

# CubeSuite+ V1.01.00

統合開発環境

ユーザーズマニュアル 解析編

対象デバイス 78K0マイクロコントローラ RL78ファミリ 78K0Rマイクロコントローラ V850マイクロコントローラ RXファミリ

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

www.renesas.com

Rev.1.00 2011.10

# ご注意書き

- 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ものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
- 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、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 て、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がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。
- 注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいい ます。

# このマニュアルの使い方

このマニュアルは, RX ファミリ, V850 マイクロコントローラ, RL78 ファミリ, 78K0R マイクロコントローラ, 78K0 マイクロコントローラ用アプリケーション・システムを開発する際の統合開発環境である CubeSuite+ について説 明します。

CubeSuite+ は、RX ファミリ、V850 マイクロコントローラ、RL78 ファミリ、78K0R マイクロコントローラ、78K0 マイクロコントローラの統合開発環境(ソフトウエア開発における、設計、実装、デバッグなどの各開発フェーズに必 要なツールをプラットフォームである IDE に統合)です。統合することで、さまざまなツールを使い分ける必要がな く、本製品のみを使用して開発のすべてを行うことができます。

- 対象者 このマニュアルは、CubeSuite+を使用してアプリケーション・システムを開発するユーザを対象として います。
- 目 的 このマニュアルは、CubeSuite+の持つソフトウ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
第3章 注意事項
付録A ウインドウ・リファレンス
付録B 索 引

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

凡	例	データ表記の重み	:	左が上位桁,右が下位桁
		アクティブ・ロウの表記	:	XXX (端子,信号名称に上線)
		注	:	本文中につけた注の説明
		注意	:	気をつけて読んでいただきたい内容
		備考	:	本文中の補足説明
		数の表記	:	2 進数 … XXXX, または XXXXB
				10 進数 XXXX
				16 進数 0xXXXX

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

資	料名	資料番号		
		和文	英文	
CubeSuite+ 統合開発環境	起動編	R20UT0727J	R20UT0727E	
ユーザーズ・マニュアル	V850 設計編	R20UT0549J	R20UT0549E	
	RL78 設計編	R20UT0728J	R20UT0728E	
	78K0R 設計編	R20UT0547J	R20UT0547E	
	78K0 設計編	R20UT0546J	R20UT0546E	
	RX コーディング編	R20UT0767J	R20UT0767E	
	V850 コーディング編	R20UT0553J	R20UT0553E	
	コーディング編(CX コンパイラ)	R20UT0825J	R20UT0825E	
	RL78, 78K0R コーディング編	R20UT0729J	R20UT0729E	
	78K0 コーディング編	R20UT0782J	R20UT0782E	
	RX ビルド編	R20UT0768J	R20UT0768E	
	V850 ビルド編	R20UT0557J	R20UT0557E	
	ビルド編(CX コンパイラ)	R20UT0826J	R20UT0826E	
	RL78, 78K0R ビルド編	R20UT0730J	R20UT0730E	
	78K0 ビルド編	R20UT0783J	R20UT0783E	
	RX デバッグ編	R20UT0769J	R20UT0769E	
	V850 デバッグ編	R20UT0734J	R20UT0734E	
	RL78 デバッグ編	R20UT0733J	R20UT0733E	
		R20UT0732J	R20UT0732E	
		R20UT0731J	R20UT0731E	
	解析編	このマニュアル	R20UT0735E	
	メッセージ編	R20UT0736J	R20UT0736E	

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

この資料に記載されている会社名、製品名などは、各社の商標または登録商標です。

目 次

## 第1章概 説 … 7

1.1 概 要 … 7
 1.1.1 解析対象 … 7
 1.1.2 解析情報の種別 … 7
 1.1.3 デバッグ・ツールの設定 … 8
 1.2 特 長 … 13

# 第2章 機 能 … 17

- 2.1 概 要 … 17 2.2 関数情報を表示する … 22 2.3 変数情報を表示する … 23 2.4 関数間の呼び出し関係(コール・グラフ)を表示する … 24 2.5 クラス/関数/変数の情報を表示する … 29 2.6 表示方法をカスタマイズする … 31 2.6.1 表示項目を設定する … 31 2.6.2 表示項目を並び替える … 33 2.6.3 特定項目を固定表示に設定する … 34 2.6.4 解析情報をソート表示する … 35 2.6.5 解析情報をフィルタ表示する … 35 2.7 定義箇所へジャンプする … 39 2.8 宣言箇所へジャンプする … 41 2.9 ブレーク・イベントを設定する … 42 2.9.1 関数にブレークポイントを設定する … 42 2.9.2 変数にブレーク・イベントを設定する … 42 2.10 ウォッチ式に登録する … 44 2.11 参照箇所を一覧表示する … 45 2.12 情報ファイルをインポート/エクスポートする … 46 2.13 解析情報をグラフ化して表示する … 51 2.13.1 変数値の推移をグラフ化する … 51 2.13.2 関数の実行時間率をグラフ化する … 58
  - 2.14 解析情報をファイルに保存する … 60

# 第3章 注意事項 … 62

- 3.1 アクティブ・プロジェクトの変更について … 62
- 3.2 カバレッジ結果について … 62
- 3.3 トレース・データ解析方式について … 62
- 3.4 プログラム実行中にパネルをオープンした場合について … 62

- 付録 A ウインドウ・リファレンス … 64
  - A.1 説 明 … 64
- 付録 B 索 引 … 166

# 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、RX ファミリ、V850 マイクロコントローラ、 RL78 ファミリ、78K0R マイクロコントローラ、78K0 マイクロコントローラ用に開発されたプログラムの動作解析を 支援します。

## 1.1.1 解析対象

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

したがって、アセンブル・ソース・ファイルは解析対象外となります。

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

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

- 2. 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。
- 【CA850】
   K&R 形式の記述には対応していません。
- 備考 情報ファイル(関数一覧ファイル(\*.mtfl)/変数一覧ファイル(\*.mtvl))をインポートすることにより, アクティブ・プロジェクト以外の C/C++ ソース・ファイル/ヘッダ・ファイルにおける関数/変数の情報 を強制的に表示することも可能です(「2.12 情報ファイルをインポート/エクスポートする」参照)。

#### 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。

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

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

RENESAS

- プロパティ パネル→ [設定] タブ→ [全般] カテゴリ→ [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

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

#### (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「RI シリーズ」が提供している実行解析ツール(「AZ シリーズ(AZ850V4/ AZ78K0R)」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、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]タブで行います。

- (1) トレース機能
- (2) カバレッジ機能
- (3) RRM 機能
- 備考 各機能についての詳細は、「CubeSuite+ デバッグ編」を参照してください。



(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	V850E1	【関数情報】(関数一覧 パネル/コール・グラフ パネル)
	V850ES	[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 <sup>注1</sup>
		[トレース] カテゴリ
		[トレース・データの選択]プロパティ→次のいずれかの指定
		- [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]
		【変数情報】(変数一覧 パネル/コール・グラフ パネル/解析グラフ パネル)
		[トレース] カテゴリ
		[トレース・データの用途]プロパティ→[トレース]
		[トレース・データの選択]プロパティ→次のいずれかの指定
		- [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]
	RL78	なし(常に有効) <b>注 2</b>
	78K0R	
	78K0	
IECUBE2	V850E2	【関数情報】(関数一覧 パネル/コール・グラフ パネル)
	シングルコア	[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 <sup>注3</sup>
		[トレース] カテゴリ
		[トレース・データの選択]プロパティ→次のいずれかの指定
		- [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]
		【変数情報】(変数一覧 パネル/コール・グラフ パネル/解析グラフ パネル)
		[トレース] カテゴリ
		[トレース・データの用途]プロパティ→[トレース]
		[トレース・データの選択]プロパティ→次のいずれかの指定
		- [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]
	V850E2	サポートしていません。
	マルチコア	
E1/E20	RX	解析ツールが必要とするトレース機能はサポートしていません。

RENESAS

デバッグ	・ツール	デバッグ・ツールの設定
MINICUBE	V850	サポートしていません。
MINICUBE2	RL78	
E1/E20	78K0R	
EZ Emulator	78K0	
シミュレータ	RX	[トレース] カテゴリ
	<sub>V850</sub> 注4	[トレース機能を使用する]プロパティ→[はい]
	RL78	
	78K0R	
	78K0	

注 1. 【IECUBE【V850E1】【V850ES】】

トレース機能/カバレッジ機能/ RRM 機能は、一部が排他使用の機能であるため、[トレース] カテ ゴリ内 [トレース・データの用途] プロパティにおいて優先的に使用する機能を指定する必要があり ますが、この際に、同プロパティに [トレース] 以外の [カバレッジ] / [RRM] を指定している場 合でも、解析ツールが動的解析情報を取得するために必要となるトレース機能は有効となります。

2. [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。

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。

**4.** [V850]

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,シミュレータはサポートしていません。

(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	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。
RL78 78K0R		なし(常に有効)
	78K0	



デバッグ	・ツール	デバッグ・ツールの設定
IECUBE2	V850E2	[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。
	V850E2	サポートしていません。
	マルチコア	
MINICUBE/MIN	ICUBE2	サポートしていません。
E1/E20		
EZ Emulator		
シミュレータ	RX	
<sub>V850</sub> 注		[カバレッジ機能を使用する] プロパティ→ [はい]
RL78		
	78K0R	
	78K0	

# 注 【V850】

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、シミュレータはサポートしていません。

## (3)RRM 機能

デバッグ	・ツール	デバッグ・ツールの設定
IECUBE	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→ [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 <i>→ 有効な数値</i>
		[リアルタイム表示更新を自動設定する]プロパティ→[はい]
	RL78	[実行中のメモリ・アクセス]カテゴリ
	78K0R	[実行を一瞬停止してアクセスする]プロパティ→[はい]
	78K0	[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 <i>→ 有効な数値</i>
IECUBE2	V850E2	- [実行中のメモリ・アクセス]カテゴリ
	シングルコア	[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 <i>→ 有効な数値</i>
	V850E2	サポートしていません。
	マルチコア	
MINICUBE	V850E1	
	V850ES	[実行を一瞬停止してアクセスする]プロパティ→[はい]
	V850E2	[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 <i>→ 有効な数値</i>



デバッグ	・ツール	デバッグ・ツールの設定
E1/E20	RX	[トレース] カテゴリ
		[トレース機能の用途]プロパティ→[リアルタイム RAM モニタ]
		[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 <i>→ 有効な数値</i>
		[リアルタイム表示更新を自動設定する]プロパティ→[はい]
MINICUBE2	V850	[実行中のメモリ・アクセス] カテゴリ
E1/E20	RL78	[実行を一瞬停止してアクセスする]プロパティ→[はい]
EZ Emulator	78K0R	[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 <i>→ 有効な数値</i>
	78K0	
		[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 <i>→ 有効な数値</i>
		[リアルタイム表示更新を自動設定する]プロパティ→[はい]
シミュレータ	RX	[実行中のメモリ・アクセス] カテゴリ
	<sub>V850</sub> 注	[実行中に表示更新を行う]プロパティ→[はい]
	RL78	[表示更新間隔 [ms]]プロパティ <i>→ 有効な数値</i>
	78K0R	
	78K0	

注 【V850】

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、シミュレータはサポートしていません。



# 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。

閧	一覧						
2	T 🔫 🛣 🕵 🔊	時間単位(山)・					
P	関数名	74 771	ル名 ▽口	属性 ▽□	戻り値の型 ▽₽	引数 ▽中	^
•	∎o AD_Init	CG_ad.c		-	void	void	
	🖕 AD_Read	CG_ad.c		-	unsigned short	unsigned…	
	💊 AD_Start	CG_ad.c		-	void	void	
	💊 AD_Stop	CG_ad.c		-	void	void	
	🚽 MD_INTAD	CG_ad_u	ser.c	-	void	void	
	💊 TMP0_Start	CG_time	r.c	-	void	void	
	🚽 TMP1_Start	CG_time	r.c	-	void	void	
	💊 ad_receive	(定義箇)	所なし)	_	-	-	
	🖕 func1	CG_mair	1.C		void	void	
	🖕 func1a	CG_mair	1.C	-	void	void	
	🖕 func2	CG_mair	1.C	-	void	void	V
<						>	

# - 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	数一覧									X	3
2	1 式 🧶 🕵 🕻	88									
P	変数名	74	ファイル名	7₽	属性	74	型	74	メンバ マセ	アドレス マ≠	2
•	🅤 g_ad_data	C	G_main.c		-		unsigned	short	-	0x03ffb116	Π
	👽 g_ad_finish	C	G_main.c		-		unsigned	char	-	0×03ffb114	
	👽 g_count_10ms	C	G_main.c		-		unsigned	int	-	0x03ffb110	ſ
	👽 g_count_1 ms	C	G_main.c		-		unsigned	int	-	0x03ffb10c	
	👽 g_flag_detect	C	G_main.c		_		unsigned	short	-	0x03ffb118	l.
	😜 _S_romp	C	G_systeminit	.C	_				-		•
	合計値	C	G_main.c		-		-		-	<u> </u>	-
	合計値	C	G_systeminit	.C	_		- <u></u>		_		-
<										>	·



# - コール・グラフの表示

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

コール・グラフ 図 武 温	
	Main 💽 対象関数名 💽 🚱
main main.c (55)	
main_sub01 main.c (166	
sub01	<static></static>
2(2)	
	sub02.c (14)
2 (2) sub02 sub01.c (23	>
	1 (2) sub02_sub01

# - クラス情報/関数情報/変数情報の表示

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





- デバッグ・ツールとの連携

解析ツールの各パネルから、デバッグ・ツールに対して次の操作を行うことができます。 - 指定した関数/変数が定義されている箇所へのジャンプ:

- 関数一覧 パネル/変数一覧 パネル/コール・グラフ パネル/クラス/メンバ パネル
- 指定した関数/変数に対するブレーク・イベントの設定:
- 関数一覧 パネル/変数一覧 パネル

- 指定した変数をウォッチ式に登録

変数一覧 パネル

#### - 関数の参照箇所/変数の参照箇所の一覧表示

指定した関数/変数を参照している箇所を検索し、参照箇所の一覧を表示します。

出力	
参照検索開始」	^
対象: sub01」 定義箇所: sub01.c(9): int <u>sub01</u> (int arg_a, int arg_b, int arg_c)↓ 参照箇所(実行回数 0): sub01.h(3): int <u>sub01</u> (int, int, int);↓ 参照箇所(実行回数 0): main.c(38): int <u>sub01</u> (int, int, int);↓ 参照箇所(実行回数 0): main.c(38): →→result = <u>sub01</u> (local_a, local_b, local_c);↓ 参照箇所(実行回数 0): main.c(129): →→result = <u>sub01</u> (slobal_a, global_b, global_c); ↓	
」 関数 sub01 で呼び出している関数の一覧:」	-
sub01_sub01_J	
」 関数 sub01 で参照(リード/ライト)している変数の一覧:」	
global al	
[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 /	-

- 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。

解析グラフ1			8
2 式 🐝 🌌 🖉			
global_a	💌 🛃 🛃	(変数値を入力してください)	
	515m557245300n5	1531m5196µ5300n5	L-53(ms20)
<b>H</b> 416 418	3)		N
値の推移、実行時間の	割合 /		•



- 関数の実行時間率を円グラフで表示します。





# 第2章機能

この章では、解析ツールの主な機能を操作手順とともに説明します。

# 2.1 概 要

解析ツールが取得した解析情報は、次のパネルで表示されます。

表 2—1	解析情報を表示するパネル
-------	--------------

パネル		説明
関数一覧 パネル	関数情報を表示	
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
クラス/メンバ パネル	クラス情報【CC-RX】 <sup>注</sup> /関数	情報/変数情報をツリー形式で表示
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)

注【CC-RX】

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

また、各パネルでは、次の種類の関数/変数を解析情報の対象とします。

表 2—2	解析情報の対象となる関数/	′変数の種類
-------	---------------	--------

種類	関数一覧	変数一覧	コール・グラフ	クラス/メンバ	解析グラフ
	パネル	パネル	パネル	パネル	パネル
グローバル関数	0	_	0	0	_
スタティック関数	0	_	0	0	_
メンバ関数【CC-RX】 <sup>注1</sup>	0	_	0	0	_
グローバル変数	_	0	0	0	0
ファイル内スタティック変数		0	0	0	0
関数内スタティック変数		0	0	_	0
ローカル変数	_	-	-	_	_
IOR/SFR <sup>注2</sup>		0	0	_	0
クラス変数【CC-RX】 <sup>注1</sup>	_	0	0	0	0
インスタンス変数【CC-RX】 <sup>注1</sup>	_	_	_	0	0



注1. 【CC-RX】

C++ ソース・ファイルを対象とする場合のみ存在する関数/変数です。

IOR [RX] [V850]
 SFR[RL78] [78K0R] [78K0]

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<sup>®</sup>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/クラス/メンバパネルにおいて、静的解析情報が表示可能となります。
     なお、静的解析情報を取得するためには、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(「1.1.2 解析情報の種別」参照)。
- (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箇所へジャンプします。

(c) 関数にブレークポイントを設定する

パネル上の関数に対して、ブレークポイントを設定します。

(d)参照箇所を一覧表示する

パネル上の関数を参照している箇所を検索し、参照箇所一覧を表示します。

- 備考 アクティブ・プロジェクト以外で定義されている関数情報を表示することもできます。
   表示方法についての詳細は、「2.12 情報ファイルをインポート/エクスポートする」を参照してください。
- (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箇所へジャンプします。

(c) 変数にブレーク・イベントを設定する

パネル上の変数に対して、アクセス系のブレーク・イベントを設定します。

#### (d) ウォッチ式に登録する

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

#### (e)参照箇所を一覧表示する

パネル上の変数を参照している箇所を検索し、参照箇所一覧を表示します。

備考 アクティブ・プロジェクト以外で定義されている変数情報を表示することもできます。 表示方法についての詳細は、「2.12 情報ファイルをインポート/エクスポートする」を参照してく ださい。

#### (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/変数が定義されている箇所へジャンプします。

#### (b) 関数/変数を検索する

コール・グラフ内に存在する関数/変数を検索します。

#### (9) クラス/関数/変数の情報を表示する

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

#### (a) 定義箇所へジャンプする

ツリー上のクラス/関数/変数が定義されている箇所へジャンプします。

#### (b) 宣言箇所へジャンプする

ツリー上のクラス/関数/変数が宣言されている箇所へジャンプします。

#### 備考 【CC-RX】

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

#### (10) 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。
- (11) 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。
- (12) 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。

メイン・ウインドウのツールバーの ジョボタンをクリックすることにより,現在表示可能な(「1.1.2 解析情報の 種別」参照)最新の関数情報が関数一覧パネルに表示されます。

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

#### 注意 【CA850】【CA78K0R】【CA78K0】

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

( <b>1</b> )	関数名	<b>∀</b> ₽	ファイル名	74	属性 ▽₽	戻り値の型 マ₽	引数 ▽中	1
ト行マーク	AD_Init		CG_ad.c		-	void	void	
	AD_Read		CG_ad.c		-	unsigned short	unsigned····	"
	💊 AD_Start		CG_ad.c		-	void	void	1
	💊 AD_Stop		CG_ad.c	1	-	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	~

#### 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
  - プログラムの実行により値が変化した情報は強調表示されます(強調表示の際の文字色/背景色は、オプションダイアログにおける[全般-フォントと色]カテゴリの設定に依存)。強調表示は、ツールバーの ボタンをクリックすることにより解除することができます。
  - 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

内容の更新を行います。

- ブレーク・イベントを設定する
- 参照箇所を一覧表示する
- 4. アクティブ・プロジェクト以外で定義され、かつアクティブ・プロジェクトから一度も参照されていない
   関数は、関数一覧パネルでその情報が表示されません。この場合の関数情報の表示方法については、「2.
   12 情報ファイルをインポート/エクスポートする」を参照してください。

RENESAS

# 2.3 変数情報を表示する

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

メイン・ウインドウのツールバーの デボタンをクリックすることにより,現在表示可能な(「1.1.2 解析情報の 種別」参照)最新の変数情報が変数一覧パネルに表示されます。

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

#### 注意 【CA850】【CA78K0R】【CA78K0】

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

ッールバーー	5 <u>3一覧</u> 図 22 09 155 55 15					×
		ファイル名 マキ	属性 ▽₽	型 🖓	メンバ マセ	アドレス マロ
カレント行マーク 🗕 🖌	▶ Se_ad_data	CG_main.c		unsigned short	-	0x03ffb116
	🖉 🖉 🖉 🖉	CG_main.c	-	unsigned char	-	0x03ffb114
	👽 g_count_10ms	CG_main.c	-	unsigned int	-	0x03ffb110
	😜 g_count_1 ms	CG_main.c	-	unsigned int	-	0x03ffb10c
	😜 g_flag_detect	CG_main.c	-	unsigned short	-	0x03ffb118
	S_romp	CG_systeminit.c	-	-	-	-
	合計値	CG_main.c	-	-	-	-
	合計値	CG_systeminit.c	-	-	-	-
	<	ļ			Anno 1997 - Anno 19	>

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
  - プログラムの実行により値が変化した情報は強調表示されます(強調表示の際の文字色/背景色は、オプションダイアログにおける[全般-フォントと色]カテゴリの設定に依存)。強調表示は、ツールバーの
     ボタンをクリックすることにより解除することができます。
  - **3.** カレント行マーク(▶)は,該当行がカレント行であることを示します。

カレント行マークのある変数を対象として、次の操作を行うことができます。

- 定義箇所へジャンプする

内容の更新を行います。

- ブレーク・イベントを設定する
- 参照箇所を一覧表示する
- 変数値の推移をグラフ化する
- 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 変数は、変数一覧パネルでその情報が表示されません。この場合の変数情報の表示方法については、「2.
   12 情報ファイルをインポート/エクスポートする」を参照してください。

# 2.4 関数間の呼び出し関係(コール・グラフ)を表示する

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

メイン・ウインドウのツールバーの ボタンをクリックすることにより,現在表示可能な(「1.1.2 解析情報の 種別」参照)最新のコール・グラフがコール・グラフ パネルに表示されます。

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

- 注意 1. デバッグ・ツールがトレース機能をサポートしていない場合、またはデバッグ・ツールのトレース機能を有効化していない場合、コール・グラフにおいて、動的解析情報(実行回数/リード回数/ライト回数)を表示することはできません。
  - 2. 【CA850】【CA78K0R】【CA78K0】 ビルド・ツールにおいてクリーンを実行すると、現在表示しているコール・グラフは消失します。



#### 図 2-3 関数間の呼び出し関係の表示(コール・グラフ パネル:全体表示)

表示されるコール・グラフに対して、次の操作を行うことができます。



#### (1) 親関数を変更して表示する

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

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

備考 【V850E2】

選択しているマイクロコントローラがマルチコア対応版の場合では, [親関数名] コンボ・ボックスに おいて "PEn"を指定することができます。この場合は, 該当 PEn で実行される関数のみを対象と したコール・グラフが表示されます。

## (2) 指定した関数の親関数/子関数を表示する

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

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

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



図 2-4 親関数/子関数の表示(コール・グラフ パネル:詳細表示)

(3) 定義箇所ヘジャンプする

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

## (4) 関数/変数の情報をポップアップ表示する

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

注意 パネル・コントロール・エリアの [1] ボタンの設定により、マウスのドラッグによるスクロールを許可している場合、この機能を使用することはできません。

#### 図 2---5 関数情報/変数情報のポップアップ表示例



(5) 関数/変数を検索する

コール・グラフ内に存在する関数/変数を検索することができます。

操作は、コール・グラフ パネルのツールバーの 満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。

				20	
			×	126	
	条件( <u>C</u> ):	を含む	*		
クラス名(ビ):			~		
	条件(1):	を含む	~		
□ 大文字と小	文字を区別す	\$3( <u>5</u> )			
親関数の個数	t ( <u>P</u> ):		~		
	条件(_):	と等しい	~		
子関数の個数	( <u>Η</u> ):		~		
	条件( <u>D</u> ):	と等しい	~		
				8 -	

#### 図 2-6 関数の検索(コール・グラフ検索 ダイアログ)



#### (a) [関数/変数名]の指定

検索対象となる関数名/変数名を指定します。

キーボードより文字列を直接入力するか(最大指定文字数:2046文字),ドロップダウン・リストより 入力履歴項目を選択します(最大履歴数:10個)。

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

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

備考 関数名/変数名は、次の操作によっても指定することができます。

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

(b) [クラス名] の指定【CC-RX】

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

キーボードより文字列を直接入力するか(最大指定文字数:2046文字),ドロップダウン・リストより 入力履歴項目を選択します(最大履歴数:10個)。

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

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

備考 クラス名は、次の操作によっても指定することができます。

- 任意の文字列をこのエリアにドラッグ・アンド・ドロップ

(c) [親関数の個数]の指定

関数の検索条件の1つとして必要な場合、検索対象の関数の親関数の個数を指定することができます。 キーボードより数値を直接入力するか(指定可能範囲:0~65535)、ドロップダウン・リストより入力

履歴項目を選択します(最大履歴数:10 個)。

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

(d) [子関数の個数]の指定

関数の検索条件の1つとして必要な場合、検索対象の関数の子関数の個数を指定することができます。 キーボードより数値を直接入力するか(指定可能範囲:0~65535)、ドロップダウン・リストより入力 履歴項目を選択します(最大履歴数:10個)。

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

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

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

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



図 2----7 関数の検索箇所

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

プログラム中のクラス情報【CC-RX】/関数情報/変数情報をツリー形式で表示することができます。 これらの情報は、メイン・ウインドウのツールバーの

# ルに表示されます。

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

注意 1. 【CC-RX】

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

2. [CA850] [CA78K0R] [CA78K0]

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

	クラス/メンバ	×
ッールバーー	[ビュー設定] コンボ・ボックス	
ーー パネル・コントロール ・エリア ー	カテゴリ・グループ表示 🛛 🗿 🕥 (対象名を入力してくだ 🔍 🍼	$\mathbf{\overline{\mathbf{b}}}$
	<ul> <li>別名定義 (typedef)</li> <li>③ グローバル関数と変数</li> <li>● ○ 名前空間</li> <li>● クラス</li> <li>クラス</li> <li>クラス</li> <li>クラス</li> <li>クラス</li> <li>カデゴリ用ツリー・エリア</li> <li>● ○ 共用体</li> <li>● ○ 列挙体</li> </ul>	
	ਵw tempfunc02 <t:2>0 ਵw tempfunc02<t:2>(char, short) ਵw tempfunc02<t:2>(short, int) ਵw templatetest01 (void)</t:2></t:2></t:2>	~
	=w templatetest02(void) <b>変数</b>	
	♥ AAA ♥ Fixed_Vectors	-
	CTRI	~

図 2----8 クラス情報の表示(クラス/メンバ パネル)

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

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

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

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

表 2---3 クラス/メンバ パネルにおけるツリーの分類方法

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

- 定義箇所へジャンプする

- 宣言箇所へジャンプする

2. [CC-RX]

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



# 2.6 表示方法をカスタマイズする

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

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

- 表示項目を設定する
- 表示項目を並び替える
- 特定項目を固定表示に設定する
- 解析情報をソート表示する
- 解析情報をフィルタ表示する
- 備考 列の選択 ダイアログ上の [デフォルト] ボタンをクリックすることにより、上記の操作により行ったカスタ マイズをすべてデフォルトの状態に戻すことができます。

## 2.6.1 表示項目を設定する

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

- (1) 表示項目を限定する場合
- (2) 表示項目を追加する場合

## (1) 表示項目を限定する場合

操作方法は、次の2通りです。

(a) パネル内での操作

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

. .

#### 図 2--9 表示項目の限定(パネル内での操作)

2	र्दे 🤗 🕵 📡	🐹 時間単位(U) -		属性・マロ
Ø	関数名 TMP0_Start	マセ ファイル名 CG_timer.c	マロ属性	g 中 戻り値の型 マ中 void
	🥃 main 🥃 PORT_Init 🛛 Γ	CG_main.c 非まー対象の項目をフ	<u>ウフにとい</u>	void void
	🖕 CG_ReadRese	非表示対象の項目を パネル外へドラッグ・	ウスにより アンド・ドロップ	void



## (b) 列の選択ダイアログによる操作

パネルのヘッダ・エリア左端の目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。





## (2) 表示項目を追加する場合

パネルのヘッダ・エリア左端の習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 2-12 表示項目の追加(列の選択ダイアログからのドラッグ・アンド・ドロップ操作)



# 2.6.2 表示項目を並び替える

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

- (1)パネル内での操作
- (2) 列の選択ダイアログからの操作

#### (1)パネル内での操作

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

1 関数名	7₽	ファイル名	\. \. \. \. \. \. \. \. \. \. \. \. \. \	属性	<b>∀</b> ₽	戻り値の型	74
TMP0_Start		CG_timer.c	-		5	void	
💊 main		CG_main.c				void	
💊 PORT_Init		CG_port.c		10 =1 +1 +	も <b>の</b> 西 ロ	<i>タナ</i> → 上 フ <i>オ</i>	=
🖉 🖕 CG_ReadRese	t	CG_system_u	ser.c	移動対象	家の項日	名をマワスで	ドフ
a 🖃 🚳 i 📧 🗹	190 2		7				
∎ <u>₹</u> % £ 5	\$ 15 .	時間単位(山)	7				
2 武 🏶 🕵 🕻 関数名	, 23 3 ₽ 7	時間単位(山)	7 )+ 7+p <sup>+</sup> +p	属性	\	戻り値の型	74
2 ( ) ( ) ( ) ) ) ( ) ) ) ( ) ) ) ) ( ) ) ) ) ) ) ) ) ) ) ) ) )	<b>⊠ 3</b> ⊽≠	時間単位(U) 時間単位(U) 「Gu_timer.c	7-11-1	属性	\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	戻り値の型 void	74
2 (現数名 (明数名 (TMP0_Start (main	₹ 7+ 7	時間単位(山 居性 T CG_timerc CG_main.c	7 7 <del>1</del> 7 <del>1</del> 1	属性	ΥÞ	戻り値の型 void void	<b>∆</b> +
3 【】 ● 「「「」 3 【】 ● 「関数名 ■ TMP0_Start ■ main	7 H	時間単位(山) 居性 CG_timer.c CG_main.c CG_port.c	7+1+1	属性	74	戻り値の型 void void void void	Δħ

#### 図 2-13 表示項目の並び替え(パネル内での操作)



#### (2)列の選択ダイアログからの操作

パネルのヘッダ・エリア左端の記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。





# 2.6.3 特定項目を固定表示に設定する

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

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

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

なお、固定表示アイコンの意味は次のとおりです。

表 2—4 固定表示アイコン

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

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

RENESAS

# 2.6.4 解析情報をソート表示する

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

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

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

なお、ソート表示を行っている項目名には次のマークが表示されます。

表 2—5 ソート表示マーク

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

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

# 2.6.5 解析情報をフィルタ表示する

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

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

各項目ごとにカスタムなフィルタの設定を行い、取得した情報値を表示します。 操作は、対象の項目名のフィルタ・アイコン (▼/▼)をクリックすることで表示される、次のメニュー 項目を選択することで行います。



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。

表 2---6 フィルタの設定項目

[(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 定を行ってくだい。



図 2-15 項目ごとのフィルタ設定(フィルタ設定 ダイアログ)

なお、フィルタ・アイコンの意味は次のとおりです。

表 2---7 フィルタ・アイコン

アイコン	説明
Y	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)。
<b>T</b>	フィルタ表示を行っていることを示します。
### (2) パネルと連携したフィルタ表示

- (a) プロジェクト・ツリー パネルとの連携
- (b) エディタ パネルとの連携
- (c) デバッグ・マネージャパネルとの連携【V850E2】
- 注意 ここで説明するフィルタ表示を行っている際に、「(1)項目ごとのカスタム設定によるフィルタ表示」 の設定を行うと、それまで行っていたパネルと連携したフィルタ表示は解除されます。

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

#### (a) プロジェクト・ツリー パネルとの連携

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

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

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

選択対象	表示対象
アクティブ・プロジェクト内の	単一ファイル内に定義されている関数/変数
単一ファイル	
アクティブ・プロジェクト内の	複数ファイル内に定義されている関数/変数
複数のファイル	
アクティブ・プロジェクト内の	単ーカテゴリ下のファイル内に定義されている関数/変数
単一のカテゴリ	
アクティブ・プロジェクト内の	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
複数のカテゴリ	
アクティブ・プロジェクト内の	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
ファイルとカテゴリの組み合わせ	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

表 2-8 プロジェクト・ツリー パネルと連携したフィルタ表示

注意 アセンブラ・ソース・ファイルは、解析対象外です。

備考 ヘッダ・ファイルを選択した場合,該当ヘッダ・ファイル内で定義されている関数/変数が表示 対象となります。 (b) エディタ パネルとの連携

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

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

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

(c) デバッグ・マネージャパネルとの連携【V850E2】

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の K ボタンをクリックすることでこのフィルタ機能を有効します(この機能を解除する場合は、再び同ボタンをクリックします)。

ただし,選択しているマイクロコントローラがマルチコア対応版でない場合,この機能は無効となります。



## 2.7 定義箇所へジャンプする

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

### 注意 【CA850】

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

- 例1. #pragma task TASK\_A
  - 2. #pragma interrupt INTP0 functionA
- (1) エディタ パネルへのジャンプ

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

パネル	対象	操作
関数一覧 パネル	関数行	ダブルクリック
変数一覧 パネル	変数行	
コール・グラフ パネル	- 関数ボックス - 変数ボックス	ダブルクリック <sup>注</sup>
クラス/メンバ パネル	クラス/カテゴリ用ツリー - <i>名前空間名</i> ノード【CC-RX】 - <i>クラス名</i> ノード【CC-RX】 - <i>インタフェース名</i> ノード【CC-RX】 - <i>構造体名</i> ノード - <i>共用体名</i> ノード - <i>列挙体名</i> ノード	コンテキスト・メニューの [ソースヘジャン プ] を選択
	関数/変数用リスト - <i>関数名</i> ノード - <i>変数名</i> ノード - マクロ名ノード - 別名定義名ノード - 列挙体メンバ名ノード	ダブルクリック

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

### (2) 逆アセンブル パネル/メモリ パネルへのジャンプ

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

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

**注 1.** メモリ パネルへのジャンプのみサポートしています。

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



## 2.8 宣言箇所へジャンプする

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

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

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

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



## 2.9 ブレーク・イベントを設定する

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

- 関数にブレークポイントを設定する

-変数にブレーク・イベントを設定する

## 2.9.1 関数にブレークポイントを設定する

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

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

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1. 該当箇所にすでにブレークポイントが設定されている場合は、次の動作となります。 - 有効状態のブレークポイントが設定されている場合: 何もしません - 無効状態のブレークポイントが設定されている場合: 有効状態にします

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

## 2.9.2 変数にブレーク・イベントを設定する

変数一覧 パネル上の変数にアクセス系のブレーク・イベントを設定します。

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

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

この際に、コンテキスト・メニュー内のテキスト・ボックスに値を指定することができます。

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

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。

変動一覧		
2 🛣 🧶 😰 🖀	変数 "global_b"	をカレント行にしたのち、コンテキスト・メ
	ニューより、[アク	クセス・ブレークの設定] → [変数に書き込み
elobal_array ma	ain.c ブレークを設定]	内に値を入力し、[Enter] キーを押下します。
● global_b i m/ global_c A すべ	ainc ここでは、変数 " ての参照す レークする設定を	global_b"に"Oxb"が書き込まれた場合にブ しています。
global_char] 峰解析	fグラフ1に包隷(12)	signed char [256]
global_char. アクセ	セス・ブレークの設定(目) 🔸	👼 変数に読み込みブレークを設定( <u>R)</u>
global_struc 🔛 ウォン	ッチ1 (2登録( <u>R</u> )	💉 変数に書き込みブレークを設定W Oxb
🔜 global_strud 🔣 yuu;	スヘジャンプ( <u>S</u> )	☞ 変数に読み書きブレークを設定(A)
🔄 🔤 🐺 XEU	ノヘジャンプ( <u>M</u> )	2
🔁 38-	–( <u>C</u> ) Ctrl+C	

図 2-16 変数一覧 パネルにおけるブレーク・イベントの設定例

- 注意1. この操作は、デバッグ・ツールと接続時のみ行うことができます。
  - 2. [RX]

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

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

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



## 2.10 ウォッチ式に登録する

変数一覧 パネル上の変数をウォッチ パネル (ウォッチ 1) のウォッチ式として登録することができます。 操作は、対象となる変数の行(複数行選択可)をウォッチ パネル (ウォッチ 1) へ直接ドラッグ・アンド・ドロッ プすることで行います。

- 注意 1. この操作は、デバッグ・ツールと接続時のみ行うことができます。
  - 2. [合計値]の行に対してこの操作は無効です。
- 備考 対象となる変数の種別により、ウォッチパネル上に登録されるウォッチ式の名称は次のように異なります。
  - グローバル変数: "変数名"
  - ファイル内スタティック変数: "ファイル名#変数名"
  - 関数内スタティック変数: "ファイル名 # *関数名* # *変数名*"
  - クラス変数: "ク*ラス名*:: *変数名*"



## 2.11 参照箇所を一覧表示する

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

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

検索結果は、次の出力パネルの[参照の検索]タブ上に出力されます。

#### 図 2-17 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):	101
〕 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」 [FOF]	~
↓ 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/C++ ソース・ファイル中の "#if" / "#ifdef" などで、コンパイル時にプリプロセッサにより除外される コードにおいて参照されている箇所は参照箇所として出力されません。
- 備考1. 出力結果上の関数名/変数名をダブルクリックすることにより、エディタパネルをオープンし、該当関数 /変数が定義されているソース・テキスト箇所へジャンプすることができます。
  - 2. 出力パネルの [参照の検索] タブにフォーカスがある状態で、[ファイル] メニュー→ [名前を付けて 出力 参照の検索 を保存...]を選択することにより、参照箇所一覧をテキスト・ファイル (\*.txt) に保存することができます。

## 2.12 情報ファイルをインポート/エクスポートする

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

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

情報ファイルには次の種類があります。

表 2---9 情報ファイルの種類

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

注選択しているマイクロコントローラが、マルチコア対応版の場合のみ対象となる項目です。

次の手順により操作を行ってください。

(1) 情報ファイルを生成(エクスポート) する

#### (a) 手動でエクスポートを行う場合

インポートする必要のある関数/変数についてのみの情報ファイルを生成します。

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

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



呂∠──10   邗ノナゴルの工队(民奴 見ノナゴルで工队)の物口0
------------------------------------

名前を付けて保存				2 🛛
(保存する場所型:	🚞 Sample		S 🕫 💌	
していたファイル	DefaultBuild			
<b>じ</b> デスクトップ				
ک ۲۲ ۴キ۱۶۷۶				
<b>ग</b> २१ २४८२-७				
	ファイル名(N): ファイルの種類(T):	FuncList_Sub2.mtfl 関数一覧ファイル (*.mtfl)	× •	1#17 (2) キャンセル

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

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

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

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

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

操作は、プロパティパネルの[設定] タブ上の[インポート/エクスポート] カテゴリ内[ビルド時に エクスポートする] プロパティを[はい] に指定したのち、同カテゴリ内の[関数用エクスポート・ファ イル名] / [変数用エクスポート・ファイル名] プロパティにおいて、生成する情報ファイル名を指定し ます(拡張子は、関数一覧ファイルの場合は "mtfl"、変数一覧ファイルの場合は "mtvl" に限ります)。 相対パスによる指定の場合は、プロジェクト・フォルダを基点として指定してください。

### 図 2—19 [インポート/エクスポート] カテゴリ

⊡ 1	ンボート/エクスボート	
田子	ンポート・ファイル	インポート・ファイル[0]
Ľ.	ルド時にエクスポートする 🧹	はい 🔌
関	数用エクスポート・ファイル名	FuncList_Sub2.mtfl
変	<b>数用エクスポート・ファイル名</b>	VarList_Sub2.mtvl



以上の設定により、情報ファイルの自動エクスポートの設定は完了です。

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

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

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

- %ProjectName% :プロジェクト名に置換します。

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

(2) 情報ファイルをインポートする

生成(エクスポート)した情報ファイルをインポートする設定を行います。

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

図 2-20 [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

□ インボート/エクスボート	<b>×</b>	
田 インポート・ファイル	インポート・ファイル[0]	

#### 図 2-21 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		<b>X</b>
パス(1行につき1つのパ	ス)( <u>P)</u> : 😱	
%ActiveProjectDir%¥F	FuncList_Sub2.mtfl	~
<		×
参照( <u>B</u> ) プレースホルダ( <u>L</u> ):		
プレースホルダ	値	説明 🔼
ActiveProjectDir ActiveProjectName MainProjectDir MainProjectName MicomToolPath	C:¥Test¥Sample 78K0R_LH3 userop C:¥Test¥Sample 78K0R_LH3 userop C:¥Test¥Sample	アクティブ・プロジェクトのフォル! アクティブ・プロジェクト名 メイン・プロジェクトのフォルダの メイン・プロジェクト名 CubeSuite+のインストール先:
<		>
	ОК	) キャンセル ヘルブ(円)



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

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

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

### 図 2-22 [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)

□ インポート		
<b>ワ</b> インポート・ファイル	インボート・ファイル[2]	
( [0]	C:¥Test¥Sample¥FuncList_Sub2.mtfl	)
[1]	C:¥Test¥Sample¥VarList_Sub2.mtvl	

以上の設定により、情報ファイルのインポートの設定は完了です。

- 備考1. パス編集 ダイアログでは、次のプレースフォルダに対応しています。
  - %ProjectName% : プロジェクト名に置換します。
  - %MicomToolPath% : CubeSuite+ のインストール・フォルダの絶対パスに置換します。
  - CubeSuite でエクスポートした情報ファイル(CubeSuite 関数一覧ファイル(\*.csfl) / CubeSuite 変数一覧ファイル(\*.csvl))をインポートすることができます。
  - **3.** インポート・ファイルとアクティブ・プロジェクト内に同名の関数/変数が存在する場合は、次の 決定規則に従います。

C ソース・ファイルの場合:

- ファイル名が異なり、かつインポート・ファイル側の属性に "static" (小文字のみ) が含まれてい る場合は、スタティック関数/スタティック変数とみなし、異なる関数情報/変数情報として取 り込みます。
- ファイル名が同名で関数名が異なり、かつインポート・ファイル側の属性に"static"(小文字のみ)が含まれている場合は、関数内スタティック変数とみなし、異なる変数情報として取り込みます。
- 上記以外の関数/変数については、グローバル関数/グローバル変数、または同一ファイル内の 同名スタティック関数/スタティック変数とみなし、同一の関数情報/変数情報としてマージし て取り込みます。

C++ ソース・ファイルの場合:

- [クラス名]【CC-RX】, [名前空間]【CC-RX】, [ファイル名], および [引数] を比較して, 1つ でも一致していない場合, 異なる関数情報として取り込みます。
- [クラス名]【CC-RX】, [名前空間]【CC-RX】, [関数名], [ファイル名], および [引数]を比較 して, 1つでも一致していない場合, 異なる変数情報として取り込みます。
- 上記以外の関数/変数については、同一の関数情報/変数情報としてマージして取り込みます。

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

項目		規則
関数情報	変数情報	
[クラス名]【CC-RX】 [名前空間]【CC-RX】 [ファイル名] [ファイル・パス] [DE 体報] 注	[クラス名] [CC-RX] [名前空間] [CC-RX] [ファイル名] [関数名]	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 免したる問題時報 (恋教情報が方方する場合)
[FE] ff fk] [スタック・サイズ[バイト]] [引数の数] [引数] [戻り値の型] [アクセス指定子] [CC-RX] [属性]	[ファイル・ハス] [PE 情報] 注 [アクセス指定子] [CC-RX] [属性] [型] [メンバ]	家となる国政情報/変数情報/行任りる場合は 最後にインポートされたファイルの関数情報/変 数情報を取り込みます。 なお、アクティブ・プロジェクト内/インポー ト・ファイル内のどちらにも値が存在しない場合 は空欄("-")となります。
[コード・サイズ[バイト]]	[サイズ]	次の優先順位に従います。 "インポート・ファイルの値" > "アクティブ・ プロジェクトの値" ただし、複数のインポート・ファイルにマージ対 象となる関数情報/変数情報が存在する場合は、 最後にインポートされたファイルの関数情報/変 数情報を取り込みます。 なお、アクティブ・プロジェクト内/インポー ト・ファイル内のどちらにも値が存在しない場合 は空欄("-")となります。

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

### (3) 情報ファイルのインポートを中止する

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



## 2.13 解析情報をグラフ化して表示する

取得した関数情報/変数情報をグラフ化して表示することができます。

グラフの表示は、メイン・ウインドウのツールバーの 

「ボタンをクリックすることでオープンする、解析グラフ
パネルで行います (

「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。

解析グラフ1	8
2 式 😼 🏄 🖉	
global_a	🛿 🙇 🙋 (変数値を入力して(ださい) 🛛 🔽 峰
global_a global_b global_c 2 a 2 a 1 1 5 c 0 c	
σ-	515ms572µ5900ns 1531ms196µ5300ns 15531ms201
	×
値の推移、実行時間の割合	

### 図 2-23 解析情報のグラフ化(解析グラフパネル)

## 2.13.1 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、[値の推移] タブを参照してください。



図 2—24 変数値の推移のグラフ化

解析グラフ1		8
2 🔁 😼 🌌		
global_a	🔊 💁 🙋 変 数値を	入力して(ださい) 🔹 🔛
global_a global_b global_c 2 1	牧名] コンボ・ボックス	【変数値】コンボ・ボックス
	3 515ms572µs900ns	1s31ms196µs300ns 1s531ms201
(H) (H) (H) (H)		N N
値の推移、実行時間の割る	\$ /	-

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
- クラス変数(C++ ソース・ファイルを対象とする場合)
- CPU レジスタ
- IOR [RX] [V850]
- SFR [RL78] [78K0R] [78K0]
- アドレス

グラフ化対象の登録方法には、次の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<sup>注</sup>
- エディタ パネル
- CPU レジスタ パネル
- IOR パネル【RX】【V850】
- SFR パネル【RL78】【78K0R】【78K0】
- ウォッチ パネル
- 注 対象となる変数の種別により、グラフ化対象として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"
  - クラス変数: "ク*ラス名*:: *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、 A 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、 レー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の ぶボタン(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

RENESAS

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)。

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

- 子ノードを持つ変数(配列/構造体など)の場合,構造体のメンバや配列のインデクスなど, 展開後の変数などについては登録対象となりません。
- 3. すでに登録済みのグラフ化対象と名前が一致する場合、登録対象となりません。
- (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のボタン により指定します。

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	<u>ii</u>	デバッグ・ツールのトレース機能により取得したトレー
[IECUBE [V850]]		ス・データを基にグラフ表示を行います。
[IECUBE [RL78]]		
[IECUBE [78K0R]]		
【IECUBE2】 <sup>注</sup>		
【シミュレータ】		
リアルタイム・サンプリング方式	D	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)。

表 2—10 グラフ・データの取得方法の選択

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きます。

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。

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 <sup>注1</sup> の場合のみ表示可	デバッグ・ツールの RRM 機能が有効 <sup>注1</sup> 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 <sup>注2</sup>	【IECUBE【V850】】【MINICUBE2【78K0】】 【EZ Emulator【78K0】】 デバッグ・ツールの RRM 機能が使用可能 な領域に依存 <sup>注 3</sup>
グラフ化可能な 対象のサイズ	- 4 バイト以下【RX】【V850】 - 2 バイト以下【RL78】【78K0R】【78K0】	4 バイト以下

表 2-11 グラフ・データの取得方法によるグラフ表示の相違



相違点	トレース・データ解析方式	リアルタイム・サンプリング方式
	【IECUBE】【シミュレータ】	
時間表示範囲	トレース・データとして記録されてい	プログラムの実行開始から実行停止までの
	る時間	実行時間(Run-Break 時間)
時間表示形式	XXXsXXXmsXXXµsXXXns	XXXsXXXms
値の遷移箇所	実際のタイミングと合致(ポップアッ	指定サンプリング間隔 <sup>注 4</sup> に依存するため
	プ表示から確認可能)	正確な時間/変異箇所の特定は不可
注意	次の場合は値の推移の解析不可	- グラフ化対象の登録個数に依存して、サン
	- コンパイラの最適化により、変数が	プリング間隔が不定となる可能性あり
	レジスタに割り当てられている区間	- 値の取得を失敗する可能性あり
	- 2 バイトの対象に 1 バイト単位で値を	失敗した場合は、時間情報のみ表示し、
	書き込んだ場合、または4バイトの	変化点と変化点を結ぶ線は非表示
	対象に1バイト /2 バイト単位で値を	
	書き込んだ場合	

- **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。
  - 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。
  - 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。

- 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,登録したグラフ化対象に関する最新のグラフが表示されます(デフォルト)。

なお、リアルタイム・サンプリング方式を指定している場合では、プログラム実行中においても、指定されているサンプリング間隔でグラフが更新されます(最新の 20 個まで表示可)。

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。

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 2-25 グラフ表示における表	表範囲の設定
-------------------	--------

Ξ	解析グラフ 1			
	表示範囲の上限/下限を設定する	はい	<b>v</b> )	
	表示範囲の上限値1	200		
	表示範囲の下限値1	-200		
	有効範囲の上限/下限を設定する1	()()/	7	
	凡例を表示する 1	いいえ		
	実行時間の割合グラフに表示する関数	10	工限/下限値を10/10進数で入力します。	
			【入力例】	
			100, -32, 0xFFFF, -0x800	

(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。

#### 図 2-26 グラフ表示における有効範囲の設定

E	日 解析グラフ 1	
	表示範囲の上限/下限を設定する1_いいえ	
	有効範囲の上限/下限を設定するしはい	~)
	有効範囲の上限値1 200	
	有効範囲の下限値1 -200	
	凡例を表示する1 いい	
	実行時間の割合グラフに表示する関数 10 上限/下限値を 10/16 進数で入	カします。
	【入力例】	
	100, -32, 0xFFFF, -0x800	

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。

 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。

RENESAS

- 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。
- 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 2-27 遷移箇所のポップアップ表示例



備考1. [場所] 情報は、トレース・データ解析方式により取得したグラフ・データで、かつ遷移箇所の情報が存在する場合のみ表示します(存在しない場合は"-"を表示します)。 また、この場合、遷移箇所上を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。

- 2. [時間] 情報の表示形式は次のとおりです。
  - トレース・データ解析方式: XXXsXXXmsXXXµsXXXns
  - リアルタイム・サンプリング方式 : XXXsXXXms
- (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 20 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 2-28 グラフ・データの検索箇所

## 2.13.2 関数の実行時間率をグラフ化する

関数の実行時間の割合を円グラフで表示します。

グラフ表示は、現在取得している動的解析情報(関数一覧パネルにおける[実行時間(割合)[%]]と同等)を基 に、解析グラフパネルの[実行時間の割合]タブで行います(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、[実行時間の割合]タブを参照してください。

- 注意 1. デバッグ・ツールがトレース機能をサポートしていない場合、またはデバッグ・ツールのトレース機能 を有効化していない場合、このグラフを表示することはできません。
  - 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—29 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 2-30 実行時間のポップアップ表示例

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



## 2.14 解析情報をファイルに保存する

関数一覧 パネル/変数一覧 パネル/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
関数一覧ファイル (*.mtfl)	関数情報をインポートするためのファイル形式(「2.12 情報ファイル
	をインポート/エクスポートする」参照)

注意 現在パネル上に表示している項目/解析情報値のみが保存対象となります。

#### (2) 変数情報を保存する

操作は、変数一覧 パネルにフォーカスがある状態で、[ファイル] メニュー→ [名前を付けて 変数一覧データ を保存…]を選択するとオープンする名前を付けて保存 ダイアログにより行います。

保存の際は、次のファイル形式を指定することができます。

テキスト・ファイル (*.txt)	テキスト形式
CSV(カンマ区切り)(*.csv)	CSV 形式
Microsoft Office Excel ブック (*.xls)	Microsoft Excel 97 - Excel 2003, および 5.0/95 ブック (*.xls) 形式
変数一覧ファイル (*.mtvl)	変数情報をインポートするためのファイル形式(「2.12 情報ファイル
	をインポート/エクスポートする」参照)

注意 現在パネル上に表示している項目/解析情報値のみが保存対象となります。

(3) 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。



## 第3章 注意事項

この章では、解析ツールを使用する際の注意事項を示します。

## 3.1 アクティブ・プロジェクトの変更について

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

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

## 3.2 カバレッジ結果について

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

カバレッジの結果をクリアするには、エディタ パネル/逆アセンブル パネルにおいて、コンテキスト・メニューの [カバレッジ情報のクリア]を選択したのち、関数一覧 パネル/変数一覧 パネルの ご ボタンをクリックしてください。

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

## 3.3 トレース・データ解析方式について

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

## 3.4 プログラム実行中にパネルをオープンした場合について

#### (1) 関数一覧パネル/変数一覧パネル

静的解析情報のみを更新して表示します。動的解析情報の更新は行いません。

#### (2) コール・グラフパネル

- CubeSuite+ 起動後,一度もパネルをオープンしていない場合

静的解析情報のみを更新して表示します。動的解析情報の更新は行いません。



- CubeSuite+ 起動後,パネルをオープンしたあとの場合 前回の内容を表示します。
- (3) クラス/メンバ パネル

静的解析情報のみを更新して表示します。動的解析情報の更新は行いません。

- (4) 解析グラフパネル
  - CubeSuite+ 起動後, 一度もパネルをオープンしていない場合

[値の推移] タブ	何も表示しません。
[実行時間の割合] タブ	"実行時間情報がありません。"を表示します。

- CubeSuite+ 起動後、パネルをオープンしたあとの場合

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



# 付録 A ウインドウ・リファレンス

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

## A.1 説 明

次に、解析ツールで使用するウインドウ/パネル/ダイアログの一覧を示します。

ウインドウ/パネル/ダイアログ名	機能概要
メイン・ウインドウ	CubeSuite+ を起動した際、最初にオープンするウインドウ
プロジェクト・ツリー パネル	プロジェクトの構成要素のツリー表示
プロパティ パネル	解析ツールの詳細情報の表示。および設定の変更
関数一覧 パネル	取得した関数情報の表示
変数一覧 パネル	取得した変数情報の表示
コール・グラフ パネル	関数間の呼び出し関係(コール・グラフ)の表示
クラス/メンバ パネル	取得したクラス情報【CC-RX】 <sup>注</sup> /関数情報/変数情報のツリー表示
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(*.mtfl)/変数一覧ファイル(*.mtvl))をイ
	ンポートする際のファイルの指定
列の選択 ダイアログ	関数一覧 パネル/変数一覧 パネルにおける表示項目の並べ替え、または表示/
	非表示の設定
コール・グラフ検索 ダイアログ	コール・グラフ パネルで表示しているコール・グラフ内に存在する関数/変数
	の検索
フィルタ設定 ダイアログ	関数一覧 パネル/変数一覧 パネルにおけるフィルタ表示の条件設定
名前を付けて保存 ダイアログ	
	ルの表示内容の新規保存、および情報ファイル(関数一覧ファイル(*.mtfl)/
	変数一覧ファイル(*.mtvl))の生成

#### 表 A-1 ウインドウ/パネル/ダイアログー覧

注【CC-RX】

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



# メイン・ウインドウ

CubeSuite+ を起動した際,最初にオープンするウインドウです。

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

## 図 A—1 メイン・ウインドウ

	🔕 Test02 - CubeSuite+ - [フロジェクト・ツリー	1				
(1) –	ファイル(E) 編集(E) 表示(V) プロジェクト(P) ビルド(B)	デバッグ(D) ツール(T) ウイ	ノドウ( <u>W)</u> ヘルプ( <u>H</u>	)		
· · _	i 🚳 スタート(S)   🚚 📄 🎒 i 💥 🐚 📸   🥠		- : 6			SE CE CE K
(						
(2) —						
	₩ ¥ ×	🚰 プロパティ 🥑 sub01.c				- x
	2 🙆 🙎	🔍 プログラム解析 のプロパ	<del>Г</del> イ			
	□ Test02 (プロジェクト)	日全般		tto		<u>^</u>
		強制的にクロス・リファレン	ス・ファイルを出力す	る (まい)		
	国 🌽 端子配置 (設計ツール)	プロトタイプ宣言を参照検	索の対象とする	(まい)		
	■ 1 コード生成 (設計ツール)	プログラム停止時に更新な	if70	(\$(.)		
		引数を改行して表示する		()()ຂ້		
		SFR/IORを変数として表	示する	いいえ		
	〒 □ ファイル	定義箇所がない関数をコ	ール・グラフの表示対	物とない		
		ノロクラム1学正時に関数1	\$\$12 を出/19つ	616176		
		□ 解析グラフ 1				<b>~</b>
		解析結果を累積する				
		解析結果を累積するかどうか	を指定します。			
(3)		設定				-
(0)		関数一覧				д×
		3 2 3 1 8 5	時間単位( <u>U</u> )▼			
		図 関数名 ▼中 フ	ァイル名 ▽口	属性 マロ 戻り値の型	▽ㅋ 引数 ▽ㅋ	コード・サイズDバイト] 🔨
		🕨 🖘 child1 👘 mair	.c –	void	void	
		v child2 mair	i.c –	void	void	
		child3 mair	.c –	void	void	<u>anananananananananan</u> y
		child_num_1 main	.c –	void	void	
		🗸 child_num_2 main	IC –	void	void	
		💗 👽 child_num_3 👘 mair	l.C  -	void	void	
		🚽 🤘 func 🛛 subl	11.c –	void	void	
		v func2 subl	n.c  -	void	void	
		<	1111			>
1	< >	💎 関数一覧 🔝 出力				
<u> </u>	F1 F2 F3 F4	F5 FG	F7	F9 F9	F10 F11	FI2
						▲非接続

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]

[オープン方法]

- Windows の [スタート] メニュー→ [プログラム] → [Renesas Electronics CubeSuite+] → [CubeSuite+] を 選択

## [各エリアの説明]

## (1) メニューバー

## (a) [表示]

解析ツール専用の[表示]メニューの各項目、および機能は次のとおりです(デフォルト)。

出力		出力 パネルをオープンします。
プログラム解析		解析ツール用の各パネルをオープンするために、次のカスケード・メニューを表示します。
	関数一覧	関数一覧 パネルをオープンします。
	変数一覧	変数一覧 パネルをオープンします。
	解析グラフ 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。
	クラス/メンバ	クラス/メンバ パネルをオープンします。

## (2) ツールバー

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

<b>S</b>	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。
Q	変数一覧 パネルをオープンします。
	[表示]メニュー→[変数一覧]の選択と同等です。
	解析グラフ パネル(解析グラフ 1)をオープンします。
(1 番目)	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。
	解析グラフ パネル(解析グラフ 2)をオープンします。
(2番目)	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。
	解析グラフ パネル(解析グラフ 3)をオープンします。
(3番目)	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。
	解析グラフ パネル(解析グラフ 4)をオープンします。
(4番目)	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。



1	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。	
<b>\$</b>	クラス/メンバ パネルをオープンします。	
	[表示] メニュー→ [クラス/メンバ] の選択と同等です。	

(3) パネル表示エリア

CubeSuite+ が使用する各種パネルを表示するエリアです。

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

- プロジェクト・ツリー パネル
- プロパティ パネル
- 関数一覧 パネル
- 変数一覧 パネル
- 解析グラフ パネル
- コール・グラフ パネル
- クラス/メンバ パネル
- 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。		
100	変数一覧	変数一覧 パネルをオープンします。		
角	 ¥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。		
[=	ュール・グラフ	コール・グラフ パネルをオープンします。		
1		クラス/メンバパネルをオープンします。		
-	プロパティ	解析ツールの詳細情報(プロパティ)をプロパティ パネルに表示します。		



# プロパティ パネル

解析ツールの詳細情報の表示、および設定の変更を行います。

□ 全股	
解析結果を累積する	いいえ
強制的にクロス・リファレンス・ファイルを出力する	いいえ
プロトタイプ宣言を参照検索の対象とする	(tu)
プログラム停止時に更新を行う	はい
時間の単位	ns
引数を改行して表示する	いいえ
SFR/IORを変数として表示する	いいえ
定義箇所がない関数をコール・グラフの表示対象とする	いいえ
プログラム停止時に関数情報を出力する。	いいえ
□ インボート/エクスボート	
田 インポート・ファイル	インポート・ファイル[0]
ビルド時にエクスポートする	いいえ
🗆 解析グラフ 1	
表示範囲の上限/下限を設定する1	いいえ
有効範囲の上限/下限を設定する1	いいえ
凡例を表示する1	(まい)
- 凡例の表示位置1	左側
実行時間の割合グラフに表示する関数の数1	10
田 解析グラフ 2	
🗄 解析グラフ 3	
田 解析グラフ 4	

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



## [コンテキスト・メニュー]

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

## (1) 文字列編集中以外の場合

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

### (2) 文字列編集中の場合

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


# [設定] タブ

[設定] タブでは、次に示すカテゴリごとに詳細情報の表示、および設定の変更を行います。

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

🔍 プログラム解析 のプロパティ	
□ 全設	
解析結果を累積する	いいえ
登制的にクロス・リファレンス・ファイルを出力する	いいえ
プロトタイプ宣言を参照検索の対象とする	(\$6)
プログラム停止時に更新を行う	(\$()
時間の単位	ns
引数を改行して表示する	いいえ
SFR/IORを変数として表示する	いいえ
定義箇所がない関数をコール・グラフの表示対象とする	いいえ
プログラム停止時に関数情報を出力する	いいえ
□ インボート/エクスボート	
田 インポート・ファイル	インポート・ファイル[0]
ビルド時にエクスポートする	いいえ
日 解析グラフ 1	
表示範囲の上限/下限を設定する1	いいえ
有効範囲の上限/下限を設定する1	いいえ
凡例を表示する1	(\$()
凡例の表示位置1	左側
実行時間の割合グラフに表示する関数の数 1	10
団 解析グラフ 2	
田 解析グラフ 3	
団 解析グラフ 4	
全般	

### 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	- 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	なる項目は次のとおりです。					
	- 関数一覧 /	パネル						
	[実行回数]	[実行回数] / [実行時間 [ <i>単位</i> ]]						
	- 変数一覧 /	- 変数一覧 パネル						
	[リード回数	[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	レ・グラフ パネル						
	実行回数/	/リード回数/ライト回数						
	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					
	指定可能值	i はい 前回のプログラム実行による計測値に今回のプロ						
			の計測値を加算して表示します。					
		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	解析ツールが	 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						
レンス・ファイルを出力	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							
する	プロパティ設	パティ設定 <sup>注 1</sup> 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	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					
	指定可能值	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	関数の参照箇	」 i所の一覧表	- 示を行う際に(「2.11 参照箇所を一覧表示する」参照)、プロ					
検索の対象とする	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	[CA850][	[CA78K0R] [CA78K0]					
		変更不可						
		[CC-RX]	[CX]					
		ドロップ	<sup>8</sup> ダウン・リストによる選択					
	指定可能值	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	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						
を行う	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				
	指定可能值	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	各パネルのツールバーの 武 ボタンが有効となります(プロ				
		定する	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	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				
	指定可能值	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 パオ	→					
	デフォルト	いいえ					
	変更方法	ドロップダウン・リストによる選択					
	指定可能值	はい	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	変数一覧 パオ	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					
て表示する	表示するか否	かを指定しる	ます。				
	デフォルト	いいえ					
	変更方法	[CA850][	CX] [CA78KR0] [CA78K0]				
		ドロップ	ダウン・リストによる選択				
		[CC-RX]					
		変更不可					
	指定可能值	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				
コール・グラフの表示対 会とする	数もコール・	グラフに表 <del>;</del>	〒するか否かを指定します。				
* と 9 つ	デフォルト	いいえ					
	変更方法	ドロップダ	ウン・リストによる選択				
	指定可能值	はい	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。				

RENESAS

プログラム停止時に関数	STF 用の情報ファイル <sup>注 2</sup> を、ビルド・ツールのプロパティ パネルの[共通オプショ						
情報を出力する	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						
【∨850】	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	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					
	指定可能值						
		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
  - [CC-RX]

[コンパイル・オプション] タブ→ [その他] カテゴリ→ [クロス・リファレンス情報を出力する] プロパティ

- [CX]

[共通オプション] タブ→ [出力ファイルの種類と場所] カテゴリ→ [クロス・リファレンス情報を 出力する] プロパティ

- [CA850]

[クロス・リファレンス・オプション] タブ→ [クロス・リファレンス・ツール] カテゴリ→ [クロ ス・リファレンス・ツールを使用する] プロパティ

- 【CA78K0R】【CA78K0】 [コンパイル・オプション] タブ→ [リスト・ファイル] カテゴリ→ [クロスリファレンス・リス ト・ファイルを出力する] プロパティ
- 2. 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, 「2.12 情報ファイルをインポート/エクスポート する」を参照してください。



インポート・ファイル	インポートす	るファイルる	を指定します。						
	次のプレース	ホルダに対応	<b>なしています。</b>						
	%ProjectNam	ne% : プロ	コジェクト名に置換します。						
	%MicomTool	%MicomToolPath%: CubeSuite+ のインストール・フォルダの絶対パスに置換します。							
	相対パスでの	相対パスでの指定はプロジェクト・フォルダを基点とします。							
	なお、同一ファイルを指定した場合は、最初に指定したファイルのみをインポートします。								
	サブプロパテ	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 タ	「イアログは、このプロパティを選択すると欄内右端に表示さ						
		れる []	ボタンをクリックすることでオープンします(プロパティ パ						
		ネル上でイ	ンポート・ファイルを指定することはできません)。						
	指定可能值	64 個までの	りファイル						
ビルド時にエクスポート	ビルド/リビ								
する	情報ファイル	(関数一覧:	ファイル(*.mtfl)/変数一覧ファイル(*.mtvl))を生成(エク						
	スポート)す	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							
	指定可能值	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・	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						
ファイル名	定した場合の	み表示される	ます。						
	生成する情報	ファイル(闇	関数一覧ファイル(*.mtfl))の名前を指定します。拡張子						
	(*.mtfl)を変	更することは	tできません(拡張子を省略した場合は、自動的に mtfl を付与						
	します)。								
	次のプレース	ホルダに対応	<b>なしています</b> 。						
	%ProjectNam	ne%	プロジェクト名に置換します。						
	%ActiveProje	ctName%	アクティブ・プロジェクト名に置換します。						
	相対パスでの	指定はプロジ	ェクト・フォルダを基点とします。						
	なお、このブ	゚ロパティが堅	空欄の場合、情報ファイルは生成しません。						
	デフォルト	%ProjectNa	ame%.mtfl						
	変更方法	キーボード	からの直接入力						
	指定可能值	259 文字ま	での文字列						



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

### (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、「2.13 解析情報をグラフ化して表示する」を参照してください。

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

表示範囲の上限/下限を	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 <i>1 ~ 4</i>	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					
	指定可能值	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 <i>1 ~ 4</i>	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	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				
	指定可能值	- 10 進数指定の場合					
		-2147483648 ~ 4294967295					
		- 16 進数指定の場合("0x"の付与)					
		-0x80000000 ~ 0xFFFFFFF					



表示範囲の下限値 <i>1~4</i>	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] を指知	\$い]を指定した場合のみ表示されます。					
	表示するグラ	ラフにおいて、Y 軸の下限値を指定します。 コ					
	デフォルト	-2147483648					
	変更方法	キーボードからの直接入力					
	指定可能值	- 10 進数指定の場合					
		-2147483648 ~ 4294967295					
		- 16 進数指	音定の場合("Ox"の付与)				
		-0x8000	0000 ~ 0xFFFFFFF				
有効範囲の上限/下限を	解析グラフィ	ペネルの[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				
設定する <i>1 ~ 4</i>	するか否かを	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				
	指定可能值	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 <i>1~4</i>	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] を指え	官した場合の	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	た数値よりフ	<b>たきい値への推移はグラフに表示されません。</b>				
	デフォルト	214748364	7				
	変更方法	キーボードからの直接入力					
	指定可能值	- 10 進数指定の場合					
		-2147483648 ~ 4294967295					
		- 16 進数指定の場合("0x"の付与)					
		-0x80000000 ~ 0xFFFFFFF					
有効範囲の下限値 <i>1~ 4</i>	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]を指え	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	た数値よりハ	いさい値への推移はグラフに表示されません。				
	デフォルト	-214748364	48				
	変更方法	キーボード	からの直接入力				
	指定可能值	- 10 進数指	行定の場合				
		-214748	3648 ~ 4294967295				
		- 16 進数指	音定の場合("0x"の付与)				
		-0x8000	0000 ~ 0xFFFFFFF				
凡例を表示する 1 ~ 4	解析グラフィ	ペネルの[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				
	指定可能值	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 <i>1 ~ 4</i>	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	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				
	指定可能值	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	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						
表示する関数の数 <i>1 ~ 4</i>	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	 た法 キーボードからの直接入力					
	指定可能值	 					



# 関数一覧 パネル

取得した関数情報を表示します。

このパネルで表示対象となる関数の種類は次のとおりです。

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

なお、関数情報を表示するための操作手順は、「2.1 概 要」を参照してください。

### 注意 【CA850】【CA78K0R】【CA78K0】

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

	開発	一覧							
ツールバー]-	2	式 🍣 🛣 🛸 🔊	時間単位	±( <u>U)</u> ▼					
(1) —	P	関数名	\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	ファイル名	\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	属性 ▽中	戻り値の型 ▽	中 引数 ▽≠	• •
F	•	🗊 AD_Init		CG_ad.c		-	void	void	
		🖕 AD_Read		CG_ad.c		-	unsigned short	unsigned····	<b>T</b> =
		💊 AD_Start		CG_ad.c		-	void	void	-
		💊 AD_Stop		CG_ad.c		-	void	void	
		🖕 MD_INTAD		CG_ad_user.c		-	void	void	
(2) -		TMP0_Start		CG_timer.c		-	void	void	Ť.
(2)		TMP1_Start		CG_timer.c		-	void	void	Ť
		💊 ad_receive		(定義箇所なし)		-	-	-	
		🖕 func1		CG_main.c			void	void	Ť
		👽 func1a		CG_main.c		-	void	void	Ť
		👽 func2		CG_main.c		-	void	void	~
L	<							)	

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



### [各エリアの説明]

### (1) ヘッダ・エリア

#### (a)項目名

取得した関数情報の項目名を表示します。

なお、各項目名に表示されるマーク(アイコン)の意味は次のとおりです。

マーク(アイコン)	意味
┶╲┶	固定表示設定の有無を示します(「2.6.3 特定項目を固定表示に設定する」参照)。
$\Delta \checkmark \nabla$	ソート表示設定の有無を示します(「2.6.4 解析情報をソート表示する」参照)。
$\nabla \times \mathbf{Y}$	フィルタ表示設定の有無を示します(「2.6.5 解析情報をフィルタ表示する」参照)。
٩	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



#### (2) 情報表示エリア

取得した関数情報を表示します。

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

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

なお, プログラムの実行により値が変化した情報は強調表示されます(強調表示の際の文字色/背景色は, オプション ダイアログにおける [全般 - フォントと色] カテゴリの設定に依存)。強調表示を解除する場合は, ツールバーの 🚱 ボタンボタンをクリックしてください。

関数情報として取得する項目と内容は次のとおりです。



項目	種別	内容
関数名	静的	C/C++ ソース・ファイルで定義/参照されているグローバル関数/ファイル内
	解析	スタティック関数/メンバ関数(クラス型で定義されている関数)の名前を表示
		します。
		最大表示文字数は次のとおりです。
		- 【CC-RX】 8189 文字
		- 【CA850】 1022 文字
		- 【CX】
		- 【CA78K0R】【CA78K0】 249 文字
		なお、表示アイコンは次の意味を示します。
		- 📦 : 関数
		- =💊 : メンバ関数【CC-RX】
		- 《 <b>T</b> 》: テンプレート関数【CC-RX】
		[CC-RX]
		const メンバ関数 /volatile メンバ関数の場合は、関数名の直後に"const"/
		"volatile"を表示します。
クラス名	静的	関数が属しているクラス名を表示します。
[CC-RX]	解析	テンプレート・クラスの場合は、" <i>クラス名</i> <t: <i="">テンプレート引数の数 &gt;"の形</t:>
		式でテンプレート引数の数も表示します。
		ただし、関数がメンバ関数以外の場合、または解析未完の場合は空欄となりま
		す。
名前空間	静的	関数が属している名前空間名を表示します。
[CC-RX]	解析	名前空間がネストしている場合は、" <i>名前空間名</i> :: <i>名前空間名</i> "の形式で表示し
		ます。また、無名名前空間の場合は、" <unnamed>"を表示します。</unnamed>
		ただし、解析未完の場合は空欄となります。
ファイル名	静的	
	解析	ただし、プロジェクト内の C/C++ ソース・ファイルで定義されていない場合
		または解析未完の場合は゛"(定義箇所なし)"を表示します。
		  なお、ヘッダ・ファイルで定義されている場合は、該当ヘッダ・ファイル名を表
		示します。
	静的	
	解析	ただし、プロジェクト内の C/C++ ソース・ファイルで定義されていない場合
	711 1/1	または解析未完の場合は空欄となります。
		のためががかった。第二後二歳とようよう。
		パスを表示します。
 PF	静的	関数が実行される PE を次にように表示します
· ⊢ I⊟ +¤ [\/850F2]	解析	
	<u> </u>	- PF 共通で実行 Common
		- <b>T</b> H
		'2'    ただ , 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ョア対応版の



項目	種別	内容
インポート <sup>注1</sup>	静的	関数情報の取得先を次のように表示します。
	解析	- アクティブ・プロジェクト内から取得した場合
		"Original"を表示
		- インポート・ファイルから取得した場合
		インポート・ファイル名のすべてを表示
		- アクティブ・プロジェクト内とインポート・ファイルから取得した場合
		"Original"とインポート・ファイル名のすべてを表示
		なお、インポート機能については、「2.12 情報ファイルをインポート/エクス
		ポートする」を参照してください。
アクセス指定子	静的	メンバ関数に指定されているアクセス指定子を表示します。
[CC-RX]	解析	ただし、解析未完の場合は"-"を表示します。
		表示可能なアクセス指定子は次のとおりです。
		public, private, protected
属性	静的	関数のシンボル属性/シンボル修飾属性を表示します。
	解析	複数の属性が存在する場合は、","で区切り表示します。
		ただし、解析未完の場合は"-"を表示します。
		表示可能な属性は次のとおりです。
		- [CC-RX]
		static, interrupt, inline, template, virtual, abstract
		- [CA850]
		static
		- [CX]
		static, interrupt, inline, delete
		- [CA78K0R]
		static, callt, interrupt, near, far, rtos task, rtos interrupt
		- [CA78K0]
		static, callt, callf, noauto, norec, interrupt, bank, rtos task, rtos interrupt
戻り値の型	静的	関数の戻り値の型を表示します <sup>注 2</sup> 。
	解析	ただし、解析未完の場合は"-"を表示します。
		表示可能な最大ポインタ数は次のとおりです。
		- 【CC-RX】 :制限なし
		-【CA850】 :6 個
		-【CX】 :8個
		- 【CA78K0R】【CA78K0】 : 7 個
引数の数 <sup>注1</sup>	静的	
	解析	可変引数を持つ関数の場合は、関数の定義箇所に定義されている引数の数を表示
		します。
		ただし,解析未完の場合は"-"を表示します。

項目	種別	内容
引数	静的	関数の引数の型と仮引数名を表示します <sup>注 2</sup> 。
	解析	複数の引数が存在する場合は、","で区切り表示します。
		可変引数を持つ関数の場合は、関数の定義箇所に定義されている引数の型と引数
		名を表示します。
		引数が存在しない場合は"void"を表示します。
		ただし,解析未完の場合は"-"を表示します。
		表示可能な最大ポインタ数は次のとおりです(配列の 1 次元目はポインタとして
		扱います)。
		- 【CC-RX】 :制限なし
		-【CA850】 :6 個
		-【CX】 :8個
		- 【CA78K0R】【CA78K0】 :8 個
コード・サイズ[バイト]	静的	関数のコード・サイズを 10 進数で表示します。
	解析	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	関数のスタック・サイズを 10 進数で表示します。
[V850]	解析	ただし,解析未完の場合は"-"を表示します。
[RL78]		[CA78K0R] [CA78K0]
[78K0R]		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
【78K0】		保するスタック・サイズを表示するため、スタック見積もりツールが表示する
		スタック・サイズの値とは異なります。また、関数内の CALL/PUSH/POP 命
		令が使用するスタック・サイズは含みません。
開始アドレス	静的	関数の開始アドレスを 16 進数で表示します。
	解析	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 <sup>注1</sup>	静的	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。
	解析	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	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。
	解析	プロトタイプ宣言も参照として計数します。
		なお、C/C++ ソース・ファイル中の"#if"/"#ifdef"などで、コンパイル時に
		プリプロセッサにより除外されるコードにおいて参照されている箇所は参照回数
		に含みません(コンテキスト・メニューの [すべての参照を検索] による検索結
		果においても出力しません)。
		ただし,解析未完の場合は"-"を表示します。
		[CA78K0R] [CA78K0]
		"#pragma 指令"内に関数名を含んでいる箇所も参照回数に計数します。
		[CC-RX] [CX]
		関数ポインタへの代入による関数の参照は参照回数に含みません。



項目	種別	内容
実行回数	動的	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
[IECUBE]	解析	で表示します <sup>注3</sup> 。
【IECUBE2】		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
【シミュレータ】		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 [ <i>単位</i> ]	動的	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 <sup>注3,4</sup> 。
[IECUBE [V850]]	解析	なお. <i>単位</i> 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
[IECUBE [RL78]]		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
[IECUBE [78K0R]]		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
【IECUBE2】		単位]プロパティを参照してください)。
【シミュレータ】		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 )[%]	動的	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
[IECUBE [V850]]	解析	時間の割合を。0.00 ~ 100.00 の範囲で表示します <sup>注 3, 4</sup> 。
[IECUBE [RL78]]		なお、セル内背景色の比率は、割合値を示します。
[IECUBE [78K0R]]		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"-"
【IECUBE2】		を表示します。
【シミュレータ】		
平均実行時間 [ <i>単位</i> ]	動的	関数の平均実行時間(" <i>実行時間</i> " ÷ " <i>実行回数</i> ")を表示します <sup>注 3, 4</sup> 。
[IECUBE [V850]]	解析	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。
[IECUBE [RL78]]		なお. <i>単位</i> 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
[IECUBE [78K0R]]		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
【IECUBE2】		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
【シミュレータ】		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 [%]	動的	関数のコード・カバレッジ率(C0:" <i>実行されたアドレス範囲のコードのバイト</i>
[IECUBE]	解析	<i>数</i> " ÷ " <i>関数のコード・サイズ</i> " × 100)を表示します <sup>注 5</sup> 。
【IECUBE2】		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。
【シミュレータ】		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、「2.6.1 表示項目を設定 する」を参照してください。
  - 2. [CC-RX] [CA850] [CA78K0R] [CA78K0]

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

[CX]

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

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

- **3.** プロパティ パネルの [設定] タブ上の [全般] カテゴリ内 [解析結果を累積する] プロパティにおい て [はい] を選択している場合, プログラム実行ごとの累積による数値を表示します。
- 4. 実行時間には、コンパイラが用意しているランタイム・ライブラリの実行時間は含まれません。
- 5. コード・カバレッジ率は、プログラム実行ごとの累積による数値で計算します。 したがって、コード・カバレッジ率をリセットしたい場合は、エディタパネル/逆アセンブルパネ ルにおいて、コンテキスト・メニューの[カバレッジ情報のクリア]を選択したのち、ツールバーの 「」「ボタンをクリックしてください。また、プログラムを修正したのちビルドを実行した結果、各関数 の配置アドレスが前回のビルド時の配置アドレスと異なる場合、未実行の関数のコード・カバレッジ率 を表示します。

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。

関数名	▽中 ファイル名 1	マーコード・サイズ マー	参照回数 又中	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	0	4	1	0
合計値	CG_main.c	114	8	0	0

### 図 A—6 [合計値]の表示(関数一覧パネル)



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 <sup>注</sup>	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 [ <i>単位</i> ]	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
1			バレッジ率

- 注 この項目は、デフォルトでは表示されません。表示方法についての詳細は、「2.6.1 表示項目を設定す る」を参照してください。
- 注意 1. 【IECUBE【V850】】【IECUBE2】

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

- 2. 【IECUBE [RL78]】 [IECUBE [78K0R]] プログラム実行時の最初のトレース・データのタイム・タグには"0"が出力されます。 このため、実行→停止、またはステップ実行を繰り返し行った場合、[実行時間 [ 単位]] / [実行 時間(割合) [%]] / [平均実行時間 [ 単位]] 項目が不正な値となります。
- 【IECUBE【78K0】】
   次の項目は表示されません。
   [実行時間[単位]] / [実行時間(割合)[%]] / [平均実行時間[単位]]
- 4. 【CC-RX】【CX】 最適化により削除された未使用スタティック関数は一覧に表示されません。
- [CA78K0R] [CA78K0]
   同名のソース・ファイルがプロジェクトに登録されている場合、ビルド・ツールによりクロス・リファレンス情報が上書きされてしまうため、同名のソース・ファイルのうち最後にコンパイルされたソース・ファイルの情報以外を取得することができません。
- 6. システム・ライブラリ関数については、次の項目の値を取得することができません。
  [戻り値の型] / [引数の数] / [引数] / [コード・サイズ[バイト]] / [スタック・サイズ[バイト]] / [終了アドレス] / [実行時間[単位]] / [実行時間(割合)[%]] / [平均実行時間[単位]] / [コード・カバレッジ[%]]
- 7. デバッグ・ツールと切断時は、動的解析情報の項目は非表示となります(デフォルト)。

- **備考1.** ヘッダ・ファイル内に "static"の関数定義を記述して、1つ以上のソース・ファイルでインクルー ドしている場合は次のように表示します。
  - [CA850]
  - ヘッダ・ファイルの情報を1行に、すべてのソース・ファイルの情報を1行にまとめて表示します。
  - [CC-RX] [CX] [CA78K0R] [CA78K0]
  - ヘッダ・ファイルとすべてのソース・ファイルの情報を1行にまとめて表示します。
  - 2. [CC-RX]

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

```
11
// テンプレート関数を使用する関数
11
int templatet_use(void)
{
    short result = 0;
   short s = 100;
    char c = 200;
   result += template_func(s, c);
   return result;
}
11
// テンプレート関数の定義
//
template <typename T1, typename T2> T1 template_func(T1 t1, T2 t2)
{
   T1 result = 10;
   result += t1 + t2;
   return result;
}
```

例1. テンプレート関数の定義用の行に表示する内容

```
[戻り値の型]: -
[引数] : -
[引数] : Template_func<T:2>
2. テンプレート関数を使用している関数の行に表示する内容
[戻り値の型] : short
[引数] : short t1, char t2
[関数名] : Template_func<T:2>
```



- 3. 各情報の表示は、次のカスタマイズを行うことができます。
  - 解析情報をソート表示する
  - 解析情報をフィルタ表示する
- 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。

	2	最新情報を取得し、表示内容を更新します。
	<u>t</u>	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)。
<	8	プログラムの実行により値が変化したことを示す強調表示を解除します。
		ただし、プログラム実行中は無効となります。
341	2	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)。
	<u> </u>	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)。
Par l	<u> </u>	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
ľ	V850E2】	します(「(c) デバッグ・マネージャ パネルとの連携【V850E2】」参照)。
		ただし、選択しているマイクロコントローラがマルチコア対応版でない場合、またはデバッグ・ツー
		ルと切断時は無効となります。
民	5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	

### [[ファイル] メニュー (関数一覧 パネル専用部分)]

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

関数一覧データを保存	このパネルの内容を前回保存したファイルに保存します(「2.14 解析情報をファイル
	に保存する」参照)。
	なお、起動後に初めてこの項目を選択した場合は、 [名前を付けて 関数一覧データ を保
	存…]の選択と同等の動作となります。
名前を付けて 関数一覧データ を	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
保存	グをオープンします(「2.14 解析情報をファイルに保存する」参照)。
印刷	現在このパネルに表示している内容を印刷するために、Windows の印刷用 ダイアログ
	をオープンします。

### [[編集] メニュー (関数一覧 パネル専用部分)]

関数一覧パネル専用の[編集]メニューの各項目、および機能は次のとおりです。

コピー	選択している行の内容をタブ区切りの文字列としてクリップ・ボードにコピーします
	(複数行選択可)。
	なお、コピーした内容をこのパネル上に貼り付けることはできません。
すべて選択	このパネルに表示されているすべての行を選択状態にします。

### [コンテキスト・メニュー]

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

すべての参照を検索	選択している行(複数行選択可)の関数を参照している箇所を検索し,出力パネルの [参照の検索]タブ上に参照箇所一覧を表示します(「2.11 参照箇所を一覧表示する」 参照)。
関数の先頭にブレークを設定	カレント行の関数の先頭行(対象関数の先頭実行可能行)にブレークポイントを設定し ます(「2.9.1 関数にブレークポイントを設定する」参照)。 ただし、デバッグ・ツールと切断時は無効となります。
ソースヘジャンプ	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(「2.7 定義箇所へジャンプする」参照)。
逆アセンブルヘジャンプ	カレント行の関数の開始アドレスに対応する逆アセンブル・データを逆アセンブル パ ネル(逆アセンブル 1)上にオープンします(「2.7 定義箇所へジャンプする」参照)。 ただし、デバッグ・ツールと切断時は無効となります。
メモリヘジャンプ	カレント行の関数の開始アドレスに対応するメモリ・リストをメモリ パネル(メモリ 1)上にオープンします(「2.7 定義箇所へジャンプする」参照)。 ただし、デバッグ・ツールと切断時は無効となります。
コピー	選択している行(複数行選択可)の内容をタブ区切りの文字列としてクリップ・ボード にコピーします。 なお、コピーした内容をこのパネル上に貼り付けることはできません。



# 変数一覧 パネル

取得した変数情報を表示します。

- このパネルで表示対象となる変数の種類は次のとおりです。
  - グローバル変数
  - ファイル内スタティック変数
  - 関数内スタティック変数
  - IOR [RX] [V850]
  - SFR [RL78] [78K0R] [78K0]
  - クラス変数(C++ ソース・ファイルを対象とする場合)
- なお、変数情報を表示するための操作手順は、「2.1 概 要」を参照してください。

### 注意 【CA850】【CA78K0R】【CA78K0】

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

### 図 A—7 変数一覧 パネル

	変動一覧											
・ルバー]-	2 🕄 🧶 🛛	R 🔉 🕅										
(1) -	一 変数名	 	ファイル名	7₽	属性	\. \. \. \. \. \. \. \. \. \. \. \. \. \	型	\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	メンバマセ	アドレス	7₽	
Ē	🕨 🦻 g_ad_dat	a	CG_main.c		-		unsigned	short	-	0x03ff	6116	
	🔪 g_ad_fini	sh	CG_main.c		-		unsigned	char	-	0x03ff	Ь114	
	👽 g_count_	10ms	CG_main.c		-		unsigned	int	-	0x03ff	Ь110	
	💊 g_count_	1 ms	CG_main.c	1	-		unsigned	int	-	0x03ff	b10c	
	💊 g_flag_de	etect	CG_main.c	ŀ	-		unsigned	short	-	0x03ffl	Ь118	
	S_romp	1	CG_systemini	t.c			-		-		-	
	合計値		CG_main.c	 T	_		_		-			
	合計値		CG_systemini	t.c	y addation to the		_		-		-	
	<			ľ							>	

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー (変数一覧 パネル専用部分)]
- [[編集] メニュー (変数一覧 パネル専用部分)]
- [コンテキスト・メニュー]

### [オープン方法]

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

### [各エリアの説明]

### (1) ヘッダ・エリア

#### (a)項目名

取得した変数情報の項目名を表示します。

なお、各項目名に表示されるマーク(アイコン)の意味は次のとおりです。

マーク(アイコン)	意味
₽∕₽	固定表示設定の有無を示します(「2.6.3 特定項目を固定表示に設定する」参照)。
$\Delta \checkmark \nabla$	ソート表示設定の有無を示します(「2.6.4 解析情報をソート表示する」参照)。
$\nabla \times \mathbf{Y}$	フィルタ表示設定の有無を示します(「2.6.5 解析情報をフィルタ表示する」参照)。
٩	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



#### (2) 情報表示エリア

取得した変数情報を表示します。

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

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

なお、プログラムの実行により値が変化した情報は強調表示されます(強調表示の際の文字色/背景色は、 オプション ダイアログにおける [全般 - フォントと色] カテゴリの設定に依存)。強調表示を解除する場合は、 ツールバーの 🚱 ボタンボタンをクリックしてください。

変数情報として取得する項目と内容は次のとおりです。

項目	種別	内容	
変数名	静的		
	解析	タティック変数/関数内スタティック変数/クラス変数(クラス型の静的なメン	
		バ変数)の名前を表示します。	
		また, プロパティ パネルの[設定]タブの[全般]カテゴリ内[SFR / IOR を	
		変数として表示する]プロパティにおいて[はい]を選択している場合、プログ	
		ラム中で使用している SFR/IOR 名を表示します(【CC-RX】を除く)。	
		最大表示文字数は次のとおりです。	
		- 【CC-RX】 8189 文字	
		- 【CA850】 1022 文字	
		- 【CX】	
		- 【CA78K0R】【CA78K0】 :249 文字	
		なお、表示アイコンは次の意味を示します。	
		- 👽 🗄 変数	
		- 🧼 : メンバ変数【CC-RX】	
		- 🚺 : IOR [RX] [V850] /SFR [RL78] [78K0R] [78K0]	
		[CC-RX]	
		無名共用体を使用している場合 " <unnamed_n>"の形式で変数名を表示しま</unnamed_n>	
		す (N:1から出現順に自動的に付与される番号)。	
クラス名	静的	変数が属しているクラス名を表示します。	
[CC-RX]	解析	なお、テンプレート・クラスの場合は、" <i>クラス名</i> <t: <i="">テンプレート引数の数 &gt;"</t:>	
		の形式でテンプレート引数の数も表示します。	
		ただし、変数がメンバ変数以外の場合、または解析未完の場合は空欄となります。	
名前空間	静的	変数が属している名前空間名を表示します。	
[CC-RX]	解析	名前空間がネストしている場合は <i>、"名前空間名</i> :: <i>名前空間名</i> "の形式で表示し	
		ます。また、無名名前空間の場合は、" <unnamed>"を表示します。</unnamed>	
		ただし、解析未完の場合は空欄となります。	
ファイル名	静的	変数が定義されている C/C++ ソース・ファイル名を表示します (パスを除く)。	
	解析	ただし、プロジェクト内の C/C++ ソース・ファイルで定義されていない場合、ま	
		たは解析未完の場合は、"(定義箇所なし)"を表示します。	
		なお、ヘッダ・ファイルで定義されている場合は、該当ヘッダ・ファイル名を表	
		示します。	
		[CA78K0R] [CA78K0]	
		同一ファイル内で、extern 宣言を記述している行と変数の定義行が存在する場	
		合、変数の定義情報を取得することはできません。このため、この場合は"(定	
		義箇所なし)"と表示します。	
関数名 <sup>注1</sup>	静的	変数が定義されている関数の関数名を表示します。	
	解析	だたし、関数内スタティック変数以外の場合は空欄となります。	
		[CC-RX]	
		上記に加え,"()"内に関数の引数の型を列挙します。	
		また。const メンバ関数 /volatile メンバ関数の場合は、関数名の直後に"const"	
		/"volatile"を表示します。	



項目	種別	内容	
ファイル・パス <sup>注1</sup>	静的	変数が定義されている C/C++ ソース・ファイルの絶対パスを表示します。	
	解析	ただし、プロジェクト内の C/C++ ソース・ファイルで定義されていない場合、ま	
		たは解析未完の場合は空欄となります。	
		なお、ヘッダ・ファイルで定義されている場合は、該当ヘッダ・ファイルの絶対	
		パスを表示します。	
PE 情報	静的	変数に対してアクセス可能な PE を次にように表示します。	
【V850E2】	解析	- PEnからアクセス : PEn	
		- PE でアクセス : Common	
		- 不明 : -	
		ただし、この項目は、選択しているマイクロコントローラがマルチコア対応版の	
		場合のみ表示します。	
ィンポート <sup>注1</sup>	静的	変数情報の取得先を次のように表示します。	
	解析	- 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	
		なお、インボート機能については、「2.12 情報ファイルをインボート/エクス	
<u>.</u>	  ,	小一下9の」を変現ししくたさい。	
アクセス指定子	静的	メンバ変数に指定されているアクセス指定子を表示します。	
[CC-RX]	解析	ただし、解析未完の場合は"-"を表示します。	
		│表示可能なアクセス指定チは次のとおりです。 │	
	 	public, private, protected	
属性	静的	変数のシンボル属性/シンボル修飾属性を表示します。	
	解析	複数の属性が存在する場合は、","で区切り表示します。	
		ただし、解析未完の場合は"-"を表示します。	
		表示可能な属性は次のとおりです。 	
		- [CC-RX]	
		Static, const, volatile, restrict	
		- [CA850] static	
		static ior const volatile delete	
		static const volatile srea rwsfr rosfr, wosfr, near, far	
		- [CA78K0]	
		static const volatile sreg rwsfr rosfr wosfr	



項目	種別	内容	
型	静的	変数の型を表示します <sup><b>注2</b></sup> 。	
	解析	なお、"#define 文"/"typedef 文"により型に別名が付与されている場合、別	
		名ではなく、基となる型を表示します。	
		ただし、または解析未完の場合は"-"を表示します。	
		表示可能な最大ポインタ数は次のとおりです(配列は最大 4 次元まで表示可)。	
		- 【CC-RX】 :制限なし	
		-【CA850】 :6 個	
		-【CX】 :8個	
		- 【CA78K0R】【CA78K0】 :8 個	
メンバ	静的	構造体/共用体のメンバを表示します。	
	解析	複数のメンバが存在する場合は、","で区切り表示します。	
		ただし、構造体/共用体以外の場合、または解析未完の場合は"-"を表示しま	
		す。	
アドレス	静的	変数の配置アドレスを 16 進数で表示します。	
	解析	表示桁数は選択しているマイクロコントローラの最大アドレス値と同等です。	
		ただし、解析未完の場合は"-"を表示します。	
サイズ[バイト]	静的	変数のサイズを 10 進数で表示します。	
	解析	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。	
参照回数	静的	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。	
	解析	変数の定義箇所も参照として計数します。また、構造体/共有体/配列の場合は、	
		変数単位で計数します(メンバごとや要素ごとの参照回数は非表示)。	
		なお、C/C++ ソース・ファイル中の"#if"/"#ifdef"などで、コンパイル時に	
		プリプロセッサにより除外されるコードにおいて参照されている箇所は参照回数	
		に含みません(コンテキスト・メニューの[すべての参照を検索]による検索結	
		果においても出力しません)。	
		ただし、解析未完の場合は"-"を表示します。	
		[CC-RX] [CX]	
		変数定義行において、代入文が記述されている行("int variable = 10"など)も	
		参照回数として計数します。	
		また, "variable++;"を記述している行は "variable = variable + 1"と解釈する	
		ため、参照回数は2回と計数します。	
リード回数	動的	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 <sup>注3,4</sup> 。	
[IECUBE]	解析	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	
[IECUBE2]		(メンバごとや要素ごとのリード回数は非表示)。	
【シミュレータ】		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	動的	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 <sup>注3、4</sup> 。	
[IECUBE]	解析	↓ ∬ 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	
[IECUBE2]		(メンバごとや要素ごとのライト回数は非表示)。	
【シミュレータ】		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
リード/ライト回数 <sup>注1</sup>	動的	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
[IECUBE]	解析	します <sup>注3, 4</sup> 。
【IECUBE2】		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
【シミュレータ】		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	動的	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 <sup>注 3</sup> 。
[IECUBE]	解析	ただし,bit 型 /boolean 型 /Bool 型/構造体/共用体/配列/ポインタの場合,デ
[IECUBE2]		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
【シミュレータ】		す。
		[CC-RX] [CA850] [CX]
		4バイト以下の変数、または IOR のみ表示可能です。
		[CA78K0R] [CA78K0]
		2バイト以下の変数、または SFR のみ表示可能です。
最大値	動的	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 <sup>注 3</sup> 。
【IECUBE】	解析	ただし、bit 型 /boolean 型 /Bool 型/構造体/共用体/配列/ポインタの場合、デ
【IECUBE2】		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
【シミュレータ】		す。
		[CC-RX] [CA850] [CX]
		4バイト以下の変数、または IOR のみ表示可能です。
		[CA78K0R] [CA78K0]
		2バイト以下の変数,または SFR のみ表示可能です。
データ・カバレッジ [%]	動的	変数のデータ・カバレッジ率(" <i>アクセス6れたアドレス範囲のバイト数</i> "÷" <i>変数</i>
[IECUBE [RL78]]	解析	<i>のサイズ</i> ゛× 100)を表示します <sup>注 5</sup> 。
[IECUBE [78K0R]]		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。
[IECUBE [78K0]]		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
【シミュレータ】		"-"を表示します。

- **注1.** この項目は、デフォルトでは表示されません。表示方法についての詳細は、「2.6.1 表示項目を設定 する」を参照してください。
  - [CC-RX] [CA850] [CA78K0R] [CA78K0]
     "#define 文"または "typedef 文"により型に別名が付与されている場合、別名ではなく、基となる型を表示します。
     [CX]
     "typedef 文"により型に別名が付与されている場合、次の型を表示します。

typedef の基の型	表示する型
- unsigned long	unsigned long
- unsigned int	



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

- **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。
- 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、ツールバーのご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	▽中 ファイル名	マヨ サイズDバイト]	▽口 参照回数	▽⊣リード回数	▽⊣ ライト回数	γÞ
合計値	CG_main.c		13	9	0	- 4
合計値	CG_systeminit	.C	0	1	0	0



表示	種別	項目	内容
合計値	_	ファイル名	対象ファイル名
		ファイル・パス <sup>注</sup>	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		対象ファイル内変数からのリード回数の合計値
ライト		ライト回数	対象ファイル内変数へのライト回数の合計値
		リード/ライト回数	対象ファイル内変数へのリード/ライト回数の合計値
		データ・カバレッジ (%)	対象ファイル内変数のファイル単位でのデータ・カバレッジ率

- **注** この項目は、デフォルトでは表示されません。表示方法についての詳細は、「2.6.1 表示項目を設定す る」を参照してください。
- 注意 1. bit 型変数 /boolean 型 /Bool 型変数に対する [リード回数] / [ライト回数] / [リード/ライト 回数] / [データ・カバレッジ [%]] 項目の計測は、バイト単位で行います(変数が割り当てられ ているアドレスへのアクセスを計測)。このため、同一アドレスに割り当てられている bit 型変数 / boolean 型 /Bool 型変数に対するこれらの項目の値は同じ値となります。
  - [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。
  - [CC-RX] [CX]
     最適化により削除された未使用スタティック変数は一覧に表示されません。
  - 【CA78K0R】【CA78K0】
     同名のソース・ファイルがプロジェクトに登録されている場合、ビルド・ツールによりクロス・リファレンス情報が上書きされてしまうため、同名のソース・ファイルのうち最後にコンパイルされたソース・ファイルの情報以外を取得することができません。
  - 5. デバッグ・ツールと切断時は、動的解析情報の項目は非表示となります(デフォルト)。
- 備考1. 同一関数内に同名の関数内スタティック変数が宣言されている場合の扱いは次のとおりです。 【CA850】【CA78K0R】【CA78K0】
  - [変数名] / [ファイル名] / [関数名] / [ファイル・パス] / [インポート] / [属性] 関数内で最初に宣言された変数の情報を表示します。
  - [参照回数] 関数内で宣言された変数のすべての参照回数を表示します。
  - 上記以外の項目

関数内で最後に宣言された変数の情報を表示します。

[CC-RX] [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。

RENESAS

- 上記以外の項目

関数内で最後に宣言された変数の情報を表示します。

- **2.** ヘッダ・ファイル内に "static"の変数定義を記述して、1つ以上のソース・ファイルでインクルードしている場合は次のように表示します。
  - (CA850)
  - ヘッダ・ファイルの情報を1行に、すべてのソース・ファイルの情報を1行にまとめて表示します。
  - 【CC-RX】【CX】 ヘッダ・ファイルとすべてのソース・ファイルの情報を1行にまとめて表示します。 また、型情報は"?"と表示します。
  - 【CA78K0R】【CA78K0】 ヘッダ・ファイルとすべてのソース・ファイルの情報を1行にまとめて表示します。 また,型情報は"-"と表示します。
- 3. 各情報の表示は、次のカスタマイズを行うことができます。
  - 解析情報をソート表示する
  - 解析情報をフィルタ表示する
- 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
  - 変数値の推移をグラフ化する

### [ツールバー]

ツールバー上の各ボタン、および機能は次のとおりです。

22	最新情報を取得し、表示内容を更新します。
4	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)。
۲	プログラムの実行により値が変化したことを示す強調表示を解除します。
	ただし、プログラム実行中は無効となります。
飘	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)。
<b>3</b> 55	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)。
20	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
【V850E2】	す(「(c) デバッグ・マネージャ パネルとの連携【V850E2】」参照)。
	ただし、選択しているマイクロコントローラがマルチコア対応版でない場合、またはデバッグ・ツールと
	切断時は無効となります。

### [[ファイル] メニュー (変数一覧 パネル専用部分)]

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

変数一覧データを保存	このパネルの内容を前回保存したファイルに保存します(「2.14 解析情報をファイ
	ルに保存する」参照)。
	なお、起動後に初めてこの項目を選択した場合は、[名前を付けて 変数一覧データ を
	保存…]の選択と同等の動作となります。
名前を付けて 変数一覧データ を保	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
存	グをオープンします(「2.14 解析情報をファイルに保存する」参照)。
印刷	現在このパネルに表示している内容を印刷するために、Windows の印刷用 ダイアロ
	グをオープンします。

### [[編集] メニュー (変数一覧 パネル専用部分)]

変数一覧パネル専用の[編集]メニューの各項目、および機能は次のとおりです。

コピー	選択している行の内容をタブ区切りの文字列としてクリップ・ボードにコピーします
	(複数行選択可)。
	なお、コピーした内容をこのパネル上に貼り付けることはできません。
すべて選択	このパネルに表示されているすべての行を選択状態にします。

### [コンテキスト・メニュー]

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

す	べての参照を検索	選択している行(複数行選択可)の変数を参照している箇所を検索し、出力パネルの [参照の検索]タブ上に参照箇所一覧を表示します(「2.11 参照箇所を一覧表示す
		◎」奓照)。
解	析グラフ1に登録	選択している行(複数行選択可)の変数を解析グラフ パネル(解析グラフ 1)に登録
		します(「2.13.1 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。
7	クセス・ブレークの設定	ブレーク関連のイベント <sup>注</sup> を設定するために、次のカスケード・メニューを表示しま
		す(「2.9.2 変数にブレーク・イベントを設定する」参照)。
		ただし、デバッグ・ツールと切断時は無効となります。
	変数に読み込みブレークを設定	カレント行の変数に、リード・アクセスのブレーク・イベントを設定します。
	変数に書き込みブレークを設定	カレント行の変数に、ライト・アクセスのブレーク・イベントを設定します。
	変数に読み書きブレークを設定	カレント行の変数に、リード/ライト・アクセスのブレーク・イベントを設定します。
ゥ	オッチ1に登録	選択している行(複数行選択可)の変数をウォッチ式としてウォッチ パネル(ウォッ
		チ 1)に登録します(「2.10 ウォッチ式に登録する」参照)。
		ただし、デバッグ・ツールと切断時は無効となります。
ソースヘジャンプ		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(「2.7 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(「2.7 定義箇所へジャンプする」参照)。
	ただし、デバッグ・ツールと切断時は無効となります。
コピー	選択している行の内容をタブ区切りの文字列としてクリップ・ボードにコピーします
	(複数行選択可)。
	なお、コピーした内容をこのパネル上に貼り付けることはできません。

### 注 【RX】

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



# コール・グラフ パネル

関数間の呼び出し関係をツリー構造の図(コール・グラフ)で表示します。

このパネルで表示対象となる関数/変数の種類は次のとおりです。

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

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

- 注意 1. デバッグ・ツールがトレース機能をサポートしていない場合、またはデバッグ・ツールのトレース機能を有効 化していない場合、コール・グラフにおいて、動的解析情報(実行回数/リード回数/ライト回数)を表示す ることはできません。
  - 2. 【CA850】【CA78K0R】【CA78K0】 ビルド・ツールにおいてクリーンを実行すると、現在表示しているコール・グラフは消失します。
  - 【CC-RX】
     C++ ソース・ファイルを対象としている場合、暗黙的に呼び出されるクラス型のコンストラクタ/デストラクタの呼び出しは、コール・グラフに表示されません。
- 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(全体表示)

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー (コール・グラフ パネル専用部分)]
- [[編集] メニュー (コール・グラフ パネル専用部分)]
- [コンテキスト・メニュー]

# [オープン方法]

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



## [各エリアの説明]

(1) パネル・コントロール・エリア

.ma	   パネルを直接ドラッグすることにより、素示内容のスクロールを注示するか不か
	を設定しより (トクル)。 ノノオルト ではスクロールを計可しません。
	クラフ内の関数ホックス/変数ホックスをクリックしても、その関数/変数を強
	調表示(選択状態)することはできません( 対象関数名] コンボ・ボックスにも
	反映されません)。また、関数/変数の情報をボッブアップ表示することもできま 
	せん。
·	コール・グラフのズーム率を変更します。
	10 ~ 109 の範囲の数値を選択することができます(デフォルトでは 100 が指定さ
	れます)。
	コール・グラフのズーム率を変更します。
	各ボタンを1回クリックすると、ズーム・スライダの値が1つ減少/増加します。
	コール・グラフのズーム率を自動で変更します。
	   パネルのサイズに合わせて、コール・グラフの全体サイズを縮小/拡大します。
Q	コール・グラフのズー / 恋をデフォルトの 100 にリセットレます
main 🖌	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
[親関数名]コンボ・ボックス	ロップダウン・リストにより選択します <sup>た</sup> 。
(コンボ・ボックス左)	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")。
対象関数名 🖌 🖌	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
[対象関数名] コンボ・ボックス	より選択します。
(コンボ・ボックス右)	- [親関数名]コンボ・ボックスが空欄の場合
	全体表示: <i>プログラム中に存在するすべての関数名</i>
	詳細表示: <i>現在表示対象となっているすべての関数名</i>
	- [親関数名]コンボ・ボックスで親関数を指定している場合
	全体表示: <i>対象親関数から呼び出されている関数名</i> (子関数/孫関数~を含む)
	詳細表示: <i>現在表示対象となっているすべての関数名</i>
	   なお、コール・グラフ内の任意の関数ボックスをクリックすることでも該当関数
	    が強調表示(選択状態)され,該当関数名がこのコンボ・ボックスに反映されま
	す(
	強調表示(選択状態)されて   sub01 c (49)
	いる関数ボックス くstatic>



1 m	現在選択している関数([対象関数名] コンボ・ボックスで指定している関数)に
	対する親関数と子関数を表示する詳細表示にコール・グラフを切り替えます(ト
	グル)。
	ただし、プログラム実行中は無効となります。

注 【V850E2】

選択しているマイクロコントローラがマルチコア対応版の場合では、"PEn"を指定することができま す。この場合、該当 PEn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。 なお、コール・グラフは、 W ボタンのクリック(トグル)により、次の2つの表示モードに切り替わります。

#### - 全体表示(デフォルト)

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

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

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

注【V850E2】

選択しているマイクロコントローラがマルチコア対応版で, [親関数名] コンボ・ボックスにおいて "PE n"を指定した場合では, 該当 PEn で実行される関数のみを対象としたコール・グラフを表示 します。

- 詳細表示

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

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

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

備考 次のいずれかの場合、コール・グラフは全体表示に切り替わります。

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





コール・グラフの構成要素の詳細は次のとおりです。

図 A-11 コール・グラフの構成要素





#### (a) 関数ボックス

関数をボックス形式で表示します。 各関数ボックスに表示する情報は次のとおりです。



図 A-12 関数ボックスの情報

### 関数名 対象関数名を示します。 [CC-RX] - 対象関数がオーバーロードされている場合、またはテンプレート関数の場合は、 "()"内に引数の型を列挙します。 - 対象関数が const メンバ関数 /volatile メンバ関数の場合は、関数名の直後に "const" / "volatile"を表示します。 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。 定義箇所 ただし、定義箇所情報が存在しない場合は、"(定義箇所なし)"を表示します。 属性 対象関数の属性を示します。 - スタティック関数: <static> - テンプレート関数: <template>【CC-RX】 - 仮想関数 <virtual> [CC-RX] - 純粋仮想関数 <abstract> [CC-RX] 上記以外の場合は空欄となります。 対象関数の現在の実行状態を次の背景色で示します。 状態 - 水色:未実行 -紫色 実行済 PE 情報 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 [V850E2] 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(「2.7 定義箇所へジャンプする」参照)。 ただし、 🦛 ボタンの設定により、マウスのドラッグによるスクロールを許可している場合はこ の機能を使用することはできません。この場合は、「対象関数名」コンボ・ボックスにより対象関 数を選択したのち、コンテキスト・メニューの〔ソースへジャンプ〕を選択してください。
#### (b)変数ボックス

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

各変数ボックスに表示する情報は次のとおりです。



図 A—13 変数ボックスの情報

変数名	対象変数名を示します。	
	なお、対象変数が関数内スタティック変数の場合は、" <i>変数名 # 関数名</i> "の形式で	
	表示します。	
	対象亦教が完美されている笛託を"ファイルタ(行数)"で示します	
た我回加		
禹性		
	- スタティック変数: static>	
	- 関数内スタティック変数: <static local=""></static>	
	上記以外の場合は空欄となります。	
参照回数	対象変数を静的に参照している回数を示します。	
リード回数,ライト回数	対象変数を動的に参照(R: リード, W: ライト)している回数を示します。	
	ただし、トレース・データが存在する場合のみ表示します。	
	なお、プロパティ パネルの[設定]タブ上の[全般]カテゴリ内[解析結果を累	
	積する]プロパティにおいて[はい]を選択している場合.プログラム実行ごとの	
	累積による数値を表示します。	
状態	対象変数の現在の使用状態を次の背景色で示します。	
	- 緑色:未使用	
	- 赤紫:使用済	
PE 情報	選択しているマイクロコントローラがマルチコア対応版の場合、対象変数が割り付	
[V850E2]	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 変数ボックスをダブルクリックすることにより、該当変数が定義されているソース・テキスト箇 所へジャンプすることができます(「2.7 定義箇所へジャンプする」参照)。

RENESAS

ただし、 「「ボタンの設定により、マウスのドラッグによるスクロールを許可している場合はこの機能を使用することはできません。この場合は、スクロールの許可をいったん解除してから操作を行ってください。

(c)接続線

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

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

各接続線に表示する情報は次のとおりです。

**備考** 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-14 接続線の情報

参照回数	対象関数を静的に呼び出している回数を示します。		
実行回数	対象関数を動的に呼び出している回数を"()"内に示します。		
	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。		
	combination01_caller main.c (238)		



循環呼び出し	たとえば、3つの関数としてA、B、Cがあり、A→B→C→Aと呼び出している場合			
	では, "A → B"と"B → C"についての接続線は表示しますが, "C → A"についての接			
	続線は表示せず、循環していることを示す次の線分のみを表示します。			
	なお、循環呼び出しとなった関数についての情報は、ポップアップ表示で確認すること			
	ができます。			
	circular02 caller05			
	v main.c (97) (→			

(d) ポップアップ表示

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

表示形式は次のとおりです。

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

- 関数ボックス



#### 関数名 : *関数名* 場所 : *ファイル名 (行番号*)

② ファイルの絶対パス
 再帰呼び出し:参照回数(実行:実行回数)
 循環呼び出し
 *関数名:参照回数(*実行:*実行回数*)
 *関数名:参照回数(*実行:*実行回数*)
 …
 動的呼び出し
 *関数名:回数 関数名:回数* …



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

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

- 変数ボックス





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

変数名	対象変数名を示します。
	なお、対象変数が関数内スタティック変数の場合は、 " <i>関数名#変数名</i> "の形式で表示します。
	[CC-RX]
	- グローバル変数/ファイル内スタティック変数の場合
	名前空間(グローバル名前空間/無名名前空間を除く)に属する場合は, " <i>名前空間名</i> :: <i>変</i>
	<i>数名</i> "の形式で変数名を表示します。
	- クラス変数の場合
	名前空間(グローバル名前空間/無名名前空間を除く)に属する場合は、" <i>名前空間名</i> :: ク
	<i>ラス名</i> :: <i>変数名</i> "の形式で、属さない場合は " <i>クラス名</i> :: <i>変数名</i> "の形式で変数名を表示
	します。
	- 関数内スタティック変数の場合:
	名前空間(グローバル名前空間/無名名前空間を除く)に属する場合は、" <i>名前空間名</i> :: <i>関</i>
	<i>数名# 変数名</i> "の形式で、属さない場合は <i>" 関数名# 変数名</i> "の形式で変数名を表示しま
	す。また、クラス変数の場合は、" <i>クラス名</i> :: <i>関数名 # 変数名</i> "の形式で表示します。
場所	対象変数が定義されている箇所を示します。

# [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。

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

## [[ファイル] メニュー (コール・グラフ パネル専用部分)]

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

コール・グラフ・データを保存	このパネルの内容を前回保存したファイルに保存します(「2.14 解析情報をファイル		
	に保存する」参照)。		
	なお、起動後に初めてこの項目を選択した場合は、 [名前を付けて コール・グラフ・		
	データ を保存 …]の選択と同等の動作となります。		
名前を付けて コール・グラフ・	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		
データ を保存	グをオープンします(「2.14 解析情報をファイルに保存する」参照)。		

RENESAS

## [[編集] メニュー (コール・グラフ パネル専用部分)]

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

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

## [コンテキスト・メニュー]

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

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



# クラス/メンバ パネル

クラス情報【CC-RX】/関数情報/変数情報をツリー形式で表示します。

このパネルで表示対象となる関数/変数の種類は次のとおりです。

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

なお、クラス情報【CC-RX】/関数情報/変数情報を表示するための操作手順は、「2.5 クラス/関数/変数の情報 を表示する」を参照してください。

注意 1. 【CC-RX】

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

2. 【CA850】【CA78K0R】【CA78K0】 ビルド・ツールにおいてクリーンを実行すると、現在このパネルに表示している内容をすべてクリアしま す。

	クラス/メンバ 🛛
[ツールバー]-	
(1) -	カテゴリ・グループ表示 🛛 💽 💿 対象名を入力してくだ 💵 🖌 🔪
(2) -	■ 別名定義 (typedef) ● グローバル関数と変数 ● ■ 名前空間 ● ■ クラス ● ■ 構造体 ■ ■ 共用体 ■ ■ 列挙体
(3) —	<pre>=w tempfunc02<t:2>0 =w tempfunc02<t:2>(char, short) =w tempfunc02<t:2>(short, int) =w templatetest01 (void) =w templatetest02(void) </t:2></t:2></t:2></pre>
	<ul> <li>AAA</li> <li>Fixed_Vectors</li> <li>_BTBL</li> <li>CTBI</li> </ul>

#### 図 A—15 クラス/メンバパネル



- ここでは、次の項目について説明します。
  - [オープン方法]
  - [各エリアの説明]
  - [ツールバー]
  - [[編集] メニュー (クラス/メンバ パネル専用部分)]
  - [コンテキスト・メニュー]

# [オープン方法]

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

## [各エリアの説明]

カテゴリ・グループ表示 🗸 🗸	ツリーの分類方法を次のドロップダウン・リストより選択します。
[ビュー設定] コンボ・ボックス	- カテゴリ・グループ表示(デフォルト)
(コンボ・ボックス左)	- アクセス・グループ表示
	- 名前空間グループ表示
	- ファイル・グループ表示
	- アルファベット・グループ表示
	1 つ前に選択したノードを選択状態にします。
	ただし、以前に選択したノードの履歴が存在しない場合、またはプログラム実行
	中は無効となります。
$\bigcirc$	ボタンをクリックする前に選択したノードを選択状態にします。
	ただし、 💽 ボタンにより選択したノードの履歴が存在しない場合、 またはプロ
	グラム実行中は無効となります。
(対象名を入力してくだ ₩ 🗸	「人」ボタンによる検索を行う際の検索対象の文字列を指定します(大文)
[対象名] コンボ・ボックス	字/小文字不問)。
(コンボ・ボックス右)	キーボードより文字列を直接入力するか(最大指定文字数:512 文字)、ドロップ
	ダウン・リストより入力履歴項目を選択します(最大履歴数:10 個)。
	ただし、プログラム実行中は無効となります。
	ている文字列を含むノードを上方向に検索し、検索結果を選択状態にします。
	- ただし, [対象名] コンボ・ボックスが空欄の場合, またはプログラム実行中は無
	効となります。
>	直前にフォーカスのあったツリーに対して、[対象名]コンボ・ボックスで指定し
-	ている文字列を含むノードを下方向に検索し、検索結果を選択状態にします。
	ただし、[対象名]コンボ・ボックスが空欄の場合、またはプログラム実行中は無
	効となります。

(1) パネル・コントロール・エリア

- 備考 検索対象の文字列入力後, [Enter] キーを押下することにより, 💽 ボタンのクリックと同等の動作を 行い, [Shift] + [Enter] キーを押下することにより, 🕥 ボタンのクリックと同等の動作を行いま す。
- (2) クラス/カテゴリ用ツリー・エリア

プログラムで定義されているクラス情報などをツリー形式で表示します。

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

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

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

ツリーで表示するノード名とその順番は次のとおりです。

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

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

(a)カテゴリ・グループ表示(デフォルト)

クラスやインタフェースなどの種類で分類して表示します。

ノード	備考
局 別名定義 (typedef)	
ਡ❤ グローバル関数と変数	
■ マクロと定数	[CA78K0R] [CA78K0]
☑ 名前空間	[CC-RX]
	[CC-RX]
基底型	
基底型(クラス/インタフェース/ 構造体)名	
派生型	
派生型(クラス/インタフェース/ 構造体)名	
内部型	
<i>内部型(クラス/構造体/共用体/列挙体)名</i>	



ノード	備考
🗁 インタフェース	[CC-RX]
<i>⊷0 インタフェース名</i> 注1	
基底型	
基底型(クラス/インタフェース/ 構造体)名	
派生型	
派生型(クラス/インタフェース/ 構造体)名	
内部型	
内部型(クラス/構造体/共用体/列挙体)名	
□ 構造体	
☆ <i>構造体名</i> 注 1, 2	
□ 共用体	
□ 列挙体	
<i>副 列举体</i> 名	

注1. 【CC-RX】

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

アクセス指定子	アイコン
public	なし
protected	9
private	8

2. [CC-RX]

必要に応じて、基底型ノード/派生型ノード/内部型ノードを表示します。

3. [CC-RX]

必要に応じて、内部型ノードを表示します。

#### (b) アクセス・グループ表示

クラスなどに設定されたアクセス指定子で分類して表示します。 なお、同じ種類のノードはアルファベット順で表示します。

ノード	備考
局 別名定義 (typedef)	
■♥ グローバル関数と変数	
マクロと定数	[CA78K0R] [CA78K0]
public	[CC-RX]
☆ クラス名 <sup>注1</sup>	
<i>⊶ インタフェース名</i> 注1	
i 列挙体名	



ノード	備考
protected	[CC-RX]
ペ クラス名注1	
<i>⊶</i> 0 <i>インタフェース名</i> 注1	
■ 列挙体名	
🗁 private	[CC-RX]
ペ クラス名注1	
<i>⊶</i> 0 <i>インタフェース名</i> 注1	
↔ 共用体名注 2	
■ 列挙体名	

注1. 【CC-RX】

必要に応じて、基底型ノード/派生型ノード/内部型ノードを表示します。

【CC-RX】
 必要に応じて、内部型ノードを表示します。

#### (c) 名前空間グループ表示

クラスなどが定義されている名前空間で分類して表示します。

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

ノード	備考
☞ 名前空間名注1	
ਡ 別名定義 (typedef)	
ਡਂ❤ グローバル関数と変数	
■ マクロと定数	[CA78K0R] [CA78K0]
☆ クラス名注2	[CC-RX]
<i>⊷● インタフェース名</i> 注2	[CC-RX]
<i>副 列举体</i> 名	

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

必要に応じて、基底型ノード/派生型ノード/内部型ノードを表示します。

3. [CC-RX]

必要に応じて、内部型ノードを表示します。

(d)ファイル・グループ表示

クラスなどが定義されているファイルで分類して表示します。

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

ノード	備考
ファイル名注1	
ਡ 別名定義 (typedef)	
ਡ️❤ グローバル関数と変数	
▶ マクロと定数	[CA78K0R] [CA78K0]
ペ クラス名注 2	[CC-RX]
<i>⊶0 インタフェース名</i> 注2	[CC-RX]
∻ 共用体名注3	
<i>■ 利挙体</i> 名	

注1. ファイルの種類により付与されるアイコンは異なります。

- 2. 【CC-RX】 必要に応じて,基底型ノード/派生型ノード/内部型ノードを表示します。
- 【CC-RX】
   必要に応じて、内部型ノードを表示します。
- (e) アルファベット・グループ表示

クラスやアクセス指定子に関与せず、アルファベット順で表示します。

ノード	備。	
局 別名定義 (typedef)		
ਡ♥ グローバル関数と変数		
📻 マクロと定数	[CA78K0R] [CA78K0]	
☆ クラス名注1	[CC-RX]	アルファベット順に表示
<i>⊶0 インタフェース名</i> 注1	[CC-RX]	
<i>3 列挙体</i> 名		

注1. 【CC-RX】

必要に応じて、基底型ノード/派生型ノード/内部型ノードを表示します。

- [CC-RX]
   必要に応じて、内部型ノードを表示します。
- 注意 別名定義 (typedef) ノード/グローバル関数と変数ノード/マクロと定数ノードは、関数/変数用リ スト・エリアに表示する情報が存在しない場合は表示されません。
- 備考 1. 名前空間名ノードについて
  - 内部に名前空間を含む場合, "包括する名前空間名:: 内部の名前空間名"を表示します。
  - 例 namespace Name : Name

内部の名前空間の場合: Name::SubName

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

クラス/カテゴリ用ツリー・エリアで選択しているノードに関する情報(該当ノードで定義されている関数 名/変数名など)を一覧表示します。

クラス/カテゴリ用ツリー・エリアで選択しているノードと、このエリアで表示する内容の関係は次のとお りです。

選択ノード	このエリアでの表示内容	備考
🗊 別名定義 (typedef)	🔓 別名定義名	
意♥ グローバル関数と変数	=>> グローバル関数名	
	≕❤ スタティック関数名	
	▼ グローバル変数名	
	💟 ファイル内スタティック変数名	
📻 マクロと定数	マクロ名	[CA78K0R] [CA78K0]
	∃ 定数	



選択ノード	このエリアでの表示内容	備考
🔧 クラス名	🗅 別名定義名	[CC-RX]
<i>⊷</i> 0 インタフェース名	=🗣 メンバ関数名	
☆ 構造体名 (C++ ソース・ファイル )	💡 クラス変数名	
	💡 インスタンス変数名	
	∃ 定数	
<> <i>共用体</i> 名 (C++ ソース・ファイル )	🔁 別名定義名	[CC-RX]
	= 🗣 メンバ関数名	
	💡 インスタンス変数名	
	□ 定数	
🧇 <i>構造体名</i> (C ソース・ファイル)	<ul><li> メンバ変数名 </li></ul>	
< <i>共用体</i> 名 (C ソース・ファイル )	<ul><li> メンバ変数名 </li></ul>	
<i>計列挙体</i> 名	🚽 列挙体メンバ名	
{} 名前空間名	なし	
🧧 ファイル名		
上記以外		

注意 表示対象となる関数/変数が存在しない場合、またはクラス/カテゴリ用ツリー・エリアでノードを 選択していない場合は、このエリアは何も表示されません。

#### 備考1. 【CC-RX】【CX】

関数名には、引数情報を付与して表示します。

- 例 int main(void): main(void) void main\_sub01(int local\_a, int local\_b, int local\_c): main\_sub01(int, int, int) template<class T> T max(T a, T b): max<T>(T, T)
- 2. [CC-RX]

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

ただし、ツリーの分類方法をアルファベット・グループ表示としている場合は、すべてアルファ ベット順となります。

アクセス指定子	アイコン
public	なし
protected	9
private	3

3. [CC-RX]

テンプレート関数は、テンプレート関数を定義している箇所と使用している箇所を個別に表示しま す。

**4.** 現在選択しているノードの定義箇所,または宣言箇所へジャンプすることができます(「2.7 定義 箇所へジャンプする」/「2.8 宣言箇所へジャンプする」参照)。

#### (4) ポップアップ表示

クラス/カテゴリ用ツリー・エリア/関数/変数用リスト・エリア上の xxx 名ノード(ファイル名ノードを 除く)にマウス・カーソルを重ねることにより、属している名前空間名をポップアップ表示します。

ただし, グローバル名前空間の場合は"-"を, 無名名前空間の場合は"<unnamed>"を表示します。

また, *名前空間名*ノードを選択している場合では,対象の名前空間が属している名前空間名(上位の名前空間名)を表示します。

#### [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。

2	最新情報を取得し、	表示内容を更新します。

## [[編集] メニュー (クラス/メンバ パネル専用部分)]

クラス/メンバパネル専用の[編集]メニューの各項目、および機能は次のとおりです。

コピー	選択しているノードの文字列をクリップ・ボードにコピーします。
すべて選択	関数/変数用リスト・エリアのノードをすべて選択状態にします。

### [コンテキスト・メニュー]

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

ソースヘジャンプ	選択しているノードが定義されているソース・ファイルをエディタ パネル上にオープ
	ンします(「2.7 定義箇所へジャンプする」参照)。
ソースの宣言へジャンプ	選択しているノードが宣言されているソース・ファイルをエディタ パネル上にオープ
	ンします(「2.8 宣言箇所へジャンプする」参照)。
逆アセンブルヘジャンプ	関数/変数用リスト・エリアで選択しているノードの開始アドレスに対応する逆アセン
	ブル・データを逆アセンブル パネル(逆アセンブル 1)上にオープンします(「2. 7
	定義箇所へジャンプする」参照)。
	ただし、デバッグ・ツールと切断時は無効となります。
	[CC-RX]
	インスタンス変数を示すノードを選択している場合は無効となります。
メモリヘジャンプ	関数/変数用リスト・エリアで選択しているノードの開始アドレスに対応するメモリ・
	リストをメモリ パネル(メモリ 1)上にオープンします(「2.7 定義箇所へジャンプ
	する」参照)。
	ただし、デバッグ・ツールと切断時は無効となります。
	[CC-RX]
	インスタンス変数を示すノードを選択している場合は無効となります。



関数/変数一覧へジャンプ	関数一覧 パネル/変数一覧 パネルをオープンし、 このパネルで選択しているノードの
	関数/変数を選択状態にします。
	[CC-RX]
	インスタンス変数を示すノードを選択している場合は無効となります。
コピー	選択している項目の内容を文字列としてクリップ・ボー ドにコピーします。



# 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,「2.13 解析情報をグラフ化して表示する」を参照してください。

注意 このパネルは、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)。

	解析グラフ1		8
(1)—	2 式 😼 🏄 🖉		
	global_a	🖌 🐼 🐼 (変数値を入力して(だざい) 🛛 🖌	
(2) —	global_a global_b global_c		
		515ms572µs900ns 1s31ms196µs300ns	15531m5201
	> H         4IA		<u>&gt;</u> H
(3) —	値の推移(実行時間の割合)	k	-

図 A-----16 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・ウインドウのツールバーの 🜄 ボタンのクリック
- [表示] メニュー→ [プログラム解析] → [解析グラフ1~4]の選択



### [各エリアの説明]

(1) ツールバー

ツールバー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(1)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 <b>ー</b> タを保存	このパネルのタブの内容を前回保存したファイルに保存します(「2.14 解析情報を	
	ファイルに保存する」参照)。	
	なお、起動後に初めてこの項目を選択した場合は、 [名前を付けて 解析グラフ・データ	
	を保存 …]の選択と同等の動作となります。	
名前を付けて 解析グラフ・デ <del>ー</del>	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	
タ を保存	イアログをオープンします(「2.14 解析情報をファイルに保存する」参照)。	



# [値の推移] タブ

グラフ化の対象として登録した変数/レジスタ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,「2.13.1 変数値の推移をグラフ化する」を参照してください。



図 A-18 解析グラフ パネル: [値の推移] タブ(凡例表示をしない場合)



RENESAS

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

注意 このエリアは、プログラム実行中は無効となります。

#### (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。

✓ [½] [½] (変数名を入力してください) NA ▶ (変数値を入力してください) v - グラフ化対象を新規に登録する場合 (変数名を入力してください) 🗸 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 [変数名] コンボ・ボックス 定文字数:2046 文字)<sup>注</sup>。 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 - クラス変数(C++ ソース・ファイル) - CPU レジスタ - IOR [RX] [V850] - SFR [RL78] [78K0R] [78K0] - 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。 N [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。





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)。

入力形式	取得值
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
<i>変数式</i> .メンバ名	構造体/共用体のメンバ値
<i>変数式</i> -> メンバ名	ポインタの指し示す構造体/共用体のメンバ値
* 変数式	ポインタの変数の値
CPU レジスタ名	CPU レジスタの値
I/O レジスタ名【RX】【V850】	I/O レジスタの値
SFR レジスタ名	SFR レジスタの値
[RL78] [78K0R] [78K0]	
ラベル、EQU シンボル、即値アド	ラベルの値、EQU シンボルの値、即値アドレ
レス	スの値
ビット・シンボル	ビット・シンボル の値

表 A-2 グラフ化対象の入力形式

- 注意 1. 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。
  - ツールバーの ジャンにより、ウォッチ パネルと連動している場合、グラフ化対象の登録を 変更することはできません(このエリアは無効となります)。
- **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)。

(変数値を入力してください) (変数名を入力してください) \* N N

(変数名を入力してください) ✔	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。
[変数名] コンボ・ボックス	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。
(変数値を入力してください) 🗸	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)。
[変数値]コンボ・ボックス	なお、値は 10 進数でのみ指定可能です。
	[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。
	[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—11 グラフ・データの取得 方法によるグラフ表示の相違」参照)。

(b) 值 (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

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

R	<b></b> _
vs global.c 値: 660, 場所: main.c(115) 時期: 219ma602.c700mg	



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。

- リアルタイム・サンプリング方式を指定している場合





(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。

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

RENESAS

## [実行時間の割合] タブ

取得した関数情報において、関数の実行時間の割合を円グラフで表示します。

なお、このタブ上でグラフを表示するための操作手順は、「2.13.2 関数の実行時間率をグラフ化する」を参照して ください。

- 注意 1. デバッグ・ツールがトレース機能をサポートしていない場合、またはトレース機能を有効化していない 場合、このグラフを表示することはできません。
  - 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-19 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・ウインドウのツールバーの [] ボタンのクリック→ [実行時間の割合] タブの選択

- [表示] メニュー→ [プログラム解析] → [解析グラフ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、その関数の情報ををポップアップ表示します。 表示形式は次のとおりです。



## 関数名

実行時間[単位]

関数名	対象関数名を示します。
	[CC-RX]
	対象関数がメンバ関数の場合は、" <i>クラス名</i> :: <i>関数名</i> "の形式で表示します。
<i>実行時間[単位</i> ]	対象関数の実行時間を示します(関数一覧 パネルにおける [実行時間 [単位 ]]] 項目の
	値と同値)。
	なお、[ <i>単位</i> ]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。



# [ツールバー]

ツールバーの各ボタン、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。

2	最新情報を取得し、表示内容を更新します。
	ただし、変数が1つも登録されていない場合は無効となります。
<u>P</u>	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)。
	このタブ上で、このボタンは機能しません。
2	このタブ上で、このボタンは機能しません。
3	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、「2.11 参照箇所を一覧表示する」を参照してください。

図 A---20 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 - <i>タブ名</i> を保存	現在選択しているタブ上に表示されている内容を前回保存したテキスト・ファイル
	(*.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 - <i>タブ名</i> を保	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
存	(*.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。
置换	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-21 出力パネル: [すべてのメッセージ] タブ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]

## [オープン方法]

- [表示] メニュー→ [出力] の選択

### [各エリアの説明]

(1) メッセージ・エリア

CubeSuite+ が提供している全コンポーネント(解析ツール含む,設計ツール/ビルド・ツール/デバッグ・ ツールなど)から出力されたメッセージを表示します。

ただし,解析ツールが解析中に出力するメッセージの表示は行いません([プログラム解析] タブ上でのみ表示)。

なお、メッセージの表示色は、出力メッセージの種別により、次のように異なります(表示の際の文字色/ 背景色は、オプション ダイアログにおける [全般 - フォントと色] カテゴリの設定に依存)。

メッセージ種別	表示例	(デフォル	、ト)	説明
通常メッセージ	AaBbCc	文字色	黒	何らかの情報を通知する際に表示されます。
		背景色	白	
警告メッセージ	AaBbCc	文字色	青	操作に対して、何らかの警告を通知する際に表示され
		背景色	標準色	ます。



メッセージ種別	表示例(デフォルト)		<pre>&gt; ト)</pre>	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
		背景色	薄グレー	な場合に表示されます。

- 備考1. 出力されたメッセージをダブルクリック,またはメッセージにキャレットを移動したのち [Enter] キーを押下することにより,エディタパネルをオープンし,該当ファイルの該当行番号を表示しま す。
  - 2. 警告メッセージ、またはエラー・メッセージを表示している行にキャレットがある状態で、コンテキスト・メニューの [メッセージに関するヘルプ]を選択、または [F1] キーを押下することにより、該当行のメッセージに関するヘルプを表示します。
  - **3.** [ファイル] メニュー→ [名前を付けて 出力 すべてのメッセージ を保存 ...] を選択することにより、出力内容をテキスト・ファイル (\*.txt) に保存することができます。



# [プログラム解析] タブ

CubeSuite+ が提供している各種コンポーネント(解析ツールを含む,設計ツール/ビルド・ツール/デバッグ・ツールなど)から出力されるメッセージのうち,解析ツールが出力するメッセージを表示します。



- ここでは、次の項目について説明します。
  - [オープン方法]
  - [各エリアの説明]

## [オープン方法]

- [表示] メニュー→ [出力] の選択

## [各エリアの説明]

(1) メッセージ・エリア

CubeSuite+ が提供している各種コンポーネント(解析ツールを含む,設計ツール/ビルド・ツール/デバッグ・ツールなど)から出力されるメッセージのうち,解析ツールが出力したメッセージを表示します。

なお,メッセージの表示色は,出力メッセージの種別により,次のように異なります(表示の際の文字色/ 背景色は,オプションダイアログにおける[全般-フォントと色]カテゴリの設定に依存)。

メッセージ種別	表示例	(デフォル	<pre>&gt; ト)</pre>	説明
通常メッセージ	AaBbCc	文字色	黒	何らかの情報を通知する際に表示されます。
		背景色	白	
警告メッセージ	AaBbCc	文字色	青	操作に対して、何らかの警告を通知する際に表示され
		背景色	標準色	ます。
エラー・メッセージ	AaBbCc	文字色	赤	致命的なエラー, または操作ミスにより実行が不可能
		背景色	薄グレー	な場合に表示されます。



- 備考1. 出力されたメッセージをダブルクリック,またはメッセージにキャレットを移動したのち [Enter] キーを押下することにより,エディタパネルをオープンし,該当ファイルの該当行番号を表示しま す。
  - 2. 警告メッセージ、またはエラー・メッセージを表示している行にキャレットがある状態で、コンテキスト・メニューの [メッセージに関するヘルプ]を選択、または [F1] キーを押下することにより、該当行のメッセージに関するヘルプを表示します。
  - **3.** [ファイル] メニュー→ [名前を付けて 出力 プログラム解析 を保存 ...] を選択することにより, 出力内容をテキスト・ファイル (\*.txt) に保存することができます。



## [参照の検索] タブ

指定した関数/変数の参照箇所一覧を表示します。

なお、関数/変数の参照箇所一覧の出力方法についての詳細は、「2.11 参照箇所を一覧表示する」を参照してくだ さい。

注意 1. このタブは、参照箇所を一覧表示する操作を一度も実行していない場合は出現しません。

- 2. C/C++ ソース・ファイル中の "#if" / "#ifdef" などで、コンパイル時にプリプロセッサにより除外される コードにおいて参照されている箇所は参照箇所として出力されません。
- 【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-23 出力パネル:[参照の検索]タブ

	出力	
(1)-	参照検索開始」 対象: sub01」 定義箇所: sub01.c(9): int <u>sub01</u> (int arg a, int arg_b, int arg_c)」 参照箇所(実行回数 0): sub01.h(3): int <u>sub01</u> (int, int, int);」 参照箇所(実行回数 0): main.c(38): int <u>sub01</u> (int, int, int);」 参照箇所(実行回数 0): main.c(38):	
	関数 sub01 で参照(リード/ライト)している変数の一覧:↓ slobal 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/	>

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]

## [オープン方法]

- [表示] メニュー→ [出力] の選択

#### [各エリアの説明]

(1) メッセージ・エリア

関数一覧パネル/変数一覧パネルにおいて、関数/変数を参照している箇所を検索した際に、その結果として、次の参照箇所一覧を表示します(「2.11 参照箇所を一覧表示する」参照)。

RENESAS

- 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-24 関数の参照箇所一覧の出力フォーマット

	参照検索開始
Γ	対象: <i>対象関数名</i>
•	定義箇所: <i>ファイル名 ( 行数</i> ): <i>該当箇所の 1 行</i>
A	参照箇所(実行回数: <i>回数</i> ): ファイル名(行数): 該当箇所の1行
L	
	関数 <i>対象関数名</i> で呼び出している関数の一覧:
в –	呼び出し関数名
Γ	関数 <i>対象関数名</i> で参照(リード/ライト)している変数の一覧:
c-	参照変数名
	参照検索終了

	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	えします。		
		ただし クロス リファ	<b>ィレンス・ファイルから情報が取得できない場合は"なし"を表示</b>		
		します。			
		ファイル名(行数)	該当箇所が存在するファイル名 <sup>注</sup> 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。		
		<u>該当箇所の1行</u> 該当箇所の1行をファイルから抜き出し表示			
A		この際、対象関数名が強調表示されます。			
	参照箇所	対象関数を参照している箇所を列挙します。動的解析情報が存在する場合は()内に対			
		象関数の実行回数を示し	<i>、</i> ます。		
		ただし、クロス・リファ	<b>ィレンス・ファイルから情報が取得できない場合は"なし"を表示</b>		
		します。			
		ファイル名 (行数)	該当箇所が存在するファイル名 <sup>注</sup> 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。		



	対象関数内で呼び出している関数の一覧				
в	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。			
	対象関数内で参照	(リード/ライト)している変数の一覧			
С	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-25 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)

関数 sub01 で呼び出している関数の一覧:」 sub01\_sub01\_ 閲数 sub01 で参照(リード/ライト)している変数の一覧:丿 global a,

#### 図 A-26 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(8): int <mark>sub01</mark> (int arg_a, int arg_b, int arg_c)」 参照箇所(実行回数 0): sub01.h(3): int <mark>sub01</mark> (int, int, int);」 参照箇所(実行回数 0): main.c(38): int <u>sub01</u> (int, int, int);」 参照箇所(実行回数 0): main.c(38): ——→result = <u>sub01</u> (local_a, local_b, local_c);」 参照箇所(実行回数 14): main.c(129): ——→result = <u>sub01</u> (global_a, global_b, global_c); 」
が 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—27 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: <i>対象変数名</i>
定義箇所: <i>ファイル名 ( 行数 ):該当箇所の 1 行</i>
参照箇所 (リード/ライト回数 : <i>回数</i> ): <i>ファイル名</i> ( <i>行数</i> ) : <i>該当箇所の1行</i>
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 <sup>注</sup> 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 <sup>注</sup> を示します。	
		<ul><li>()内には、ファイル内における行数(行番号)を示します。</li><li>該当箇所の1行をファイルから抜き出し表示します。</li></ul>	
	該当箇所の1 行		
		この際、対象変数名が強調表示されます。	

- **注** 変数一覧 パネルにおいて、[ファイル・パス] 項目を表示している場合(デフォルトでは表示され ません)、ファイルの絶対パスを表示します。
- 備考1. extern 宣言が記述されている変数の行の扱いは、使用するコンパイラにより次のように異なり ます。
  - [CC-RX] [CX]

extern 宣言が記述されている変数の行を参照箇所として扱います。

- 【CA850】

extern 宣言が記述されている変数の行を定義箇所として扱います。

- [CA78K0R] [CA78K0]

extern 宣言が記述されている変数の行を参照箇所として扱います。

また, 同一ファイル内で extern 宣言が記述されている行と変数の定義行が存在する場合は, 定義箇所は"なし"と表示します。

2. [CX]

変数定義行において、代入文が記述されている行("int variable = 10"など)も参照箇所として扱います。

図 A—28 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 <mark>global_a</mark> ;」
定義箇所: main.c(10): int <mark>global a</mark> = 10;山
参照箇所: main.c(95): →global_pointer = & <mark>global_a</mark> ;↓
参照箇所: main.c(112): <del>&gt;global_a</del> = 0;」
参照箇 <u>所</u> : main.c(125): →
参照箇 <u>所</u> : main.c(129): →———→result <u>= sub01(global_a</u> , global_b, global_c);」
参照箇所: sub01.c(15): →result = tmp + <mark>global_a</mark> ;↓

### 図 A-29 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 <mark>global_a</mark> ;」
定義箇所: main.c(10): int <mark>global_a</mark> = 10;」
参照箇所(リード/ライト回数 0): main.c(95): ―→global_pointer = & <mark>global_a</mark> ;↓
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): →―――→ <mark>global_a</mark> ++;」
参照箇所(リード/ライト回数 14): main.c(129): →―――→result <u>=_sub01(global_a</u> , global_b, global_c);↓
参照箇所(リード/ライト回数 14): sub01.c(15): →result = tmp + <mark>global_a</mark> ;」



# パス編集 ダイアログ

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

なお, インポート機能についての詳細は, 「2.12 情報ファイルをインポート/エクスポートする」を参照してくだ さい。

	パス編集			
Г	パス(1行につき1つのパス)(P): 🜉			
	%ActiveProjectDir%¥F	funcList_Sub2.mtfl		~
(1)	5			>
L	参照( <u>B</u> ) プレースホルダ( <u>L</u> ):			-
	プレースホルダ	値	説明	^
(2)	ActiveProjectDir ActiveProjectName MainProjectDir MainProjectName MicomToolPath	C:¥Test¥Sample 78K0R_LH3 userop C:¥Test¥Sample 78K0R_LH3 userop C:¥Test¥Sample	アクティブ・プロジェクトのフォ アクティブ・プロジェクト名 メイン・プロジェクトのフォル! メイン・プロジェクト名 CubeSuite+のインストール	Fル? <u>-</u> ダの ·先: <mark>- ◇</mark>
L	<			>
[機能ボタン] -		ОК	キャンセル ヘルプ	<u>и</u> Ш)

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- プロパティ パネルの [設定] タブにおいて, [インポート/エクスポート] カテゴリ内 [インポート・ファイル] プロパティを選択することにより表示される [...] ボタンをクリック



### [各エリアの説明]

#### (1) パス編集エリア

パスを含むファイル名の編集,追加を行います。

(a) [パス (1 行につき 1 つのパス )] エリア

インポートする関数一覧ファイル(\*.mtfl) /変数一覧ファイル(\*.mtvl)のすべてを、1行に1つずつ、 パスを含めて指定します(1行に259文字/64行まで指定可)。相対パスによる指定の場合は、プロジェク ト・フォルダを基点として指定します。

なお、パスを含むファイル名の追加は、次の方法でも行うことができます。

- [参照 ...] ボタンをクリックしたファイルの選択
- エクスプローラなどからファイルをドラッグ・アンド・ドロップ

(b)ボタン



注意 絶対パスで非常に長いパスを相対パスで指定すると、[OK] ボタンのクリック時にエラーになる場合 があります。その場合は、絶対パスで指定してください。

#### (2) プレースホルダ・エリア

このダイアログの呼び出し元に指定可能なプレースホルダの一覧を表示します(昇順)。 行をダブルクリックすると、プレースホルダの前後に"%"を付加してパス編集エリアに表示します。

(a) [プレースホルダ]

プレースホルダを表示します。

(b)[值]

プレースホルダの置換後の文字列を表示します。

(c) [説明]

プレースホルダの説明を表示します。

- 注意 このエリアは、このダイアログの呼び出し元がプレースホルダに対応している場合のみ表示されます。
- 備考 指定可能なプレースホルダは、このダイアログの呼び出し元によって異なります。 具体的なプレースホルダについては、呼び出し元の説明を参照してください。



# [機能ボタン]

ボタン	機能
ок	指定したファイルをインポート・ファイルに設定し、このダイアログをクローズしま す。
キャンセル	ファイルの指定を中止し、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



# 列の選択 ダイアログ

関数一覧 パネル/変数一覧 パネルに表示する項目(列)の並び替え,または表示/非表示を設定します。

また、各パネルにおいて表示方法をカスタマイズする操作を行っている場合では、このダイアログより、それらのカ スタマイズをすべてデフォルトの状態に戻すことができます。



図 A---31 列の選択 ダイアログ(関数一覧パネル用)

図 A---32 列の選択 ダイアログ(変数一覧 パネル用)





- ここでは、次の項目について説明します。
  - [オープン方法]
  - [各エリアの説明]
  - [機能ボタン]

### [オープン方法]

- 関数一覧 パネルにおいて、ヘッダ・エリア上の 🛃 ボタンをクリック
- 変数一覧 パネルにおいて、ヘッダ・エリア上の 🗐 ボタンをクリック

### [各エリアの説明]

(1)項目名リスト・エリア

関数一覧 パネル/変数一覧 パネルで表示可能な項目(列)のすべてをリスト表示します(表示可能な項目 (列)は、使用するマイクロコントローラ/デバッグ・ツールにより異なります)。

なお、リスト内の各項目の表示順、およびチェック・ボックスの状態は、対応するパネルにおける現在の表 示順、および表示/非表示の状態と同等です。

各項目のチェック・ボックスの指定により、パネル上での表示/非表示を設定することができます。

>	この項目をパネル上に表示します。
	この項目をパネル上に表示しません。
•	該当項目の情報に関するメッセージを出力 パネルに出力していることを示します。マウス・カーソルを
	重ねることにより、出力した最新のメッセージをポップアップ表示します。

### [機能ボタン]

ボタン	機能	
デフォルト	関数一覧 パネル/変数一覧 パネル上の各項目の表示順、および表示/非表示の設定を	
	デフォルトの状態に戻します。	
閉じる	このダイアログをクローズします。	

備考 各項目のデフォルトの表示状態は次のとおりです。

表内[項目]の表記順は、各パネル上の項目(列)の並び順に相当します。

なお、固定表示/フィルタ表示などのカスタマイズ設定はすべて解除されます。

また、使用するマイクロコントローラ/デバッグ・ツールにより、表示項目が限定されます。

### 表 A---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	
クラス名	表示	表示	
[CC-RX]			
名前空間	非表示	非表示	
[CC-RX]			
ファイル名	表示	表示	
ファイル・パス	非表示	非表示	
PE 情報 <sup>注</sup>	表示	表示	
[V850E2]			
インポート	非表示	非表示	
アクセス指定子	表示	表示	
[CC-RX]			
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	
[V850]			
[RL78]			
【78K0R】			
[78K0]			
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	非表示	表示	
[IECUBE]			
【IECUBE2】			
【シミュレータ】			
実行時間 [ <i>単位</i> ]	非表示	表示	
[IECUBE [V850]]			
[IECUBE [RL78]]			
[IECUBE [78K0R]]			
【IECUBE2】			
【シミュレータ】			
実行時間 ( 割合 )[%]	非表示	表示	
[IECUBE [V850]]			
[IECUBE [RL78]]			
[IECUBE [78KOR]]			
[IECUBE2]			
【シミュレータ】			



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	
平均実行時間 [ <i>単位</i> ]	非表示	表示	
[IECUBE [V850]]			
[IECUBE [RL78]]			
[IECUBE [78K0R]]			
[IECUBE2]			
【シミュレータ】			
コード・カバレッジ [%]	非表示	表示	
[IECUBE]			
[IECUBE2]			
【シミュレータ】			

### 注 【V850E2】

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

項目	表示状態			
	デバッグ・ツールと切断時	デバッグ・ツールと接続時		
変数名	表示	表示		
クラス名	表示	表示		
[CC-RX]				
名前空間	非表示	非表示		
[CC-RX]				
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 <sup>注</sup>	非表示	表示		
[V850E2]				
インポート	非表示	非表示		
アクセス指定子	表示	表示		
[CC-RX]				
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		

### 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		
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 [RL78]】 【IECUBE [78K0R]] 【IECUBE [78K0]] 【IECUBE2】 【シミュレータ】	非表示	表示		

注 【V850E2】

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



# コール・グラフ検索 ダイアログ

コール・グラフパネルで表示しているコール・グラフ内に存在する関数/変数を検索します。

注意 コール・グラフとして表示している関数/変数(関数ボックス/変数ボックス)のみが検索の対象となります。

	コール・グラフ検索	×	
Γ	検索条件の指定 関数/変数名(1):		]
	◆ 条件(型): を含む ◆		
(1) —	条件型: を含む 🗸		- (2)
	□ 大文字を小文字を区別する(S) 親関数の個数(P):		(2)
	条件(Q): と等しい ▼ 子関数の個数(H):		
	条件(型): と等しい 🗸		
[機能ボタン]-	前を検索(E) 次を検索(E) キャンセル ヘルプ(H)		-

- ここでは、次の項目について説明します。
  - [オープン方法]
  - [各エリアの説明]
  - [機能ボタン]

# [オープン方法]

- コール・グラフ パネルのツールバーの 🚠 ボタンのクリック

- コール・グラフ パネルにフォーカスがある状態で, [編集] メニュー→ [検索 ...]を選択

### [各エリアの説明]

- (1) [検索条件の指定] エリア
  - 検索条件を指定します。

なお、複数の検索条件を指定した場合、すべての条件を満たす関数/変数のみを検索します。



#### (a) [関数/変数名]

検索対象となる関数名/変数名を指定します。

キーボードより文字列を直接入力するか(最大指定文字数:2046文字),ドロップダウン・リストより 入力履歴項目を選択します(最大履歴数:10個)。

デフォルトでは、前回指定した関数名/変数名を表示します(存在しない場合は"空欄")。

- [条件]

[関数/変数名]で指定した文字列に対する条件を次のドロップダウン・リストより選択します。

と等しい	指定した文字列と完全に一致する関数名/変数名を検索します。
で始まる	指定した文字列で始まる関数名/変数名を検索します。
で終わる	指定した文字列で終わる関数名/変数名を検索します。
を含む	指定した文字列を含む関数名/変数名を検索します(デフォルト)。

備考 関数名/変数名は、次の操作によっても指定することができます。

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

(b) [クラス名]【CC-RX】

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

キーボードより文字列を直接入力するか(最大指定文字数:2046 文字),ドロップダウン・リストより 入力履歴項目を選択します(最大履歴数:10 個)。

デフォルトでは、前回指定したクラス名を表示します(存在しない場合は"空欄")。

- [条件]

[クラス名]で指定した文字列に対する条件を次のドロップダウン・リストより選択します。

と等しい	指定した文字列と完全に一致するクラス名に属する関数名/変数名を検索します。
で始まる	指定した文字列で始まるクラス名に属する関数名/変数名を検索します。
で終わる	指定した文字列で終わるクラス名に属する関数名/変数名を検索します。
を含む	指定した文字列を含むクラス名に属する関数名/変数名を検索します(デフォルト)。

備考 クラス名は、次の操作によっても指定することができます。

- 任意の文字列をこのエリアにドラッグ・アンド・ドロップ

#### (c) [大文字と小文字を区別する]

[関数/変数名]/[クラス名]【CC-RX】で指定した文字列に対して、大文字と小文字を区別する否か を指定します。

<b>V</b>	大文字と小文字を区別して検索します。
	大文字と小文字を区別せず検索します(デフォルト)。

#### (d) [親関数の個数]

関数の検索条件の1つとして必要な場合、検索対象の関数の親関数の個数を指定します。

キーボードより数値を直接入力するか(指定可能範囲:0~65535),ドロップダウン・リストより入力 履歴項目を選択します(最大履歴数:10個)。

デフォルトでは、前回指定した個数を表示します(存在しない場合は"空欄")。

#### - [条件]

[親関数の個数]で指定した数値に対する条件を次のドロップダウン・リストより選択します。

より大きい	指定した数値より大きい(指定値を含まない)個数の親関数を持つ関数名を検索します。
以上	指定した数値以上(指定値を含む)の個数の親関数を持つ関数名を検索します。
と等しい	指定した数値と等しい個数の親関数を持つ関数名を検索します(デフォルト)。
以下	指定した数値以下(指定値を含む)の個数の親関数を持つ関数名を検索します。
より小さい	指定した数値より小さい(指定値を含まない)個数の親関数を持つ関数名を検索します。

#### (e) [子関数の個数]

関数の検索条件の1つとして必要な場合、検索対象の関数の子関数の個数を指定します。

キーボードより数値を直接入力するか(指定可能範囲:0~65535),ドロップダウン・リストより入力 履歴項目を選択します(最大履歴数:10個)。

デフォルトでは、前回指定した個数を表示します(存在しない場合は"空欄")。

#### - [条件]

[子関数の個数]で指定した数値に対する条件を次のドロップダウン・リストより選択します。

より大きい	指定した数値より大きい(指定値を含まない)個数の子関数を持つ関数名を検索します。
以上	指定した数値以上(指定値を含む)の個数の子関数を持つ関数名を検索します。
と等しい	指定した数値と等しい個数の子関数を持つ関数名を検索します(デフォルト)。
以下	指定した数値以下(指定値を含む)の個数の子関数を持つ関数名を検索します。
より小さい	指定した数値より小さい(指定値を含まない)個数の子関数を持つ関数名を検索します。

(2) [プレビュー] エリア

コール・グラフ全体のプレビューを表示します。

現在コール・グラフ パネルで表示している領域が、コール・グラフ全体の一部である場合は、その領域を赤 枠で示します。

RENESAS

## [機能ボタン]

ボタン	機能
前を検索	指定した条件でコール・グラフ内の最下段から上段方向へ関数名/変数名の検索を行い
	該当関数ボックス/変数ボックスを強調表示します。
	なお、コール・グラフ パネル上で関数ボックス/変数ボックスを選択している場合は、対
	象関数/変数から上段方向へ検索を開始します。
	ただし、不正な値を指定している場合、またはプログラム実行中は、メッセージを表示
	し、関数名/変数名の検索は行いません。
	また、コール・グラフ パネルが非表示の場合、または他のパネルにフォーカスがある状態
	からこのダイアログへフォーカスを移動した場合、このボタンは無効となります。
次を検索	指定した条件でコール・グラフ内の最上段から下段方向へ関数名/変数名の検索を行い
	該当関数ボックス/変数ボックスを強調表示します。
	なお、コール・グラフ パネル上で関数ボックス/変数ボックスを選択している場合は、対
	象関数から下段方向へ検索を開始します。
	ただし、不正な値を指定している場合、またはプログラム実行中は、メッセージを表示
	し、関数名/変数名の検索は行いません。
	また、コール・グラフ パネルが非表示の場合、または他のパネルにフォーカスがある状態
	からこのダイアログへフォーカスを移動した場合、このボタンは無効となります。
キャンセル	関数の検索を中止し、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



# フィルタ設定 ダイアログ

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

注意 このダイアログにより設定するフィルタ表示とパネルと連携したフィルタ表示は排他使用の機能です。 このため、これら2つのフィルタ表示機能を同時に有効化することはできません(どちらか一方のフィルタ表 示を行っている際に、もう一方のフィルタ表示の設定を行った場合、それまで行っていたフィルタ表示はすべ て解除されます)。

フィルタ表示についての詳細は、「2.6.5 解析情報をフィルタ表示する」を参照してください。

	フィルタ設定					×
_	フィルタ条件					
(1) -	(ここに比較値を入力してください)	~				~
(2) —	→					
(1) -	(ここに比較値を入力してください)	~				*
[機能ボタン] -			ок С	キャンセル	~,IJ	

図 A---34 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(マ/マ)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 <sup>注</sup>	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
ОК	指定したフィルタ条件で、関数一覧 パネル/変数一覧 パネル上の解析情報を表示しま
	す。
	なお、パネルと連携したフィルタ表示を行っている場合では、それまで行っていたパ
	ネルと連携したフィルタ表示をすべて解除します。
キャンセル	フィルタ条件の設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



# 名前を付けて保存 ダイアログ

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

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:
- 解析グラフパネル/出力パネルの場合: 現在表示しているタブの内容のみ

	名前を付けて保存						2 🛛	
(1)-	保存する場所型:	🚞 sample		~	0 1	• 🖽 🍤		
	していたつたファイル							
	<b>び</b> デスクトップ							
(2)	<u>א</u> לאנדאיז די							
	ער דא פאר איז							
	マイ ネットワーク		Ĩ.					-
(3)	•	ファイル:名( <u>N</u> ):			*	「保存	- <u>@</u>	—[機能
(4) —		ファイルの理究(1):			*	445		

図 A---35 名前を付けて保存 ダイアログ

このダイアログで指定

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
- 出力 パネルにフォーカスがある状態で, [ファイル] メニュー→ [名前を付けて 出力 タブ名 を保存 ...]を選択

### [各エリアの説明]

(1) [保存する場所] エリア

ファイルを保存するフォルダを選択します。

(2) ファイル一覧エリア

[保存する場所] エリア,および [ファイルの種類] エリアで選択された条件に合致するファイルの一覧を表示します。

(3) [ファイル名] エリア

保存する際のファイル名を指定します。

(4) [ファイルの種類] エリア

ドロップダウン・リストより選択した次のファイル形式で、パネルの内容をファイルに保存します。

(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
関数一覧ファイル (*.mtfl)	関数情報をインポートするためのファイル形式(「2.12 情報ファイ
	ルをインポート/エクスポートする」 参照)

### (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
変数一覧ファイル (*.mtvl)	変数情報をインポートするためのファイル形式(「2.12 情報ファイ
	ルをインポート/エクスポートする」 参照)

#### (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・データ解析方式の場合

【保存項目】: 時間 値(変数名) 値(変数名) … 場所

備考 値(変数名): 値が変化した変数のみ出力(ただし,値が不明の場合は空欄) 場所: 情報が存在しない場合は空欄

リアルタイム・サンプリング方式の場合

【保存項目】: 時間 値(変数名) 値(変数名) …

備考 値(変数名): 各変数の値を常に出力(ただし,値が不明の場合は空欄)

- [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。

(e) 出力パネルの場合

テキスト・ファイル (\*.txt) テキスト形式

# [機能ボタン]

ボタン 機能	
保存	指定したファイル名でファイルを保存します。
キャンセル	このダイアログをクローズします。



### 付録 B 索 引

#### [R]

RRM 機能 … 8

### 【あ行】

アクティブ・プロジェクト … 46 異常値の切り捨て … 56 インポート … 46 ウインドウ・リファレンス … 64 ウォッチ式 … 44 エクスポート … 46 円グラフ … 58,133 親関数 … 105 折れ線グラフ … 51,127

### 【か行】

解析グラフ … 51 解析グラフパネル … 125 [値の推移] タブ … 127 [実行時間の割合] タブ … 133 書き込みブレーク … 42 カバレッジ機能 … 8 カレント行 … 90,100 カレント行マーク … 90,100 関数一覧 パネル … 81 関数一覧ファイル … 147 関数の実行時間率 … 58 共用体 … 96 クラス/メンバ パネル … 115 グラフ化対象 … 52 グラフ・データの取得方法 … 54 クロス・リファレンス情報 … 7,74 降順表示 … 35 構造体 … 96 コード・カバレッジ率 … 86 コール・グラフ検索 ダイアログ … 155 コール・グラフ パネル … 103 固定表示 … 34 固定表示アイコン … 34

### 【さ行】

再帰呼び出し … 110 参照箇所一覧 … 45,142 実行時間の割合 … 58 出力パネル … 136 [参照の検索]タブ … 142 [すべてのメッセージ]タブ … 138 [プログラム解析]タブ … 140 循環呼び出し … 111 詳細表示 … 106 昇順表示 … 35 情報ファイル … 46 静的解析情報 … 7 全体表示 … 106 ソート表示 … 35

### 【た行】

ツールバー … 66
定義箇所 … 39,41
データ・カバレッジ率 … 97
動的解析情報 … 8
特長 … 13
トレース機能 … 8,54
トレース・データ解析方式 … 54

【な行】 名前を付けて保存 ダイアログ … 162

### 【は行】 パス編集 ダイアログ … 147 表示範囲 … 55 ファイルに保存 … 162 フィルタ設定 ダイアログ … 159 フィルタ表示 … 35 フィルタ・アイコン … 35 ブレークポイント … 42 ブレーク・イベント … 42 プロジェクト・ツリーパネル … 68



プロパティ パネル … 70 [設定]タブ … 73 変数一覧 パネル … 92 変数一覧ファイル … 147 ポップアップ表示 … 57

### 【ま行】

メイン・ウインドウ … 65 メニューバー … 66

### 【や行】

有効範囲 … 56 読み書きブレーク … 42 読み込みブレーク … 42

### 【ら行】

ランタイム・ライブラリ … 87 リアルタイム RAM モニタ機能 … 54 リアルタイム・サンプリング方式 … 54 列の選択 ダイアログ … 150



### 改訂記録

Rev.	発行日	改訂内容			
		ページ	ポイント		
1.00	2011.10.01	_	初版発行		

CubeSuite+ V1.01.00 ユーザーズマニュアル 解析編

- 発行年月日 2011年10月1日 Rev.1.00
- 発行 ルネサス エレクトロニクス株式会社〒211-8668 神奈川県川崎市中原区下沼部 1753

# RENESAS

ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

http://www.renesas.com

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2(日本ビル)

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。 総合お問合せ窓口: http://japan.renesas.com/inquiry

CubeSuite+ V1.01.00

