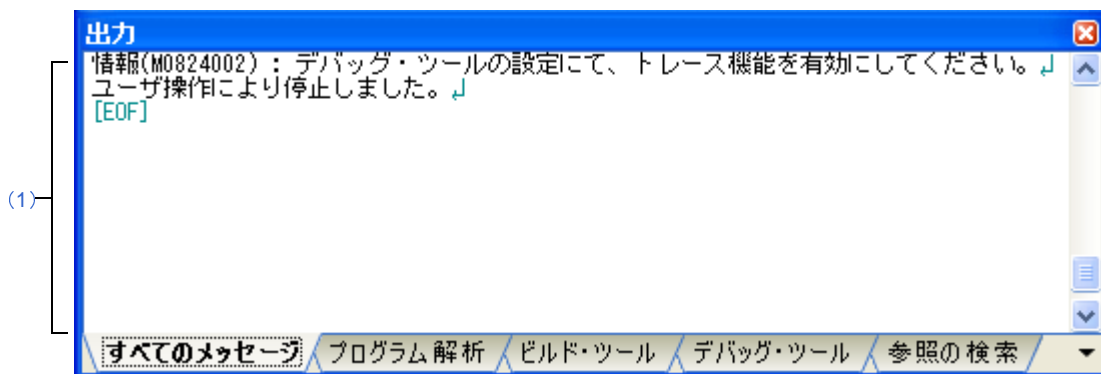
このパネル上において、マウスを右クリックすることにより表示されるコンテキスト・メニューの各項目、および機能は次のとおりです。

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無効となります。
メッセージに関するヘルプ	現在のキャレット位置のメッセージに関するオンライン・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—18 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] → [すべてのメッセージ] タブの選択

[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 背景色 白	何らかの情報を通知する際に表示されます。
	AaBbCc	文字色 青 背景色 標準色	
警告メッセージ	AaBbCc	文字色 青 背景色 標準色	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。

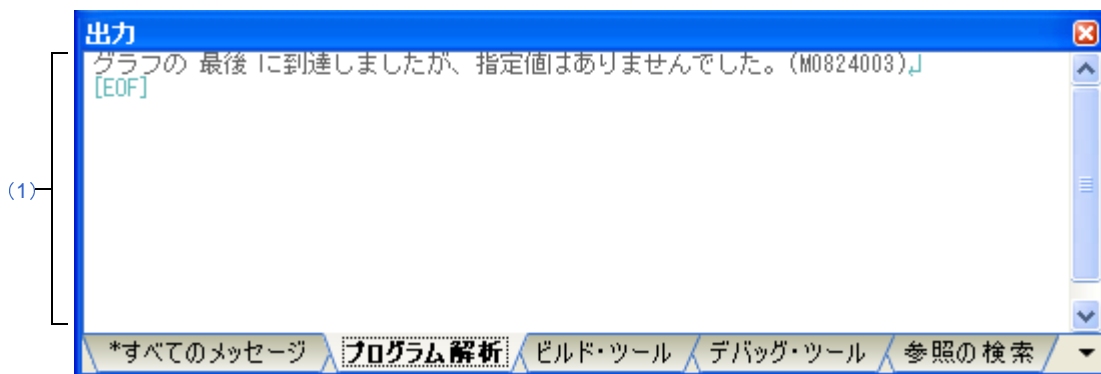
メッセージ種別	表示例（デフォルト）		説明
エラー・メッセージ	AaBbCc	文字色 赤	致命的なエラー，または操作ミスにより実行が不可能な場合に表示されます。
		背景色 薄グレー	

- 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caret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caret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オンライン・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—19 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] → [プログラム解析] タブの選択

[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caret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caret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オンライン・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。

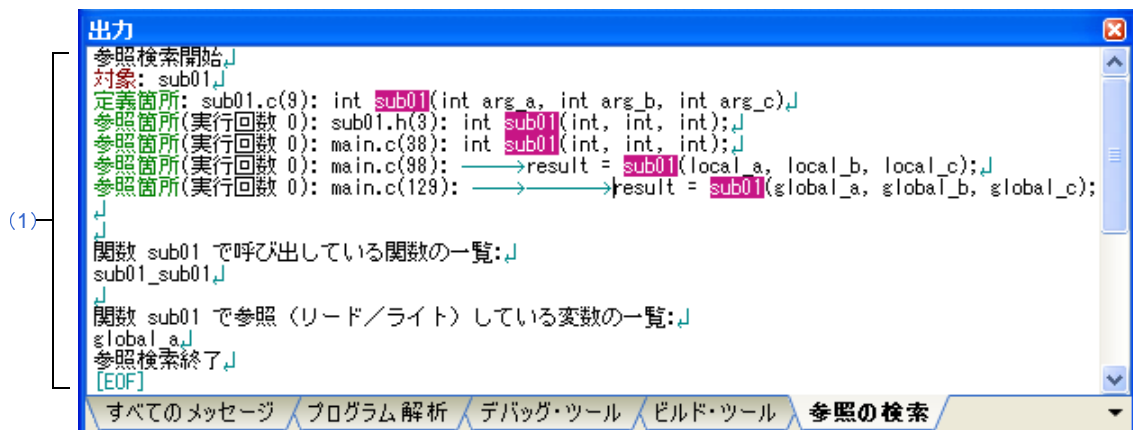
[参照の検索] タブ

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、「[2.9 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 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. **[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—20 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] → [参照の検索] タブの選択

[各エリアの説明]

(1) メッセージ・エリア

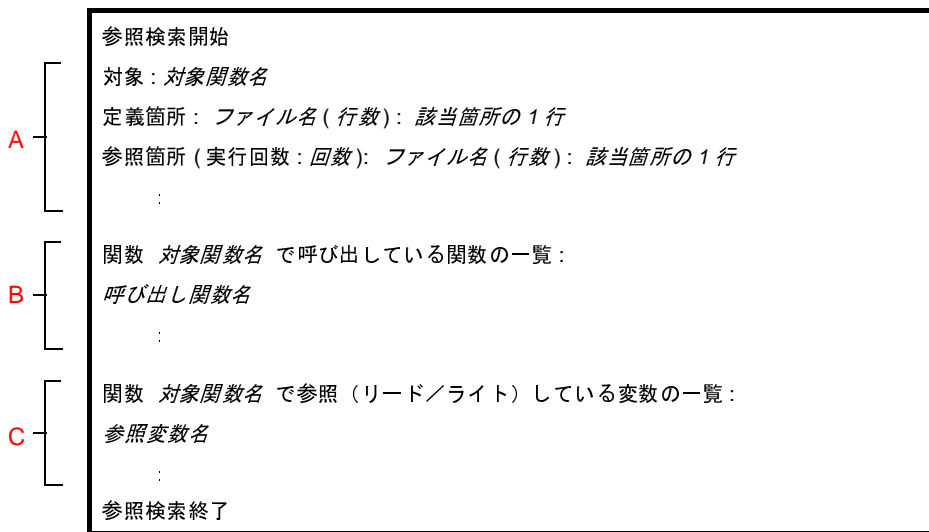
[関数一覧パネル](#)／[変数一覧パネル](#)において、関数／変数を参照している箇所を検索した際に、その結果として、次の参照箇所一覧を表示します（[2.9 参照箇所を一覧表示する](#)参照）。

- 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—21 関数の参照箇所一覧の出力フォーマット



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 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表示します。 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。 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 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。

図 A—22 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—23 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—24 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。

- 【CA78K0】【CA78K0R】

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。

- 【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。

- 【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。

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—25 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）

```

対象: global_a;
定義箇所: main.h(3): extern int global_a;
定義箇所: main.c(10): int global_a = 10;
参照箇所: main.c(95): →global_pointer = &global_a;
参照箇所: main.c(112): →global_a = 0;
参照箇所: main.c(125): →global_a++;
参照箇所: main.c(129): →result = sub01(global_a, global_b, global_c);
参照箇所: sub01.c(15): →result = tmp + global_a;

```

図 A—26 変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: global_a;
定義箇所: main.h(3): extern int global_a;
定義箇所: main.c(10): int global_a = 10;
参照箇所(リード/ライト回数 0): main.c(95): →global_pointer = &global_a;
参照箇所(リード/ライト回数 0): main.c(112): →global_a = 0;
参照箇所(リード/ライト回数 28): main.c(125): →global_a++;
参照箇所(リード/ライト回数 14): main.c(129): →result = sub01(global_a, global_b, global_c);
参照箇所(リード/ライト回数 14): sub01.c(15): →result = tmp + global_a;

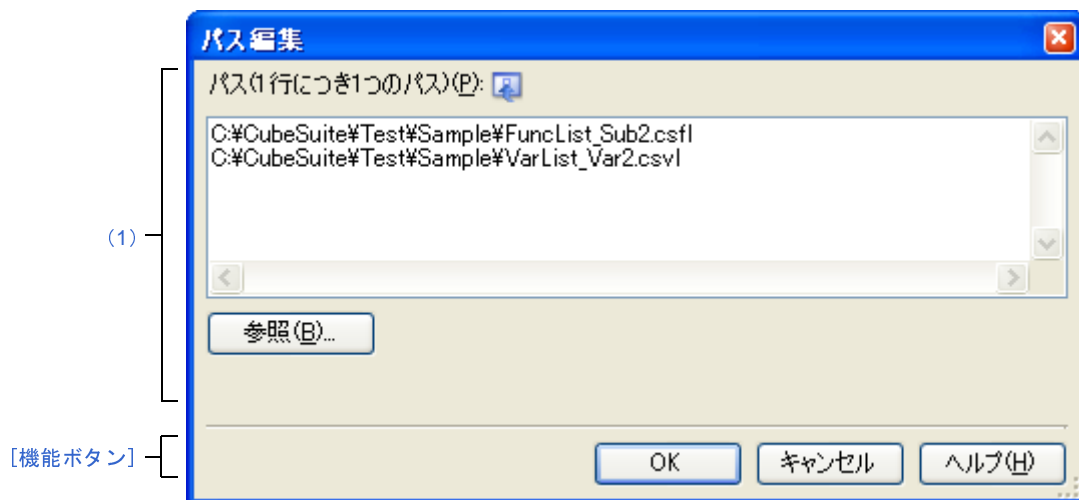
```

パス編集 ダイアログ

情報ファイル（関数一覧ファイル（*.csfl）／変数一覧ファイル（*.csvl））をインポートする際の、ファイルを指定します。

なお、インポート機能についての詳細は、「2.10 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。

図 A—27 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

(a) [パス (1 行につき 1 つのパス)] エリア

インクルードする関数一覧ファイル (*.csfl) / 変数一覧ファイル (*.csvl) 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、この際に、次の埋め込みマクロ名を使用することができます。

マクロ名	機能
%ProjectName%	プロジェクト名に置換します。
%CubeSuitePath%	CubeSuite のインストール・フォルダの絶対パスに置換します。

(b) ボタン

参照 ...	インクルード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。
--------	---

[機能ボタン]

ボタン	機能
OK	指定したファイルをインクルード・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	このダイアログのオンライン・ヘルプを表示します。

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—28 列の選択 ダイアログ（関数一覧パネル用）

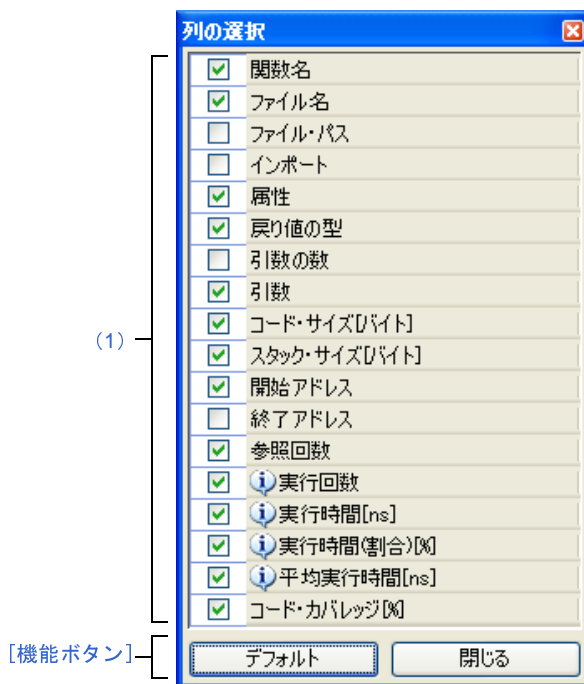
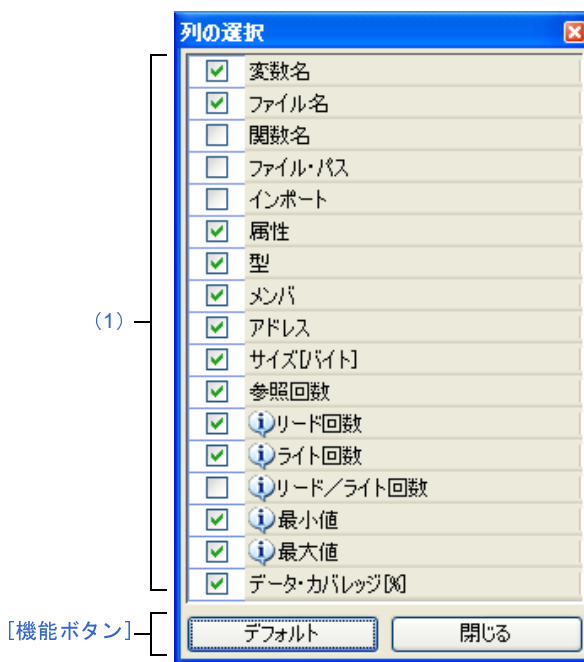




図 A—29 列の選択 ダイアログ（変数一覧パネル用）



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。

<input checked="" type="checkbox"/>	この項目をパネル上に表示します。
<input type="checkbox"/>	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—3 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
ファイル名	表示	表示
ファイル・パス	非表示	非表示
PE 情報 ^注 【V850E2M】	表示	表示
インポート	非表示	非表示
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]	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 【IECUBE】 【IECUBE2】 【シミュレータ】	非表示	表示
実行時間[単位] 【IECUBE 【78K0R】】 【IECUBE 【V850】】 【IECUBE2】 【シミュレータ】	非表示	表示
実行時間(割合)[%] 【IECUBE 【78K0R】】 【IECUBE 【V850】】 【IECUBE2】 【シミュレータ】	非表示	表示
平均実行時間[単位] 【IECUBE 【78K0R】】 【IECUBE 【V850】】 【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 【V850E2M】

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—4 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
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
PE 情報 ^注 【V850E2M】	非表示	表示
インポート	非表示	非表示
属性	表示	表示
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 【IECUBE】 【IECUBE2】 【シミュレータ】	非表示	表示
ライト回数 【IECUBE】 【IECUBE2】 【シミュレータ】	非表示	表示
リード/ライト回数 【IECUBE】 【IECUBE2】 【シミュレータ】	非表示	非表示
最小値 【IECUBE】 【IECUBE2】 【シミュレータ】	非表示	表示
最大値 【IECUBE】 【IECUBE2】 【シミュレータ】	非表示	表示
データ・カバレッジ [%] 【IECUBE 【78K0】】 【IECUBE 【78K0R】】 【IECUBE2】 【シミュレータ】	非表示	表示

注 【V850E2M】

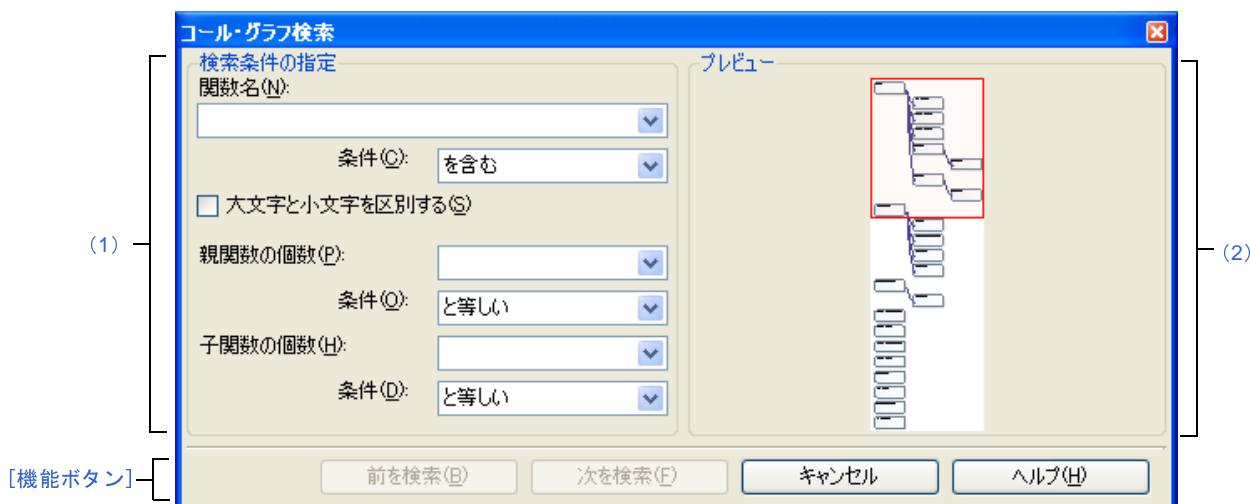
選択しているマイクロコントローラが、マルチコア対応版の場合のみ表示される項目です。

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（関数ボックス）のみが検索の対象となります。

図 A—30 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上の  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。

検索条件として、次の (b) [親関数の個数]、または (c) [子関数の個数] も指定した場合、すべての条件を満たす関数のみを検索します。

(a) [関数名]

検索対象となる関数名を指定します。

キーボードより文字列を直接入力するか、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を表示します（存在しない場合は“空欄”）。

備考 関数名の指定は、次の操作によっても指定することができます。

- 関数一覧パネルの任意の行をこのダイアログ上にドラッグ・アンド・ドロップ
- 任意の文字列をこのコンボ・ボックスにドラッグ・アンド・ドロップ

- [条件]

[関数名] 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を検索します。
で始まる	指定した文字列で始まる関数名を検索します。
で終わる	指定した文字列で終わる関数名を検索します。
を含む	指定した文字列を含む関数名を検索します（デフォルト）。

- [大文字と小文字を区別する]

<input checked="" type="checkbox"/>	大文字と小文字を区別して検索します。
<input type="checkbox"/>	大文字と小文字を区別せず検索します（デフォルト）。

(b) [親関数の個数]

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。

(c) [子関数の個数]

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の検索を行い、該当関数ボックスを強調表示します。 なお、 コール・グラフパネル 上の [対象関数名] コンボ・ボックス において、関数を指定している場合は、対象関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の検索を行い、該当関数ボックスを強調表示します。 なお、 コール・グラフパネル 上の [対象関数名] コンボ・ボックス において、関数を指定している場合は、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	このダイアログのオンライン・ヘルプを表示します。

フィルタ設定 ダイアログ

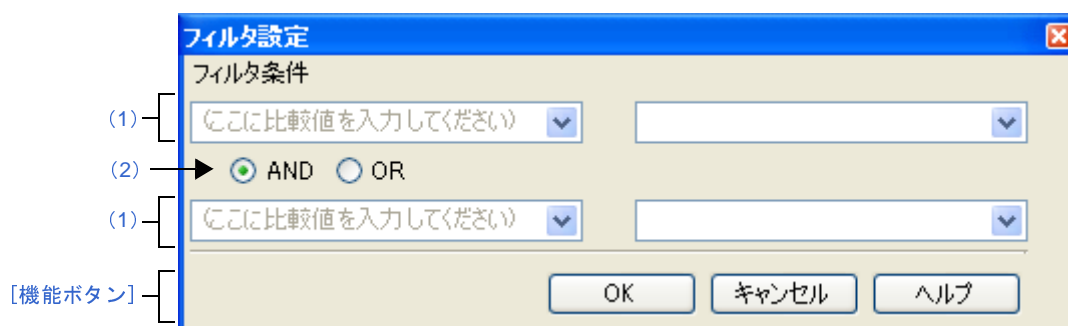
関数一覧パネル／変数一覧パネル上の解析情報を表示する際のフィルタ条件を設定します。

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、「[2.5.5 解析情報をフィルタ表示する](#)」を参照してください。

図 A—31 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	このダイアログのオンライン・ヘルプを表示します。

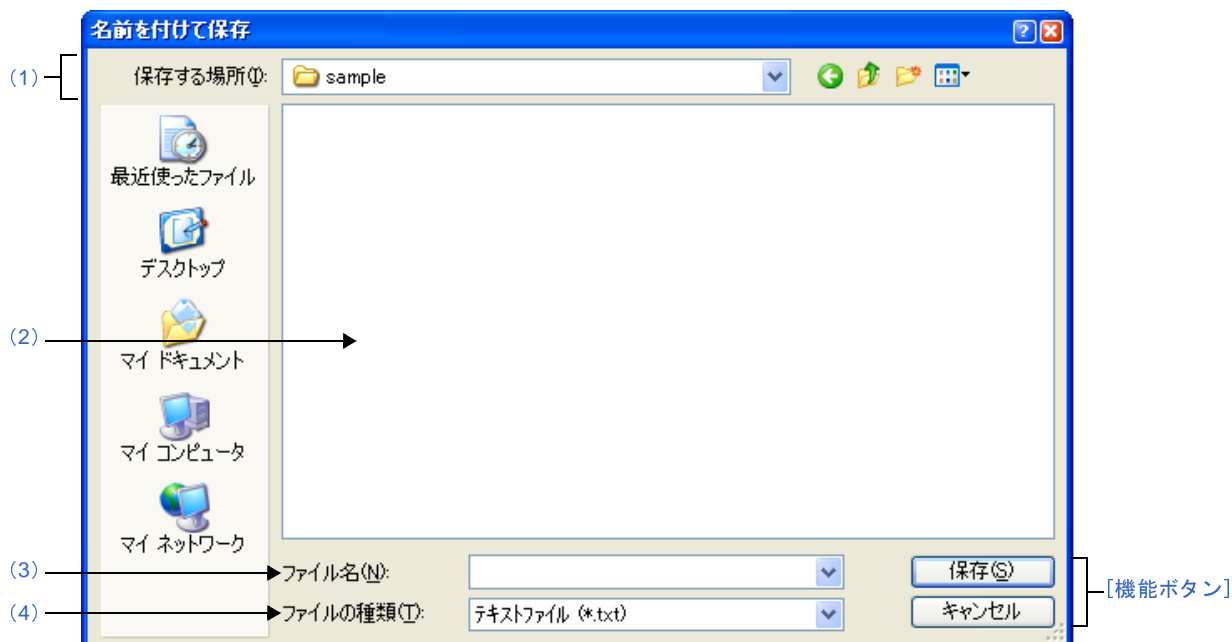
名前を付けて保存 ダイアログ

関数一覧パネル／変数一覧パネル／コール・グラフパネル／解析グラフパネルの内容を名前を付けてファイルに保存します。

ただし、保存する対象は次に従います。

- 関数一覧パネル／変数一覧パネルの場合： 現在表示している項目（列）／解析情報（行）の内容のみ
- コール・グラフパネルの場合： このダイアログで指定
- 解析グラフパネルの場合： 現在表示しているタブの内容のみ

図 A—32 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
- 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
- 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

[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。

(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 Excel 97 - Excel 2003、および 5.0/95 ブック (*.xls) 形式
CubeSuite 関数一覧ファイル (*.csfl)	関数情報をインポートするためのファイル形式 (「 2.10 情報ファイルをインポート/エクスポートする 」参照)

(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 Excel 97 - Excel 2003、および 5.0/95 ブック (*.xls) 形式
CubeSuite 変数一覧ファイル (*.csvl)	変数情報をインポートするためのファイル形式 (「 2.10 情報ファイルをインポート/エクスポートする 」参照)

(c) 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。
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)。

(d) 解析グラフ 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 Excel 97 - Excel 2003, および 5.0/95 ブック (*.xls) 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)

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。

- [値の推移] タブ

トレース・データ解析方式により取得したグラフの場合					
保存項目	時間	値 (変数名)	値 (変数名)	...	場所
備考	—	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)			情報が存在しない場合は空欄

リアルタイム・サンプリング方式により取得したグラフの場合					
保存項目	時間	値 (変数名)	値 (変数名)	値 (変数名)	...
備考	—	各変数の値を常に出力 (ただし、値が不明の場合は空欄)			

- [実行時間の割合] タブ

保存項目	関数名	割合 [%]	時間
------	-----	--------	----

備考 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。

[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

【R】

RRM 機能 … 10

【あ行】

アクティブ・プロジェクト … 39

異常値の切り捨て … 48

インポート … 39

ウインドウ・リファレンス … 54

ウォッチ式 … 37

エクスポート … 39

円グラフ … 50, 105

親関数 … 90

折れ線グラフ … 43, 99

【か行】

解析グラフ … 43

解析グラフ パネル … 97

 [値の推移] タブ … 99

 [実行時間の割合] タブ … 105

書き込みブレーク … 36

カバレッジ機能 … 10

カレント行 … 77, 87

カレント行マーク … 77, 87

関数一覧 パネル … 70

関数一覧ファイル … 39, 119

関数の実行時間率 … 50

共用体 … 83

グラフ化対象 … 44

グラフ・データの取得方法 … 46

クロス・リファレンス情報 … 9, 64

降順表示 … 31

構造体 … 83

コード・カバレッジ率 … 75

コール・グラフ検索 ダイアログ … 126

コール・グラフ パネル … 89

固定表示 … 30

固定表示アイコン … 30

【さ行】

再帰呼び出し … 93

参照箇所一覧 … 38, 114

実行時間の割合 … 50

出力パネル … 108

 [参照の検索] タブ … 114

 [すべてのメッセージ] タブ … 110

 [プログラム解析] タブ … 112

循環呼び出し … 94

昇順表示 … 31

情報ファイル … 39

静的解析情報 … 9

ソート表示 … 31

【た行】

ツールバー … 56

定義箇所 … 35

データ・カバレッジ率 … 84

動的解析情報 … 10

特長 … 14

トレース機能 … 10, 46

トレース・データ解析方式 … 46

【な行】

名前を付けて保存 ダイアログ … 132

【は行】

パス編集 ダイアログ … 119

表示範囲 … 47

ファイルに保存 … 132

フィルタ設定 ダイアログ … 129

フィルタ表示 … 31

フィルタ・アイコン … 31

ブレークポイント … 36

ブレーク・イベント … 36

プロジェクト・ツリー パネル … 58

プロパティ パネル … 60

 [設定] タブ … 63

変数一覧 パネル … 80
変数一覧ファイル … 39, 119
ポップアップ表示 … 49

【ま行】

メイン・ウインドウ … 55
メニューバー … 56

【や行】

有効範囲 … 48
読み書きブレーク … 36
読み込みブレーク … 36

【ら行】

ランタイム・ライブラリ … 75
リアルタイム RAM モニタ機能 … 46
リアルタイム・サンプリング方式 … 46
列の選択 ダイアログ … 121

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2010.09.01	—	初版発行

CubeSuite Ver.1.40 ユーザーズマニュアル
解析編

発行年月日 2010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/inquiry>

CubeSuite Ver.1.40