RENESAS

CS+ for CC V3.00.00

リリースノート

R20UT3159JJ0100 Rev. 1.00 2014. 8.22

この度は,統合開発環境 CS+をご使用いただきまして,誠にありがとうございます。 この添付資料では,本製品をお使いいただく上での制限事項および注意事項等を記載しております。 ご使用の前に,必ずお読みくださいますようお願い申し上げます。

第1章	対象デバイスについて	2
第2章	ユーザーズ・マニュアルについて	3
第3章	アンインストール時の選択キーワード	4
第4章	変更点	5
第5章	注意事項	9
第6章	制限事項	24
第7章	ドキュメント訂正	25



第1章 対象デバイスについて

統合開発環境 CS+がサポートする対象デバイスに関しては、WEB サイトに掲載しています。

こちらをご覧ください。

CS+製品ページ:

http://japan.renesas.com/cs+



第2章 ユーザーズ・マニュアルについて

本製品に対応したユーザーズ・マニュアルは、次のようになります。本文書と合わせてお読みください。

マニュアル名	資料番号
CS+ V3.00.00 インストーラ編	R20UT3094JJ0100
CS+ V3.00.00 エディタ編	R20UT3096JJ0100
CS+ V3.00.00 Python コンソール編	R20UT3097JJ0100
CS+ V3.00.00 アップデータ編	R20UT3098JJ0100
CS+ V3.00.00 メッセージ編	R20UT3099JJ0100
CS+ V3.00.00 プロジェクト操作編	R20UT3101JJ0100
CubeSuite+ V2.02.00 解析編	R20UT2868JJ0100
CS+ V3.00.00 RH850 デバッグ・ツール編	R20UT3082JJ0100
CS+ V3.00.00 RX デバッグ・ツール編	R20UT3083JJ0100



第3章 アンインストール時の選択キーワード

本製品をアンインストールする場合は、2 つの方法があります。

- ・統合アンインストーラを使用する(CS+自体をアンインストールする)
- ・個別にアンインストールする(本製品のみをアンインストールする)

個別にアンインストールを行なう場合、コントロールパネルの

・「プログラムと機能」

から, 「CS+ for CC」を選択してください。



第4章 変更点

本章では、CS+の V2.02.01 から V3.00.00 の変更点について説明します。

4.1 CS+の全体の機能改善

4.1.1 名称の変更

本バージョンより、CubeSuite+をCS+に変更しました。

ドキュメントによっては、「CubeSuite+」の表現が残っている部分がありますが、「CS+」に読み替えてご 参照ください。

4.1.2 起動メニューの分割

本バージョンより、Windowsのスタートメニューからの起動を2つに分割しました。

• CS+ for CC (RX, RH850)

• CS+ for CA,CX (78K,RL78,V850)

使用するコンパイラにより、上記の起動メニューを使い分けてください。

ファミリ		コンパイラ	Windowsのスタートメニュー
RX		CC-RX	CS+ for CC (RX, RH850)
RH850		CC-RH	CS+ for CC (RX, RH850)
78K	78K0R	CA78K0R	CS+ for CA,CX (78K,RL78,V850)
	78K0	CA78K0	CS+ for CA,CX (78K,RL78,V850)
RL78		CA78K0R	CS+ for CA,CX (78K,RL78,V850)
V850	V850E	CA850	CS+ for CA,CX (78K,RL78,V850)
	V850E2	СХ	CS+ for CA,CX (78K,RL78,V850)

なお、プロジェクト・ファイルをダブルクリックして起動した場合は、プロジェクト種別に応じた CS+を起 動します。

4.1.3 分割パネルエリアの機能追加

本バージョンより、分割パネルエリアのフォーカスのあるタブの強調表示機能を追加しました。

4.1.4 メモリ・マッピング・プロファイラの機能追加

本バージョンより、メモリ・マッピング・プロファイラ機能を追加しました。

4.1.5 オプションダイアログ機能の改善

本バージョンより、「ビルド/デバッグ」を「ビルド」と「デバッグ」に分離しました。 「ビルド」に「プロジェクト間のパラレル・ビルドを有効にする」のチェックボックスを追加しました。

4.1.6 外部ツール起動機能の改善

本バージョンより、外部ツールの起動オプションで指定できるプレースホルダを追加しました。

プレースホルダ名	說明
ActiveProjectDir	アクティブ・プロジェクト・フォルダの絶対パスに置換します。
ActiveProjectName	アクティブ・プロジェクト名に置換します。
FileDir	現在, プロジェクト・ツリー パネルで選択しているファイルが存在す
	るフォルダの絶対パスに置換します。
MainProjectDir	メイン・プロジェクト・フォルダの絶対パスに置換します。
MainProjectName	メイン・プロジェクト名に置換します。
MicomToolPath	本製品のインストール・フォルダの絶対パスに置換します。
ProjectDir	プロジェクト・フォルダの絶対パスに置換します。
ProjectName	プロジェクト名に置換します。
TempDir	テンポラリ・フォルダの絶対パスに置換します。
WinDir	Windowsシステム・フォルダの絶対パスに置換します。

4.1.7 UM、オンラインヘルプ構成の変更

- 本バージョンより,UM,オンラインヘルプ構成を変更しました。主な構成は次のようになります。 インストーラ編(UMのみ,オンラインヘルプはありません)
 - プロジェクト操作編
 - コード生成端子図操作編
 - コード生成周辺機能操作編
 - コード生成 API リファレンス編
 - エディタ編
 - コンパイラ編
 - ビルド・ツール操作編
 - デバッグ・ツール編
 - 解析ツール編
 - Python コンソール編
 - アップデータ編
 - メッセージ編

4.2 ビルド・ツールの機能改善

4.2.1 パラレル・ビルド機能追加

本バージョンより、パラレル・ビルド機能を追加しました。

パラレル・ビルド機能を使用することで、ホストマシンの複数の CPU コアで同時にコンパイルを実行し、ビルド時間を短縮できます。また、複数のプロジェクトを同時にビルドするプロジェクト間のパラレル・ビルドも行えます。

RENESAS

- 4.3 デバッグ・ツールの機能改善
 - 4.3.1 デバッグ・コンソールパネルの表示フォントの変更【RX】 デバッグ・コンソールパネルの表示フォントを等幅フォントに変更しました。
 - 4.3.2 内蔵フラッシュを書き換えるプログラムをデバッグする機能の改善【RX】 「内蔵データフラッシュを書き換えるプログラムをデバッグする」または「内蔵フラッシュ ROM を書き換え るプログラムをデバッグする」を有効にした場合、「内蔵データフラッシュ」「内蔵フラッシュ ROM」にソ フトウェアブレーク設定できないよう仕様を変更しました。
 - 4.3.3 RX64M シリーズの Trusted memory 機能対応【RX】 RX64M シリーズの Trusted memory 機能に対応しました。
 - 4.3.4 アクションイベントの機能改善【RX】 【RH850】 Printf イベントにて、変数式が空欄でも設定できるよう変更しました。
 - 4.3.5 ダウンロード機能の改善【RX】【RH850】 エラー発生時にエラー対象アドレスを表示するよう変更しました。
- 4.4 Python コンソールの機能改善

4.4.1 Python 関数の追加

以下の Python 関数を追加しました。

関数名	機能概要
common.PythonInitialize	Python を初期化します。
debugger.ActionEvent.Delete	アクション・イベントを削除します。
debugger.ActionEvent.Disable	アクション・イベントの設定を無効にします。
debugger.ActionEvent.Enable	アクション・イベントの設定を有効にします。
debugger.ActionEvent.Get	アクション・イベントの結果を参照します。
debugger.ActionEvent.Information	アクション・イベント情報を表示します。
debugger.ActionEvent.Set	アクション・イベントを設定します。
debugger.GetIORList	IOR レジスタ の一覧を表示します。



4.4.2 Python クラスの追加

以下の Python クラスを追加しました。

クラス名	機能概要
ActionEventCondition	アクション・イベントの条件を作成します。
ActionEventInfo	アクション・イベント情報を保持します。
ActionInfo	アクション・イベントの結果情報を保持します。
IORInfo	IOR レジスタ の情報を保持します。

4.4.3 Python プロパティの追加

以下の Python プロパティを追加しました。

プロパティ名	機能概要	
common.ExecutePath	実行しているCS+ のexe ファイルのフォルダの絶対パスを参	
	照します。	
build.Compile.Macro	アクティブ・プロジェクトのコンパイル・オプションである。	
	定義マクロの設定/参照を行います。	
build.Version	コンパイラ・パッケージのバージョンを参照します。	
debugger.ActionEvent.GetLine	アクション・イベント結果を保持する数を設定/参照します。	
debugger.Option.AccessStopExecution	デバッグ・ツールのオプションを設定/参照します。	
debugger.Option.AfterTraceMemoryFull	デバッグ・ツールのオプションを設定/参照します。	



第5章 注意事項

本章では、注意事項について説明します。

5.1 CS+全体の注意事項

5.1.1 ファイル名に関する注意事項

フォルダ名、ファイル名に関しては次の注意事項があります。

・フォルダ名、ファイル名
 Windows のエクスプローラで作成することのできないフォルダ名とファイル名は、使用しないでください。

・ソース・ファイル名とロード・モジュール・ファイル名とプロジェクト・ファイル名
 ファイル名は、a-z、A-Z、O-9、(ピリオド)、_(アンダスコア)、+、-のいずれかの文字で構成されます。
 ファイル名の先頭と最後に、.(ピリオド)の文字は使えません。
 ファイル名の先頭に「+」(プラス)/「-」(マイナス)は使えません。
 英大文字(A - Z)、英小文字(a - z)は区別されません。
 ファイル名は、パスを含めて最大259 文字です。

ファイル名が同じソース・ファイルは使用しないで下さい。異なるパスに存在していても区別できません。

・上記以外のファイル名

Windows のファイル名規約に準拠します。

なお、ファイル名には次の文字は使えません。

¥ / : * ? " < > | ;

ファイル名の先頭と最後に.(ピリオド)とスペースは使えません。

英大文字(A-Z), 英小文字(a-z)は区別されません。

ファイル名は、パスを含めて最大259文字です。

・フォルダ名

Windows のファイル名規約に準拠します。

なお、ファイル名には次の文字は使えません。

() , =

5.1.2 パネル表示に関する注意事項

使用するハードウエア環境が CS+ の推奨サポート環境を下回るスペックである場合, [プロパティ]パネ ルのサイズを小さくすると表示内容が乱れることがあります。

その場合には、分割パネル領域から[プロパティ]パネルを外に出してください。

- ・ドッキング可能をONにして、ドッキング・パネル化する
- ・フローティングを ON にして、フローティング・パネル化する

RENESAS

5.1.3 ユーザーアカウント制御(UAC)機能に関する注意事項

Windows Vista / Windows 7 において UAC 機能を無効にした場合,管理者権限をもたないユーザでプロジェ クトを作成や開いた場合で,かつ,デバイス依存情報をインストールしていない場合,デバイス依存情報のイ ンストールが開始されますがインストールに失敗します。UAC 機能を無効にする場合は,管理者権限でログ インしてプロジェクトを作成してください。

5.1.4 分割パネル・カテゴリに含まれるコマンドのアクセラレータに関する注意 事項

分割パネル・カテゴリに含まれるコマンドのメニューにアクセラレータが表示されているが、キーを押して も反応しません。メニューを使用する場合には、マウスで選択してください。

5.1.5 Windows の更新プログラムに関する注意事項

マイクロソフト株式会社より公開された, Windows 用の更新プログラム(KB2393802)を適用している場合, パソコンがブルースクリーンになる障害に該当することがあります。この障害に対しては, パソコン等の 各メーカより提供される修正プログラムを適用してください。

5.1.6 弊社製リアルタイム OS に関する注意事項

弊社製の RX ファミリ用のリアルタイム OS を使用する場合には、CS+のインストール・フォルダを括弧が ないフォルダに変更してインストールしてください。64bit 版の Windows にインストールする場合には、 ¥Program Files (x86) がデフォルトのインストール・フォルダになり、フォルダ名に括弧がある場合エラーに なります。

5.1.7 マイクロコントローラ変更に関する注意事項

マイクロコントローラを変更する場合には、次の注意事項があります。

- ・ 同じファミリ(RH850, RX)内の,同じビルド・ツールに対応しているマイクロコントローラへのみ 変更が可能です。
- ・ マイクロコントローラを変更する際は、デバッグ・ツールを接続していない状態にしてください。
- · マイクロコントローラを変更する前に、プロジェクトを保存する必要があります。
- ・ 端子配置(設計ツール),コード生成(設計ツール),デバッグ・ツール(ウォッチ登録情報除く)の 情報は、マイクロコントローラの変更後、引き継がれません。

5.1.8 プラグイン管理機能に関する注意事項

プラグインの管理ダイアログの基本機能タブにおいて,開発対象となるマイクロコントローラ用プラグイン のチェックは,外さないことを推奨します。

開発対象ではないマイクロコントローラ用のビルド・ツール・プラグイン,デバッグ・ツール・プラグイン のチェックを外してください。たとえば、ビルド・ツール・プラグインのみチェックを外すとデバッグツール でダウンロードするファイルが見つからずエラーとなります。



- 5.1.9 エディタ・パネルに関する注意事項
 - ページ設定ダイアログが使用できません。
 - ・ 印刷プレビューのツールバーにコピーボタンがありますが、使用できません。
 - ・ 変数, ラベルを選択して, コンテキストメニューの「関数ヘジャンプ」機能を使用した場合, 変数, ラベルにジャンプしません。
 - ・ 関数ヘジャンプ機能で、別ファイルに定義されている static 関数には移動できません。
 - メインプロジェクトとサブプロジェクトに、パスの違う同名のソースファイルが登録されていて、メインプロジェクトとサブプロジェクトのロードモジュールを両方ダウンロードしたとき、次のようになります。
 - ▶ 当該ファイルでは、メインプロジェクトのアドレスが表示される
 - 当該ファイルの逆アセンブルから「ソースへジャンプ」を行うと、メインプロジェクトに登録されているファイルが開く
 - ▶ どちらのプロジェクトから当該ファイルを開いても1つのファイルしか開けない
 - ・ 無名の構造体ではスマートエディットが正しく動作しません。
 - ・ 関数の引数に関数呼び出しが含まれる場合、ツール・チップに誤った情報が表示されます。
 - クラスの配列やクラスのポインタの配列に対して、メンバ変数、メンバ関数の補完が正しく動作しません。
 - ・ メンバ名を途中まで入力して、ctrl+"を入力しても補完機能が正しく動作しません。
 - アウトライン(折り畳み/展開)は、プロジェクトに登録したファイルのみが対象です。そのため、プロジェクトに登録していないファイルを CS+のエディタで表示しても、アウトライン表示にはなりません。
 - ・ Windows 8 では、アンチエイリアスが有効になり表示が不鮮明になる場合があります。
 - ・ 混合表示モードにて、行番号を指定してジャンプすると、逆アセンブルのコードが挿入されて表示されるため、指定行が画面上に表示されない場合があります。
 - ・ 構造体がネストしている場合に、3段目以降は、スマート・エディットの機能は使用できません。また、 ツール・チップの情報も表示されません。
 - 「#ifdef #endif」の直後のコードでは、スマート・エディットでメンバの候補が表示されません。また、 ツール・チップが表示されません。
 - 「#ifdef #endif」中の最初の変数については、「#endif」以降のコードで、スマート・エディットでメンバの候補が表示されません。また、ツール・チップが表示されません。
 - プロジェクトに含まれないファイルを開き、ブックマークを設定し、プロジェクトを閉じます。そして、
 そのファイルのブックマークの設定を変更した後、プロジェクトを開きなおします。ブックマークダイ
 アログを開いたとき、ブックマークダイアログには、プロジェクトを閉じたときのブックマークが表示
 され、ソース・ウインドウでは、プロジェクトを閉じた後のブックマークが表示され、差異が生じます。
 このような場合には、一旦、ファイルを閉じ、再度開いてください。ブックマークダイアログに表示されていたブックマークがソース・ウインドウに表示されます。
 - 矩形選択(Alt キーを押下してマウスによる範囲選択)を行い、最終行以降に行が追加されて貼り付け される場合、貼り付け位置によらず行の先頭から貼り付けられます。貼り付け後に必要なスペースを挿 入してください。
 - ・ 名前を付けて保存ダイアログにおいてファイルを保存する場合,拡張子を入力しない場合には、ファイ ルの種類ドロップ・ダウン・リストで選択されている最初の拡張子が自動的に付加されます。ただし、

ファイルの種類ドロップ・ダウン・リストで選択されている拡張子および Windows で登録されている 拡張子を付加してファイル名を入力した場合,拡張子は付加されません。 自動的に拡張子が付加されてしまった場合は,エクスプローラー等でファイルをリネームしてください。

5.1.10 PM+から CS+プロジェクトへの移行に関する注意事項

PM+ V6.00/V6.10/V6.11 で作成したCA850のプロジェクトに対して、ビルド・モードを新規追加した場合、 そのプロジェクトを CS+で読み込むと以下のようになります。

1)Debug Build または Release Build が選択されている場合:

新規追加したビルド・モードの情報が変換されません。

2)新規追加したビルド・モードが選択されている場合:

エラーとなります。

回避策として, PM+ V6.20 以上でプロジェクトを開いて保存し, 保存後のプロジェクトを CS+で読み込んで 下さい。

5.1.11 プロジェクト流用時のデバッグ・ツールの設定に関する注意事項

プロジェクトを流用作成する時,作成するプロジェクトにてデフォルトで選択されているデバッグ・ツール に対してのみ,流用した設定を反映します。

ただし, RX ファミリについては, 内部処理がエミュレータ, シミュレータで共通となっている為, デバッグ・ツールの選択状態に関わらず流用した設定を反映します。

5.1.12 オンライン・ヘルプに関する注意事項

オンライン・ヘルプにおいて、検索タブ(S)を表示した状態で閉じ、再度オンライン・ヘルプを表示し、目次(C)タブを表示した場合、コーディング編とビルド編が表示されない場合があります。

このようになった場合には、目次(C)タブを表示したままオンライン・ヘルプを閉じてから、再度オンライン・ヘルプを表示しなおしてください。

5.1.13 プロジェクト変換時の注意事項

High-performance Embedded Workshop / PM+ / 旧 CubeSuite を開いた時の〔プロジェクト変換設定〕ダイ アログで、プロジェクトの変換先デバイスを切り替えた時、〔プロジェクトの種類〕で選択されていた値を初 期値であるコンボボックスの先頭の値へ戻ります。

例えば、デバイスを選択し直すとプロジェクトの種類が先頭の(例えば〔アプリケーション〕)に切り替わります。



5.1.14 High-performance Embedded Workshop プロジェクト変換時の注意事項 High-performance Embedded Workshop のプロジェクトを CS+環境で読み込んだ場合、プロジェクト変換 ができずエラーとなったり、ビルド実行時にエラーが発生する場合があります。

(1) CS+用のプロジェクトへ変換ができない

- ・ ルネサス エレクトロニクス社製ツールチェインが使用されていないプロジェクト
- High-performance Embedded Workshop 環境の設定ファイル(tps ファイル)が存在していない プロジェクト(tps ファイルは、High-performance Embedded Workshop 環境で一度開くと自動生成さ れます。) プロジェクト変換前に一度プロジェクトを High-performance Embedded Workshop 環境で開くことで 解決できます
- ルネサス エレクトロニクス社製リアルタイム OS の設定ファイル (CFG ファイル)が複数存在してい るプロジェクト

(2) CS+用のプロジェクトへ変換はできるが、ビルド実行でエラーが発生

プレースホルダ(\$(TCINSTALL))を使用しているプロジェクト
 \$(TCINSTALL)は、変換後のプロジェクトにそのまま残ります。

CS+は、\$(TCINSTALL)を解釈できません。オプションのパラメータに\$(TCINSTALL)を使用していた場合は、そのままオプションに渡されますので意図したビルド結果を得られない可能性があります。(ビルドでエラーが発生するなど) \$(TCINSTALL)をプロジェクト変換後に、お客様自身で変更してください。

・ プレースホルダ(\$(WORKSPDIR))を使用しているプロジェクト
 プロジェクト・ファイル(拡張子 hwp)を指定して変換した場合,「%ProjectDir%¥..」
 (プロジェクトフォルダの1つ上のフォルダ)に固定で変換します。

プロジェクトフォルダの1つ上のフォルダにワークスペースがない場合は、正しいフォルダを示さなく なりますので、ビルドでエラーが発生することがあります。 その場合、プロジェクト変換後に「%ProjectDir%¥..」を、お客様自身で変更してください。

カスタムビルドフェーズを使用しているプロジェクト
 カスタムビルドフェーズは、削除されます。

カスタムビルドフェーズは、ビルド時に実行されなくなります。 よって、カスタムビルドフェーズで生成されたファイル出力を使用している場合はビルドエラーとなる 可能性があります。 プロジェクト変換後に、カスタムビルドフェーズのコマンドを、各フェーズの前後実行コマンドに必要 に応じて登録してください。

RENESAS

カスタムプレースホルダを使用しているプロジェクト
 カスタムプレースホルダは変換しません。

CS+は、カスタムプレースホルダを解釈できません。オプションのパラメータにカスタムプレースホル ダを使用していた場合は、そのままオプションに渡されますので意図したビルド結果を得られない可能 性があります。(ビルドでエラーが発生するなど)

プロジェクト変換後に、カスタムプレースホルダを、お客様自身で変更してください。

(3) その他

(a)\$(FILEDIR)は、%FileDir%へ変換します。

変換後、パス編集ダイアログでパス編集するとき、%FileDir%により以下のエラーが発生します。

指定したパスに存在しないフォルダが含まれています。(W0205012)

プロパティを編集する場合、%FileDir%を別のプレースホルダまたはディレクトリに置き換えてください。 (b)\$(WINDIR)は、%WinDir%へ変換します。

(c)フォルダの表示順が異なる場合があります。

(d)High-performance Embedded Workshop のプロジェクトでダウンロードファイルを指定している場合,

変換後は各デバッグ・ツールのダウンロードファイル一覧の2番目以降に登録します。

- (e)コンパイル・オプション-output=srcは,-output=obj(デフォルト)へ変換します。
- (f)ライブラリ・プロジェクトを変換時、ライブラリが標準ライブラリをリンクしていた場合は、そのリンク 設定を破棄します。(変換ログに出力されます)
- (g)ライブラリジェネレータで「既存標準ライブラリファイル指定」を指定していた場合、「標準ライブラリ・ ファイル指定なし」に変更します。結果、指定していたライブラリがリンクされません。(変換ログに出 力されます)

(h)High-performance Embedded Workshop の「全般」タブのオプションは変換せずに破棄します。

- (i)リンカでサブコマンドファイルを指定していた場合、変換後「サブコマンドファイルを使用する」設定を 破棄してリンカのオプション設定をデフォルトにします。
- (j)リンカの-library, -input, -binary オプションで指定したファイルは、リンク順設定ダイアログのファイル リストに表示しません。リンク順の指定対象外になります。
- (k)RTOS のコンフィグレーションファイルは、変換後に「Configuration file」カテゴリ・ノードの下に表示 しません。
- (I)RTOS オプションは変換せずに破棄します。オプション設定はデフォルトになります。
- (m)RTOS プロジェクトのビルド・モードは、プロジェクト変換後「DefaultBuild」になります。
 変換後、ビルドモードを変更してください。
- (n)RTOS プロジェクトのアセンブラ出力ファイル(ritbl.obj)のリンク順がプロジェクト変換後
 High-performance Embedded Workshop と異なります。



5.1.15 新規にプロジェクトを作成する際の注意事項

【対象】RX

RX 開発環境にて、"空のアプリケーション(CC-RX)"のプロジェクトを作成してビルドした場合、

以下のエラーが出る場合があります。

** L2132 (E) Cannot find "D" specified in option "rom"

** L2132 (E) Cannot find "D_1" specified in option "rom"

** L2132 (E) Cannot find "D_2" specified in option "rom"

エラーが発生した場合は, リンク・オプションの"ROMから RAM ヘマップするセクション"の設定を変更し てください。

5.1.16 マイクロソフト株式会社 IME に関する注意事項

マイクロソフト株式会社製の Office 2010 付属の Microsoft Office IME 2010 を使用している場合に, CS+使 用時に, E2000006 エラーが出力される場合があります。

Microsoft Office IME 2010 に起因する可能性がありますので、Windows 標準の IME に戻すか、マイクロソ フト株式会社より提供されている Microsoft Office IME 2010 の KB2687611 を解決するためのモジュールを インストールしてください。

5.1.17 チュートリアルの注意事項

チュートリアルでは、コード生成プラグイン、端子配置プラグイン、プログラム解析プラグインを使用しま す。プラグイン管理ダイアログで使用するプラグインを有効にして下さい。

5.1.18 CS+の複数起動の注意事項

CS+は、同じホストマシン上で複数起動が可能ですが、次の注意事項があります。

- ・CS+を複数起動した場合、パソコンのユーザ毎の情報ファイルは最後に書き込んだ情報が保存されます。
- ・CS+を複数起動した場合、スタック見積もりツール(CallWalker 含む)の情報ファイルは最後に書き込んだ情報が保存されます。
- ・複数起動した CS+で, 同一プロジェクト・ファイルを使用した場合, 最後に書き込んだ情報が保存され ます。
- ・複数起動した CS+で,同一プロジェクト・ファイルを使用した場合,同時にビルドしないでください。 出力ファイルが同一のためです。

5.1.19 CS+を起動する際のオプションの注意事項

CS+W.exe を起動する際に、オプションを指定できますが、次のオプションを指定しないでください。正常なエディタ機能が動作しなくなります。

- ▶ /npall オプション
- /npオプションでエディタを指定



5.2 設計ツールの注意事項

5.2.1 パッケージの変更に関する注意事項

端子配置のプロパティでパッケージ名を変更した場合,端子配置図および端子配置表の入力データはクリア されます。

5.2.2 プロジェクト保存に関する注意事項

サブプロジェクトが存在するプロジェクトにて,端子配置図または端子配置表パネルが開いた状態でプロ ジェクトの保存を行った場合に,プロジェクト・ツリー上の最後のサブプロジェクトの端子配置図,端子配置 表が必ず表示されます。

5.3 デバッグ・ツールの注意事項

文中において以下の略称を使用しています。 OCD(シリアル) : E1 エミュレータ(シリアル), E20 エミュレータ(シリアル) OCD(JTAG) : E1 エミュレータ(JTAG), E20 エミュレータ(JTAG)

5.3.1 サブプロジェクトの追加に関する注意事項

【対 象】 全デバッグ・ツール, 全デバイス共通

メインプロジェクトと異なるデバイスを扱うサブプロジェクトを追加する場合,デバッグ・ツールを切断し てから行ってください。

5.3.2 低消費電力モードに関する注意事項

【対 象】全デバッグ・ツール、RX

スリープモード,ストップモードおよびスタンバイモードなどの低消費電力モード中に強制ブレークを行った場合や,ステップ実行で低消費電力モードに移行する命令を実行した場合,シミュレータとエミュレータでは以下のような動作の差があります。

- ・エミュレータ:強制ブレークにより低消費電力モードは<u>解除されます</u>。また、ステップ実行では低消費電力 モードに<u>移行します</u>。
- ・シミュレータ:レジスタなどによる低消費電力モードへの移行はサポートしていません。WAIT 命令実行時にはブレークし、PC は次の命令のアドレスとなります。また、ステップ実行では低消費電力モードに移行せず、PC は次の命令のアドレスとなります。

5.3.3 任意区間のトレースに関する注意事項

【対象】シミュレータ、全デバイス共通

トレース開始イベントからトレース終了イベントまでをトレースする場合,シミュレータではトレース終了 イベントがトレース結果として表示されません。このため,シミュレータを使用する場合はトレース終了イベ ントをトレース・データとして表示させる範囲の1行下に設定してください。 5.3.4 サブプロジェクトの追加について

【対 象】全デバッグ・ツール, 全デバイス

デバッグ・ツール接続中にサブプロジェクトを追加すると、ダウンロード等に失敗することがあります。サ ブプロジェクトの追加は、デバッグ・ツール切断中にしてください。

5.3.5 ブレークポイントの設定等が不正になる注意

【対象】全デバッグ・ツール、全デバイス

関数名や変数名を,先頭のアンダー・バーの有無などで使い分けている場合,デバッガが誤認識してしまい, シンボル変換や,ブレークポイントの設定が不正になる場合があります。

例えば_reset と__reset という2 つの関数が存在していた場合などが該当します。

5.3.6 同名の変数の取り扱いに関する注意事項

【対象】全デバッグ・ツール、RX

異なるソースファイルに無名名前空間を記述し、その中に同名の変数を定義した場合、ウォッチパネルでは、 最初に見つかる変数の情報を表示します。

5.3.7 メンバ変数ポインタの取り扱いに関する注意事項

【対 象】全デバッグ・ツール, RX

下記のプログラムに定義されたメンバ変数ポインタ"mp1"をウォッチパネルおよびローカル変数パネルに登録した場合,型名に"int Foo::*"ではなく"int *"と表示されます。

class Foo {
 int m1;
};

int Foo::*mp1 = &Foo::m1;

5.3.8 レジスタ割付された共用体の取り扱いに関する注意事項

【対 象】全デバッグ・ツール, RX

共用体がレジスタに割り付いている場合,共用体のメンバはレジスタの下位バイトから割り付いているとみ なします。このため、ビッグエンディアンの場合はメンバの値を正しく表示できません。

5.3.9 char 型の引数を持つ同名の関数の取り扱いに関する注意事項

【対 象】全デバッグ・ツール, RX

下記のように char 型を使用した 3 つの関数を定義した場合, "Func(signed char)"のアドレスを正しく表示 できません。("Func(char)"のアドレスを表示します。)

void Func(char);

void Func(signed char);

void Func(unsigned char);



5.3.10 char 型の一次元配列の取り扱いに関する注意事項

【対象】全デバッグ・ツール, RX

下記のような char 型の一次元配列がレジスタやメモリの複数個所に割り付いていた場合は、ウォッチパネ ルおよびローカル変数パネルに配列"array"を登録しても値のカラムに文字列を表示できません。("" が値の カラムに表示されます。)

char array[5] = "ABCD";

5.3.11 オーバーレイ・セクションの優先セクションの変更に関する注意事項 【対象】全デバッグ・ツール, RX オーバーレイ・セクションの優先セクションを変更しても、デバッガの機能には直ぐには反映されません。 例えば、エディタ上のアドレス表示については、ファイルを一旦閉じ、再度開くことにより反映されます。 また、ウォッチパネル上の変数表示については、1回ステップを実行することにより反映されます。

5.3.12 レジスタ割付された変数の取り扱いに関する注意事項

【対象】全デバッグ・ツール、RX

ローカル変数パネルの[スコープ]にて"カレント"以外を選択中は、レジスタに割りついた変数の値は正しく 表示できません。また、その変数の値を編集することも出来ません。

5.3.13 変数の割り付き位置表示の取り扱いに関する注意事項

【対象】全デバッグ・ツール, RX

以下の条件を全て満たす変数を定義した場合、ウォッチパネル、ローカル変数パネルでは、対象のメンバ変 数の割り付き位置文字列が変数全体の割り付き位置文字列で表示されます。

<条件>

(1)定義した変数が複数のアドレスやレジスタに割りついている。

(アドレスカラムに2つ以上のアドレスやレジスタ名が表示される場合)

(2)変数に以下の型のメンバが定義されている。

- 構造体、クラス、配列、共用体のいずれか

```
<例>
struct Mem {
long m_base;
};
struct Sample {
long m_a;
struct Mem m_b; <-条件(2)に該当
};
main () {
struct Sample obj;
}
```



表示結果:

"obj"	-	{ R1:REG, R2:REG }	(struct Sample)
L m_a	0x00000000	{ R1:REG }	(long)
L m_b	-	{ R1:REG, R2:REG }	(struct Base)
L m_base	0x00000000	{ R2:REG }	(long)

5.3.14 変数をキャストする際の取り扱いに関する注意事項

```
【対 象】全デバッグ・ツール, RX
```

```
ウォッチパネルで変数を他の型へキャストした場合、Cスタイルのキャストを行ないます。
```

このため、例えば仮想継承クラスの基底クラスへのキャスト結果は、プログラム中で記述したキャスト結果 と異なります。

```
class AAA [
```

int m_aaa;

```
} objA;
```

class BBB : public AAA { //BBB は AAA を継承している

int m_bbb;

} objB;

```
class CCC { //CCC は AAA を継承していない
```

int m_ccc;

} objC

```
class AAA* pa = objA;
```

class BBB* pb = objB;

```
class CCC* pc = objC;
```

```
"(AAA*)pa" ・・・使用可能
```

- "(BBB*)pb" ・・・使用可能
- "(AAA*)pb" ・・・使用可能
- "(CCC*)pc" ・・・使用可能
- "(AAA*)pc"・・・pcの指すアドレスを型"AAA"の先頭アドレスと見做してキャストします。
 - プログラミング上のキャストイメージ: (AAA*)((void*)pc)

5.3.15 PC スリープ状態からの復帰に関する注意事項

【対 象】OCD(JTAG), OCD(シリアル), RX

Windows Vista または Windows 7 でデバッグ中に PC がスリープ状態または休止状態に移行した場合,復帰後にデバッグを継続できません。

Windows Vista または Windows 7 で使用する場合は, PC がスリープ状態および休止状態に移行しない設定 でご使用ください。



5.3.16 プログラム実行中のトレース停止,再開の注意事項

【対 象】全デバッグ・ツール, RX

トレース開始イベント, あるいはトレース終了イベントを設定している場合, プログラム実行中のトレース停止・再開はできません。

5.3.17 トレースのタイムスタンプについての注意事項

【対 象】OCD(JTAG), OCD(シリアル), RX

トレース情報に付加されるタイムスタンプは、フレーム間の経過時間がトレースクロックの 20 ビット分を超 える場合、および、トレース出力でロストが発生した場合、正しい時間となりません。

5.3.18 CC-RX コンパイラのリンクオプションの注意事項

【対 象】全デバッグ・ツール, RX

CC-RX コンパイラはリンク・オプション-sdebug には対応していません。

デバッグの際は, CC-RX コンパイラプロパティの[リンク・オプション]タブ→[出力カテゴリ]→[デバッグ情報 を出力する]を-debug に設定してください。

5.3.19 リターンアウト実行時の注意事項

【対象】全デバッグ・ツール, RX

再帰呼び出しの関数でリターンアウト実行した場合,呼び出し元ではなく呼び出し先関数のリターン命令を 実行したところで停止する場合があります。

5.3.20 スタートアッププログラム保護機能の注意事項

【対象】OCD(シリアル), RX100

ユーザ・プログラム実行中に下記いずれかを実施してブレーク後、「CPU のリセット」をすると、デバッガ が表示する ROM 内容と MCU の ROM 内容が一致しません。

この場合、再度ユーザ・プログラム実行して停止すると一致します。

・R_FCL_ChangeSwapState 関数をコールして即座にスワップする

・フラッシュ初期設定レジスタ(FISR)を操作して即座にスワップする

5.3.21 カバレッジ計測機能に関する注意事項

【対 象】E20エミュレータ(JTAG)、RX64M

(1) ホットプラグイン接続ではカバレッジ計測できません。

ホットプラグイン接続する場合は、[デバッグ・ツール設定]タブ上の[カバレッジ]カテゴリで、[コード・カ バレッジ機能を使用する]を[いいえ]に設定してください。

(2) プログラム実行中に「システムリセットを発行しました。」というエラーメッセージが表示された場合、 プログラム実行開始からシステムリセット発生時点までの実行はカバレッジ測定されません。



5.3.22 for 文やインライン関数内のブレークポイント設定に関する注意事項 【対象】 全デバッグ・ツール、RX

以下のプログラムをCソースに記述するとソースプログラム1行に対する命令が複数箇所に配置されます がエディタ上にはそのうちの1箇所のアドレスのみを表示しています。

この行に対しブレークポイントを設定した場合、ブレークするのはエディタに表示したアドレス実行時の みとなります。

- 1. インライン関数(*1)
- 2. テンプレート関数
- 3. for 文、do-while 文の先頭行
- *1:最適化によりインライン展開された関数も含みます

5.3.23 DMAC/DTC のトレース設定に関する注意事項

【対 象】E20エミュレータ(JTAG)、RX64M

デバッグ・ツールプロパティの[デバッグ・ツール設定]タブ内の[トレース]カテゴリの[データアクセスのバス・マスタ]を[DTAC/DTC]に設定した場合、[トレース]カテゴリの[外部トレース出力]に[トレース出力優先]を 設定するとトレースが正常に動作しない可能性があります。

[CPU 実行優先]または[外部出力しない]を設定してください。

5.3.24 PLL 動作時のメイン・クロック・ソース設定に関する注意事項

【対 象】OCD(JTAG), OCD(シリアル), RX64M

ユーザ・プログラムで PLL を使用する場合、以下の設定を行ってください。

・デバッグ・ツールプロパティの[接続用設定]タブの[クロック]カテゴリの[メイン・クロック・ソース]を [EXTAL]に設定してください。

・PLL のクロックソースに EXTAL を使用する場合は、[メイン・クロック周波数]に EXTAL の周波数を設 定してください。

・PLL のクロックソースに HOCO を使用する場合は、[メイン・クロック周波数]に HOCO の周波数を設定 してください。

5.4 解析ツールの注意事項

5.4.1 解析グラフパネルに関する注意事項

- ・ 解析グラフの実行時間の割合は使用できません。(E1/E20,RH850)
- ・ 解析グラフの値の推移機能で、トレース・データ解析方式は使用できません。(E1/E20,RH850)
- エミュレータにて、内蔵トレースのタイム・タグをサポートしていない場合は、解析グラフパネルは使用できません。(E1/E20,RX)
- 値の推移グラフにて、デバッグ・ツールにシミュレータを指定している場合、IOR のリアルタイム・サンプリング方式はサポートしていません。
- 実行時間の割合に表示する結果は正確でない可能性があります。これはトレースのタイム・ラグ計測用 カウンタが小さくオーバフローする可能性があるためです。オーバフローしているかはトレースパネル のタイムスタンプで確認してください。(E1/E20,RX)

- 5.4.2 関数一覧/変数一覧パネルに関する注意事項(E1/E20,RH850)
 - ・ **関数一覧パネルで**,実行時間/実行時間(割合)/平均実行時間/コード・カバレッジはサポートされて いません。
 - 変数一覧パネルで、データ・カバレッジはサポートされていません。

5.5 Python コンソールの注意事項

5.5.1 日本語入力に関する注意事項

Python コンソールでは日本語入力機能を有効にする事ができません。日本語を入力する場合は、外部エディ タ等で作成しコピーし貼り付けてください。

5.5.2 プロンプト表示に関する注意事項

Python コンソールのプロンプトが>>>であるところが>>>>>というように複数表示される場合や>>> の後に結果が表示され、キャレットの前に>>>がない場合があります。このような状態でも継続して関数 を入力することが可能です。

5.5.3 フォルダやファイルへのパスに関する注意事項

IronPython では、¥(バックスラッシュ)を制御文字として認識します。例えば、先頭がtで始まるフォルダ名 やファイル名の場合¥tで TAB 文字と認識してしまいます。これを回避するには次のように、""(パス指定)の前 にrを記載してください。IronPython は""の中がパスと認識します。

- (例) r"c:¥test¥test.py"
- なお、パスの指定には¥(バックスラッシュ)ではなく/(スラッシュ)も使用可能です。

5.5.4 ロードモジュールがないプロジェクトのスクリプト実行に関する注意事 項

ロードモジュール・ファイルがないプロジェクトを使用して起動オプションでスクリプト指定した場合,も しくはプロジェクト・ファイル名.pyをプロジェクト・ファイルと同じフォルダにおいてある場合は,通常プ ロジェクト読み込み後に自動的にスクリプトを実行しますが,ロードモジュール・ファイルがない場合は実行 しません。

5.5.5 強制終了に関する注意事項

無限ループしているようなスクリプトを実行中に以下の操作を行うと,強制的に関数の実行を終了させるため,関数の実行結果がエラーになる場合があります。

1. Python コンソールのコンテキストメニューの「強制終了」や Ctrl+D で強制終了

2. 複数のプロジェクトをもつプロジェクトでアクティブプロジェクトを変更した場合

5.5.6 強制停止に関する注意事項

コンテキストメニューの[強制停止]を実行した場合,実行中のスクリプトや関数を強制停止しますが,[強制 停止]した時点で実行が開始していないHook 関数やCallback 関数がある場合は,[強制停止]後順次実行します。



5.5.7 ビルド中の Python コマンドの実行に関する注意事項 ビルド中に Python コマンドを使用しないでください。

5.6 I/O ヘッダ生成ツールの注意事項

5.6.1 I/O ヘッダ・ファイル生成オプション設定機能

ビルド・ツールのプロパティの 1/0 ヘッダ・ファイル生成オプションタブで、下記の 1/0 ヘッダ・ファイル 生成オプションの設定が可能です。

・ビルド時に1/0ヘッダ・ファイルを更新する

・その他の追加オプション

5.6.2 I/O ヘッダ・ファイルの生成

プロジェクト・ツリーのビルド・ツール・ノードのコンテキスト・メニューから、I/O ヘッダ・ファイル生 成を選択することで I/O ヘッダ・ファイルを生成することが可能です。



第6章 制限事項

本章では、制限事項について説明します。

6.1 デバッグ・ツールの制限事項

文中において,以下の略称を使用しています。 OCD(シリアル) : E1 エミュレータ(シリアル), E20 エミュレータ(シリアル) OCD(JTAG) : E1 エミュレータ(JTAG), E20 エミュレータ(JTAG)

6.1.1 デバッグ・ツールの制限事項一覧

No.	対象ツール	対象デバイス	制限事項	備考
1	OCD(シリアル)	RX64M	認証切れエラーに関する制限事項	
	OCD(JTAG)			

- 6.1.2 デバッグ・ツールの制限事項詳細
- No.1 認証切れエラーに関する制限事項
- 【対 象】OCD(シリアル)、OCD(JTAG) RX64M
- 【内 容】以下の条件を全て満たした場合、認証切れエラーが発生し、デバッグを継続することができなくな ります。
 - [条件]

1. ID コードにオール FF 以外を設定しているデバイスをユーザブートモードでデバッグしている。

2. オプション設定メモリ領域へのデータが含まれるプログラムをダウンロードした後に、リセット コマンドや端子リセットまたは内部リセットが発生した場合。

【回避策】回避策はございません。



第7章 ドキュメント訂正

本章では、CS+のドキュメントの訂正について説明します。

7.1 フック処理に関するドキュメント訂正事項

フック処理に関するドキュメント訂正について説明します。フック処理の説明は各種デバッグ編に 記載があります。

フック処理の設定は、デバッグ・ツールプロパティパネルの[フック処理設定]タブで行います。 フック処理を使用すると、ダウンロード前、実行開始前、リセット後等様々なタイミングで I/O レジスタの設定、CPU レジスタの設定、python スクリプトの実行が行え、以下が実現できます。

(1)プログラム開発中で、マイコンの I/O レジスタの設定プログラムが未完成でも、実行開始前に I/O レジスタ設定を行うことによりデバッグを行うことができます。

(2)ダウンロード前に I/O レジスタを設定することにより、ダウンロードを高速に行うことができます。

(3)ダウンロード前に I/O レジスタを設定することにより外部 RAM へのダウンロードが容易に行 えます。

デバッガのフック処理から Python スクリプトを実行する場合、以下のコマンドが記載可能です。

debugger.Register.GetValue debugger.Register.SetValue debugger.Memory.GetValue debugger.Memory.SetValue

それ以外の Python コマンドを使用したい場合 Python コンソールの Hook コマンドを使用してください。

7.2 CubeSuite+ V2.02.00 解析編ドキュメント訂正

CubeSuite+ V2.02.00 解析編(R20UT2868JJ0100)のドキュメントの訂正について説明します。

7.2.1 動的解析の設定方法に関して

(1)変更箇所 1

[場所] 10 ページ 1.1.2 解析情報の種別 (2)動的解析情報 (a) トレース機能 備考2の後

なし。

[変更後]

以下の内容を追加。

トレース機能を利用した動的解析の結果を取得するには、デバッグ・ツールのプロパティにて、設定を変更 する必要があります。以下に、各デバッグ・ツールと設定が必要なプロパティの一覧を示します。



[[]変更前]

デバッグ・ツール	プロパティ	説明
【RH850】	[デバッグ・ツール設定]タブ →	「はい」に設定してください。
【シミュレータ】	[トレース]カテゴリ →	
[RX]	[トレース機能を使用する]プロパティ(※)	
【シミュレータ】	[デバッグ・ツール設定]タブ →	リストから、トレースのサイズを選択し
	[トレース]カテゴリ →	てください。
	[トレース・メモリ・サイズ[フレーム]]プロパティ	
【RH850】	[デバッグ・ツール設定]タブ →	関数情報のみを取得する場合は、「分岐
[Full-spec Emulator]	[トレース]カテゴリ →	命令」を選択してください。
	[トレース・データの選択]プロパティ	変数情報のみを取得する場合は、[デー
		タ・アクセス]を選択してください。
		関数/変数情報を取得する場合は、[分
		岐命令とデータアクセス]を選択してく
		ださい。
	[デバッグ・ツール設定]タブ →	リストから、トレースのサイズを選択し
	[トレース]カテゴリ →	てください。
	[トレース・メモリ・サイズ[フレーム]]プロパティ	
【RH850】	[デバッグ・ツール設定]タブ →	関数情報のみを取得する場合は、「分岐
【E1】/【E20】	[トレース]カテゴリ →	命令」を選択してください。
	[トレース・データの選択]プロパティ	変数情報のみを取得する場合は、[デー
		タ・アクセス]を選択してください。
		関数/変数情報を取得する場合は、[分
		岐命令とデータアクセス]を選択してく
		ださい。
[RX]	[デバッグ・ツール設定]タブ →	関数情報のみを取得する場合は、「分
[E1] / [EZ Emulator]	[トレース]カテゴリ →	岐」を選択してください。
	[トレース・データ種別]プロパティ	変数情報のみを取得する場合は、[デー
		タアクセス]を選択してください。
		関数/変数情報を取得する場合は、[分
		岐 + データアクセス]を選択してくだ
		さい。
		ただし、選択可能な設定は、デバイスに
		依存します。
[RX]	[デバッグ・ツール設定]タブ →	「トレース」を選択してください。
[E20]	[トレース]カテゴリ →	
	[トレース機能の用途]プロパティ(※)	
	[デバッグ・ツール設定]タブ →	関数情報のみを取得する場合は、「分
	[トレース]カテゴリ →	岐」を選択してください。
	[トレース・データ種別]プロパティ	変数情報のみを取得する場合は、[デー
		タアクセス]を選択してください。
		関数/変数情報を取得する場合は、[分
		岐 + データアクセス]を選択してくだ
		さい。



CS+ for CC V3.00.00

リリースノート

	ただし、選択可能な設定は、デバイスに 依存します。
[デバッグ・ツール設定]タブ →	外部トレースの機能がサポートされて
[トレース]カテゴリ →	いる場合は、リストから、トレースのサ
[トレース・メモリ・サイズ[M バイト]]プロパティ	イズを選択してください。
	ただし、トレース機能をサポートしてい
	ないデバイスの場合は、本プロパティは
	表示されません。

(※)プログラム解析プラグインの[設定]タブ→[全般]カテゴリ→[動的解析を有効にする]プロパティを「はい」 に設定すると、デバッグ・ツールへの接続時に、自動的に設定を変更します。

(2)変更箇所 2

[場所] 11 ページ 1.1.2 解析情報の種別 (2)動的解析情報 (b) RRM 機能/RAM モニタ(疑似 RRM)

機能 注意の後

[変更前]

なし。

[変更後]

以下の内容を追加。 < RRM機能/RAMモニタ(疑似RRM)の設定>

RRM機能/RAMモニタ(疑似RRM)機能を利用した動的解析の結果を取得するには、デバッグ・ツールの プロパティにて、設定を変更する必要があります。以下に、各デバッグ・ツールと設定が必要なプロパティの 一覧を示します。

デバッグ・ツール	プロパティ	説明
【RH850】	[デバッグ・ツール設定]タブ →	「はい」に設定してください。
【シミュレータ】	[実行中のメモリ・アクセス]カテゴリ →	
[RX]	[実行中に表示更新を行う]プロパティ(※)	
【シミュレータ】	[デバッグ・ツール設定]タブ →	更新間隔を指定してください。
	[実行中のメモリ・アクセス]カテゴリ →	100~65500 の範囲で指定可能です。
	[表示更新間隔[ms]]プロパティ	
【RH850】	[デバッグ・ツール設定]タブ →	「はい」に設定してください。
【Full-spec Emulator】 /	[実行中のメモリ・アクセス]カテゴリ →	
【E1】/【E20】	[実行中にアクセスする]プロパティ(※)	
[RX]	[デバッグ・ツール設定]タブ →	「はい」に設定してください。
[E1]/[E20]/[EZ Emulator]	[実行中のメモリ・アクセス]カテゴリ →	
	[実行中に表示更新を行う]プロパティ	
	[デバッグ・ツール設定]タブ →	更新間隔を指定してください。
	[実行中のメモリ・アクセス]カテゴリ →	100~65500 の範囲で指定可能です。
	[表示更新間隔[ms]]プロパティ	
[RX]	[デバッグ・ツール設定]タブ →	「リアルタイム RAM モニタ」に設定
[E20]	[トレース]カテゴリ →	してください。
	[トレース機能の用途]プロパティ	



CS+ for CC V3.00.00

 「デバッグ・ツール設定]タブ →	「はい」に設定してください。
[美打中のメモリ・アクセス]カナコリ →	
[リアルタイム表示更新を自動設定する]プロパ	
ティ	
[デバッグ・ツール設定]タブ →	更新間隔を指定してください。
[実行中のメモリ・アクセス]カテゴリ →	100~65500 の範囲で指定可能です。
[表示更新間隔[ms]]プロパティ	

(※)プログラム解析プラグインの[設定]タブ→[全般]カテゴリ→[動的解析を有効にする]プロパティを「はい」に設定すると、デバッグ・ツールへの接続時に、自動的に設定を変更します。

(3)変更箇所 3

[場所]11 ページ 解析情報の種別 (2)動的解析情報 (c) カバレッジ機能 注意3の後

[変更前]

なし。

[変更後]

以下の内容を追加。

カバレッジ機能を利用した動的解析の結果を取得するには、デバッグ・ツールのプロパティにて、設定を変 更する必要があります。以下に、各デバッグ・ツールと設定が必要なプロパティの一覧を示します。

デバッグ・ツール	プロパティ	説明
【RH850】	[デバッグ・ツール設定]タブ →	「はい」に設定してください。
【シミュレータ】	[カバレッジ]カテゴリ →	
[RX]	[カバレッジ機能を使用する]プロパティ(※)	
【シミュレータ】		
【RH850】	-	カバレッジ機能はサポートしてい
【Full-spec Emulator】 /		ません。
【E1】 /【E20】		
[RX]		
【E1】 / 【EZ Emulator】		
[RX]	[デバッグ・ツール設定]タブ →	「はい」に設定してください。
[E20]	[カバレッジ]カテゴリ →	ただし、カバレッジ機能をサポート
	[コード・カバレッジを使用する]プロパティ	していないデバイスの場合は、本プ
		ロパティは表示されません
	[デバッグ・ツール設定]タブ →	カバレッジの計測範囲を指定して
	[カバレッジ]カテゴリ →	ください。
	[カバレジ計測範囲]プロパティ	

(※)プログラム解析プラグインの[設定]タブ→[全般]カテゴリ→[動的解析を有効にする]プロパティを「はい」 に設定すると、デバッグ・ツールへの接続時に、自動的に設定を変更します。

すべての商標および登録商標は、それぞれの所有者に帰属します。

ご注意書き		
1.	本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計におい	
	て、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三	
	者に生じた損害に関し、当社は、一切その責任を負いません。	
2.	本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報	
	の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。	
3.	本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権	
	に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許	
	諾するものではありません。	
4.	当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。	
5.	当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、	
	各品質水準は、以下に示す用途に製品が使用されることを意図しております。	
	標準水準: コンピュータ、OA機器、通信機器、計測機器、AV機器、	
	家電、工作機械、パーソナル機器、産業用ロボット等	
	高品質水準:輸送機器(自動車、電車、船舶等)、交通用信号機器、	
	防災・防犯装置、各種安全装置等	
	当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム(生命維持装置、人体に埋め込み使用するもの等) 、もしくは多大な物的損害を発生さ	
	せるおそれのある機器・システム(原子力制御システム、軍事機器等)に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用	
	途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。 なお、ご不明点がある場合は、当社営業にお問い	
6.	当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製	
_	品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。	
7.	当社は、当社製品の品質および信頼性の同上に努めていますが、半導体製品はある確率で改障が発生したり、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、当在製品は耐放射線設計については行っておりません。当在製品の改厚または誤動作が生した場合も、人身事故、火災事故、在会的損害等を生しさせ ないよう。な実体の表だにないて、同意部門、び体社体部門、調整化性に部門体の中へ部門などになって、どの原理体、な実体の機関、シュニノルトでの世界に対	
	ないよう、お各体の負任において、几長設計、延焼対策設計、誤測作防止設計等の女主設計およびエーシング処理等、お各体の機器・システムとしての工何休祉	
0	を行うしくたさい。特に、マイコンソフトウェアは、単独での検証は困難なため、お各様の機器・ンステムとしての女王検証をお各様の頁仕で行うしくたさい。 ※社制見の連接案会性策の詳細につきましては、制見用則に必ず※社営業空日までな問会せておさい。「使用に際しては、社会の物質の会友、使用を掲制する	
0.	ヨ社裂血の環境適当社寺の評種につきましては、裂血値がに必りヨ社呂来念いまでの向当せてたさい。こ実用に际しては、特定の物質の召有・実用を規制する Daulo指金箔、落田されて連接眼道は会ましい読木のまま、かかては会に落合すてしまった田ノポさい、か安祥がかかては会ま落立したいこしにトリナドも得実に	
	RODS拍中寺、週田される環境関連広中を十方調査のうえ、かかる広中に週日するようご使用ください。お各様かかかる太中を運ずしないことにより生した損害に 期上で、当社は、一根をの吉佐た色いません。	
0	周しし、当社は、一切ての貝讧を見いよと心。 太姿約に記載されている当社制具なとびは後た国内Mの注合なとび規則にとり制造・体田・販売た禁止されている機器・システムに体田することけできません。ま	
5.	本資料に記載されている当社委品のより次間を国行わりの本作のよりのよりが表記により表記。「使用」「熟児を来立されている版語」 アスナムに使用することはできません。よ た、当社創品お上げ技術を大量破壊反映の開発等の日的、軍事利用の日的その仲軍事田途に使用」 たいでください、当社創品またけ技術を輸出する場合け、「外	
	え、当社教師のない技術を八重吸る英語の病光等の目的、半季が小の日前での他半季が遮に使用しないでくたとい。当社教師なたは技術を補出する場合は、「パー国為基本の代表国智慧法」その他輸出関連法会を道守し、かかる法会の定めるところにより必要な手続を行ってください	
10	日初日次の7月日次初辺」での他物田肉座広りを建すり、2.2 などののためのためのためのためのためのためです。) お客様の転売等により、太ご注音書き記載の誘条件に折触して当社創品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負	
10	相して頂きますのでご了承ください。	
11	1. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。	
	注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社がその総株主の議決権の過半数	
	を直接または間接に保有する会社をいいます。	

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

RENESAS

ルネサスエレクトロニクス株式会社

http://www.renesas.com

٦

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

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

■営業お問合せ窓口

Г

© 2014 Renesas Electronics Corporation and Renesas Solutions Corp.

※営業お問合せ窓口の住所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。