RL78開発環境 移行ガイド

R8C、M16CからRL78への移行（オンチップ・デバッグ編）

ルネサス エレクトロニクス株式会社
MCU事業本部　ソフトウェア統括部
MCUツール技術部

2012/06/29　　Rev. 1.00

R20UT2150JJ0100

00000-A

はじめに

本資料はR8C/M16Cファミリ用High-performance Embedded WorkshopとRL78ファミリ用CubeSuite+との違いやCubeSuite+でのE1,E20エミュレータ操作方法について記載しています。

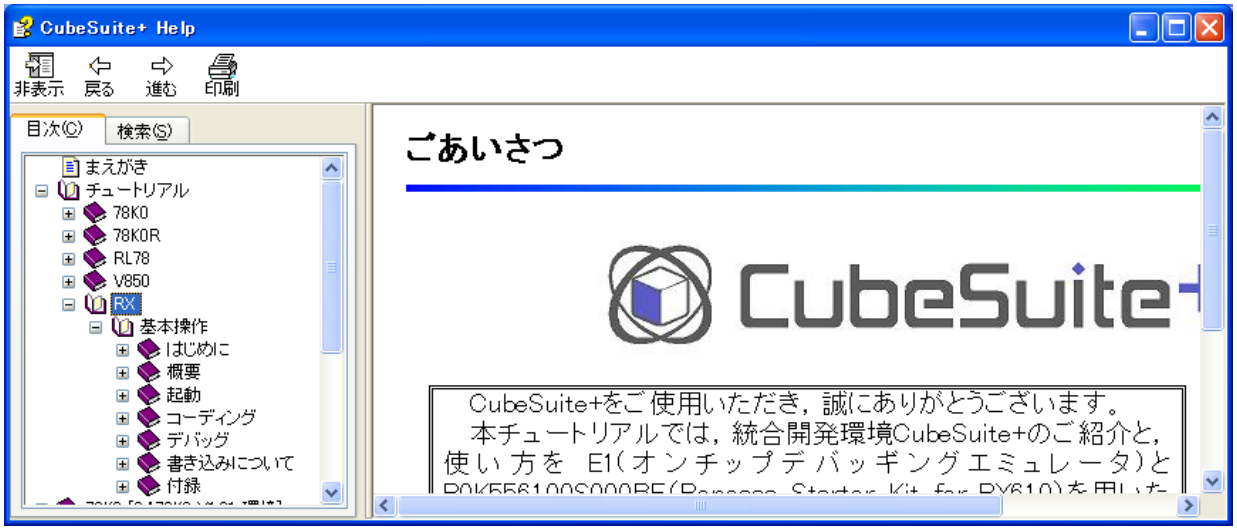
本資料はCubeSuite+V1.02.00をベースに説明しています。

ツールチェイン等について下記資料を参照ください。

- 「RL78開発環境移行ガイド R8C/M16C,H8S/H8SXから RL78への移行（コーディング編）」
- 「RL78開発環境移行ガイド R8C/M16C,H8S/H8SXから RL78への移行（ビルド編）」
- 「RL78開発環境移行ガイド R8C/M16C,H8S/H8SXから RL78への移行（起動編）」

また、CubeSuite+はツールの使い方を記載したチュートリアルガイドを用意していますので参照ください。

チュートリアルガイドはCubeSuite+のメニューから［ヘルプ］→［チュートリアル］を選択で参照できます。



チュートリアルガイド

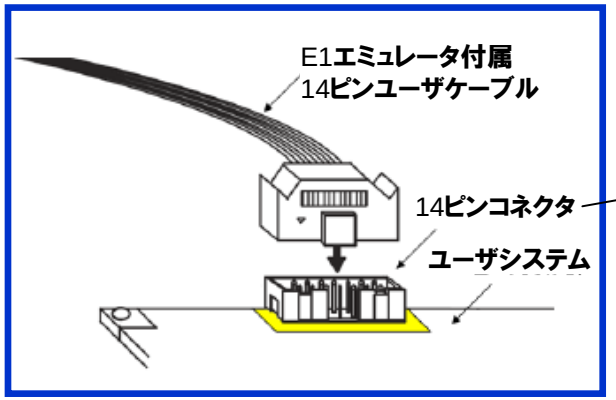
目次

1.統合開発環境とエミュレータ

	R8C,M16C	RL78
統合開発環境	High-performance Embedded Workshop4 High-performance Embedded Workshop	CubeSuite+ ルネサスの8ビットから32ビットマイコン用統合開発環境を1つに統一した新統合開発環境
オンチップデバッキングエミュレータ (OCD)	 E8a(R8C,M16C) E1(R8C)	 基本的なデバッグ機能を安価に提供 オンチップデバッギングエミュレータ E1
フルスペック・エミュレータ	CPE (R8C,M16C)   E100 (R8C,M16C)	 より高度なデバッグ機能を装備した高性能フルスペックエミュレータ フルスペックエミュレータ IECUBE

＊この資料ではOCDを使用する場合の移行方法を紹介しています。

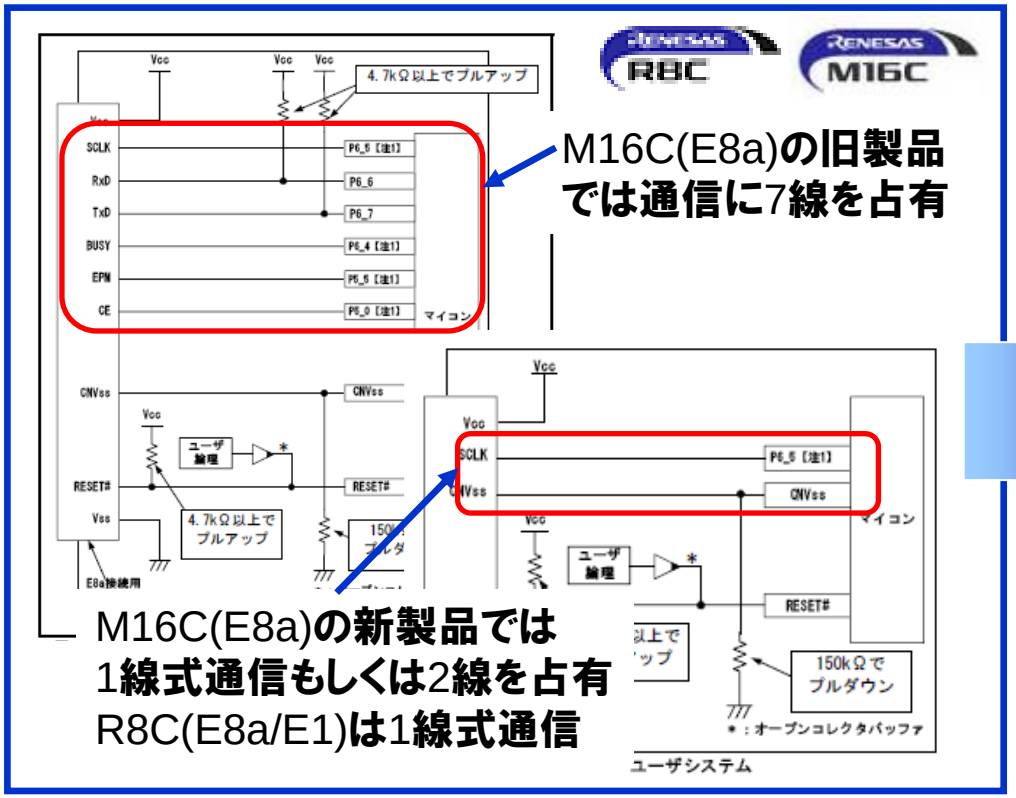
2.ターゲットインタフェースの違い(OCD)



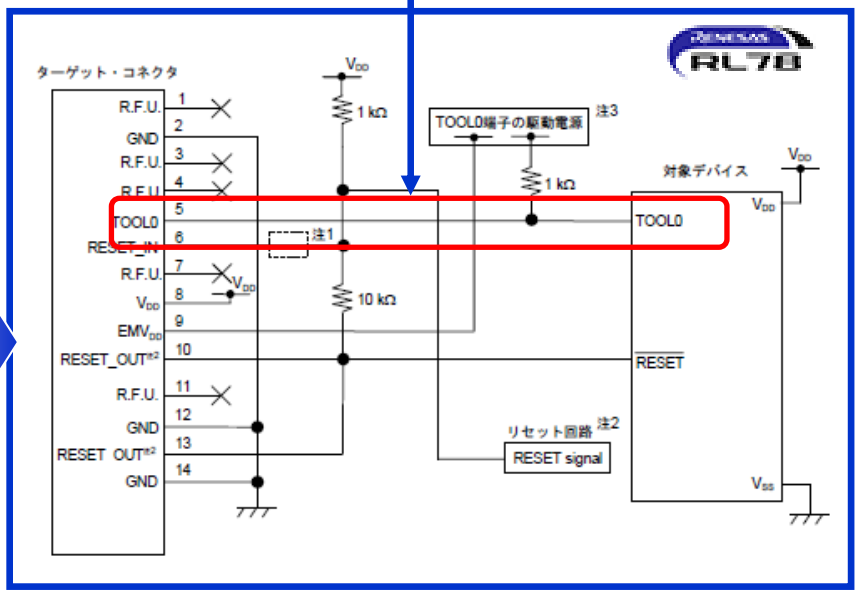
RL78とE1エミュレータは14ピンコネクタを使用して接続します。

	型 名	メーカ	仕様
14 ピン コネクタ	7614-6002	住友スリーエム株式会社	14 ピンストレートタイプ（国内推奨）
	2514-6002	3M Limited	14 ピンストレートタイプ（海外推奨）

コネクタはR8C,M16C用E8a/E1で使用しているコネクタと同じですが、下図のように通信インタフェースは異なります。



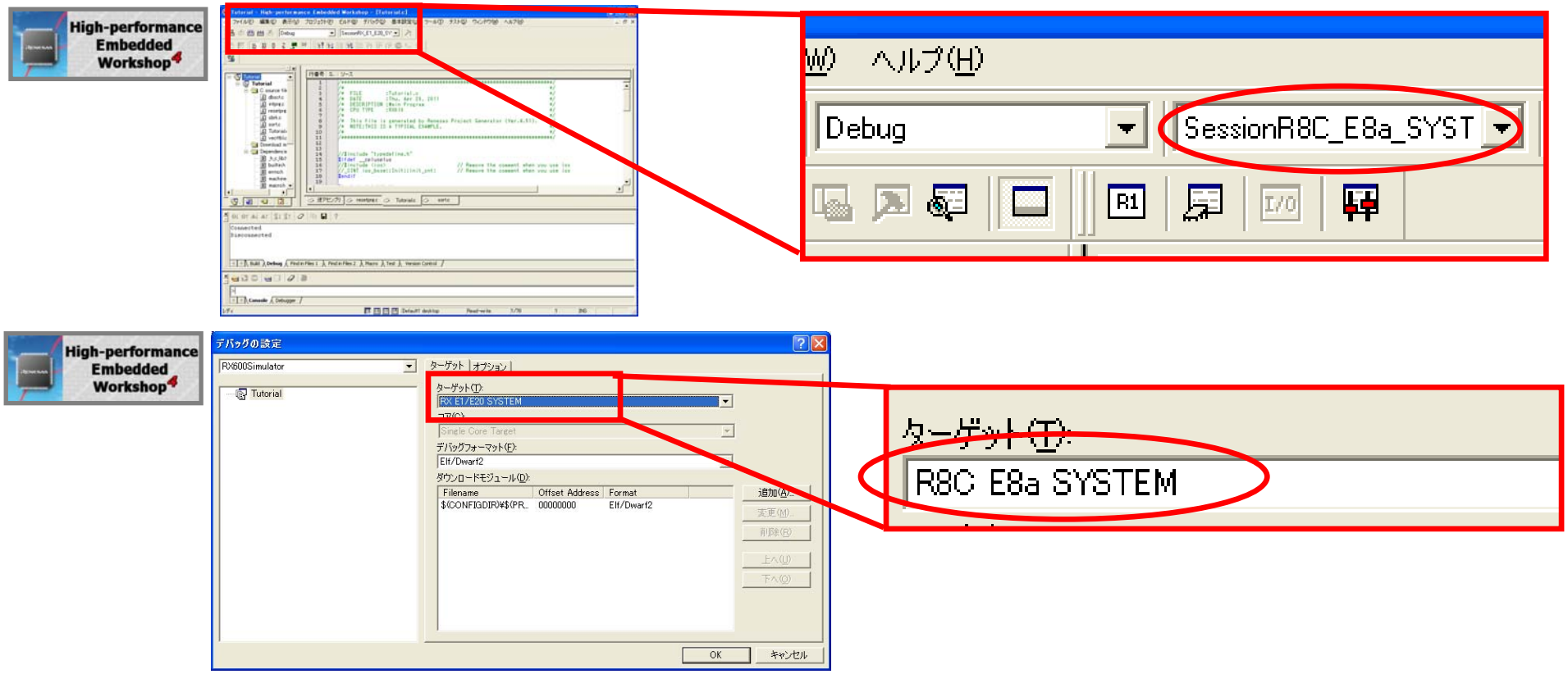
RL78(E1)は1線式通信



デバイスによっては結線回路が異なる場合があるので詳細はE1/E20エミュレータユーザーズマニュアル 別冊(RL78接続時の注意事項)を確認してください。

3.デバッガの変更方法

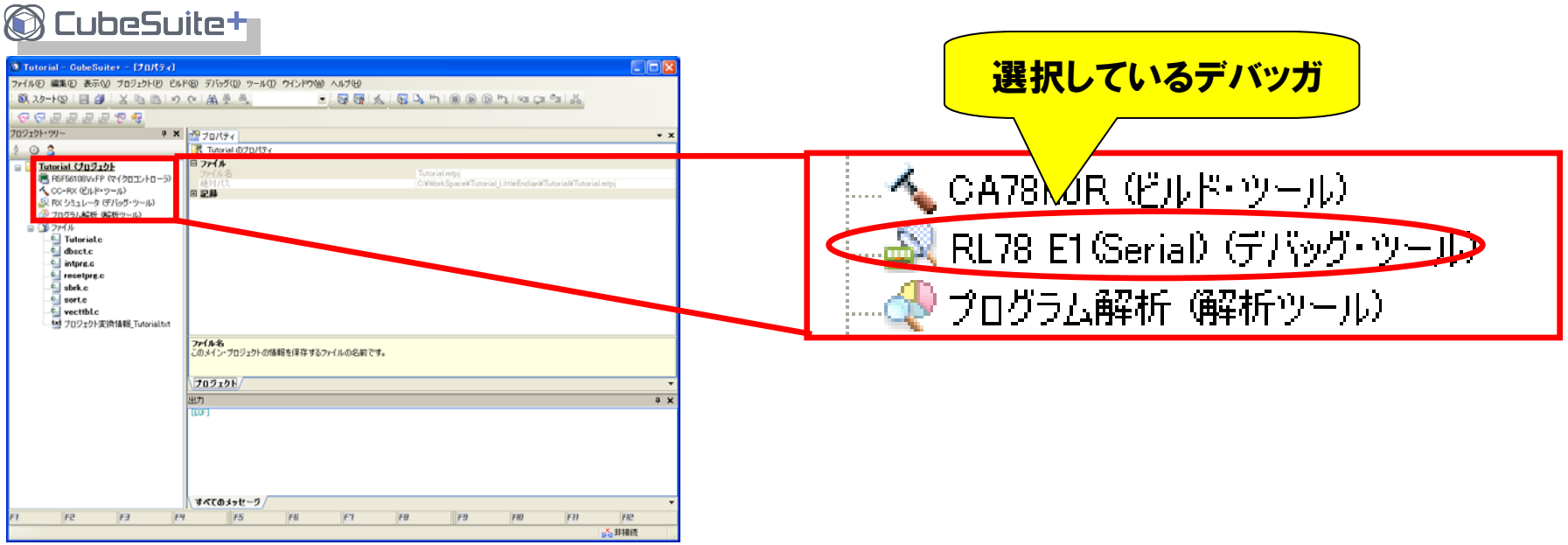
HEWではデバッグセッションの変更や [デバッグの設定] ダイアログのターゲットの変更でデバッガ (E8a、E1エミュレータやシミュレータ) の選択を行っていましたが、CubeSuite＋ではプロジェクトツリー上でデバッガ (E1エミュレータやシミュレータ) の変更をおこないます。次ページにその手順を示します。



HEWでのデバッガ (E8a,E1エミュレータやシミュレータ) 選択方法

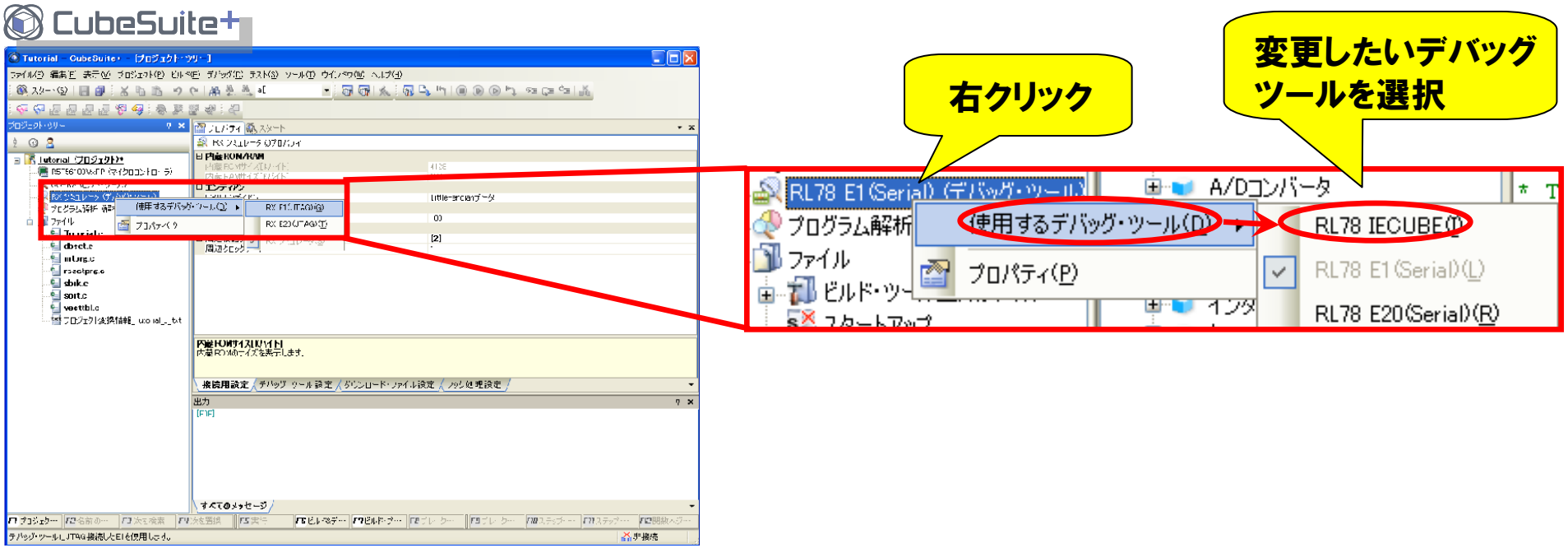
3.デバッガの変更方法

(1) CubeSuite+では現在選択しているデバッガはプロジェクトツリーパネル上の
デバッグ・ツール名 (デバッグ・ツール) で確認できます。
以下の例ではE1エミュレータが選択されています。



3.デバッガの変更方法

(2) デバッガを変更する場合、デバッグ・ツール名 (デバッグ・ツール) を右クリックしてポップアップメニューを開いて、[使用するデバッグツール] を選択して変更したいデバッグツールを選択してください。



4.IDコード入力方法

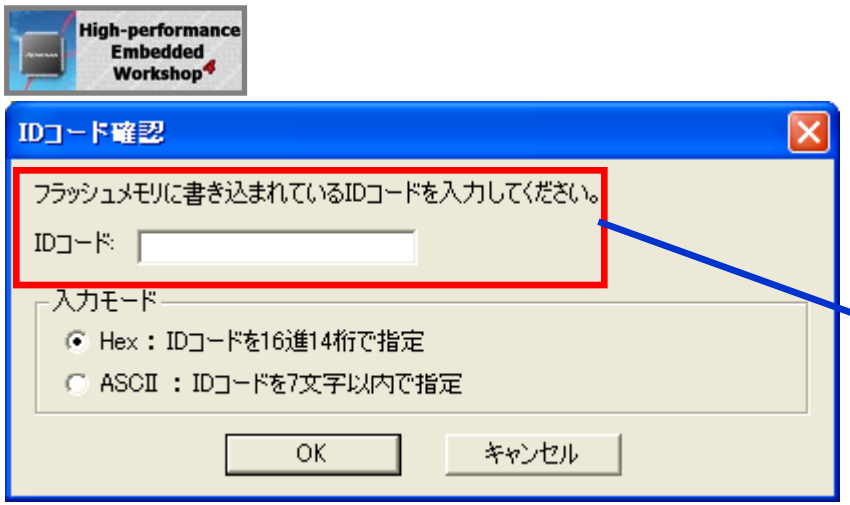
R8C,M16CとRL78ではエミュレータ使用時にどちらもIDコードの設定を行いますが、設定方法、照合方法、不一致だった場合の動作は異なります。
以下の仕様の違いを記載します。

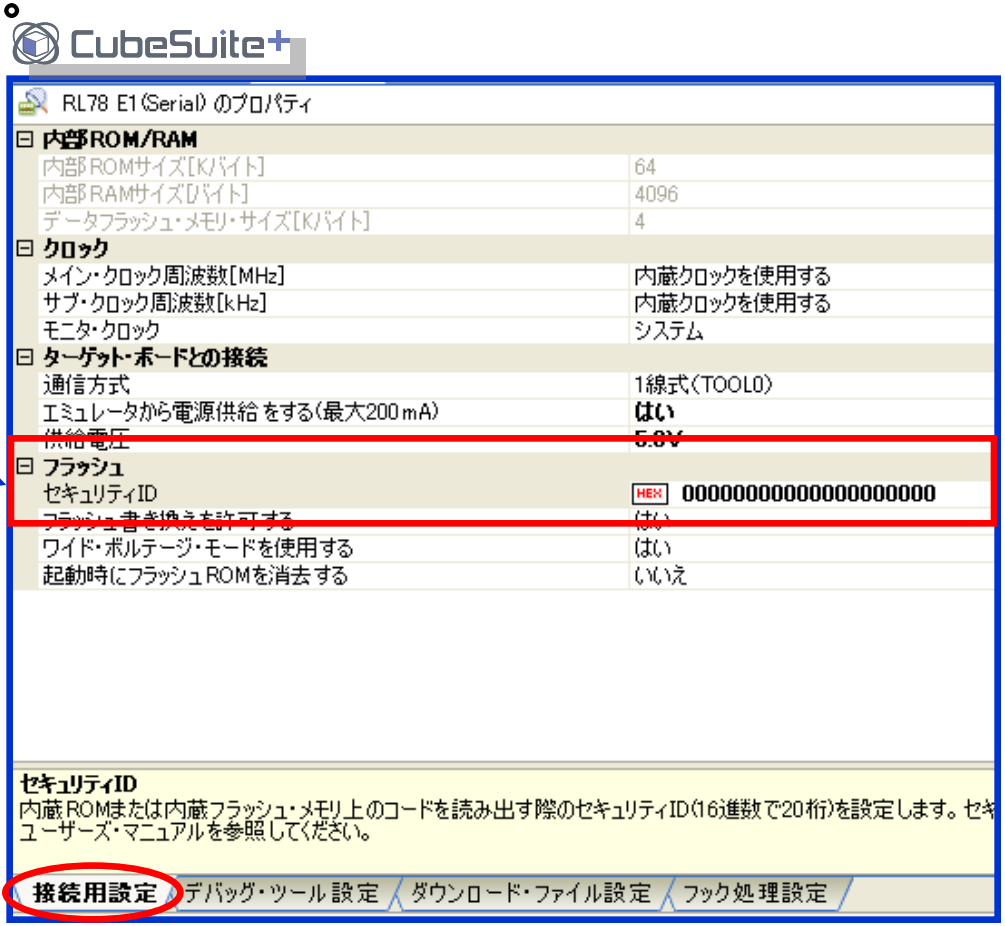
	IDコード桁数(16進数)	IDコードのアドレス	設定方法	照合方法	不一致だった場合の動作	オンボード・ライタでの有効性
R8C M16C	14桁	0xFFDF, E3,EB,EF, F3,F7,FB番地	ビルド時にユーザプログラムに埋め込む	デバッガ起動時にIDがオールFFhの場合は自動照合、それ以外の場合はダイアログで入力	デバッガは起動しない(フラッシュメモリの内容は保持される)	有り (オンボード・ライタ起動時もIDの照合が必要)
RL78	20桁	0xC4～ 0xCD 番地	ビルド時にユーザプログラムに埋め込む	デバッガに予め照合するIDコードを設定しておく	オンチップ・デバッグ・オプションバイトの設定による*	無し (デバッグ時のみ有効)

*詳細はE1/E20エミュレータユーザーズマニュアル 別冊(RL78接続時の注意事項)を参照してください。

4.IDコード入力方法

マイコンにIDコードが書き込まれている場合、HEWではエミュレータ起動時に[IDコード確認]ダイアログが表示されますが、CubeSuite+ではエミュレータ起動前に[プロパティ]パネルでIDコードを設定しておく必要があります。
以下を参考にIDコードを設定してください。



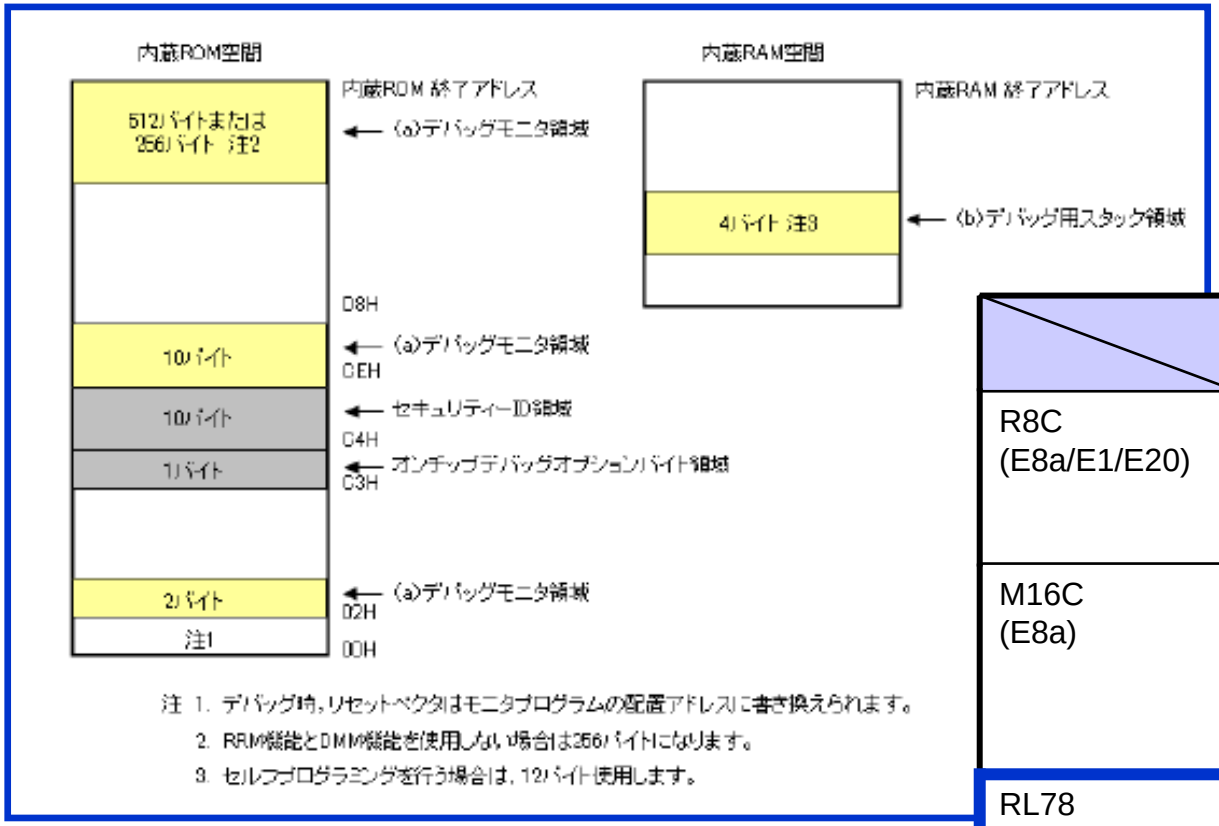


注：R8C,M16C(HEW)ではIDがFFFFFFFFFFFFFFFFFF(全てFFh)の場合は接続操作時に自動的に照合されるのでIDコード確認ダイアログは表示されない

5.リソース確保の方法

RL78ではOCD使用時にユーザリソースを占有します。それらの領域はユーザプログラムで使用しないようにビルドツール等で領域を確保する必要があります。

E1/E20**で使用する予約領域**(RL78)*



CubeSuite+**での設定方法は次ページを参照してください。**

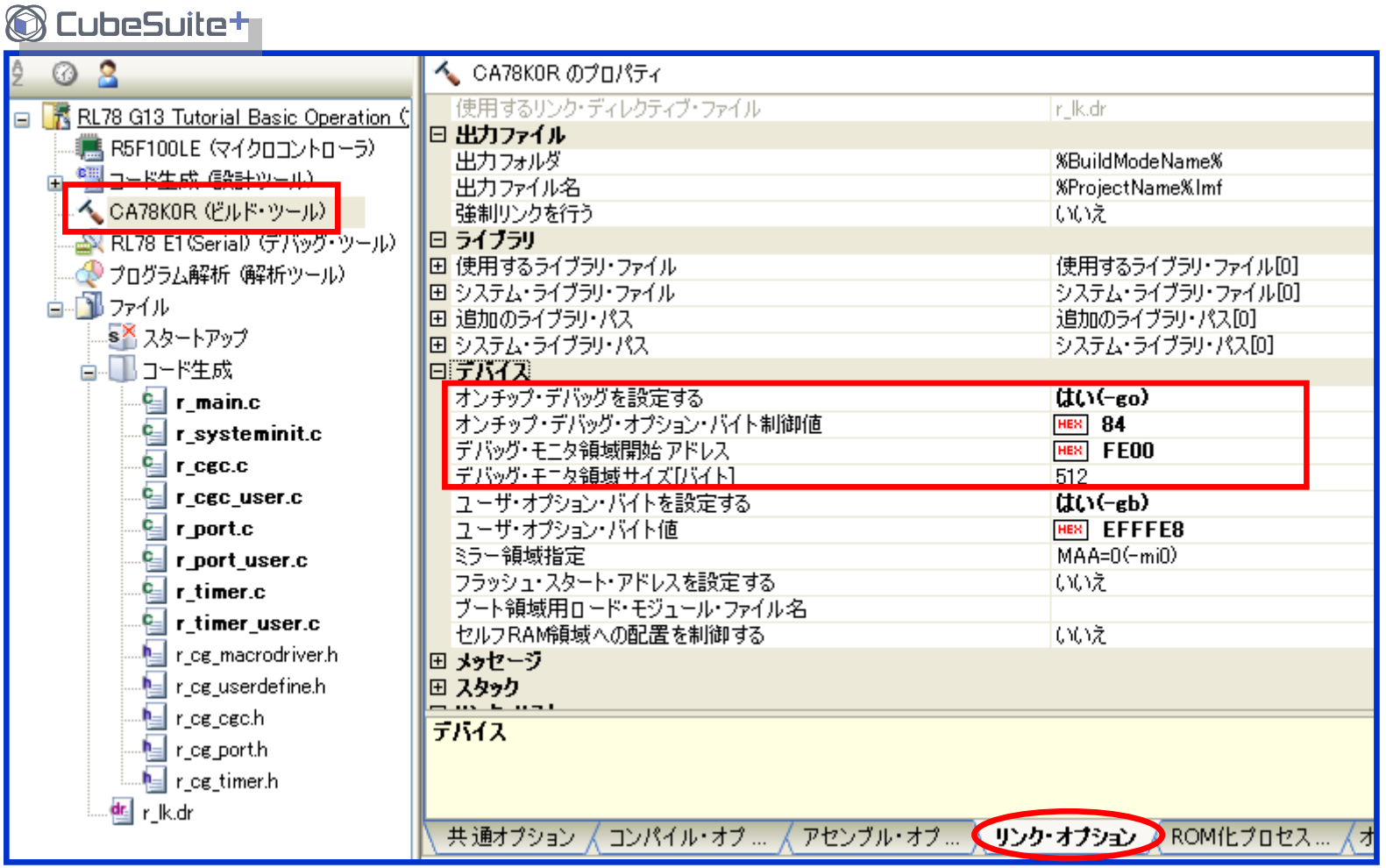
R8C,M16C**との比較**

	リソース占有	占有容量	確保方法
R8C (E8a/E1/E20)	有り (デバイスによる)	最大 ROM2K スタック8バイト ベクタの一部	ビルド時に確保 (リンカで占有領域のアドレスを避けるよう設定する)
M16C (E8a)	有り (デバイスによる)	最大 ROM2K RAM128バイト スタック14バイト ベクタの一部	
RL78 (E1/E20)	有り	左図参照	ビルド時に確保 (CubeSuite+のGUIで設定可能)

***詳細は**E1/E20**エミュレータユーザーズマニュアル 別冊** (RL78**接続時の注意事項**)**を参照してください。**

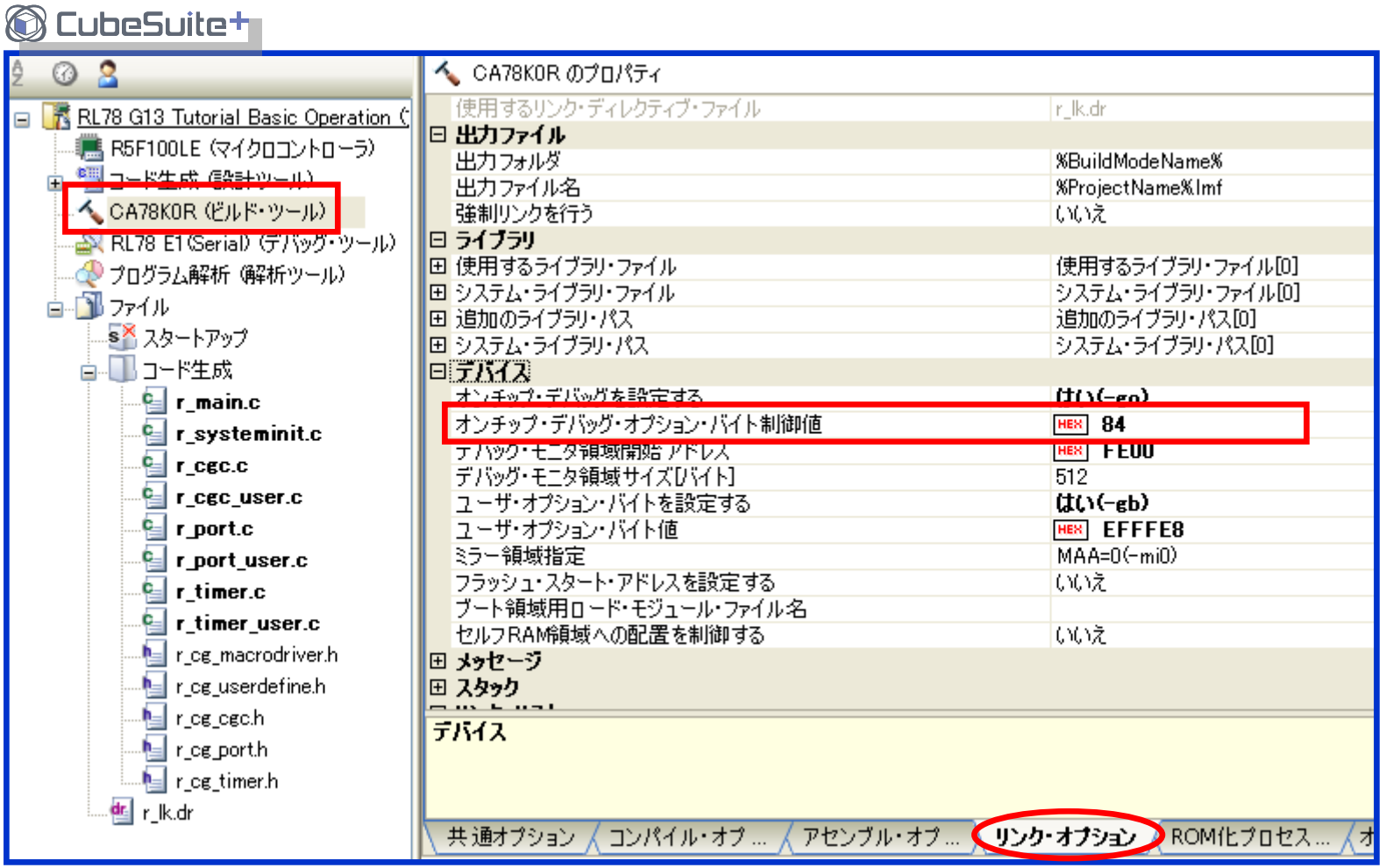
5.リソース確保の方法

デバッグ・モニタ領域のアドレスはビルド・ツールのプロパティからリンク・オプションで設定できます。



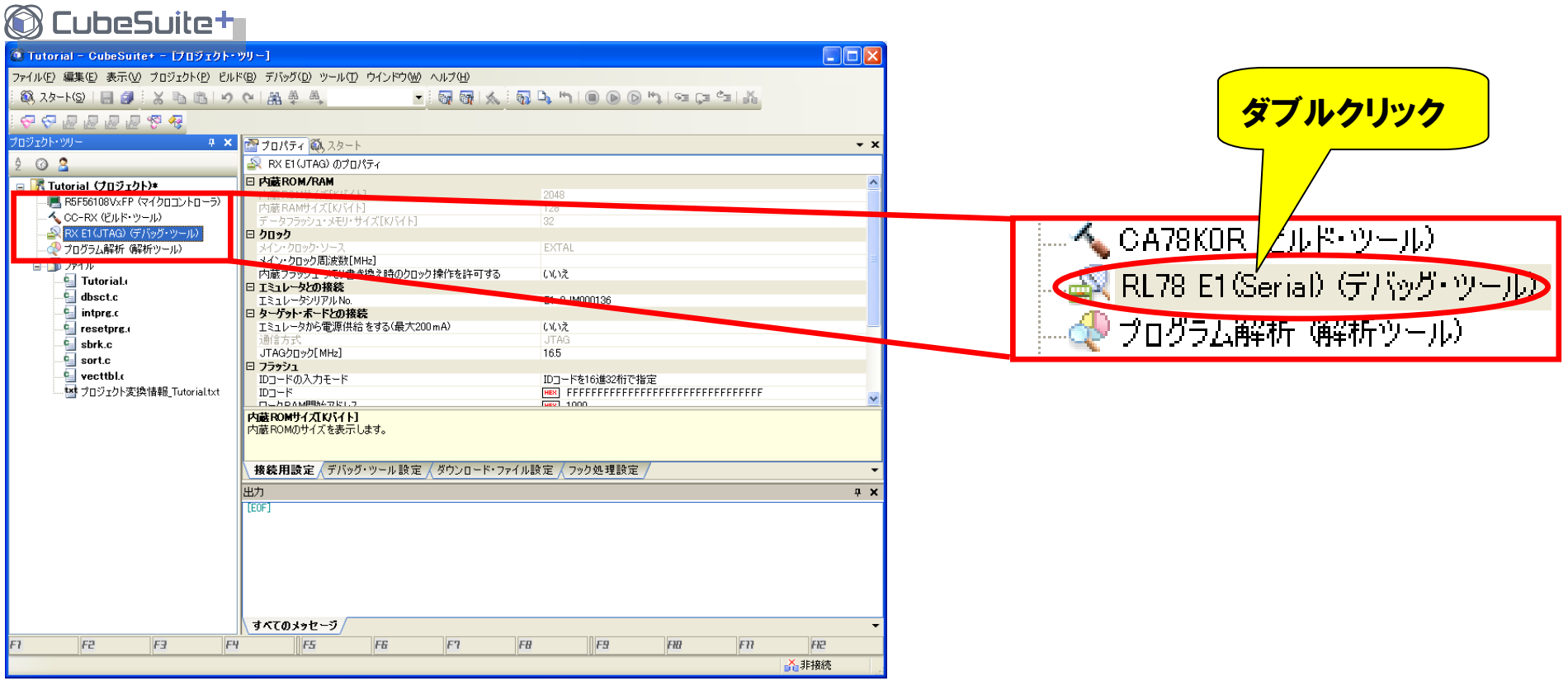
6.オンチップ・デバッグ・オプションバイトの設定方法

オンチップ・デバッグ・オプションバイトはビルド・ツールのプロパティからリンク・オプションで設定できます。



7.エミュレータ接続時の設定はどこで行うのか？

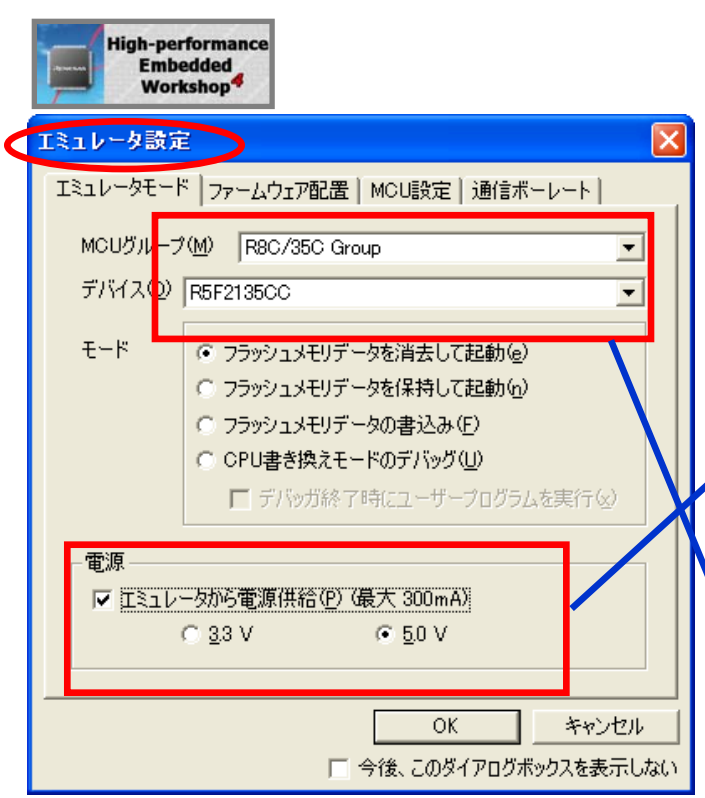
HEWではエミュレータ接続時に[起動設定]ダイアログ、[コンフィグレーションプロパティ]ダイアログが表示され設定を行いますが、CubeSuite＋ではエミュレータ**接続前**に[プロパティ]パネルで設定を行う必要があります。以下に手順を示します。
[プロジェクト・ツリー]パネルのデバッグツール名（デバッグ・ツール）をダブルクリックしてください。デバッグツールのプロパティが開きます。



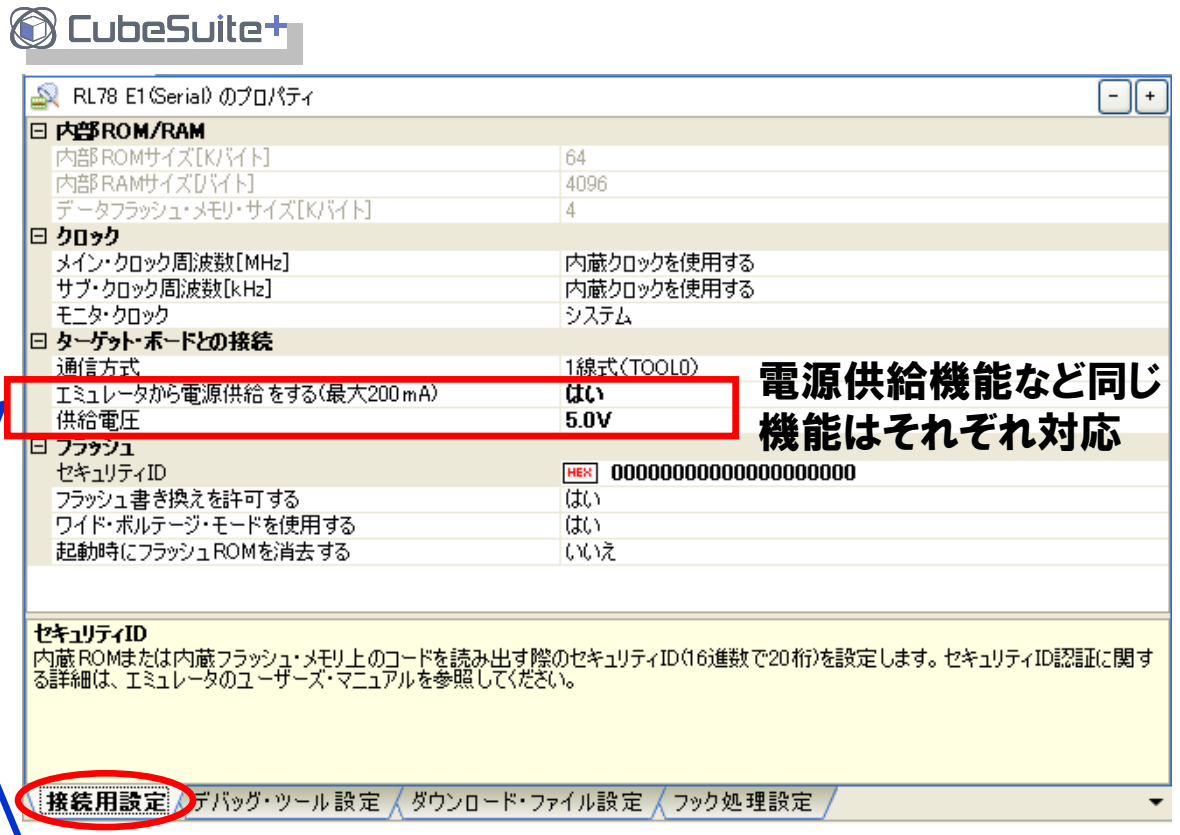
7.エミュレータ接続時の設定はどこで行うのか？

HEWはエミュレータ接続時に[エミュレータ設定]ダイアログで行いますが、CubeSuite＋ではエミュレータ接続前にデバッガの[プロパティ]パネルで設定しておきます。

(1) HEW[エミュレータ設定]ダイアログ⇒CubeSuite+[接続用設定タブ]に相当



例：R8C用E8aでのエミュレータ設定ダイアログ

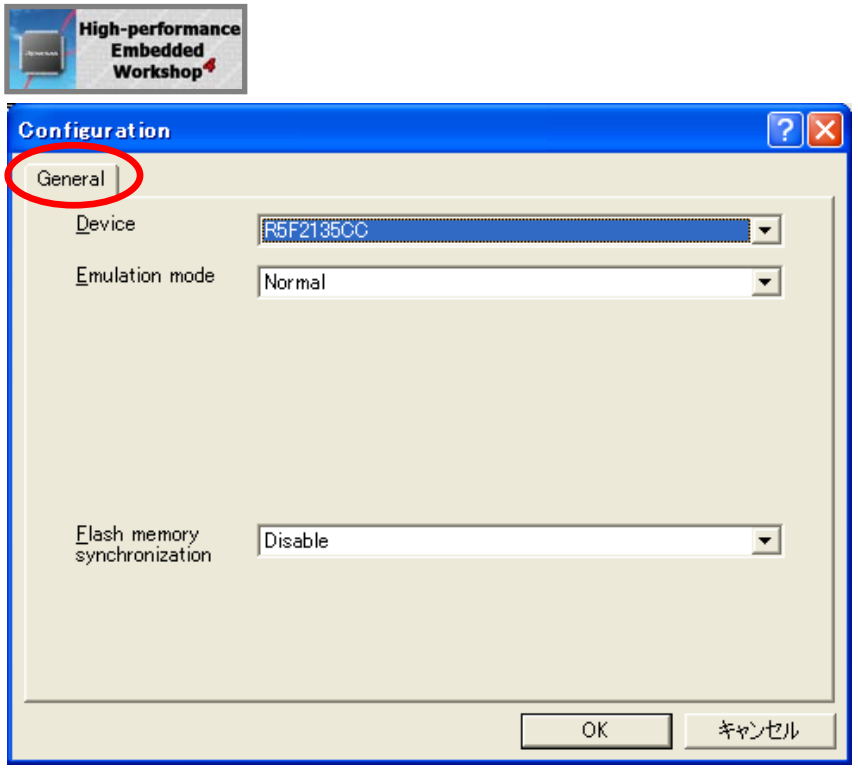


電源供給機能など同じ機能はそれぞれ対応

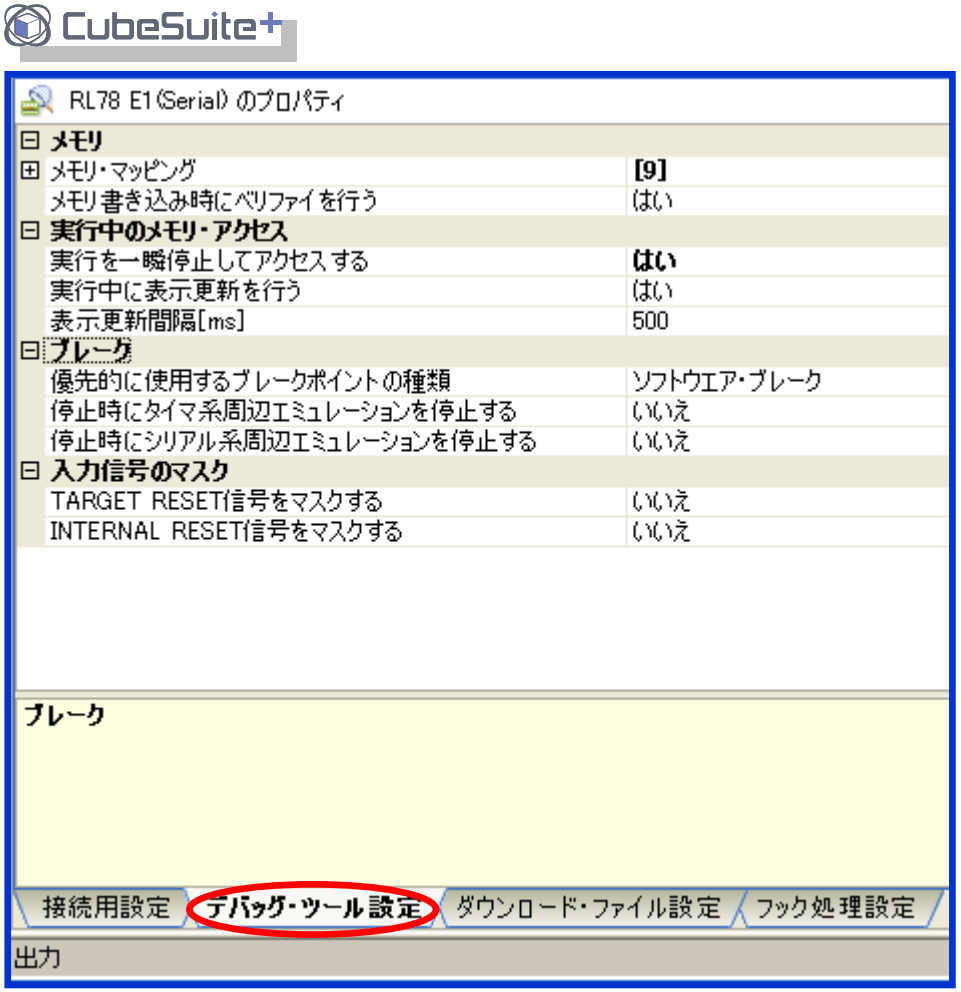
CubeSuite+ではプロジェクト作成時にデバイスを設定。

7.エミュレータ接続時の設定はどこで行うのか？

(2) HEW [コンフィグレーション] ダイアログ⇒CubeSuite+ [デバッグ・ツール設定] に相当

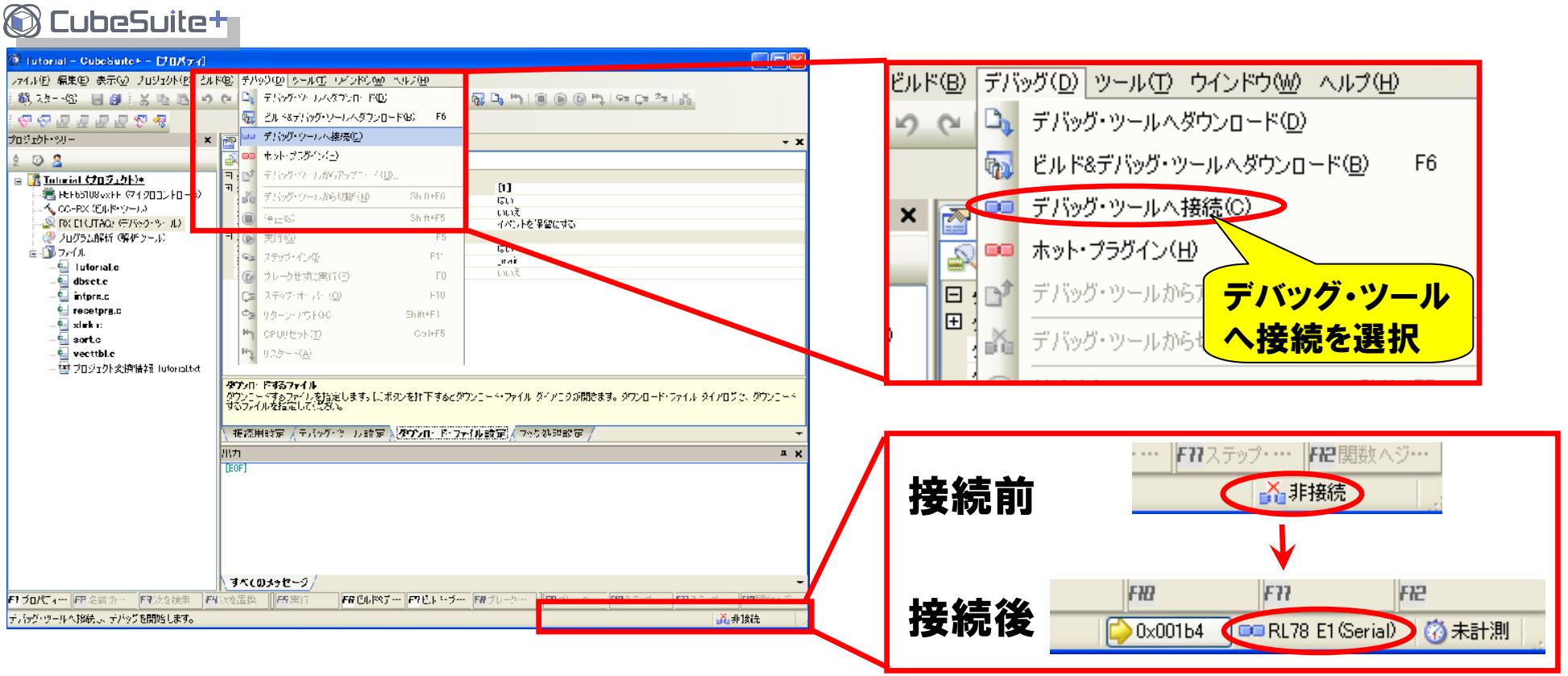


例：R8C用E8aでのエミュレータ設定ダイアログ



8.エミュレータの接続方法

(1) CubeSuite＋のメニューから [デバッグ] → [デバッグ・ツールへ接続] を選択すると設定したエミュレータ (デバッグ・ツール) に接続します。
接続が完了すると画面右下のステータスバーにデバッグツール名が表示されます。

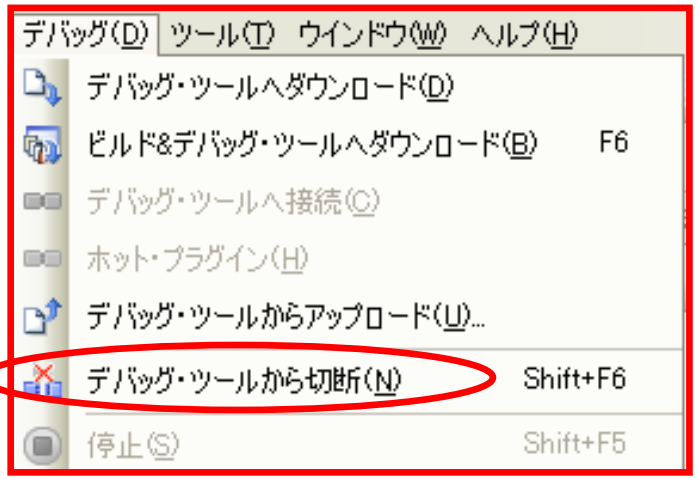


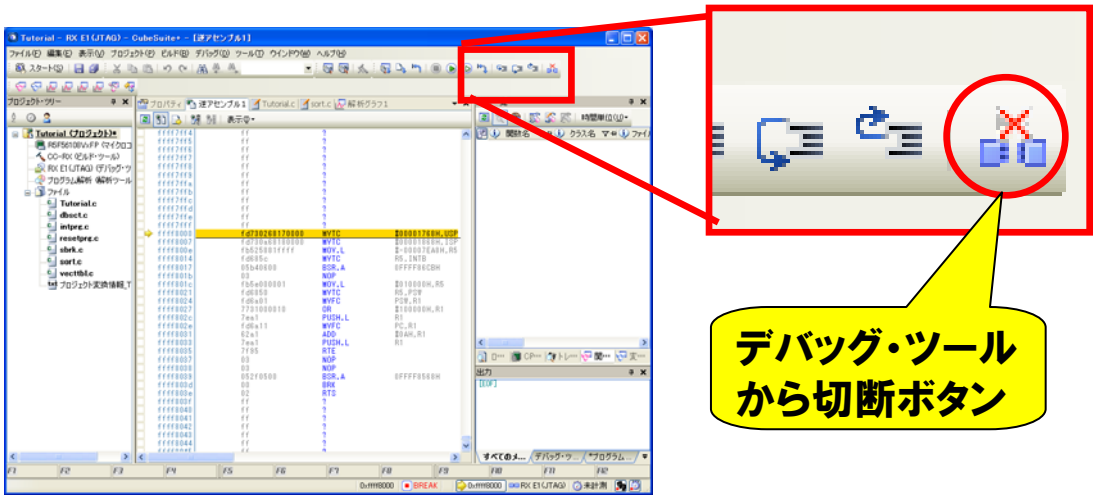
注：マイコンにIDコードを書き込んでいる場合は、「2.IDコード入力方法」を参考にして事前にIDコードを設定してください。

9.エミュレータの接続解除方法

(2) 接続を解除する場合はメニューから [デバッグ・ツールから切断] を選択するかデバッグツールバーの  ボタンをクリックしてください。

CubeSuite+



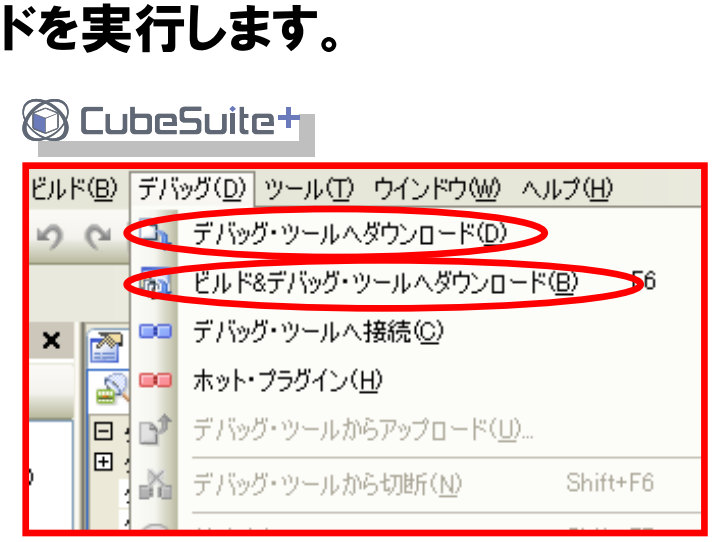


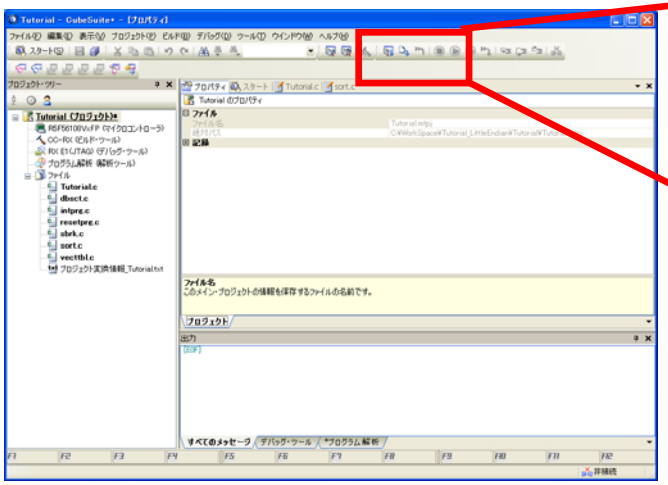
10.プログラムのダウンロード方法

メニューから [デバッグ] → [デバッグ・ツールへダウンロード] を選択するかデバッグツールバーの  ボタンをクリックすると、指定ファイルのダウンロードを実行します。

またメニューから [デバッグ] → [ビルド＆デバッグツールへダウンロード] を選択するかデバッグツールバーの  ボタンをクリックすると、プロジェクトのビルドを行った後、指定ファイルのダウンロードを実行します。

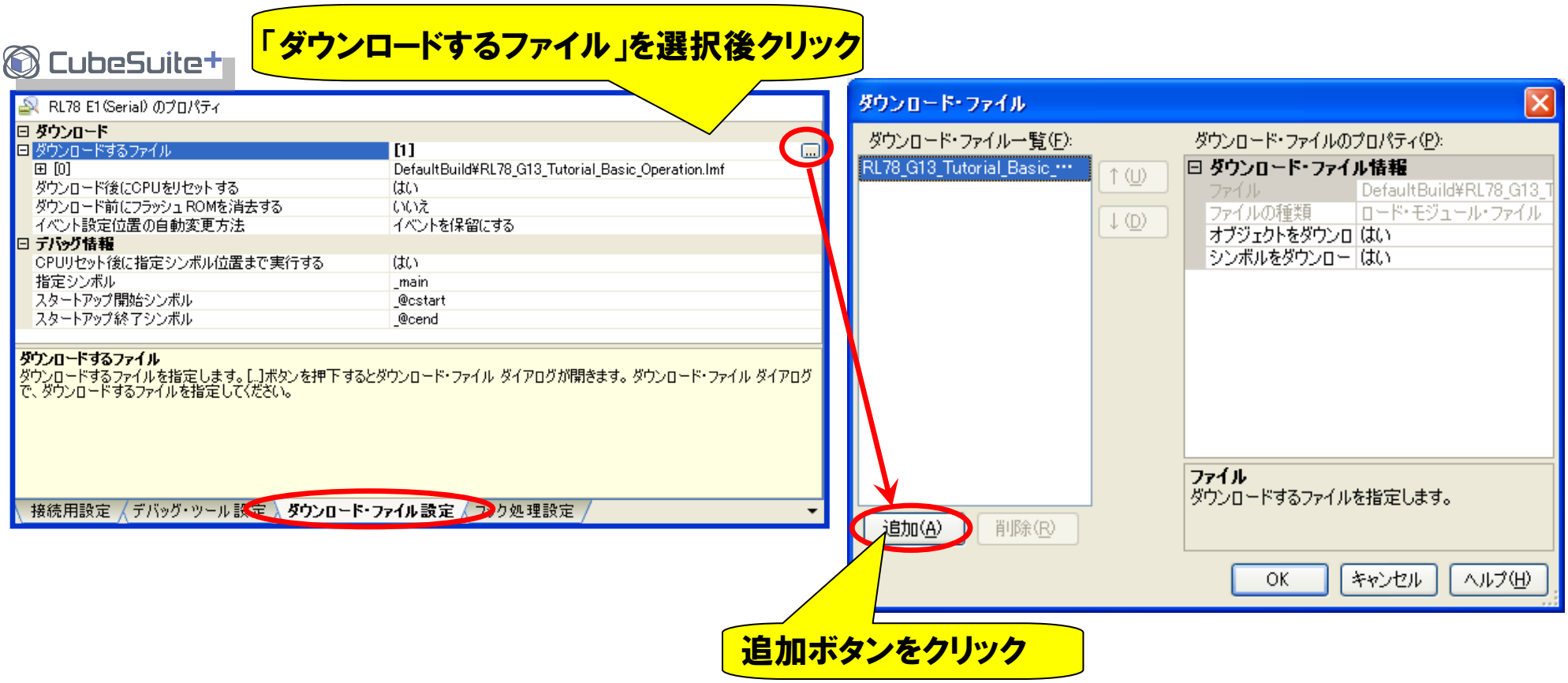
デバッグツールと接続していない場合はデバッグツールと接続を行った後ダウンロードを実行します。





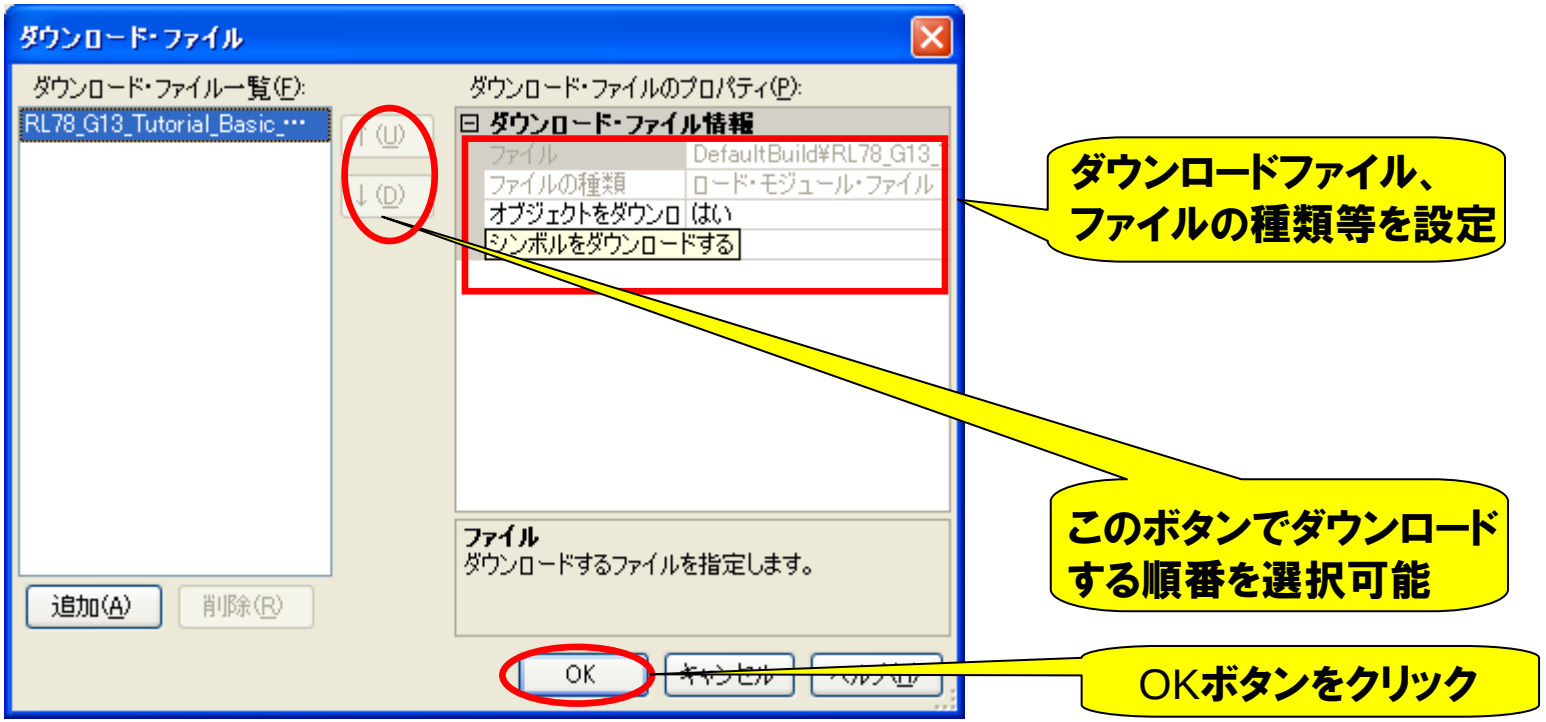
11.ダウンロードファイルを追加登録する方法

ダウンロードファイルの追加は [プロパティ] パネルの [ダウンロード・ファイル設定] タブで行います。
(1)「ダウンロードするファイル」を選択して右側の　...　をクリックしてください。
(2) ダウンロード・ファイルダイアログが開きますので追加ボタンを押してください。



11.ダウンロードファイルを追加登録する方法

(3) ダウンロード・ファイル情報にダウンロードするファイル名、ファイルの種類等を設定後OKボタンを押してください。



注:ダウンロード実行時、登録したファイルはすべてダウンロードされます。
任意のファイルのみダウンロードする場合は上記設定画面でダウンロードしたいファイルのみ「オブジェクトをダウンロードする」、「シンボルをダウンロードする」を「はい」に設定してください。

11.ダウンロードファイルを追加登録する方法

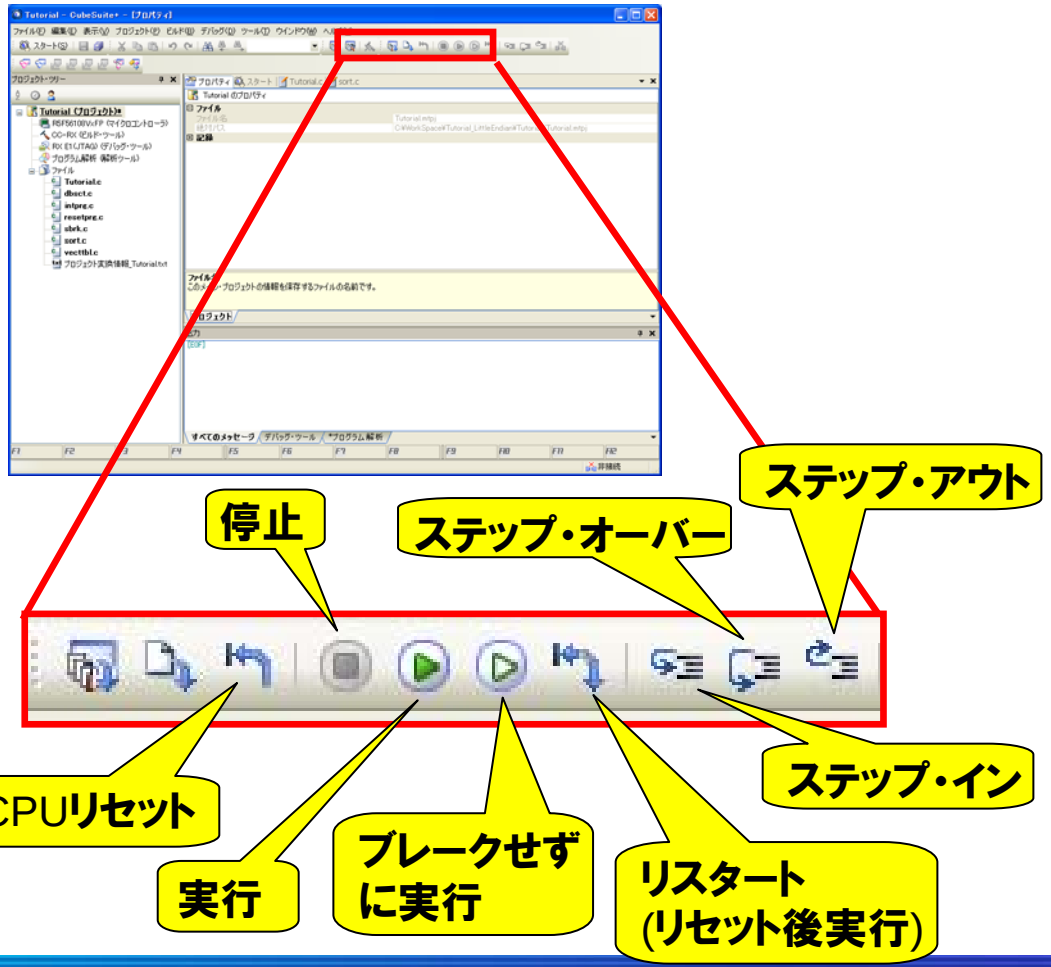
(4) R8C,M16CとRL78でダウンロード可能なファイルフォーマット(拡張子)が異なります。詳細は下図を参照してください。

	ロードモジュール フォーマット	ヘキサファイル	バイナリファイル
R8C,M16C	***.x30 *.abs**	***.mot *.hex**	***.bin**
RL78	***.lmf**	***.hex**	***.bin**
用途	ソースレベルデバッグ時にダウンロードするファイル	ROMライタ等で書き込む場合に使用するファイル	データファイル

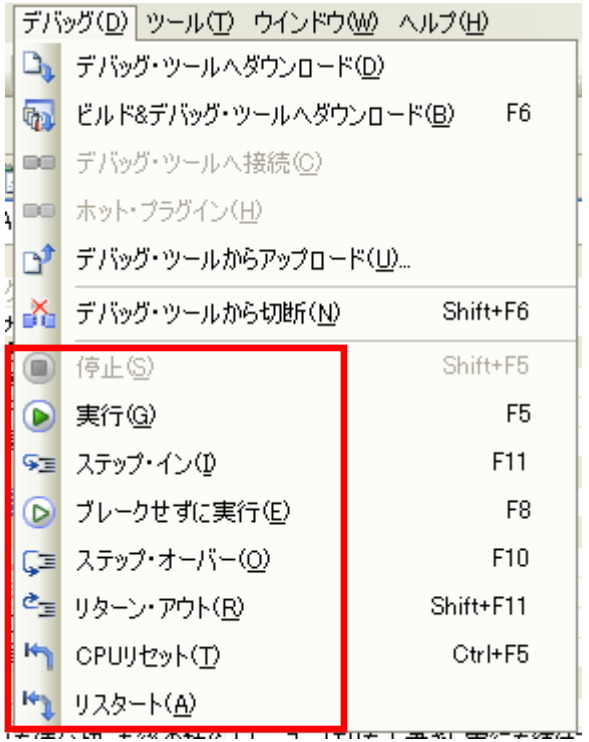
12.プログラムの実行、停止方法

プログラムの実行、停止やCPUリセットはHEWと同様にメニューおよびツールバーから行うことが出来ます。以下を参照ください。

CubeSuite+

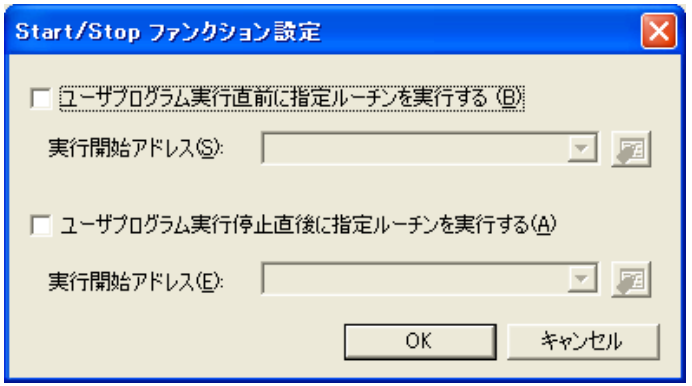


CubeSuite+のメニュー



13.ブレーク時のマイコンの動作の違い(周辺ブレーク機能)

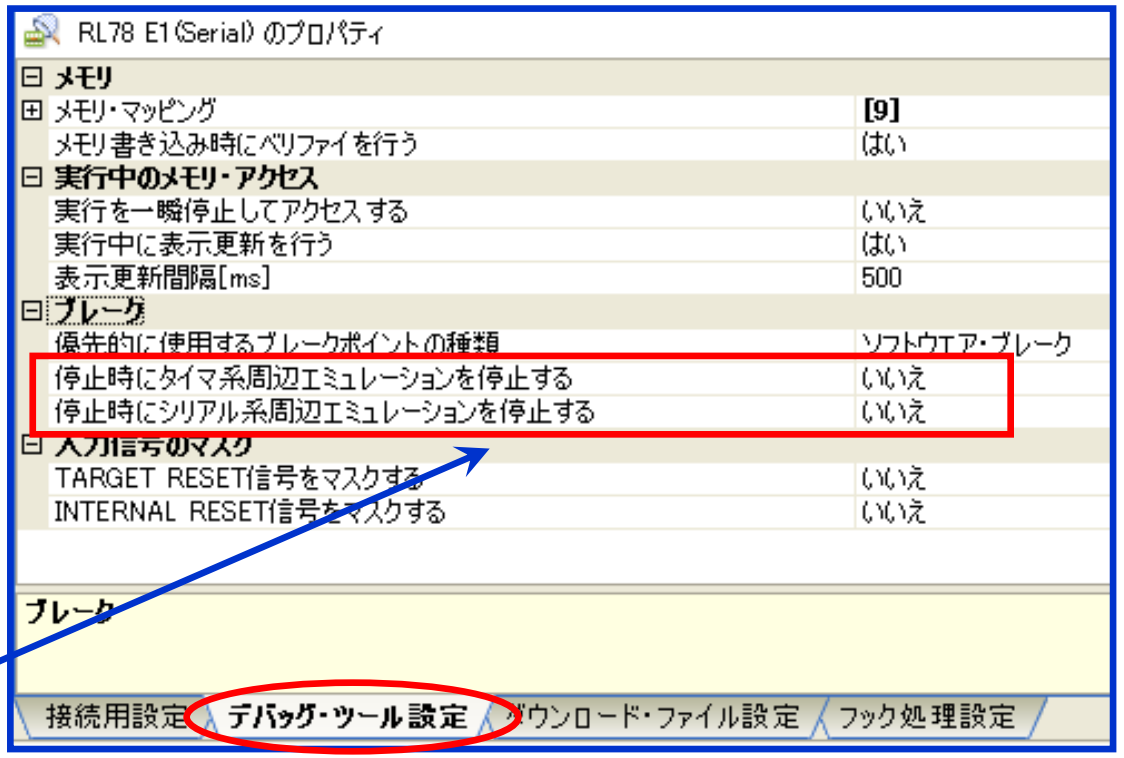
R8C,M16Cではエミュレータでブレーク時にタイマ、シリアルは継続して動作していましたが*、RL78ではCubeSuite+の[デバッグ・ツール設定]でブレーク時に動作か停止の選択ができるようになっています。



R8C,M16C**では上図にあるような**
Start/Stop**ファンクション機能を使用し周辺を停止させるコードを作成し埋め込む必要があった。**

RL78**では**CubeSuite+**で設定可能**

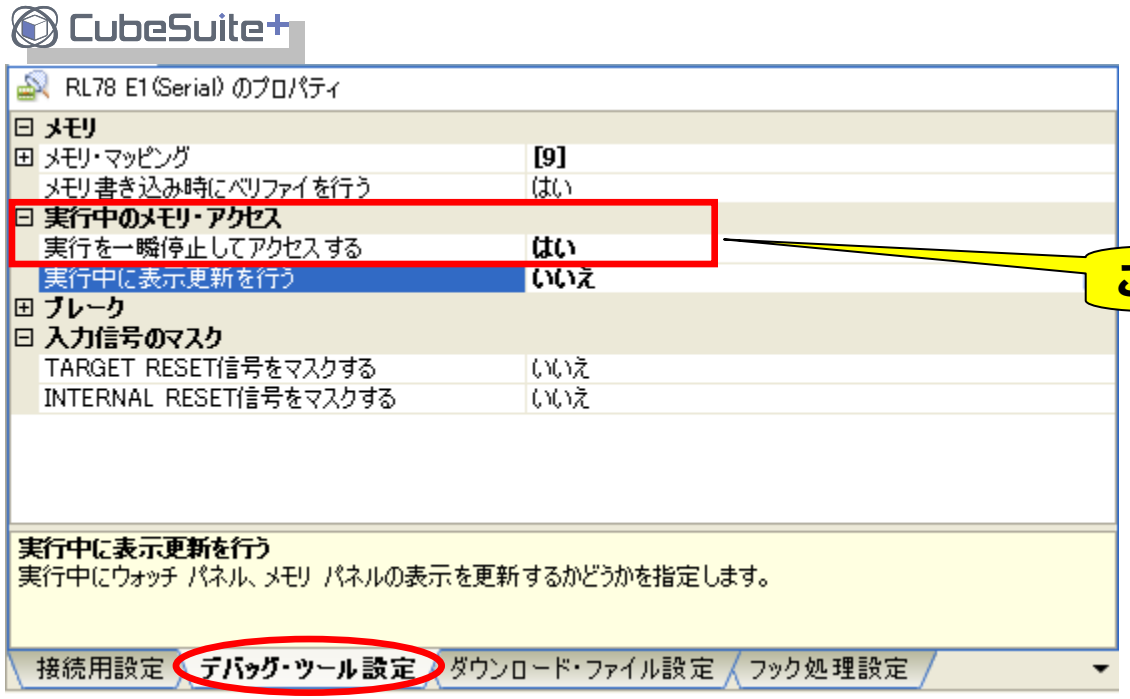
CubeSuite+



*R8C/5x**シリーズはブレーク時に停止させることが可能です**

14.プログラム実行中にメモリや変数を参照・変更する方法

CubeSuite+でプログラム実行中にメモリや変数の内容を参照、変更するには [プロパティ] パネルの設定が必要です。
以下の手順で設定を行ってください。
(1) 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] タブを開いてください。
(2) 実行中のメモリ・アクセスの [実行を一瞬停止してアクセスする] を [はい] に設定してください。プログラム実行中にメモリ、変数の内容を参照・変更できるようになります。

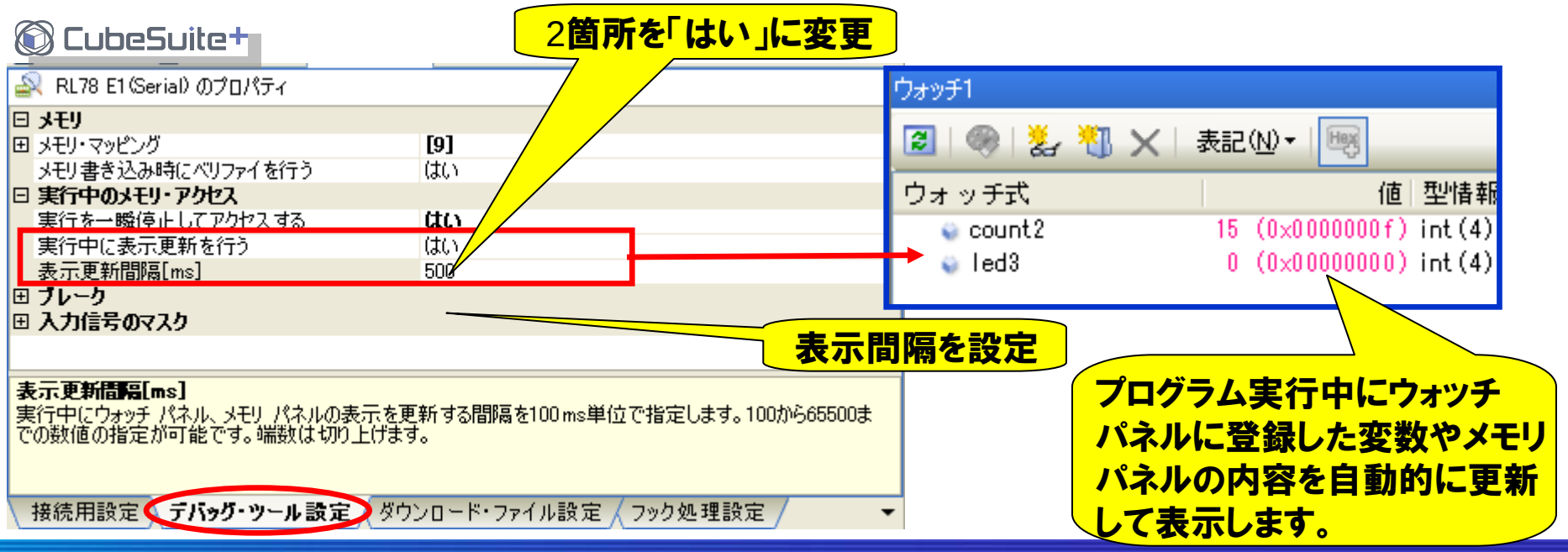


ここを「はい」に変更

本設定を「いいえ」にした場合、プログラムの実行中メモリパネルは**を表示します。

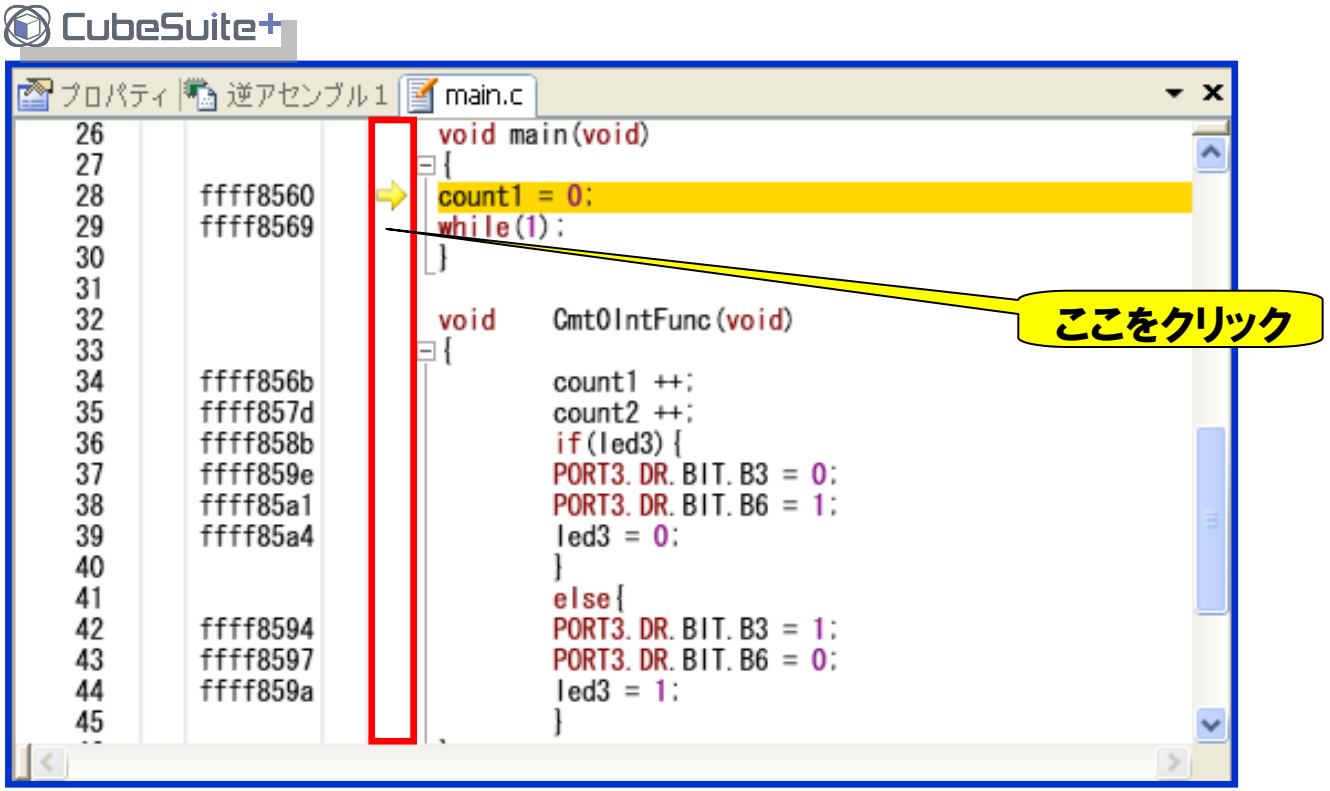
15.プログラム実行中にメモリや変数を自動更新する方法

CubeSuite+でメモリや変数の内容を自動更新するには [プロパティ] パネルの設定が必要です。以下の手順で設定を行ってください。
(1) 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] タブを開いてください。
(2) 実行中のメモリ・アクセスの [実行を一瞬停止してアクセスする] と [実行中に表示更新を行う] を [はい] に設定してください。
プログラム実行中にメモリパネル、ウォッチパネルの表示内容を自動更新します。
表示間隔を変更したい場合は [表示更新間隔] の値を修正してください。



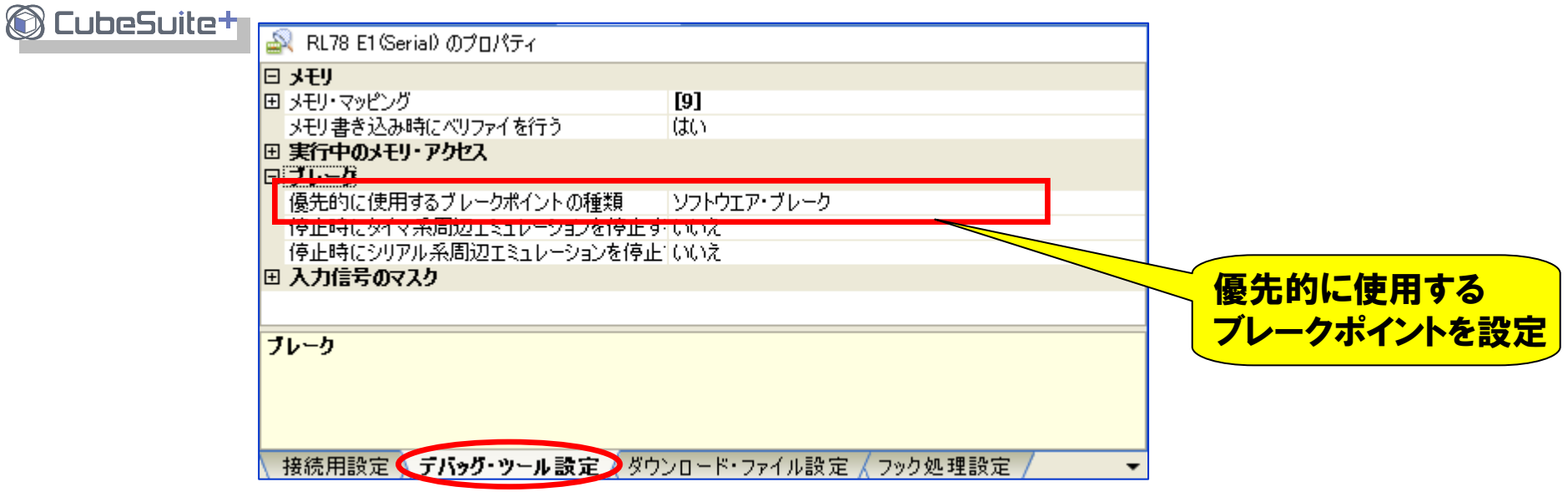
16.ブレークポイントの設定方法

(1) CubeSuite+でのブレークポイント設定はエディタパネルのメイン・エリア (下図の赤で囲っている部分) で行います。
アドレス表示のある行を1クリックすることでブレークポイントを設定できます。
ブレークポイント設定済みの行を1クリックした場合はブレークポイントを削除します。



16.ブレークポイントの設定方法

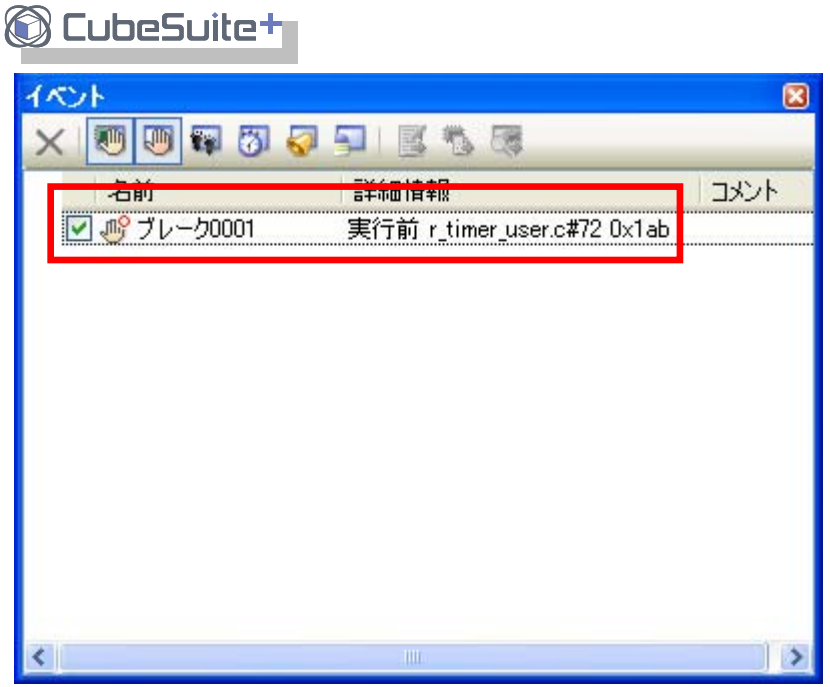
(2) ソフトウェアブレーク、ハードウェアブレークのうちどちらを設定するかは [プロパティ] パネルの [デバッグ・ツールの設定] タブの [優先的に使用するブレークポイント] で選択します。(下図の例ではソフトウェア・ブレークを設定しています)



(3) 上記で選択したブレークポイントの設定数が制限を越える場合、もう一方の種類のブレークポイントが使用されます。
どちらのブレークが割り当てられたかは、イベントマークで区別できます。
 がソフトウェアブレーク、  がハードウェアブレークのイベントマークです。

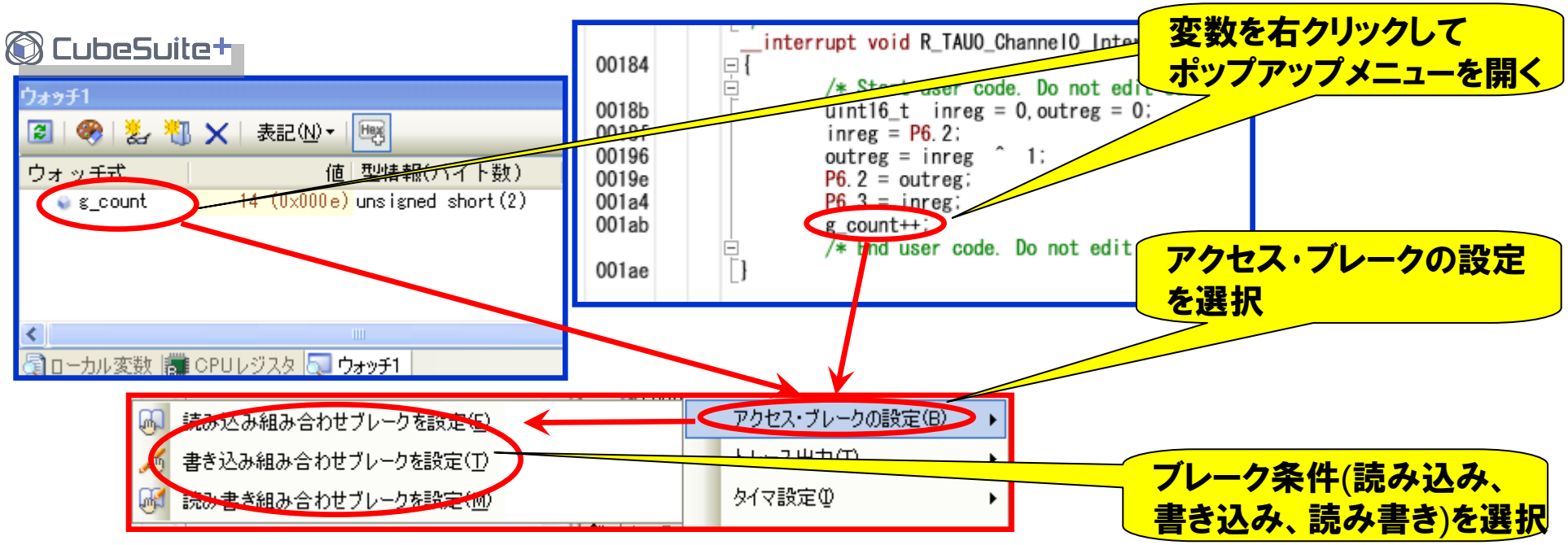
16.ブレークポイントの設定方法

(4) ブレークポイントの設定状態は [イベント] パネルで確認できます。
[イベント] パネルはCubeSuite+のメニューから [表示] → [イベント] で開きます。
[イベント] パネルで不要なブレークポイントを削除したり、無効にすることができます。



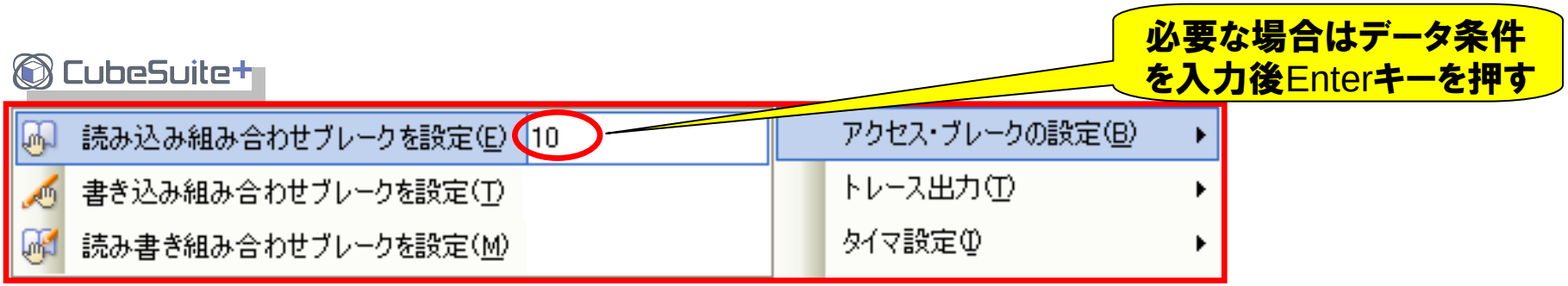
17.変数へのアクセスでブレークする方法

変数へのアクセスでブレークする設定はウォッチパネルまたはエディタパネルで行います。
(1) ウォッチパネルまたはエディタパネル上のアクセス時にブレークしたい変数を右クリックしてポップアップメニューを開いてください。
(2) アクセス・ブレークの設定 (エディタパネルの場合はブレークの設定) を選択して、「読み込み組み合わせブレークを設定」「書き込み組み合わせブレークを設定」「読み書き組み合わせブレークを設定」のいずれかを設定してください。



17.変数へのアクセスでブレークする方法

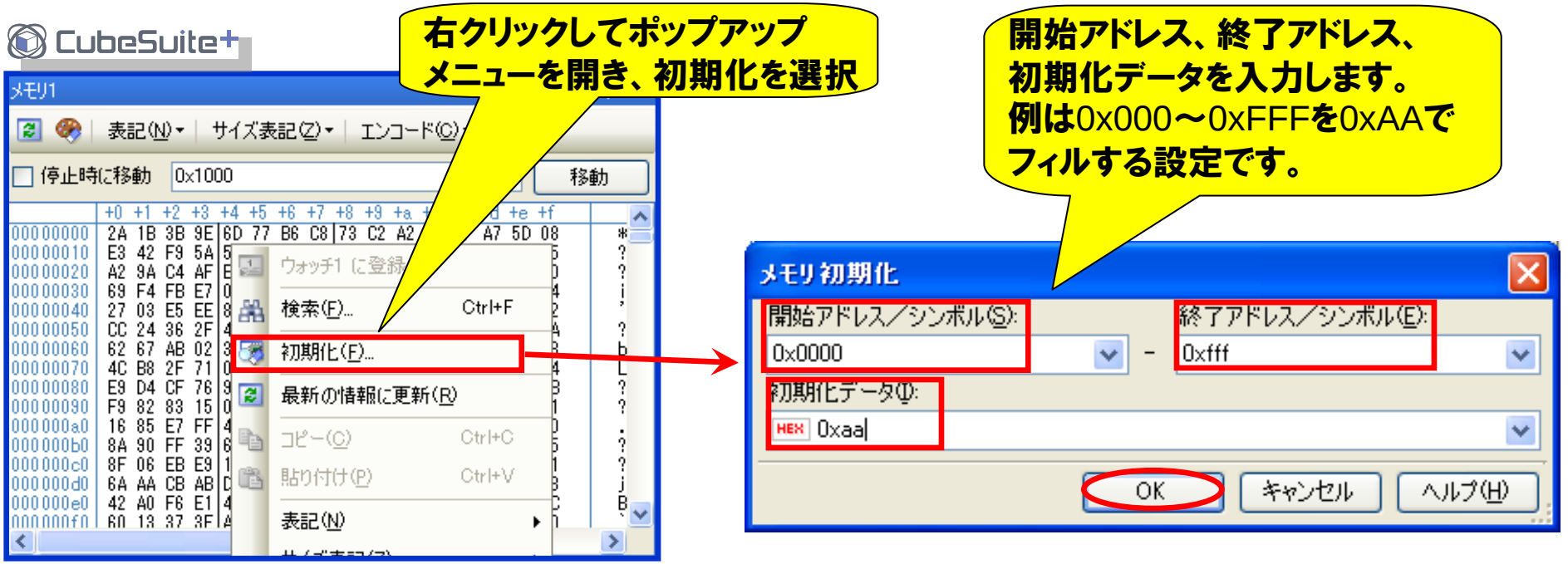
(3) データ条件を設定する場合は数値を入力してください。(不要な場合は入力しなくてもかまいません)



注:ここで入力する数値は10進数です。16進数で数値入力する場合は、0xAAのように先頭に"0x"を付けてください。

18.メモリをフィルする方法

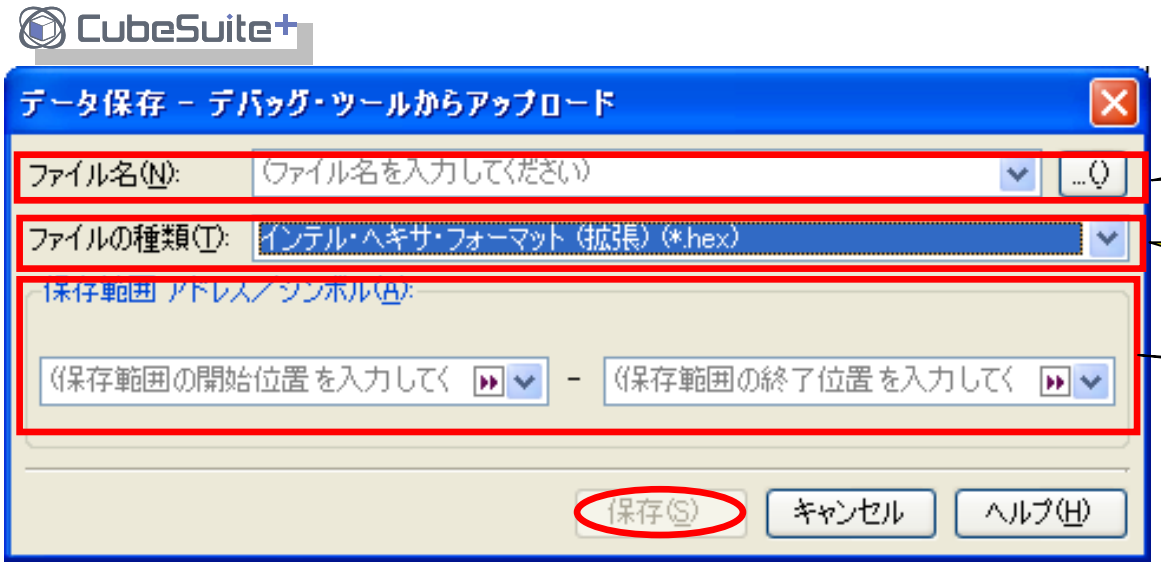
メモリフィル（一括して変更）はメモリ初期化ダイアログで行います。
（1）［メモリ］パネル内を右クリップしてポップアップメニューを開いて、［初期化］を選択してください
（2）メモリ初期化ダイアログが開きますので、初期化したいアドレス（開始アドレスと終了アドレス）を初期化データを設定後OKボタンを押してください。



注：ここで入力する数値は10進数です。16進数で数値入力する場合は、先頭に”0x”を付けてください。

19.メモリ内容をセーブする方法

CubeSuite+でメモリ内容をセーブする場合は、[デバッグ・ツールからアップロード] を使用します。
メニューから [デバッグ] → [デバッグ・ツールからアップロード] を選択してください。
データ保存ダイアログが開きますので、保存したいファイル名、ファイルの種類、保存範囲を設定して、保存ボタンを押してください。



保存ファイル名を設定します。

ファイルの種類(インテルヘキサ、モトローラS、バイナリ)を設定します。

保存範囲を設定します。

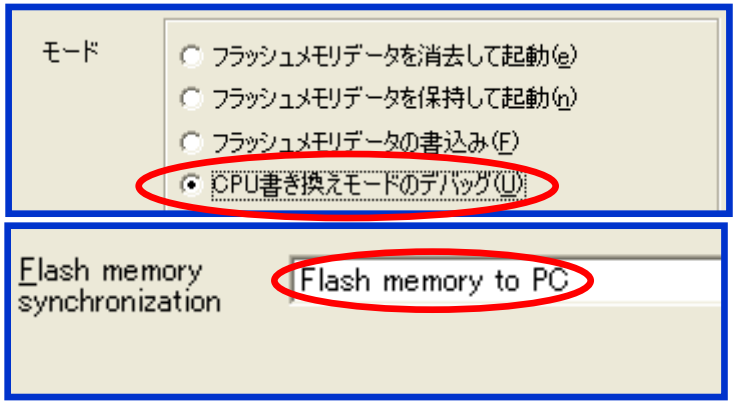
注:ここで入力する数値は10進数です。16進数で数値入力する場合は、先頭に"0x"を付けてください。

20.フラッシュ・セルフ・プログラミングについて

RL78は，ユーザ・プログラムでフラッシュ・メモリの書き換えを行うためのセルフ・プログラミング機能をサポートしています。この機能はRL78セルフ・プログラミング・ライブラリを利用することによりユーザ・アプリケーションでフラッシュ・メモリの書き換えを可能にします。

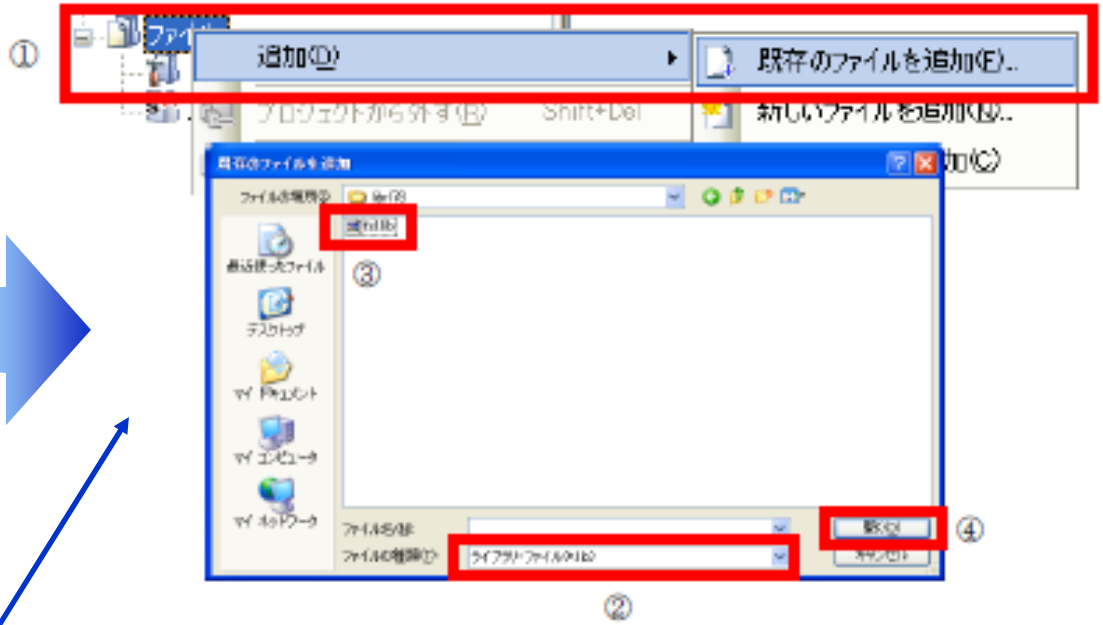
High-performance Embedded Workshop

R8C,M16C(HEW)**ではデバッガを専用のモードに変更し、フラッシュ書き換えプログラムは基本的には自作する。**



RL78(CubeSuite+)**では無償で提供されるセルフ・プログラミング・ライブラリをプロジェクトに組み込む。**
組み込み方法の詳細は以下のURL**を参照。**
http://japan.renesas.com/products/tools/flash_programming/flash_libraries/

CubeSuite+

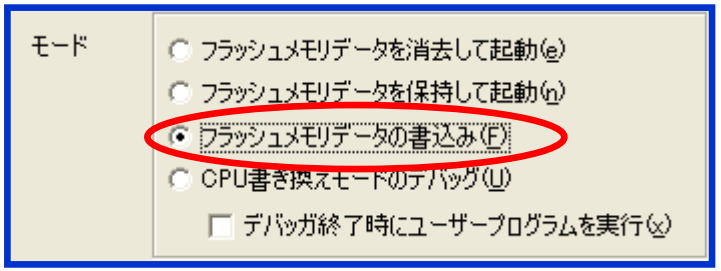


CubeSuite+**の[プロジェクト・ツリー]にある，[ファイル]を右クリックして表示されるリストから[追加]を選択し[既存のファイルを追加]画面を表示する。次に[ファイルの種類]のプルダウンメニューをクリックしファイルの種類を選択しフラッシュ・セルフ・プログラミング関連のファイルを登録する。**

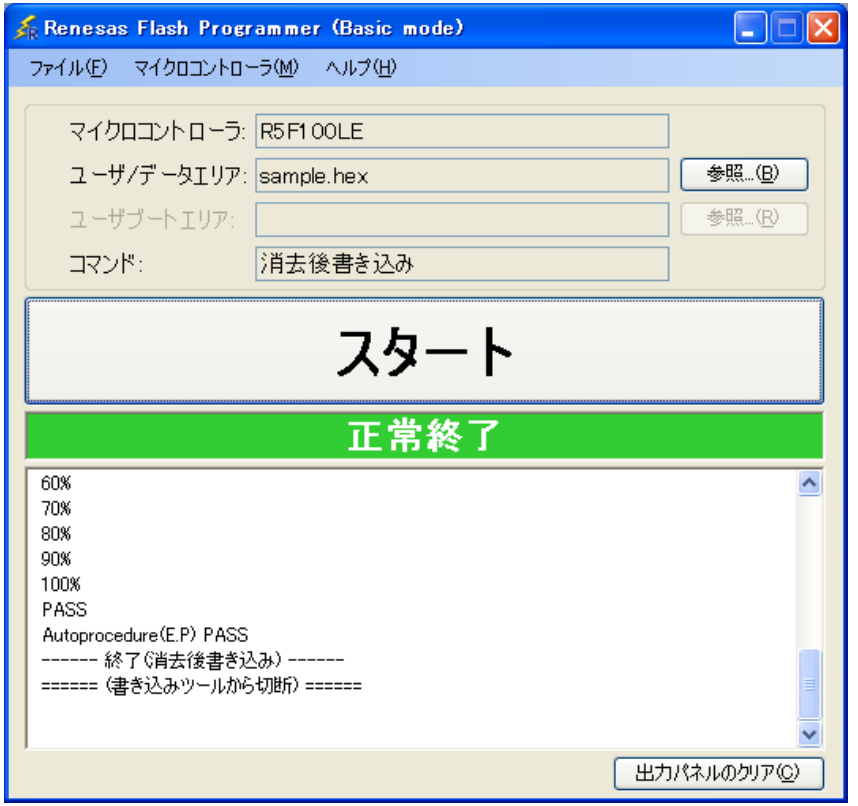
21.デバイス単体で動作確認するための書き込み方法

RL78ではデバッグ後にデバイス単体で動作確認を行う場合は、CubeSuite+ではなくフラッシュ書き込みソフトRenesas Flash Programmerを使用して書き込みを行ってください。

R8C,M16C(HEW)ではデバッガ用のモードの他にオンボードライタと同等の機能を持つモードを選択する。



RL78ではCubeSuite+ではなくRenesas Flash Programmerを使用して書き込む



Renesas Flash Programmerは、ルネサス製フラッシュ内蔵マイコンのフラッシュメモリに対し、書き込みを行うためのソフトウェアです。書き込みに特化した操作性・機能を提供します。

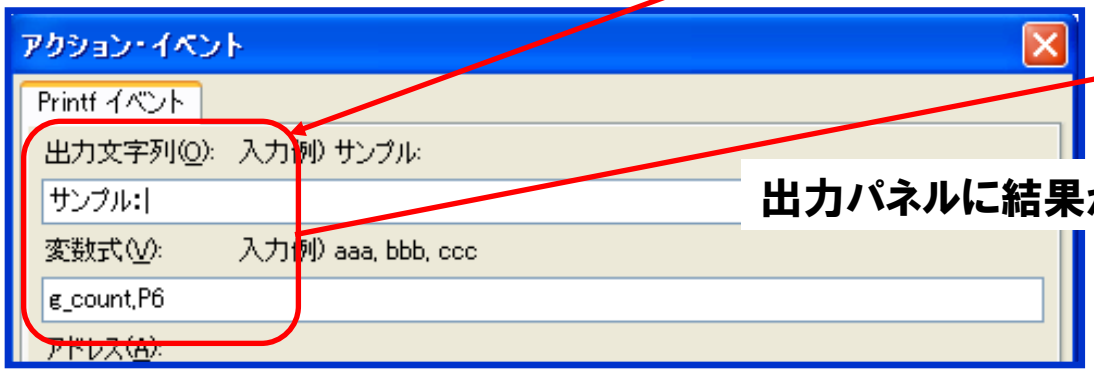
22.アクション・イベント機能(Printfイベント)

CubeSuite+ではアクション・イベントとしてPrintfイベントの設定が可能です。
Printfイベントとは，プログラムの実行を指定した箇所で一瞬停止させ，ソフトウエア処理によりコマンド(printf)を実行させる機能です。Printfイベントを設定すると，このイベントを設定した箇所の命令実行直前にプログラムが一瞬停止し，このダイアログで指定した変数式の値を出力パネルに出力します。

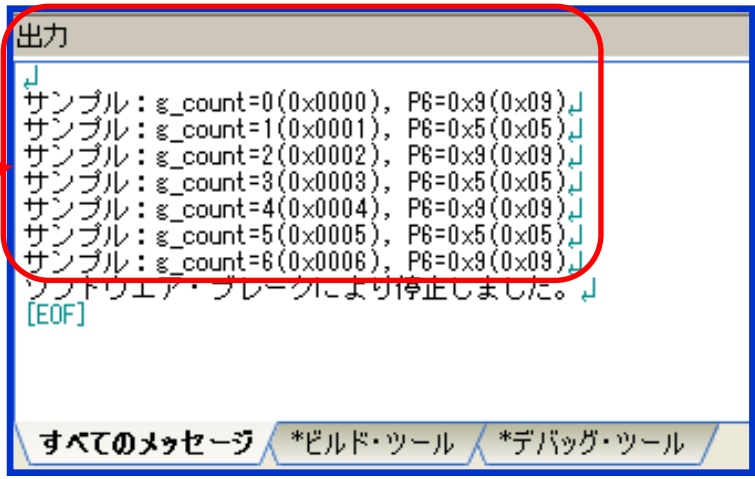
CubeSuite+

```
69      00196           outreg = inreg     1;
70      0019e           P6.2 = outreg;
71      001a4           P6.3 = inreg;
72      001ab           g_count++;
73                      /* End user code. Do not edit comment generated here */
74      001ae       }
75
```

この行にprintf("サンプル: g_count = %d, P6 = %d¥n",g_count,P6)に相当するコードを挿入したい ⇒ アクションイベントの設定



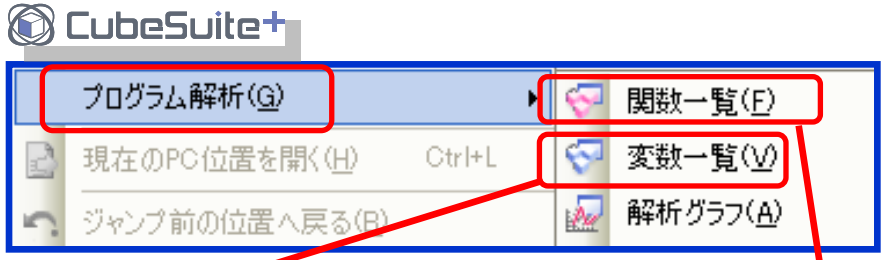
出力パネルに結果が出力

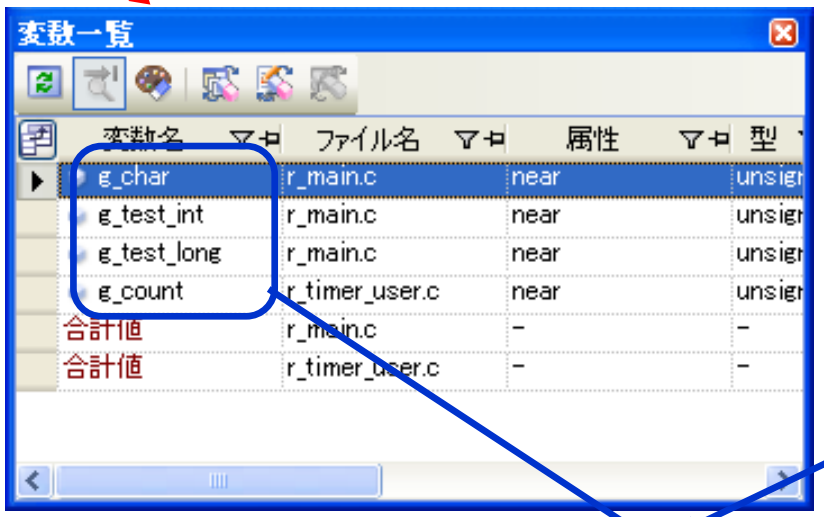


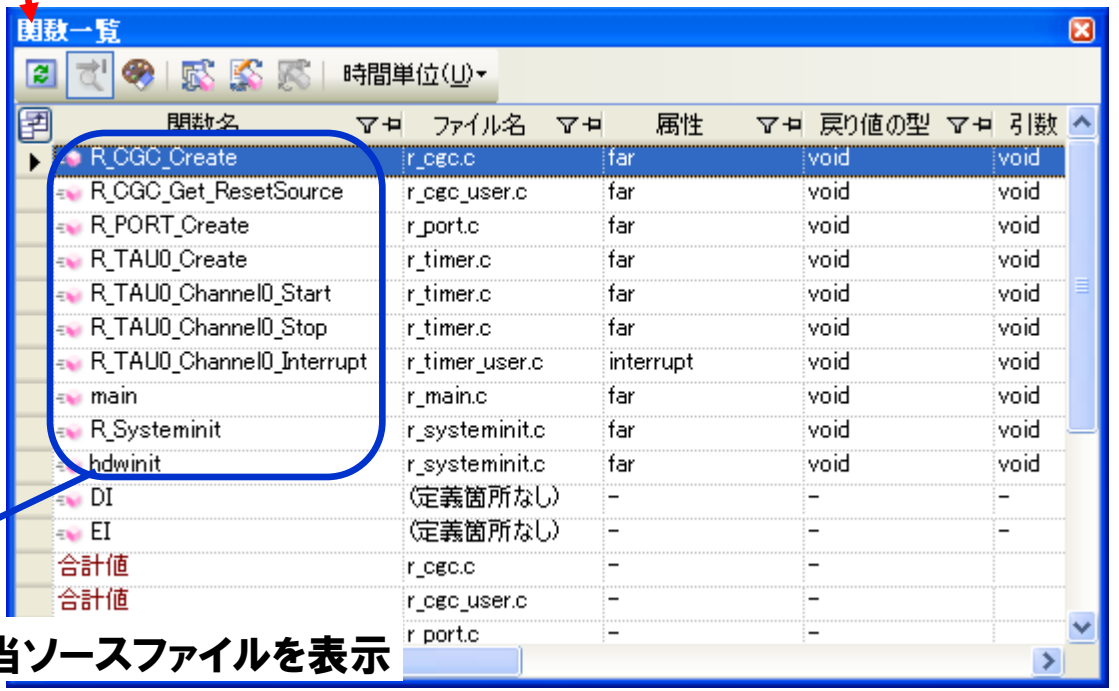
23.変数と関数の一覧表示機能

CubeSuite+でプロジェクトで使用している変数と関数の一覧を自動的に表示することができます。

メニューから[表示]→[プログラム解析]を選択してください。





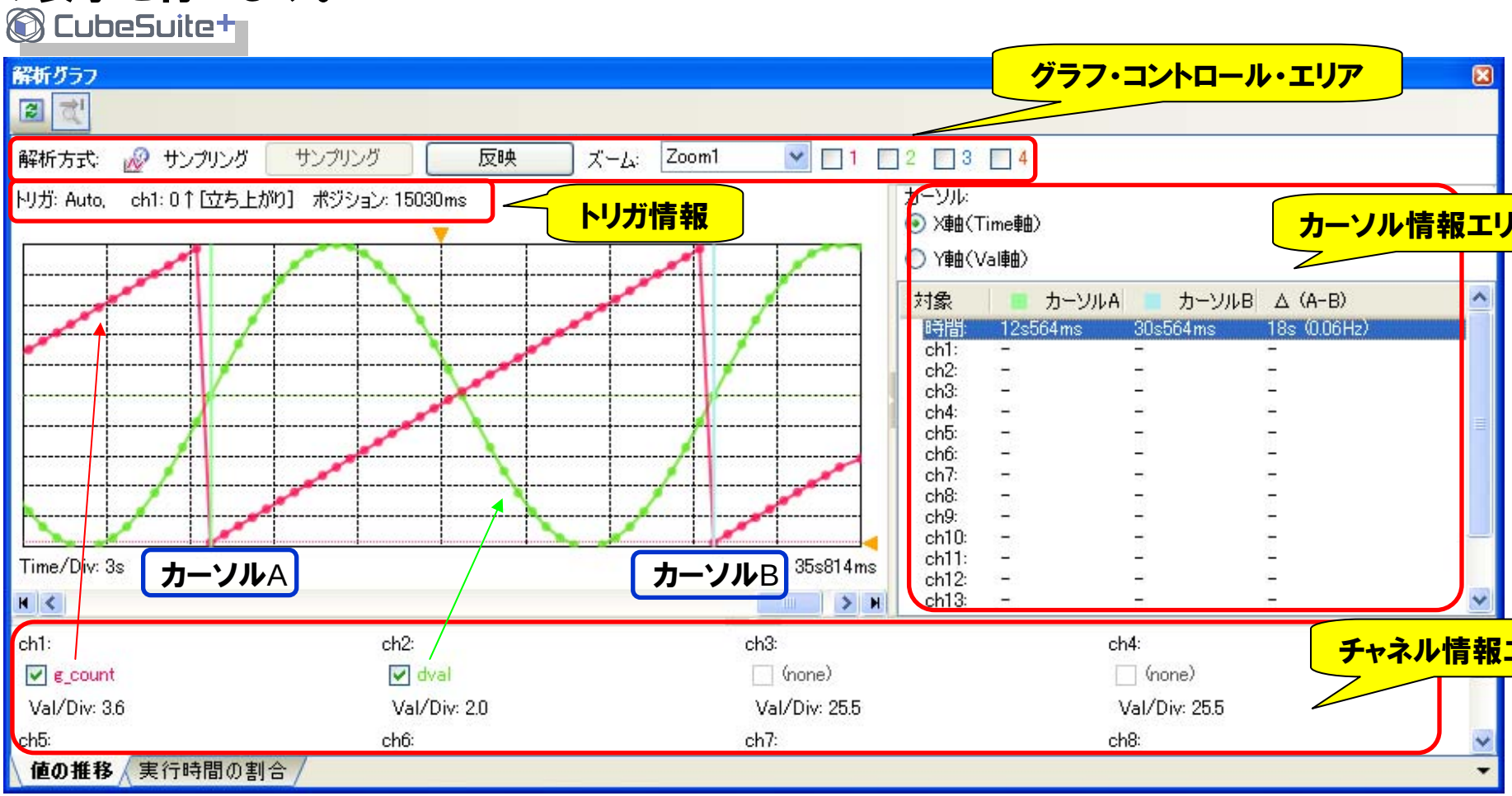


変数名や関数名をクリックすると該当ソースファイルを表示

24.解析グラフ機能

CubeSuite+にはグラフ化の対象として登録した変数／レジスタ／アドレス等の値と時間の関係を折れ線グラフで表示する解析グラフ機能があります。

RL78のオンチップ・デバッグでは疑似RRM機能により取得したデータを基にグラフの表示を行います。



25.エミュレータデバッグ機能比較(OCD)

デバッグ機能		RL78(E1/E20)	R8C(E8a/E1/E20)	M16C(E8a)
ブレーク	ソフトウェア・ブレーク	2000 点	255点	255点
	ハードウェア・ブレーク	実行・アクセス兼用で 1～2 点*	実行・アクセス兼用で 2点～10点*	実行・アクセス兼用で 6点～10点*
	強制ブレーク	可能	可能	可能
イベント	設定可能数	実行・アクセス兼用で 1～2 点*	1点～2点*	0点～2点*
	イベント使用機能	ハードウェア・ブレークのみ	ハードウェア・ブレークのみ	ハードウェア・ブレークのみ
トレース		分岐トレース*	分岐トレース/ データトレース*	分岐トレース/ データトレース*
パフォーマンス測定	測定項目	実行開始～停止	実行開始～停止	実行開始～停止
	性能	分解能 100μs, 最大測時間 100 時間	E8a：分解能 1ms 注:リソースはPCのタイマ E1/E20：分解能1μs	分解能 1ms 注:リソースはPCのタイマ
疑似リアルタイムRAMモニタ(RRM)		可能 （モニタ時にCPU を占有）	可能（モニタ時にCPU を占有/デバッグDMAC）	可能（モニタ時にCPU を占有/デバッグDMAC）
Dynamic Memory Modification (DMM)		可能 （変更時に CPU を占有）	可能（変更時に CPU を占有/デバッグDMAC）	可能（変更時に CPU を占有/デバッグDMAC）
ホットプラグイン		不可	不可	不可
セキュリティ		10 バイト ID 認証 **	7バイトID認証**	7バイトID認証**
占有端子数		1本(TOOL0端子)	1本(MODE端子)	1本or2本or7本*
周辺ブレーク		可能	不可(R8C/5xは可)	不可

***対象デバイスにより機能が異なります。　　**IDコードの仕様の違いに関しては本資料の「IDコードの入力方法」を参照してください。**

ルネサス エレクトロニクス株式会社