

e² studio

統合開発環境

ユーザーズマニュアル 入門ガイド

対象デバイス

RX, RL78, RH850, RZ ファミリ

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

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

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

このマニュアルの使い方

このマニュアルは、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 e² studio について説明します。

e² studio は RX ファミリ、RL78 ファミリ、RZ ファミリ、および Renesas Synergy 用の統合開発環境で、ソフトウェア開発における、設計、実装、デバッグなどの各開発フェーズに必要なツールを単一のプラットフォームに統合しています。Synergy 開発プラットフォームについては、以下の Synergy Gallery のサイトにあるドキュメントを参照してください。
<https://synergygallery.renesas.com/>

統合することで、さまざまなツールを使い分けずにすべての開発を本製品のみで行うことができます。

対象者 このマニュアルは、e² studio を使用してソフトウェアやハードウェアのアプリケーション・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、ユーザがターゲットデバイスを使用してハードウェアやソフトウェアのシステム開発を始める際の e² studio の機能を説明します。

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 インストール

第 3 章 プロジェクトの生成

第 4 章 ビルド

第 5 章 デバッグ

第 6 章 ヘルプ

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	: 10 進数 ... XXXX 16 進数 ... 0XXXXX

目次

第1章 概説	1
1.1. システム構成	1
1.2. 動作環境	1
1.2.1. システム要件	1
1.2.2. サポートするツールチェーン	2
第2章 インストール	3
2.1. e ² studioのインストール	3
2.2. e ² studioのアンインストール	4
2.3. e ² studioのバージョンアップ	4
2.4. コンパイラパッケージのインストール	6
第3章 プロジェクトの作成	7
3.1. 新規プロジェクトの作成	7
3.2. ワークスペースへの既存プロジェクトのインポート	11
3.3. HEWプロジェクトのインポート	17
3.4. CS+プロジェクトのインポート	19
第4章 ビルド	21
4.1. ビルドオプションの設定	21
4.2. サンプルプロジェクトのビルド	23
4.3. ビルド構成のレポート	24
第5章 デバッグ	25
5.1. 既存デバッグ構成の変更	26
5.2. 新規デバッグ構成の作成	30
5.3. 基本的なデバッグ機能	32
5.3.1. ブレークポイントの使い方	33
5.3.2. 式ビュー	35
5.3.3. レジスタービュー	36
5.3.4. メモリービュー	37
5.3.5. 逆アセンブルビュー	39
5.3.6. 変数ビュー	41
5.3.7. イベントポイントビュー	42
5.3.8. IOレジスタビュー	45
5.3.9. トレースビュー	46
第6章 ヘルプ	49

第1章 概説

ルネサス製 Eclipse ベースの組み込みスタジオ(通称 e² studio)は、ルネサス製マイクロコントローラをサポートする開発環境です。これは、ビルドフェーズ(エディタ、コンパイラ、リンカコントロールなど)及び拡張 GDB インタフェースをサポートするデバッグフェーズをサポートしたオープンソース Eclipse CDT (C/C++ 開発ツール)プロジェクトに基づき開発されています。

この章では、RX ファミリシリーズマイクロコントローラ用のアプリケーション開発を例として、e² studio のシステム構成と動作環境を説明します。

本ドキュメントの e² studio に関する説明は、特に注意書きのない限り e² studio V5.0 に基づくものです。

1.1. システム構成

一般的なシステム構成の例を以下に示します。



図 1-1 システム構成

1.2. 動作環境

本製品に対するシステムの必要条件を以下に示します。

1.2.1. システム要件

PC ハードウェア環境:

プロセッサ: 1GHz 以上(ハイパースレッディング及びマルチコア CPU をサポートする)

メインメモリ: 1GB 以上(特に Windows 64 ビット OS に関しては 2GB 以上が望ましい)

ディスプレイ: 解像度 1,024 x 768 ピクセル以上; 65,536 色以上

インタフェース: USB 2.0 (ハイスピードまたはフルスピード) ハイスピードが望ましい

PC ソフトウェア環境:

Windows 7 (32/64 ビット OS), Windows 8.1 (32/64 ビット OS), および Windows 10 (32/64 ビット OS)

1.2.2. サポートするツールチェーン

1.2.2.1. サポートするコンパイラ

[RX ファミリー用ルネサス C/C++コンパイラパッケージ](#)
[RL78 ファミリー用ルネサス C/C++コンパイラパッケージ](#)
[GCC for Renesas GNURX Windows ツールチェーン](#)
[GCC for Renesas GNURL78 Windows ツールチェーン](#)
[Renesas GNUARM-NONE Windows ツールチェーン](#)

注: Renesas 製コンパイラパッケージには「統合開発環境付き」、「統合開発環境なし」がありますが、e² studio にはいずれも適用可能です。

1.2.2.2. サポートするエミュレータ

E2 エミュレータ Lite (RX, RL78), E1 (RX, RL78, RH850), E20 (RX), IECUBE (RL78),
Segger J-Link (RX, RZ)

1.2.2.3. サポートするシミュレータ

Renesas シミュレータ (RX, RL78)

第2章 インストール

この章では、e² studio のインストールおよびアンインストールについて説明します。

最新の e² studio 統合開発環境インストーラパッケージはルネサスウェブサイト (http://www.renesas.com/e2studio_download) から無償でダウンロードできます。ダウンロードするには無料で登録できる MyRenesas アカウントが必要です。

- 注意:
1. インストーラは二種類ありますが、Web インストーラ(サイズの小さな方)を使用するにはインターネット接続が必要です。
 2. e² studio ではメジャーバージョン間のアップデートをサポートしていません(例: V4.x(V4.0.0 から V4.3.1 のいずれか)から V5.x(V5.0.0 以降)へのアップデート)。古いバージョンをアンインストールしてから新しいバージョンをインストールしてください。あるいは、新しいバージョンの e² studio を新しいフォルダにインストールしてください。
 3. e² studio インストーラ(V4 以降)には、インストール済みの e² studio の機能を削除または追加する [修正] 機能があります。
 4. [ヘルプ] メニューの [更新の検査] による更新機能はプラグインのアップデートを行うための機能です。e² studio 全体を更新するにはインストーラを御使用ください。

2.1. e² studio のインストール

- (1) e² studio インストーラをダブルクリックして e² studio インストールウィザードページを開いてください。
[Next] をクリックします。
- (2) **Install Folder**
デフォルトのインストール先は "C:\Renesas\e2_studio" に設定されています。変更する場合は、インストールするフォルダをテキストボックスに直接入力するか [Browse...] ボタンをクリックして指定してください。
また e² studio をインストールする Windows ユーザを選択してください。
[Next] をクリックします。
- (3) **Device Families**
インストールするデバイスファミリを選択してください。[Next] をクリックします。
- (4) **Extra Components**
インストールする追加コンポーネント(言語パック、SVN および Git サポート、Micrium、RTOS サポート)を選択してください。[Next] をクリックします。
- (5) **Components**
必要なコンポーネントを選択し、[Next] をクリックします。
- (6) **Additional Software**
追加ソフトウェア(コンパイラ、ユーティリティ)を選択し、[Next] をクリックします。
- (7) **License Agreement**
ライセンス契約を読んで同意した後、[Next] をクリックします。
ライセンス契約に同意しない場合、インストールは続行できません。

(8) Shortcuts

スタートメニューに表示するショートカット名を選択し、[Next] をクリックします。

(9) Summary

[Install] をクリックし、ルネサス e² studio をインストールしてください。

(10) Installing...

インストールが実行されます。Additional Software で選択した項目に応じて、ソフトウェアをインストールするためのダイアログボックスが開きます。

(11) Results

[Finish] ボタンをクリックしてインストールを完了してください。

2.2. e² studio のアンインストール

e² studio のアンインストールは、Windows OS での通常のプログラムアンインストール手順で行えます。

- (1) [スタート] → [コントロールパネル] → [プログラムと機能]を選択します。
- (2) インストール済みプログラムのリストから、"e² studio" を選択し、[アンインストール(U)] ボタンをクリックします。
- (3) [アンインストール] ダイアログボックスの [アンインストール] ボタンをクリックして削除を確認してください。

アンインストールの最後に、e² studio はインストール先から削除され、ショートカットメニューも削除されます。

2.3. e² studio のバージョンアップ

本節では、インストーラのバージョンアップ機能を用いてバージョンアップする手順を説明します。

- (1) 下記のルネサスページからインストールしたいバージョンの e² studio インストーラをダウンロードしてください。http://www.renesas.com/e2studio_download

注：差分アップデートプログラムを使用したバージョンアップは、バージョン 3.X 以下の e² studio でのみ可能です。

- (2) (1)でダウンロードしたインストーラファイルをダブルクリックして実行してください。インストーラがインストール済みのバージョンを検出します。インストール済みの e² studio を更新するか、あるいは別のフォルダに新しいバージョンの e² studio をインストールするかを選択できます。更新する場合は[アップグレード] を選択し、[次へ(N)>] をクリックしてバージョンアップを開始してください。表示されたフォルダに上書きしない場合は[インストール]を選択して別フォルダを指定してください。

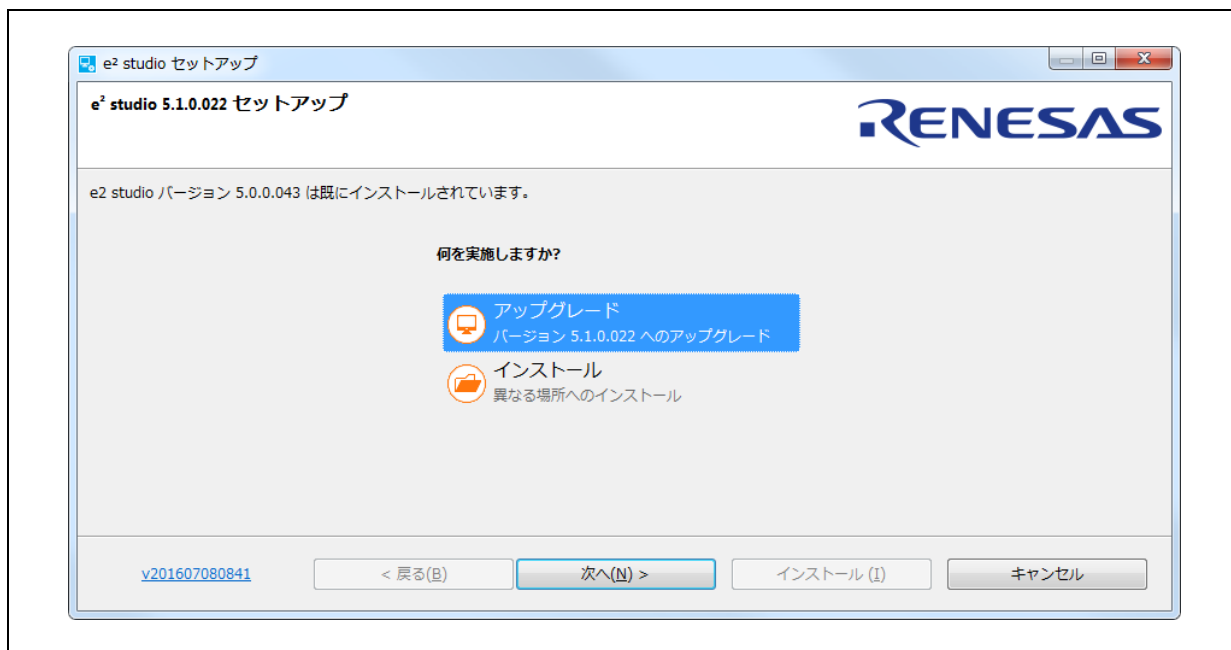


図 2-1 インストーラを使用した e² studio のバージョンアップ

- (3) 「2.1 章 e² studio のインストール」の手順を実行してください。[アップグレード] を選択した場合は既存の e² studio と同じフォルダにインストールしますので、「2.1 章の(2) Install Folder」の手順は不要になります。

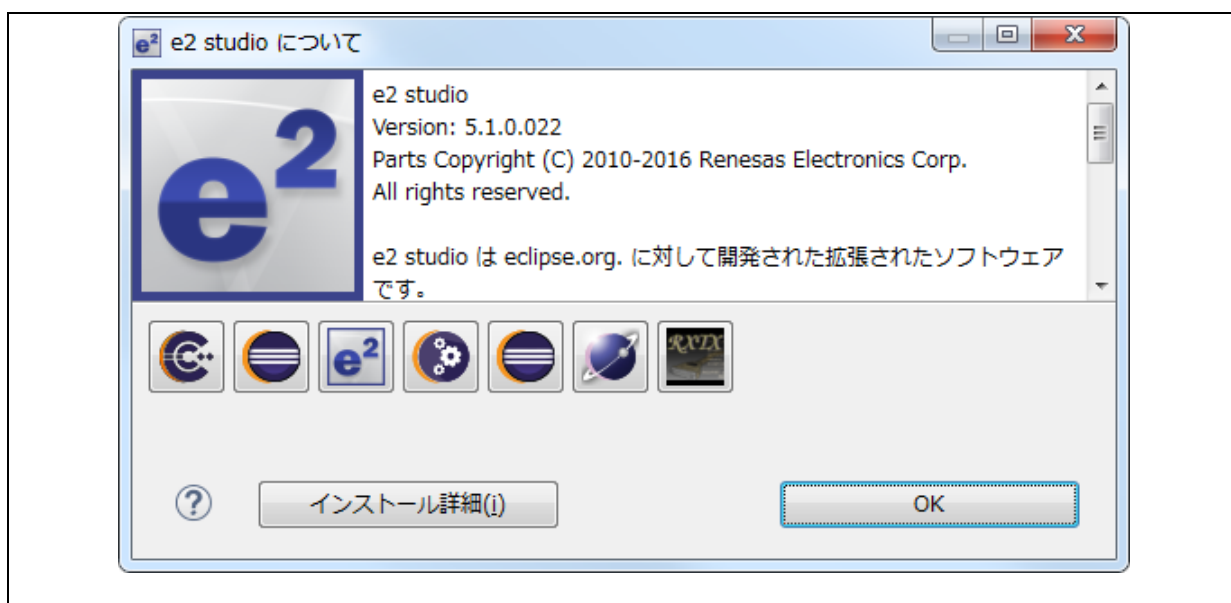


図 2-2 e² studio – [e2 studio について] パネル

- (4) [ヘルプ] メニューの [e2 studio について] で e² studio のバージョンが確認できます。

2.4. コンパイラパッケージのインストール

V4.0 以降の e² studio インストーラでは、インターネット接続環境ではインストール時に自動的にコンパイラパッケージをインストールできますが、インターネット接続のない環境、または V3.1 以前の e² studio インストーラを使用した場合にはコンパイラパッケージがインストールされません。


コンパイラパッケージを別途インストールする必要がある場合には、それぞれ下記のサイトからダウンロードしてください。コンパイラパッケージのインストール方法については各パッケージのダウンロードページをご覧ください。

Renesas 製コンパイラパッケージのダウンロードサイト:

RX ファミリー用 http://www.renesas.com/rx_c
 RL78 ファミリー用 http://www.renesas.com/rl78_c

GCC for Renesas ツールチェーンのダウンロードサイト:

<https://gcc-renesas.com/ja/>

コンパイラがインストール済みであるかを確認するには、ツールバーの  をクリックするか、[ヘルプ] メニューから[Renesas ツールチェーンの追加] をクリックして下図に示す [Renesas ツールチェーン管理] を開きます。e² studio に統合したいツールチェーンを確認してください。

使用したいコンパイラがリストにない場合は、[追加] ボタンをクリックしてインストールされている場所を指定してください。

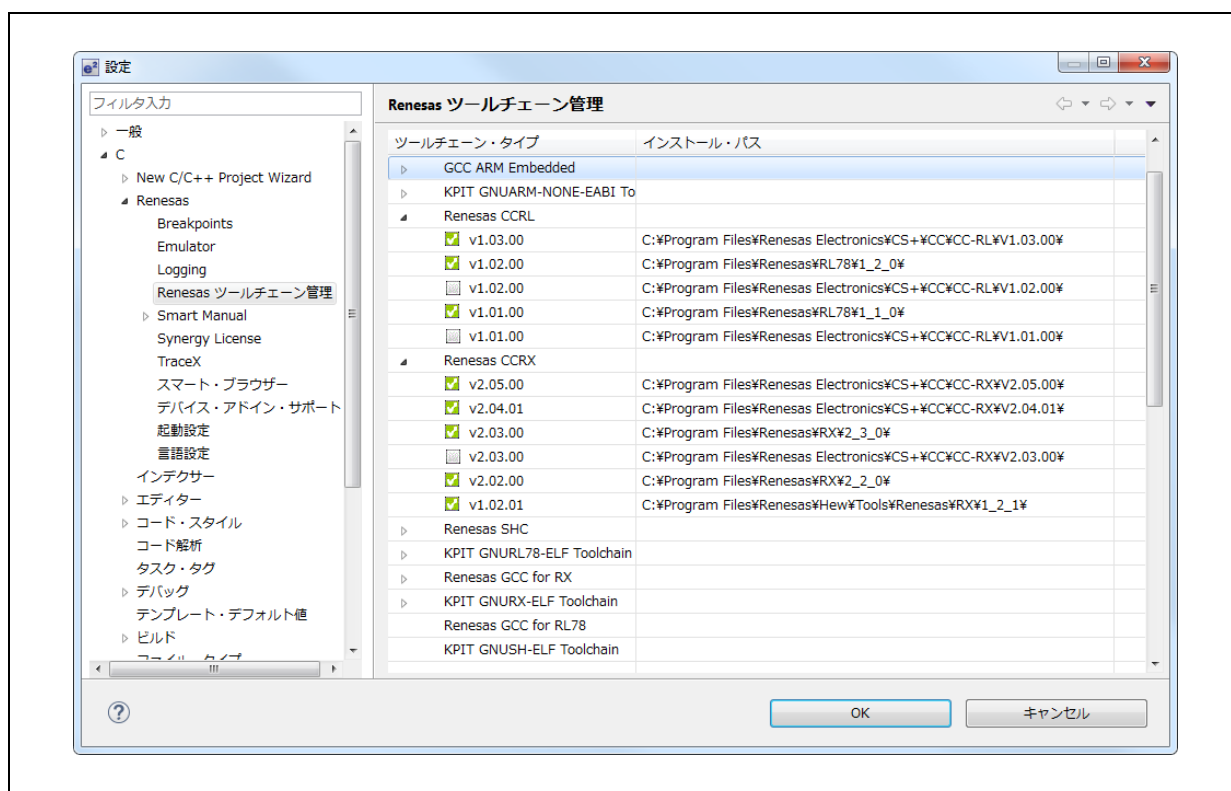


図 2-3 ツールチェーン管理

第3章 プロジェクトの作成

この章では、新規プロジェクトの作成、および既存の e² studio プロジェクト、HEW 統合開発環境プロジェクト、CS+プロジェクトの e² studio へのインポートについて説明します。

- 注意:
1. ご利用の PC に e² studio をインストールして使用する場合は、別途提供されるコンパイラパッケージをインストールする必要があります。
 2. e² studio インストールフォルダ名、ワークスペースフォルダ名、プロジェクト名、ビルド構成名、ソースファイル名に多バイト文字は使用できません。ソースファイルを配置するファイルパスにも多バイト文字が含まれないように留意してください。

3.1. 新規プロジェクトの作成

ルネサス RXC ツールチェーンで新規プロジェクトを作成するには、Windows の [スタート] メニューから e² studio を起動し、ワークスペースディレクトリを指定します。

- (1) [ファイル] → [新規] → [C Project] の順にクリックして新しい C プロジェクトを作成します。下図に示す新規プロジェクト作成ウィザードが起動します。

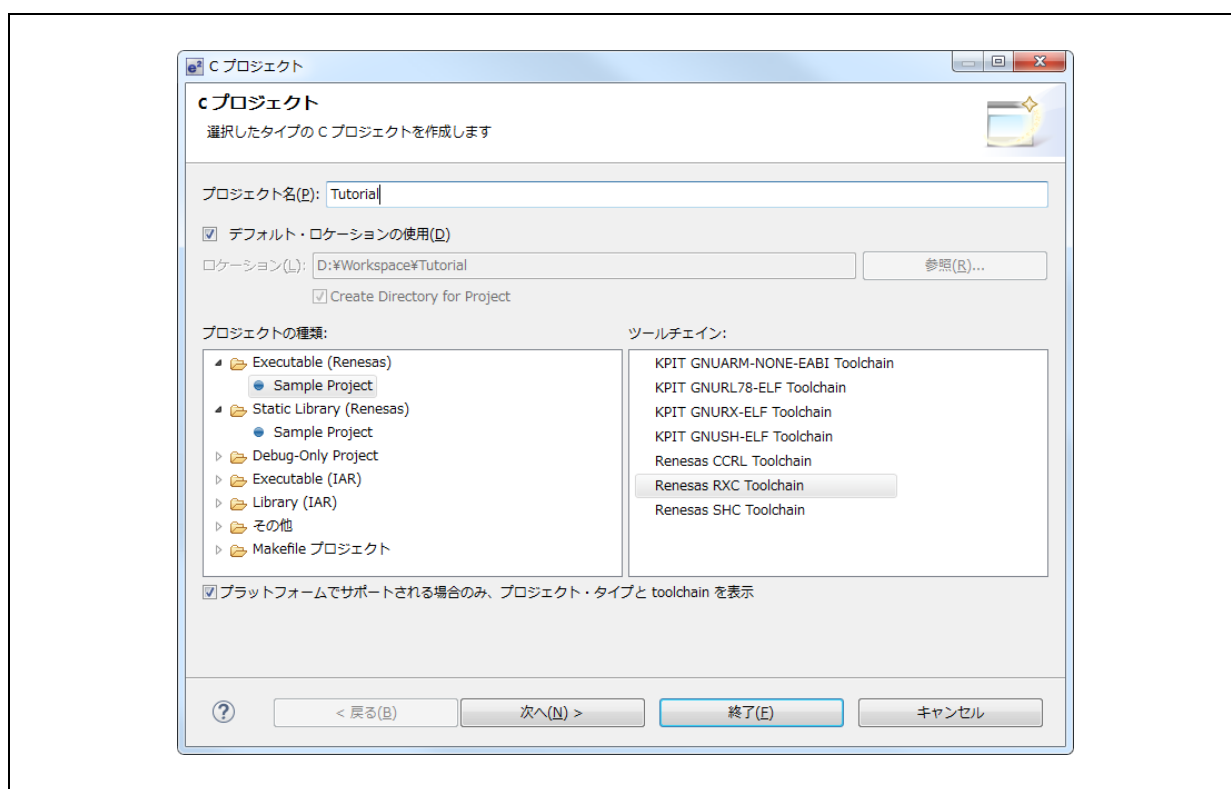


図 3-1 新規プロジェクト作成ウィザード (1/4)

- (2) プロジェクト名を入力し、“Renesas RXC Toolchain” を選択します。[次へ(N)]> をクリックしてください。“Renesas RXC Toolchain” が表示されない場合は「2.5 コンパイラパッケージのインストール」の手順にしたがってください。

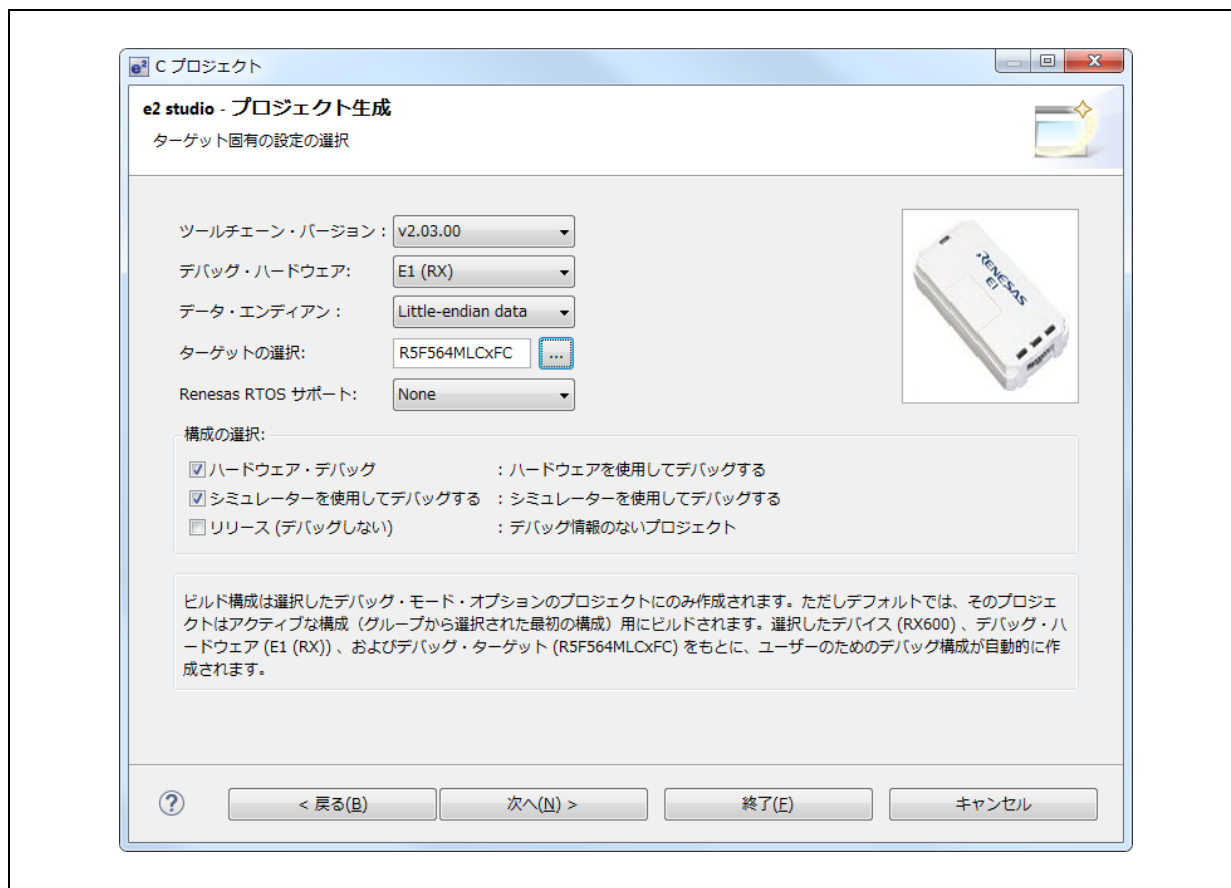


図 3-2 新規プロジェクト作成ウィザード (2/4)

- (3) [ツールチェーン・バージョン]、[デバッグ・ハードウェア]、[ターゲットの選択] を指定してください(例: ツールチェーン・バージョン: “v2.04.00”、デバッグ・ハードウェア: “E1”、ターゲットの選択: “RX64M (176 ピンデバイス, 型番: R5F564MLCxFC)”)。[次へ(N)>] をクリックして次へ進みます。

注意: E1 と同様に “E2 Lite” も [デバッグ・ハードウェア] のプルダウンメニューから選択できません。



図 3-3 新規プロジェクト作成ウィザード (3/4)

- (4) [コード生成を使用する] または[FIT モジュールを使用する]オプションが選択可能なデバイスでは、この画面でチェックマークをつけることでそれぞれの機能が利用可能なプロジェクトが作成できます。[FIT モジュールを使用する]をチェックした場合は、更にここでモジュールのダウンロード操作が行えます(FIT は RX ファミリーでのみ利用可能です)。

注：コード生成の利用可否は「インストール詳細」ダイアログの「Renesas デバイス・サポート」で確認できます。

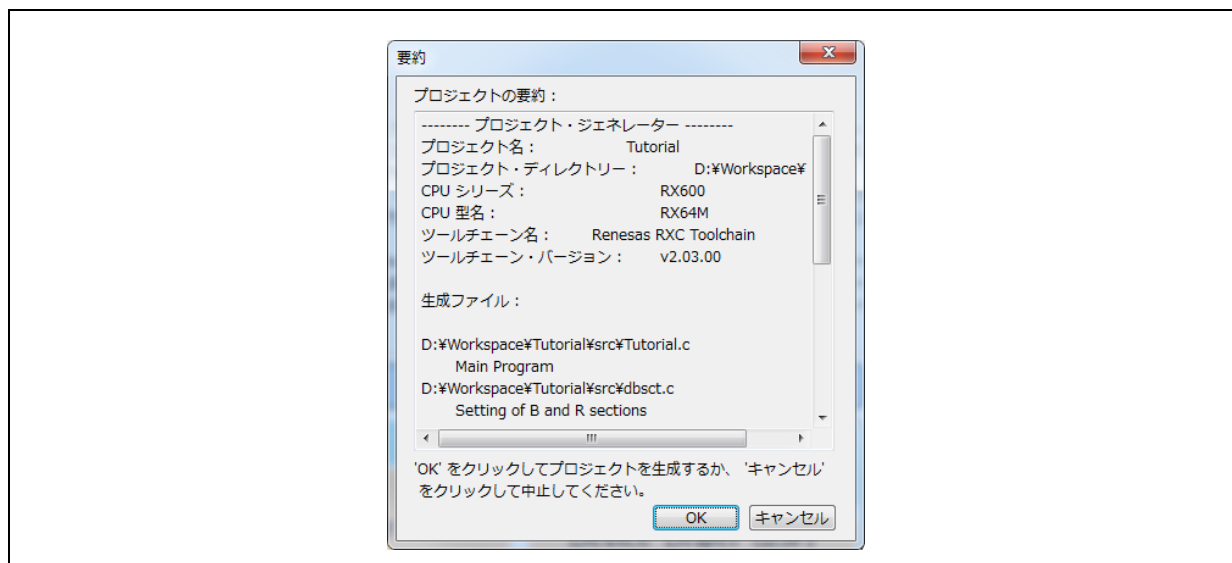


図 3-4 新規プロジェクト作成ウィザード (4/4)

- (5) 最後に[終了(E)]ボタンを押すとプロジェクトの要約が表示されます。更に[OK] をクリックするとプロジェクトが生成され、下記の画面例のように表示されます。

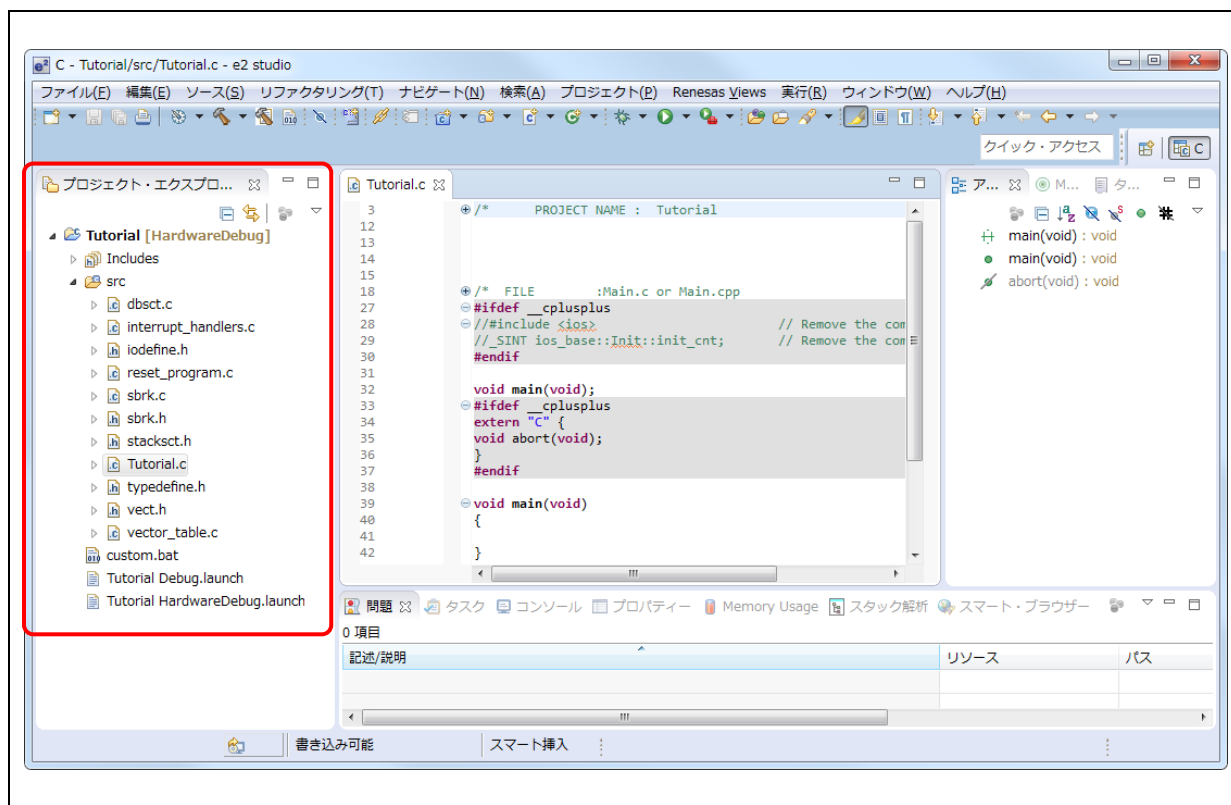


図 3-5 作成された新しい C プロジェクト

- (6) この例では「Tutorial」という新しい C プロジェクトが生成された様子を示します。

このプロジェクトは、「Tutorial.c」というアプリケーションファイルと、標準的なスタートアップファイル（dbsct.c, interrupt_handlers.c, sbrk.c など）から構成されています。[プロジェクト・エクスプローラ] パネルではすべてのプロジェクトおよびソースファイルを Windows エクスプローラと同様のフォルダ階層として表示します。

プロジェクトをバックアップする際の注意事項:

- 「.」(ドット)で始まる名前のファイルやフォルダ（例：.project）にはプロジェクトの設定情報が含まれるので、バックアップを取る際にはこれらのファイルやフォルダを含めてプロジェクトのフォルダ全体を圧縮するなどしてください。
- 他のプロジェクトのファイルを参照する設定など、プロジェクト間で共有される設定を保存するためには、ワークスペース全体をバックアップする必要があります。

3.2. ワークスペースへの既存プロジェクトのインポート

本節では、既存のプロジェクトをディレクトリまたはアーカイブからワークスペースへインポートする手順を説明します。

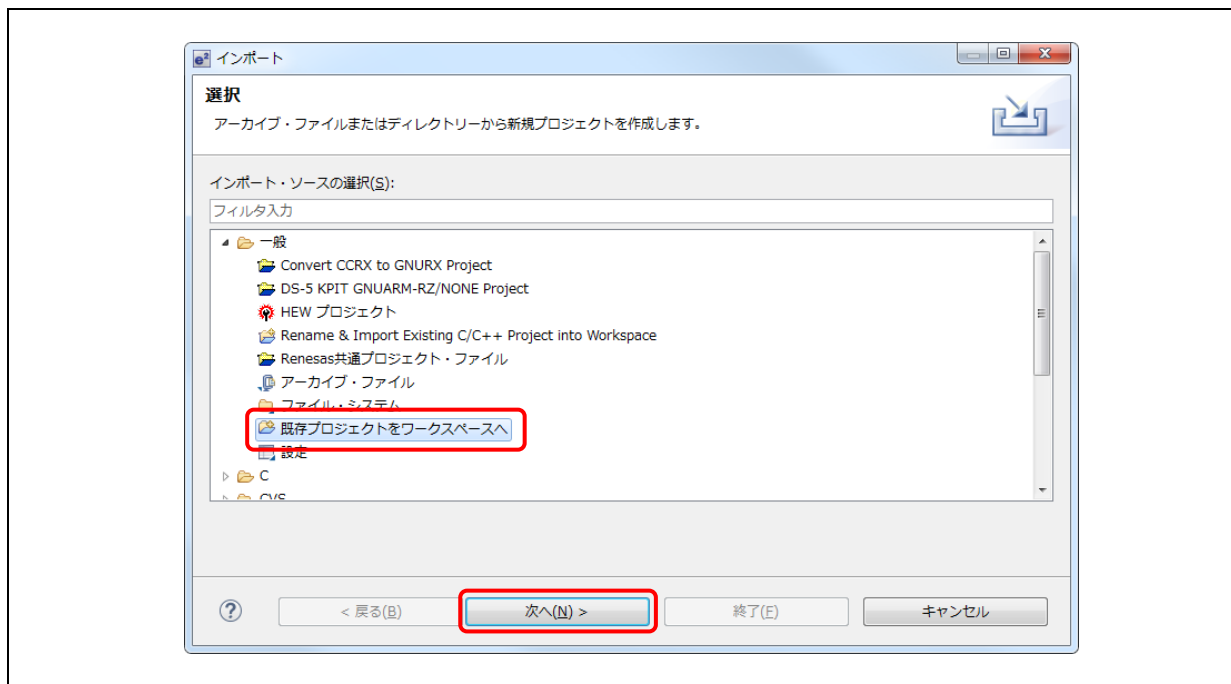


図 3-6 既存プロジェクトインポートウィザード

- (1) e² studio で、[ファイル] → [インポート] をクリックし、HEW プロジェクトインポートウィザードを開きます。「既存プロジェクトをワークスペースへ」を選択し、「次へ(N)>」ボタンをクリックして [プロジェクトのインポート] ウィンドウを開いてください。

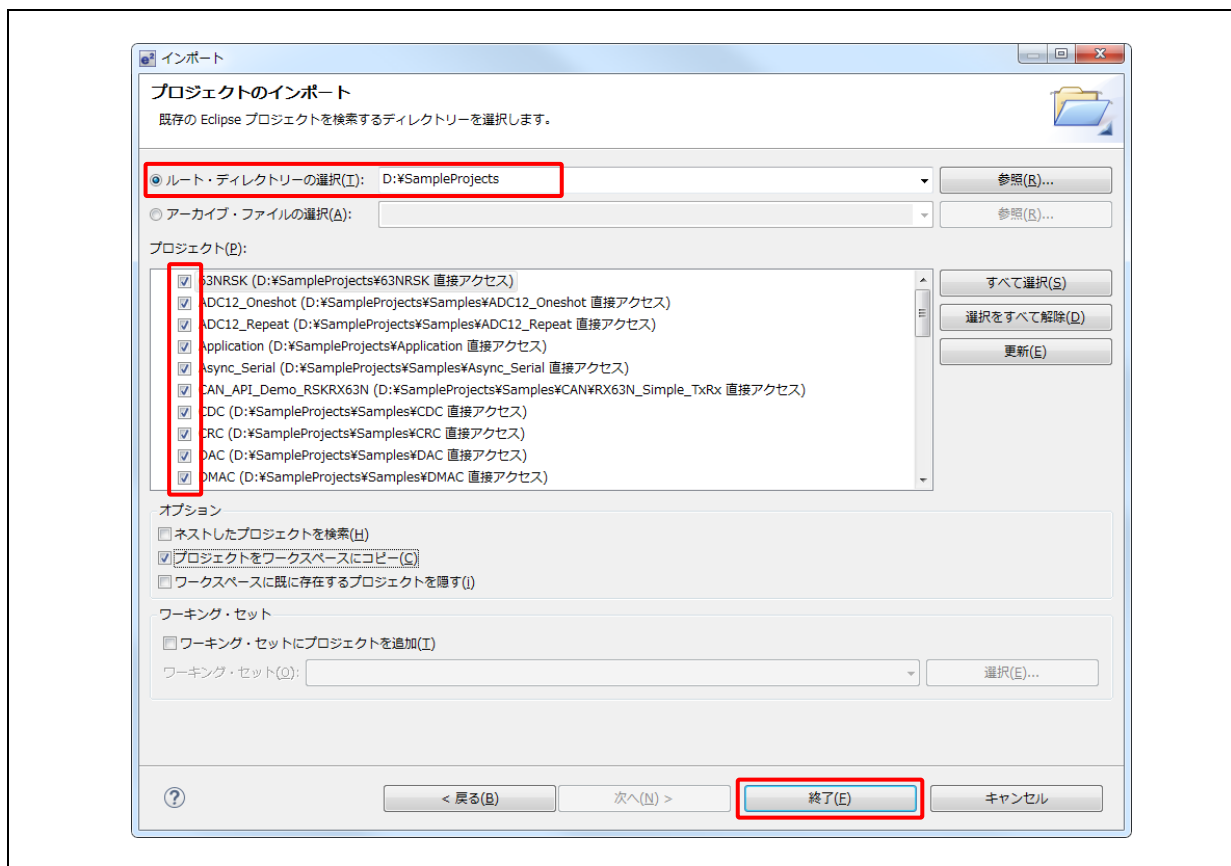


図 3-7 e² studio の [プロジェクトのインポート] ウィンドウ

- (2) プロジェクトが保管されているディレクトリまたはアーカイブを検索して選択します。既存のプロジェクトがすべて表示されますので、インポートするプロジェクトを選択してください。[終了(F)] をクリックしてプロジェクトのインポートを完了します。

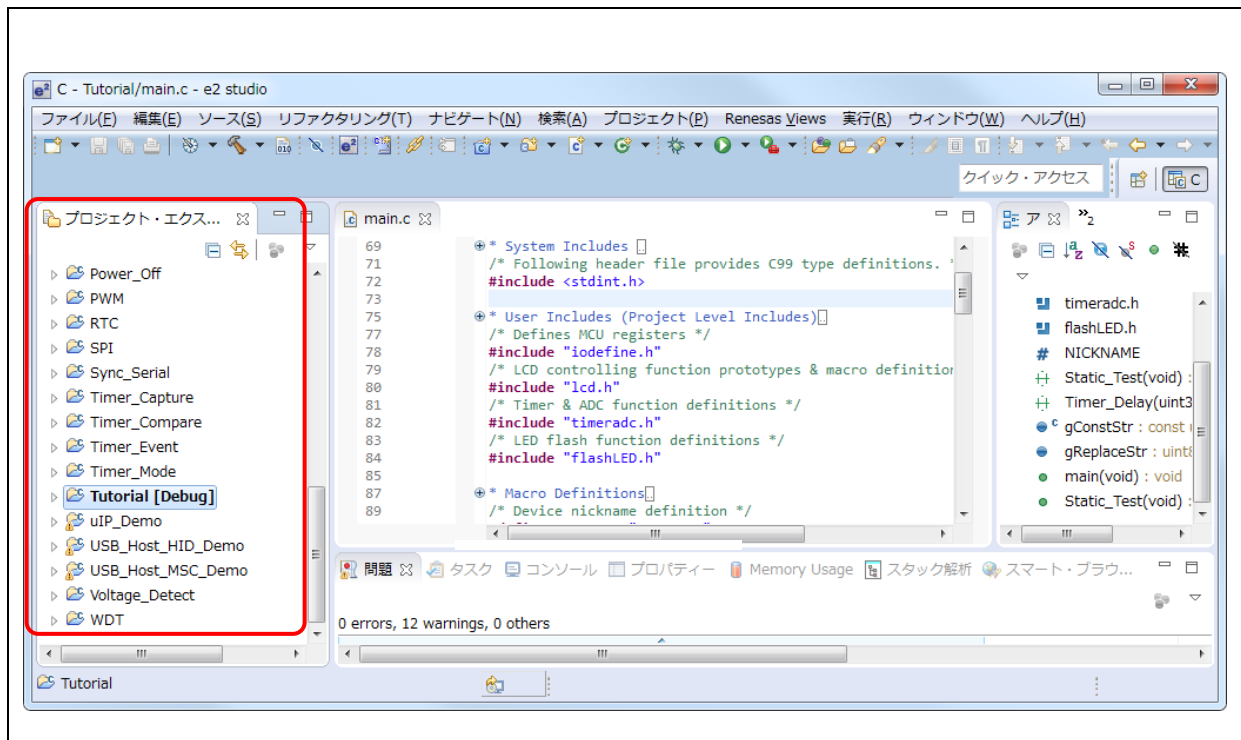


図 3-8 e² studio でのプロジェクトのインポート

(3) e² studio へのプロジェクトのインポートが完了しました。

既存のプロジェクト名のままインポートするのではなく、プロジェクト名を変更してインポートすることもできます。このオプションを使用する場合、一度にインポートできるプロジェクトは一つのみです。

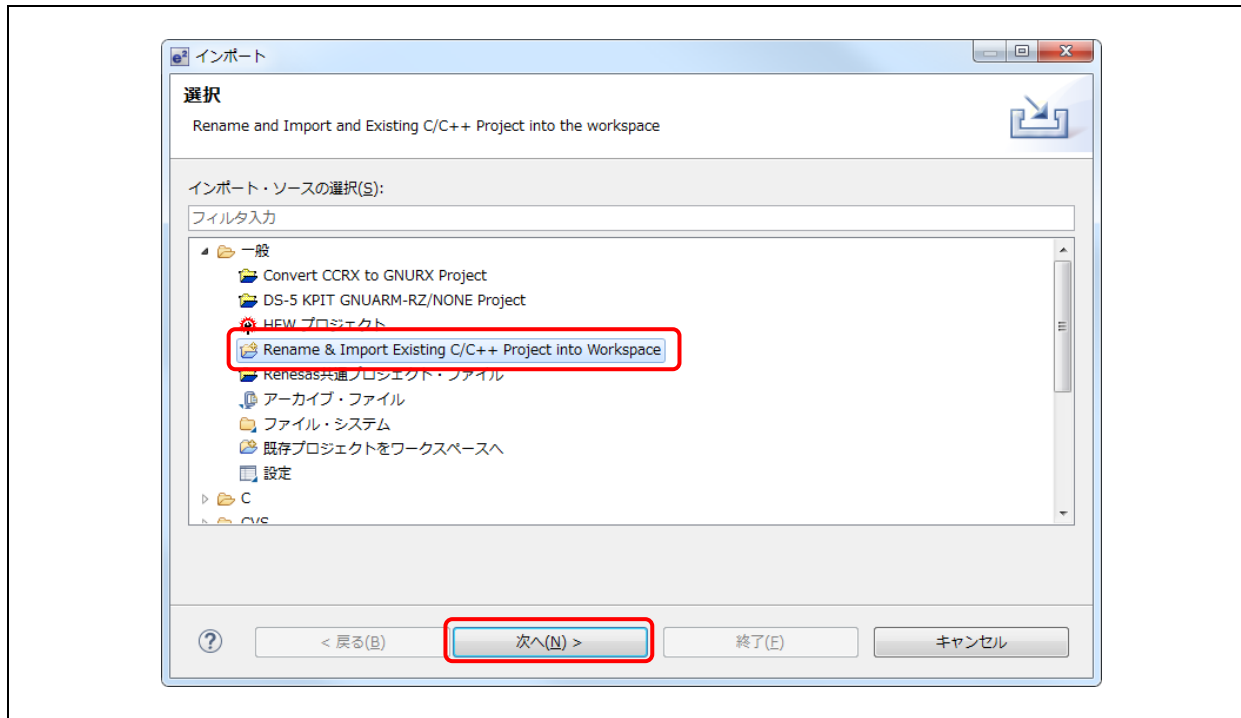


図 3-9 名称変更とプロジェクトのインポートウィザード

- (1) e² studio で、[ファイル] → [インポート] をクリックし、HEW プロジェクトインポートウィザードを開きます。“Rename & Import Existing C/C++ Project into Workspace” を選択し、[次へ(N)>] ボタンをクリックして [名称変更とプロジェクトのインポート] ウィンドウを開いてください。

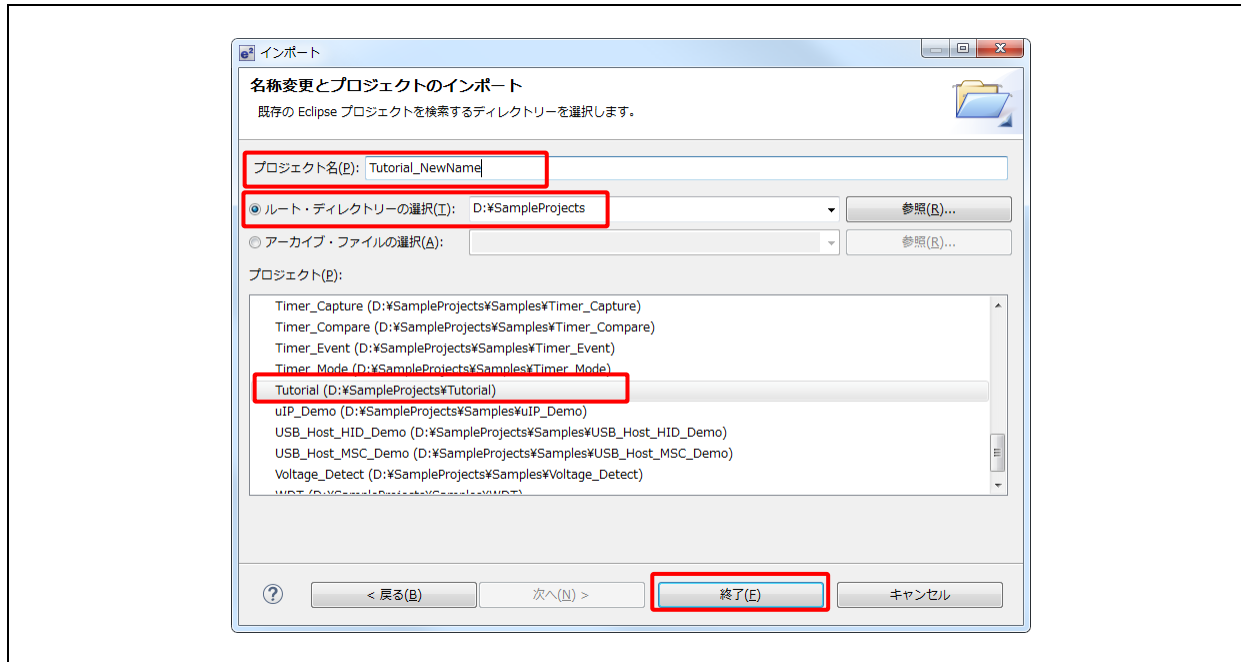


図 3-10 e² studio の [名称変更とプロジェクトのインポート] ウィザード

- (2) プロジェクトが保管されているディレクトリまたはアーカイブを検索して選択します。既存のプロジェクトがすべて表示されますので、インポートするプロジェクトを選択し、新しい名称を入力してください。[終了(F)] をクリックしてプロジェクトのインポートを完了します。

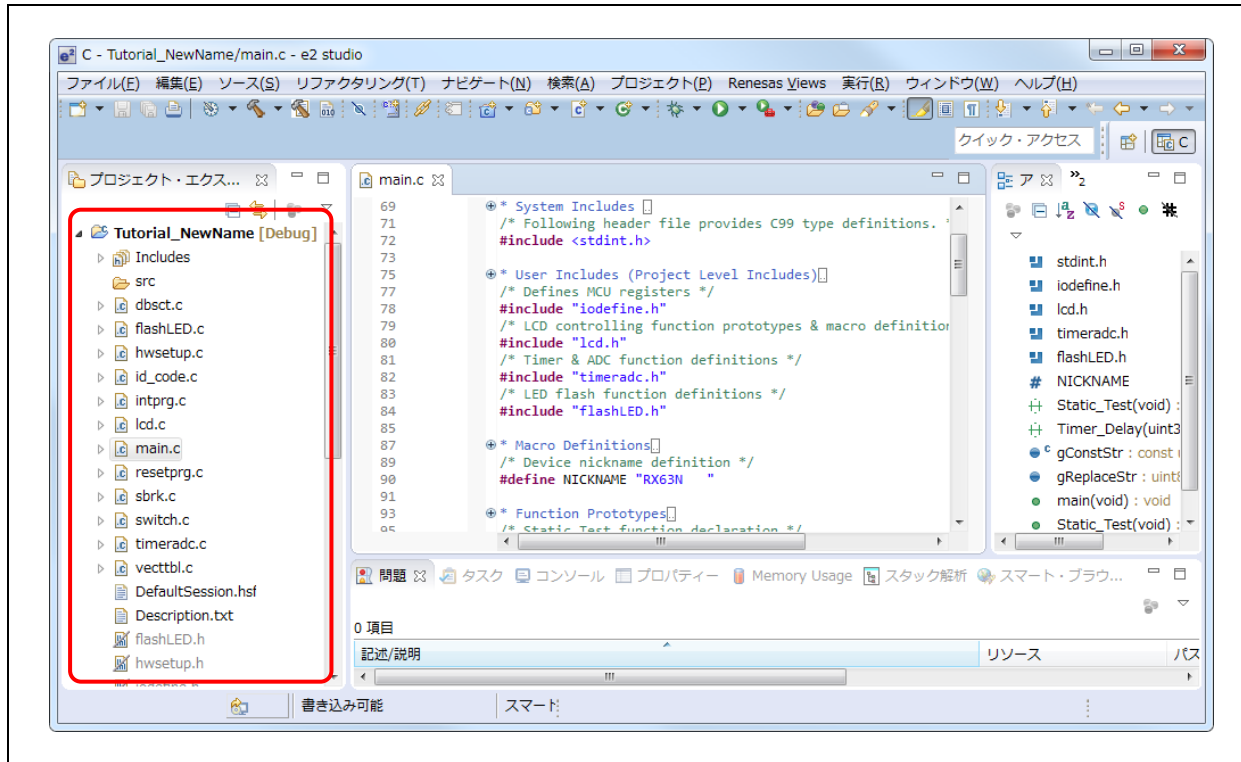


図 3-11 e² studio での名称変更とプロジェクトのインポート

(3) プロジェクトの名称変更と e² studio へのインポートが完了しました。

3.3. HEW プロジェクトのインポート

本節では、既存のプロジェクトワークスペースを e² studio に移行する HEW プロジェクトのインポート機能を説明します。これにより、HEW で作成した既存のアプリケーションプログラムやワークスペースを再利用することができます。

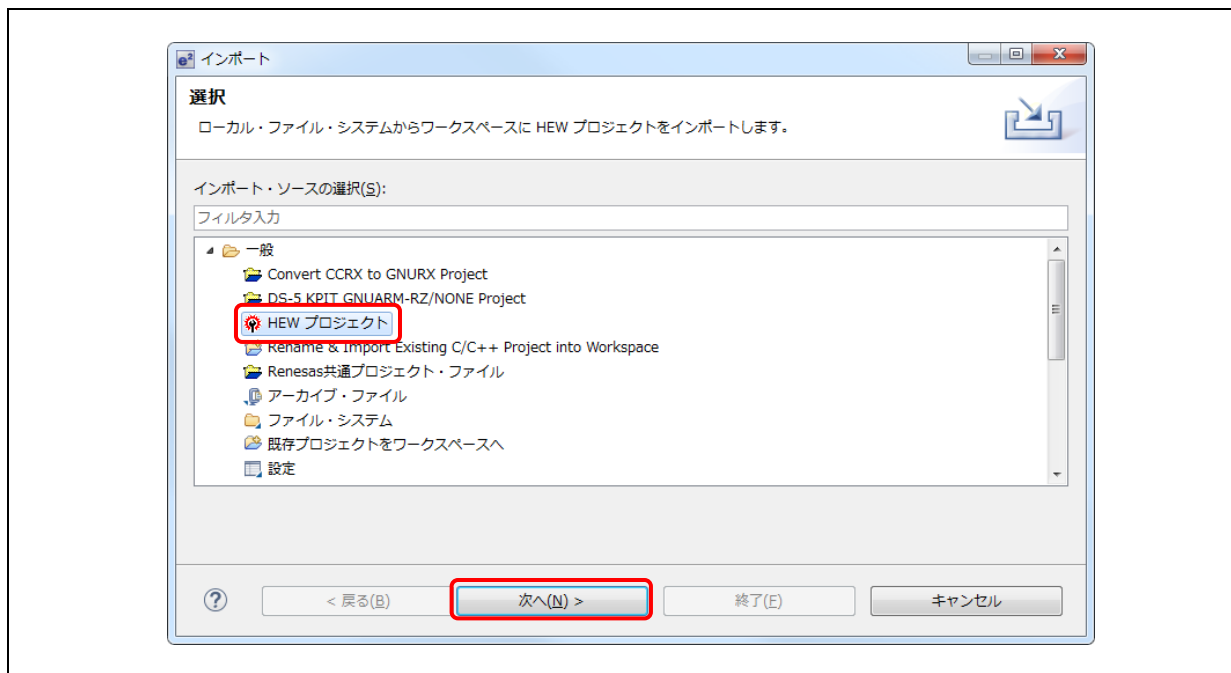


図 3-12 HEW プロジェクトインポートウィザード

- (1) e² studio で、[ファイル] → [インポート] をクリックし、HEW プロジェクトインポートウィザードを開きます。“HEW プロジェクト” を選択し、[次へ(N)>] ボタンをクリックして [HEW インポート] ウィンドウを開いてください。
- (2) HEW プロジェクトファイル (.hwp) を検索して選択し、[終了(F)] ボタンをクリックしてこのプロジェクトをインポートします。

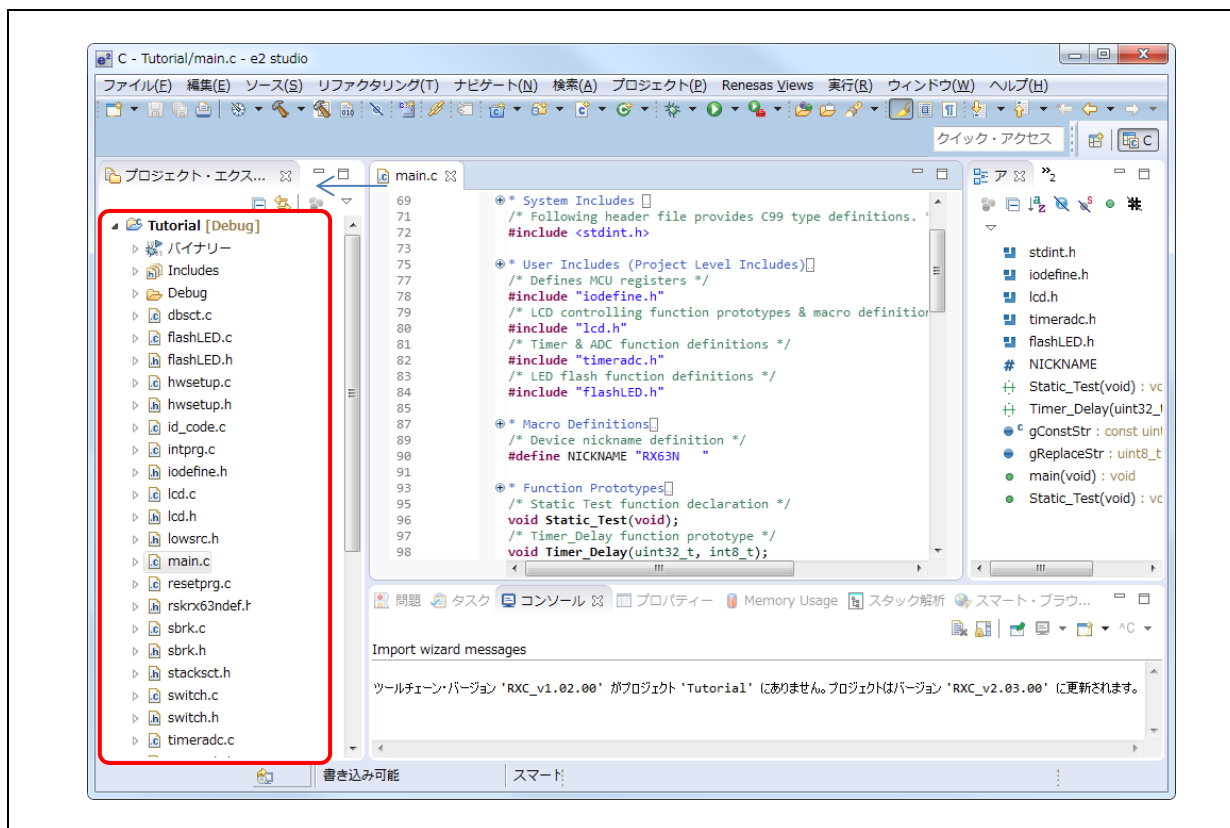


図 3-13 HEW から e² studio へのプロジェクトの移行

(3) HEW プロジェクトの e² studio へのインポートが完了しました。

変換の後、すべてのオリジナルプロジェクトとソースファイルは、新たに作成されたプロジェクトワークスペースとともに e² studio に保存されます。さらに、“`.cproject`”、“`*.linker`”、“`.info`” 及び “`.project`” が作成され、追加されます。

HEW と e² studio プロジェクトのワークスペースは物理的に同じ場所のファイルを共有します。

HEW プロジェクトのインポートが失敗した場合、以下の 2 つの前提条件を確認してください。

- (i) HEW プロジェクトワークスペースはバージョン v4.07 以降でなければなりません。
- (ii) 再度 HEW プロジェクトをインポートする場合はその前に、“`.cproject`”、“`*.linker`”、“`.info`” 及び “`.project`” などのファイルを手動で削除しておく必要があります。

参考: v4.07 より古い HEW プロジェクトのワークスペースをインポートしたい場合は、まず HEW v4.07 以降を使用してワークスペースを更新し、その後にインポートしてください。

3.4. CS+プロジェクトのインポート

プログラムを再利用するために、本節では、既存のプロジェクトワークスペースを e² studio に移行する CS+ のインポート機能を説明します。

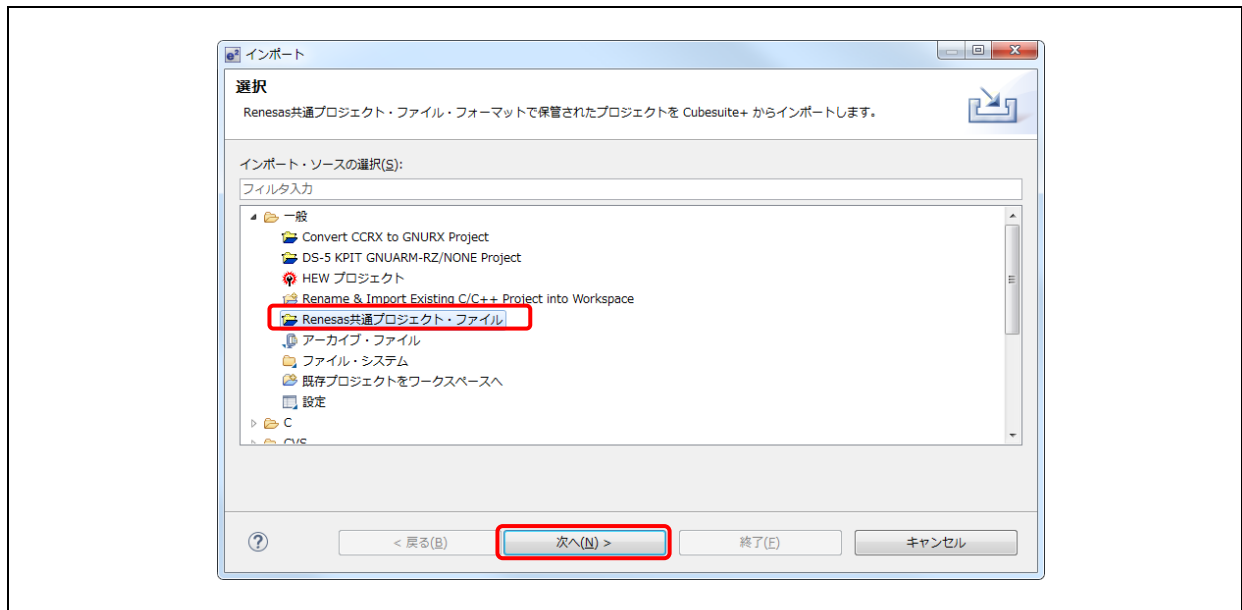
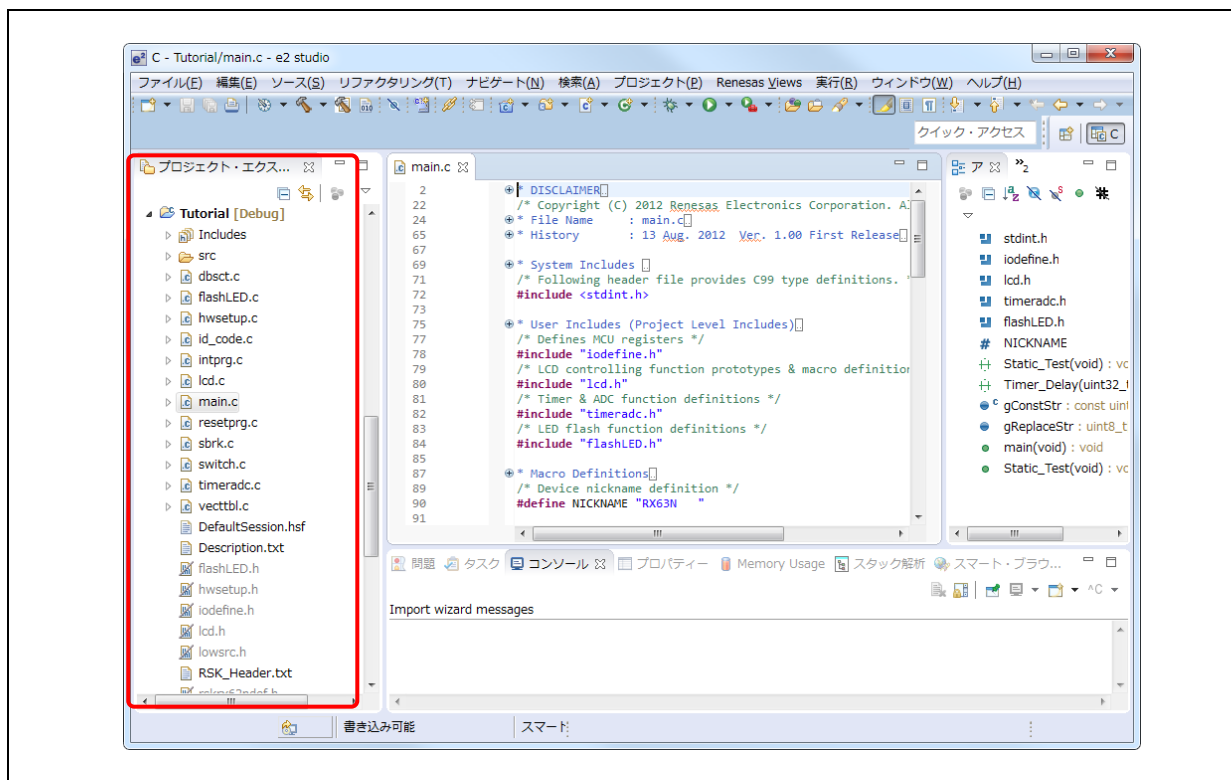


図 3-14 CS+プロジェクトインポートウィザード

e² studio で、[ファイル] → [インポート] をクリックします。“Renesas 共通プロジェクト・ファイル” を選択し、[次へ(N)>] ボタンをクリックしてください。[プロジェクトのインポート] ウィンドウが開きます。

- (1) CS+プロジェクトファイル (.rcpe) を検索して選択し、[終了(F)] ボタンをクリックしてこのプロジェクトをインポートします。

図 3-15 CS+から e² studio へのプロジェクトの移行

(2) CS+プロジェクトの e² studio へのインポートが完了しました。

変換の後、すべてのオリジナルプロジェクトソースファイルは、新たに作成されたプロジェクトワークスペースとともに e² studio に保存されます。さらに、“`.cproject`”、“`*.linker`”、“`.info`” 及び “`.project`” が作成され、追加されます。

CS+と e² studio プロジェクトのワークスペースは物理的に同じ場所のファイルを共有します。

第4章 ビルド

この章では、e² studio 統合開発環境のためのビルド構成と主要なビルド機能について説明します。

4.1. ビルドオプションの設定

プロジェクトを作成するとデフォルトのビルドオプション設定が生成されます。通常このデフォルトオプションでプロジェクトをビルドできます。しかしビルドオプションの変更が必要な場合は(ツールチェーンのバージョンや最適化オプションの変更など)、プロジェクトのビルド前に以下の手順で変更してください。

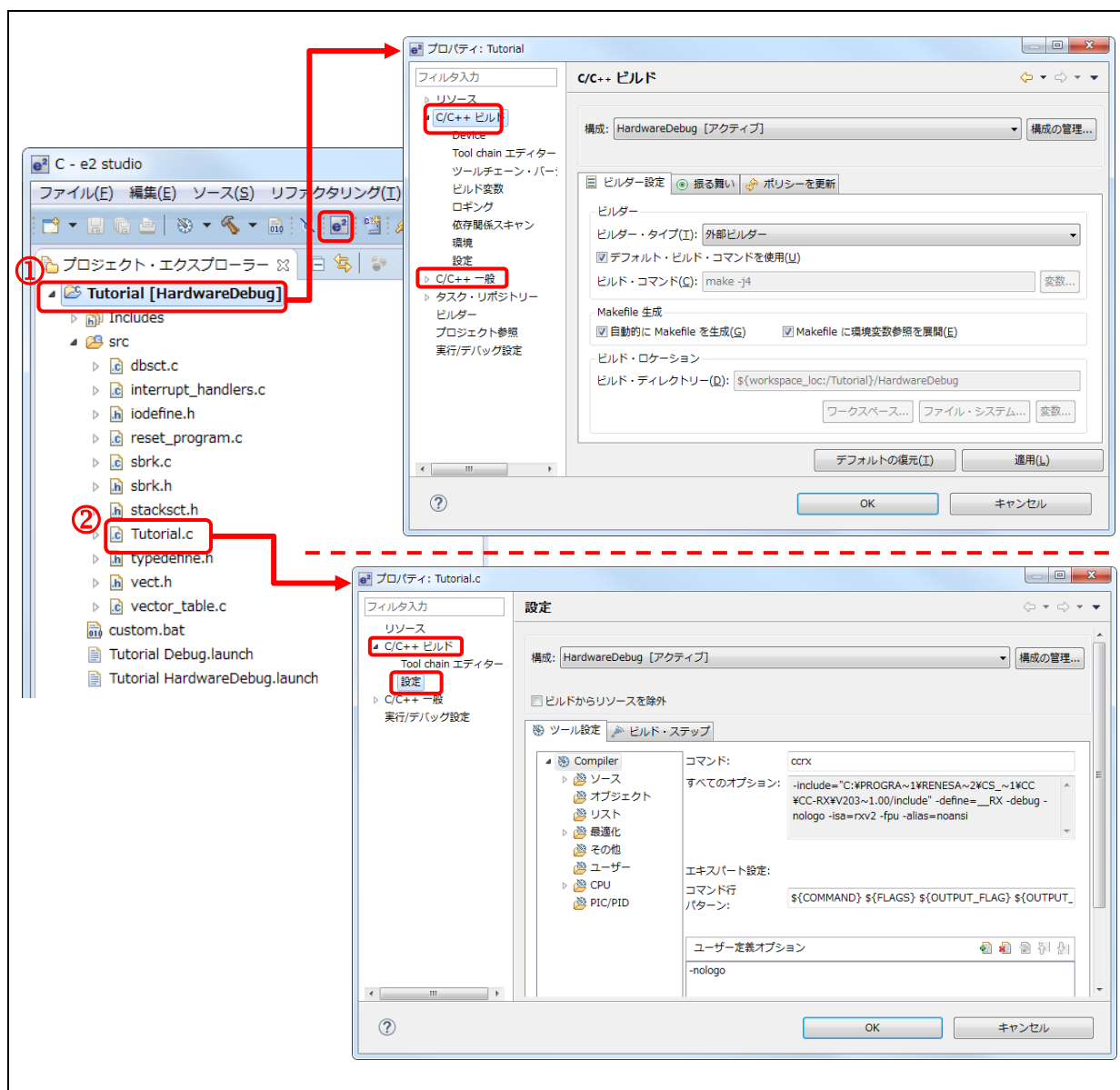



図 4-1 チュートリアルプロジェクトのプロパティと Tutorial.c ソースファイルのプロパティ

ビルドオプションは、プロジェクトのプロパティウィンドウまたはソースファイルのプロパティウィンドウで変更できます。

- (1) ① “Tutorial” プロジェクトまたは ② ソースファイル “Tutorial.c” を選択します。
- (2) アイコン  をクリックして(あるいは右クリックして [プロパティ] を選択するか、ショートカットキーの [Alt]+[Enter] もしくは [Alt]+[T] を使って)プロパティダイアログを開いてください。
- (3) “C/C++ ビルド” オプションをクリックし、構成の設定を表示、編集してください。

プロパティウィンドウはワークスペース、プロジェクト、ソースの各レベル別に設定できますが、ワークスペース内のファイル共通に適用できる項目の他に、個別のプロジェクトにはさらに多くの設定項目があります。

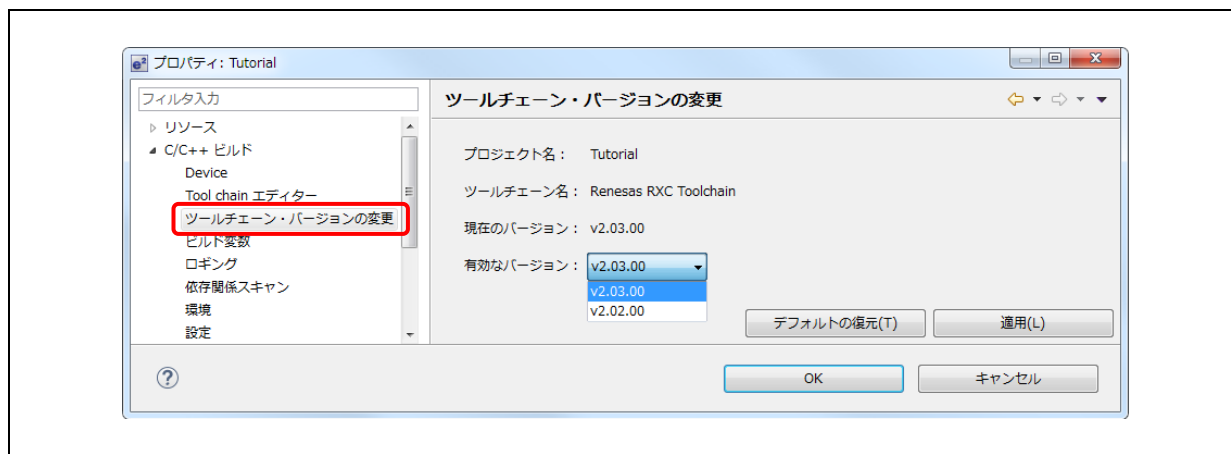


図 4-2 ツールチェーン・バージョンの変更

- (1) [C/C++ ビルド] → [ツールチェーン・バージョンの変更] をクリックして、ツールチェーン・バージョンを表示、変更してください。
[有効なバージョン]のドロップダウンリストからバージョンを選択できます。表示されるバージョンは図 2-3 で示したツールチェーン管理の設定に依存します。

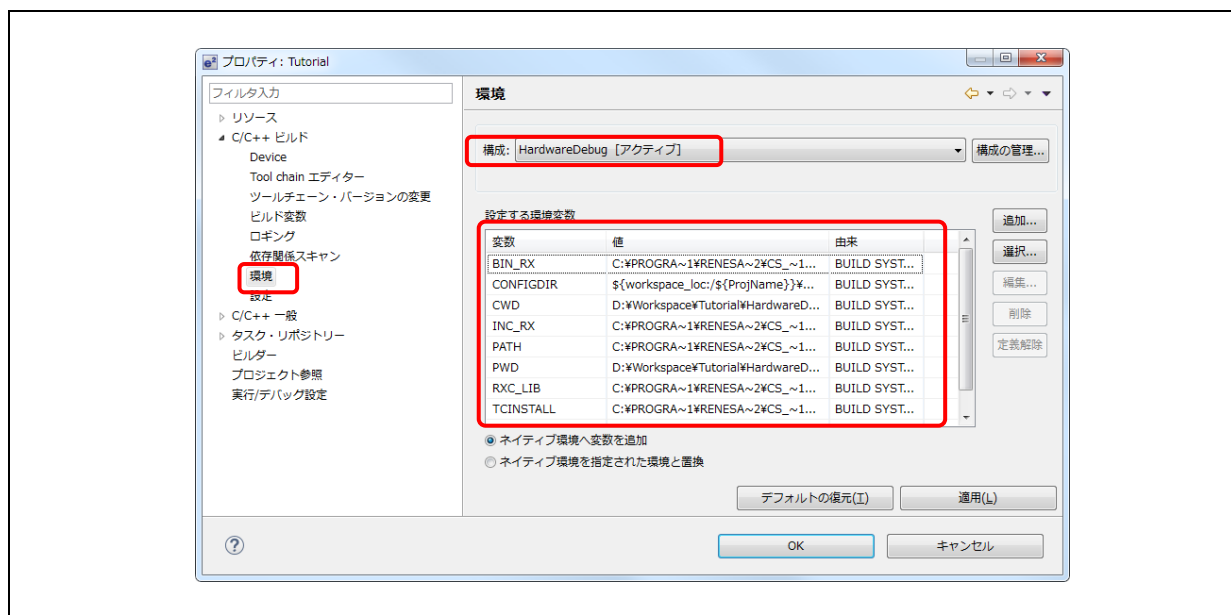


図 4-3 コンパイラのビルド設定: 環境

- (2) [C/C++ビルド] → [環境] の順にクリックし、ビルドオプションを設定し、環境変数を追加、編集してください。

ビルドオプションにより、環境変数を用いて指定したパスを含むすべてのツールチェーンの構成設定を保存することができます。現行のビルド構成は図 4-3 に示すように“HardwareDebug [アクティブ]”です。

ビルドオプションの詳細は、以下の場所にあるコンパイラユーザーズマニュアルを参照してください。
 “{コンパイラをインストールしたディレクトリ}\doc”（例： C:\Program Files\Renesas\RX\2_3_0\doc\）

4.2. サンプルプロジェクトのビルド

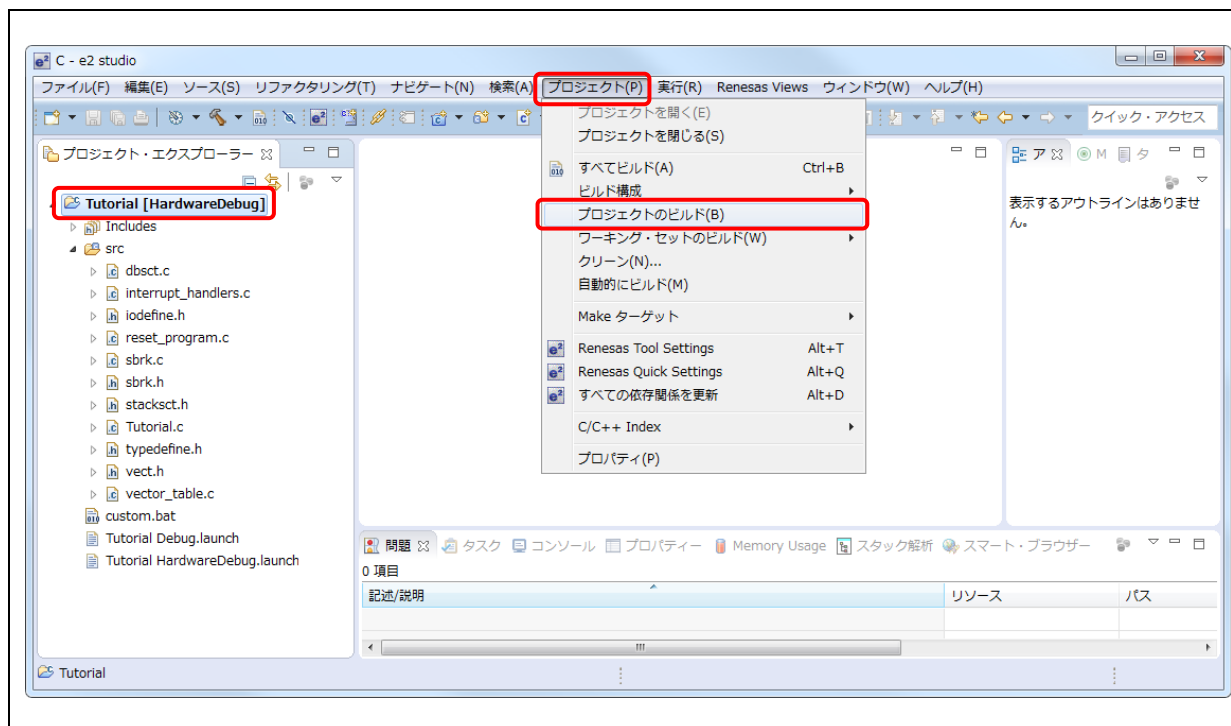



図 4-4 “チュートリアル” プロジェクトのビルド例

- (1) e² studio 環境下において、“Tutorial” という名前で新規プロジェクトを作成します(あるいは既存のプロジェクトのいずれかを開きます)。
- (2) [プロジェクト・エクスプローラー] で、当該のプロジェクトをクリックします。
- (3) [プロジェクト] → [プロジェクトのビルド] の順でクリックするか、 のアイコンをクリックして、本プロジェクトをビルドします。

[コンソール] に“ビルドが完了しました”というメッセージが表示されます。このビルドの最後に
 \${CONFIGDIR} ディレクトリに出力されたファイルは、“makefile”、“Tutorial.abs”、“Tutorial.map”、
 “Tutorial.mot”、“Tutorial.x”などで構成されます。

“Tutorial.abs” はデバッグに使われる ELF/DWARF フォーマット(*.abs)のルネサス標準ロードモジュールです。GDB は別の ELF/DWARF 仕様(*.x)のロードモジュールフォーマットをサポートしているため、e² studio のデバッグ用に“Tutorial.abs”を“Tutorial.x”に変換されます。

4.3. ビルド構成のレポート

プロジェクトレポート機能により、プロジェクトとビルド構成を e² studio からファイルに出力して、プロジェクトやビルド環境の設定を容易にチェックし、比較することができます。

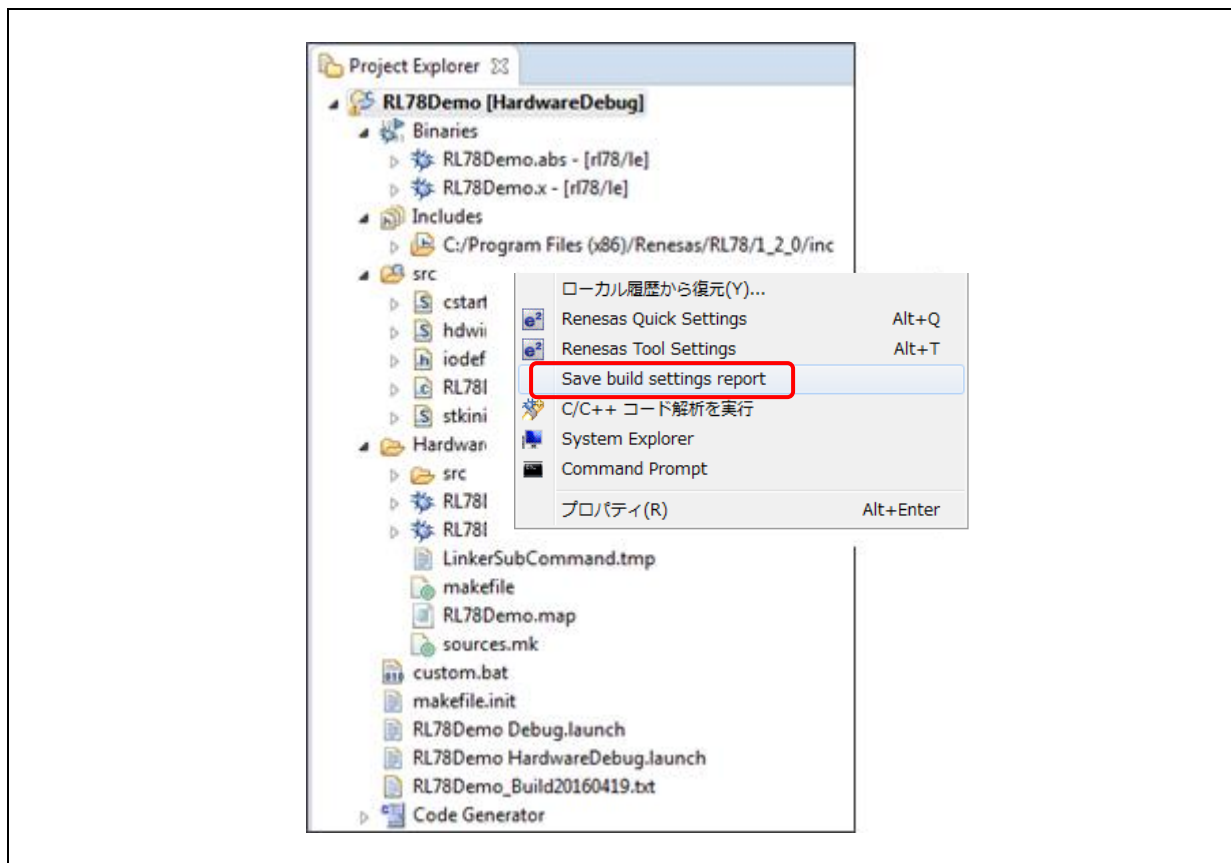


図 4-5 プロジェクトレポート

- (1) [プロジェクト・エクスプローラー] を右クリックしてコンテキスト・メニューを開きます。
- (2) [Save build setting report] を選択してビルド構成レポートを保存します。

第5章 デバッグ

この章では、e² studio 統合開発環境のデバッグ構成と主要なデバッグ機能の使い方について説明します。以下の説明は、E1 エミュレータまたは E2 エミュレータ Lite と RSK RX64M ボードのハードウェア構成に基づきビルドされた “Tutorial” プロジェクト(4.2 節)に関するものです。

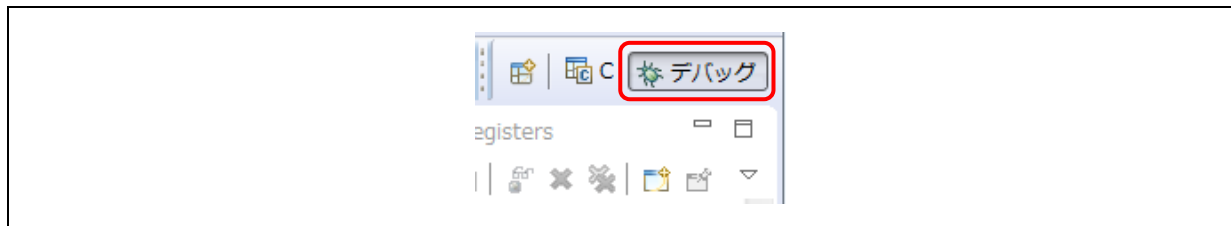


図 5-1 [デバッグ] パースペクティブへの切り替え

- (1) e² studio の “Tutorial” プロジェクトワークスペースを開き、[デバッグ] パースペクティブをクリックします。

パースペクティブはワークベンチの表示レイアウトを目的別に定義するものです。それぞれのパースペクティブは、ビュー、メニュー、ツールバーの組み合わせで構成され、パースペクティブを切り替えることで目的に適した画面レイアウトに切り替えることができます。

例えば、[C/C++] パースペクティブには C/C++プログラム開発に必要なビューが定義され、[デバッグ] パースペクティブにはプログラムのデバッグに必要なビューが定義されています。ユーザが [C/C++] パースペクティブ表示中にデバッガに接続しようとする場合、e² studio は [デバッグ] パースペクティブに切り替えるようユーザを促します。

ワークベンチは複数のパースペクティブを持つことができ、それぞれカスタマイズしたり、新しいパースペクティブを追加することもできます。

注意： デバッグについての詳細は、6 章で説明する「e² studio デバッグ・ヘルプ」を参照してください。

5.1. 既存デバッグ構成の変更

初めてデバッグを行う際にはデバッグ構成(デバッグに必要な一連のオプション)を設定してください。デフォルトの設定内容は以下の操作で変更できます。

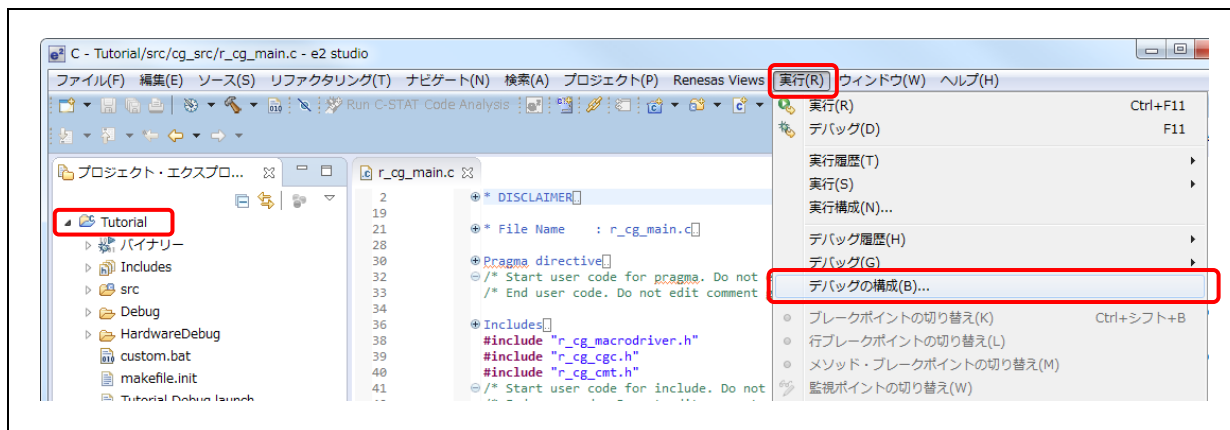


図 5-2 [デバッグ構成]ウィンドウを開く

(1) [プロジェクト・エクスプローラー] の“Tutorial” プロジェクトをクリックします。

[実行] → [デバッグの構成...] あるいは アイコン(下向き矢印) → [デバッグの構成] の順にクリックし、[デバッグ構成] ウィンドウを開きます。

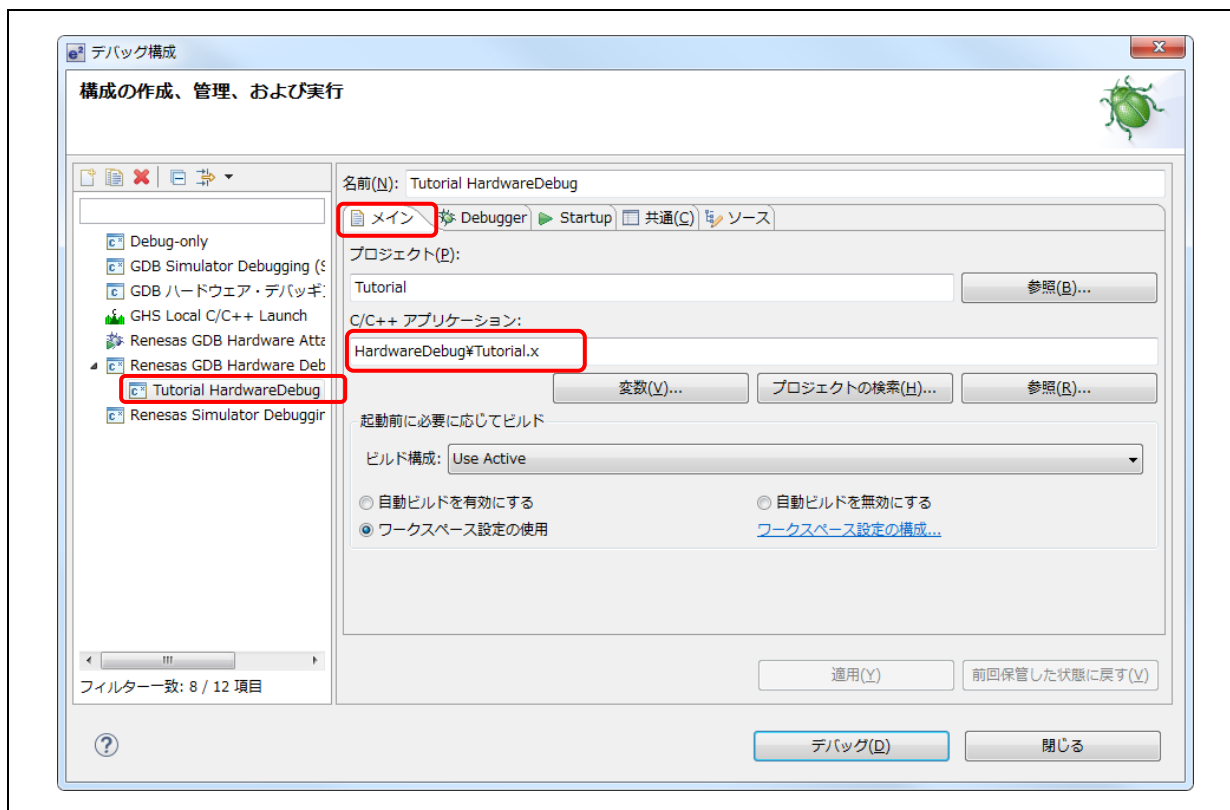


図 5-3 ロードモジュールの選択

- (2) [デバッグ構成] ウィンドウで、“Renesas GDB Hardware Debugging” デバッグ構成の表示を展開し、既存のデバッグ構成をクリックしてください(例: “Tutorial HardwareDebug”)。
- (3) [メイン] タブを選択し、プロジェクトビルドフォルダの中にあるロードモジュール “Tutorial.x” を追加します。

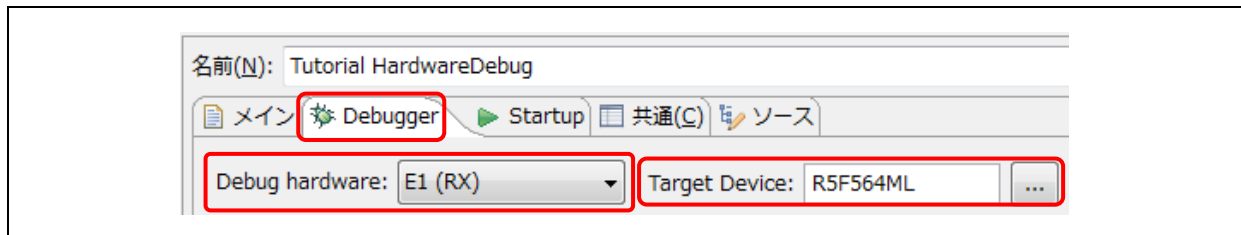


図 5-4 ターゲットデバイスの選択

- (4) [Debugger] タブに切り替え、Debug hardware を “E1”、Target Device を “R5F564ML” に設定します。
- Debug Hardware: “E1”
 - Target Device: “R5F564ML”

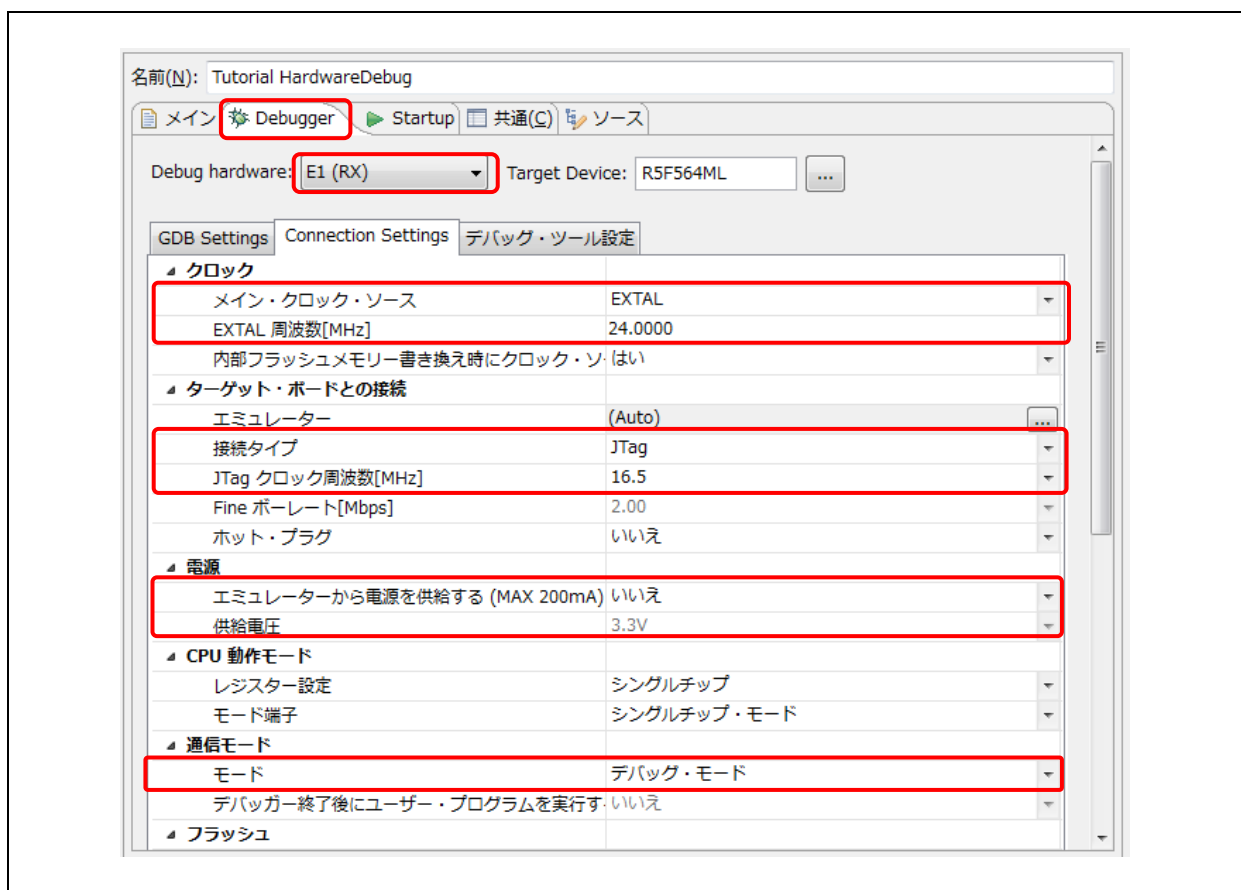


図 5-5 接続設定の変更

(5) [Debugger] タブの中の [Connection Settings] サブタブを選択し、E1 エミュレータと RSK RX64M ボードの設定に基づいて、以下のように設定します。

- クロック
 - メイン・クロック・ソース = “EXTAL”
 - EXTAL 周波数[MHz] = “24.0000”
- ターゲット・ボードとの接続
 - 接続タイプ = “JTag”
 - JTag クロック周波数[MHz] = “16.5”
- 電源
 - エミュレータから電源を供給する(MAX 200mA) = “いいえ”
- 通信モード
 - モード = “デバッグ・モード”

“エミュレータから電源を供給する(MAX 200mA)” を “はい” に設定すると、エミュレータは外部電源を使用することなくターゲットボードに電源を供給します(最大電流 200 mA)。

注意: 図 5-5 のデバッグ構成は例として示されています。誤った設定は誤作動やハードウェアの故障の原因となりますので、接続の前にボードとエミュレータの設定を慎重に確認してください。

さらに e² studio には既存のプロジェクトデバッグ構成を新規プロジェクトに複製する機能があります。この機能は、同一のデバイスとデバッグ設定を用いるプロジェクトに使用できます。

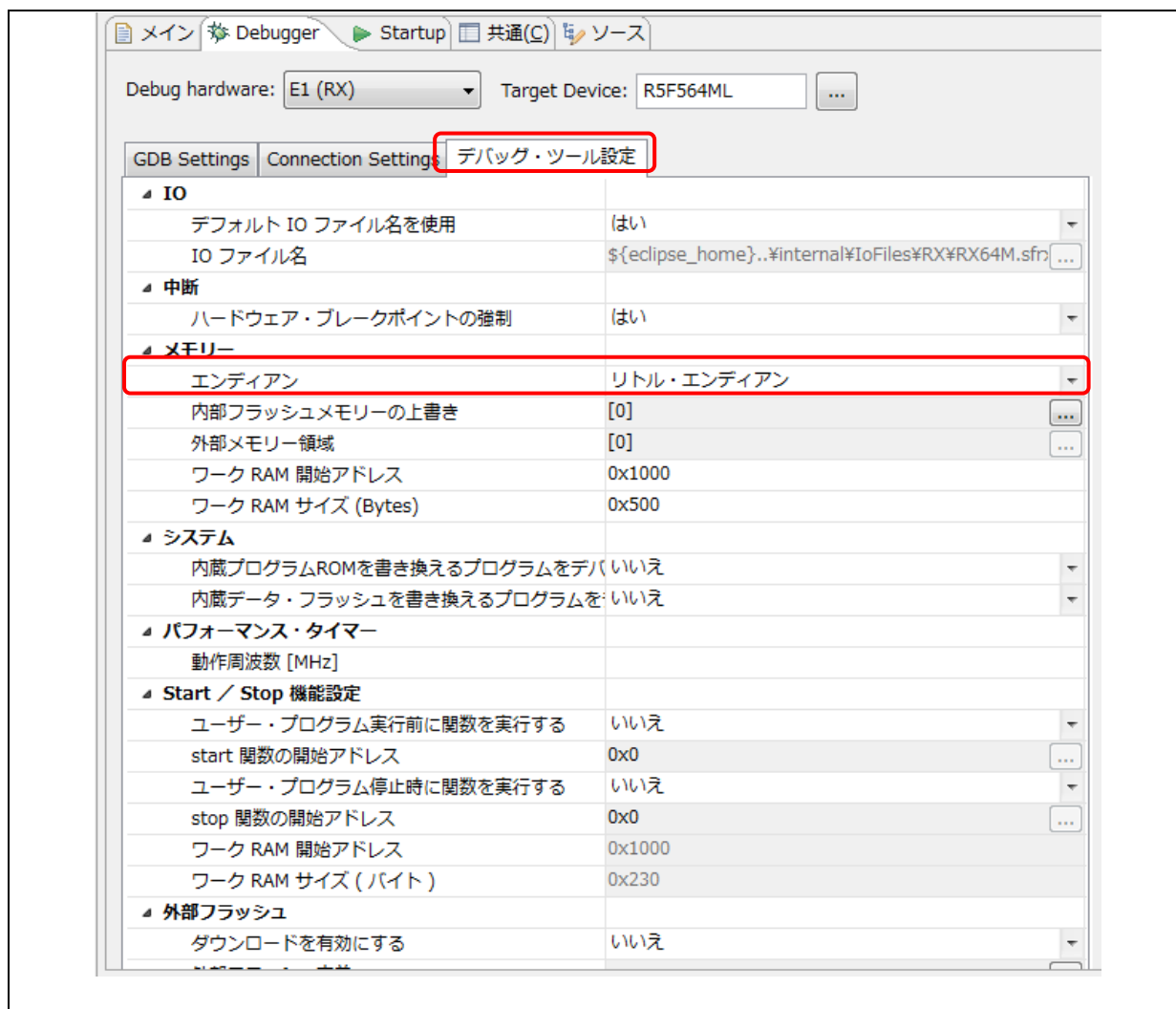


図 5-6 デバッグツール設定の変更

(6) サブタブ [デバッグ・ツール設定] に切り替え、RSK RX64M ボードに基づいて以下を確認してください。

- メモリ
エンディアン = “リトル・エンディアン”

(7) [適用] ボタンをクリックし、設定を確認します。

(8) [デバッグ] をクリックし、デバッグ起動構成を実行して E1 または E2 Lite と RSK RX64M ボードに接続します。

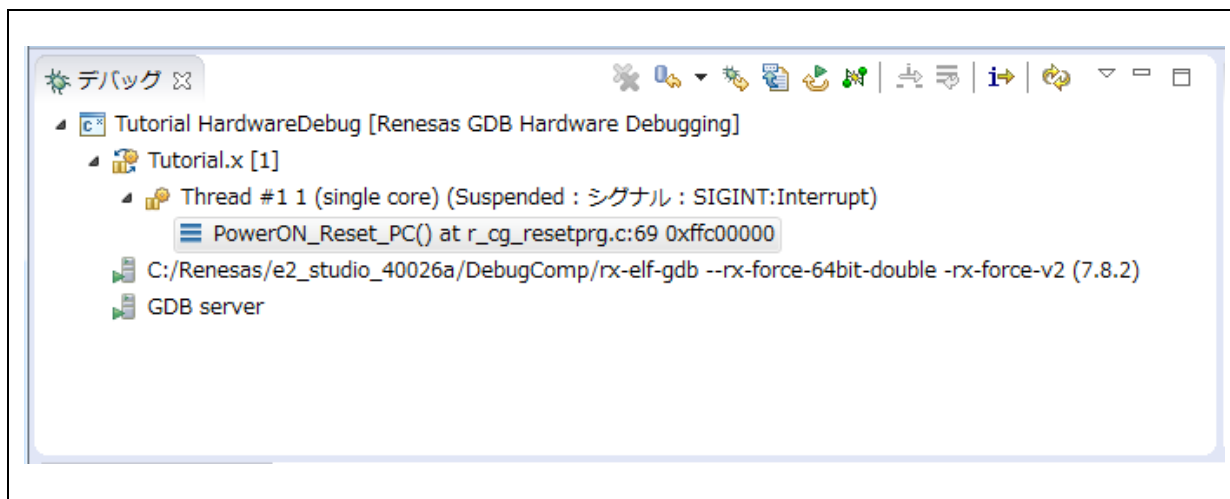


図 5-7 [デバッグ] ビューのユーザーターゲット接続

正しく接続できた場合は、図に示すような[デバッグ] ビュー画面が表示されます。プログラムのエントリポイントは、“r_cg_resetprg.c” の “PowerON_Reset_PC()” に設定されています。

5.2. 新規デバッグ構成の作成

新規デバッグ構成を作成する簡単な方法は、既存の構成を複製する方法です。以下の手順で行います。

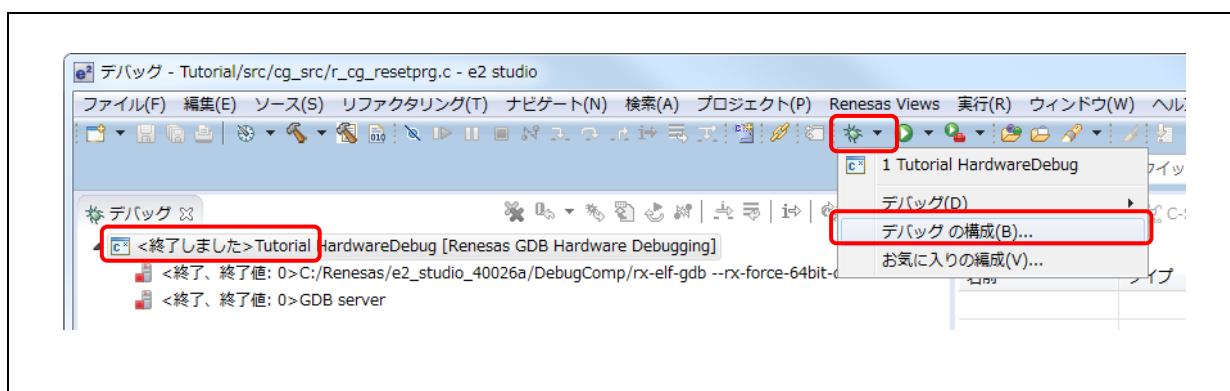


図 5-8 [デバッグ構成]ウィンドウを開く

(1) [プロジェクト・エクスプローラー] の “Tutorial” プロジェクトをクリックします。

[実行] → [デバッグの構成...] あるいは アイコン (下向き矢印) → [デバッグの構成] の順にクリックし、[デバッグ構成] ウィンドウを開きます。

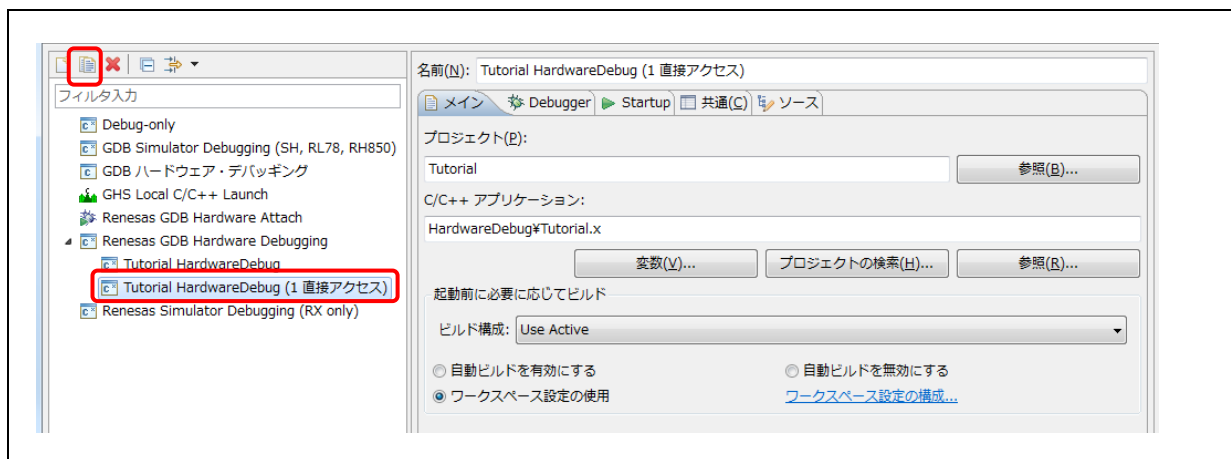



図 5-9 選択したデバッグ起動構成の複製

(2) [デバッグ構成] ウィンドウで、デバッグ構成 (例: “Tutorial HardwareDebug”) を選択し  アイコンをクリックします (現在選択しているデバッグ構成をコピーします)。新規デバッグ起動構成 (例: “Tutorial HardwareDebug (1)”) が作成されます。

(3) デバッグ構成は 5.1 節と同じ手順で変更できます。

RL78 デバッグ時の注意事項:

- ホット・プラグイン接続は、RL78/F1A, F13, F14, および F15 に対してのみサポートします。

5.3. 基本的なデバッグ機能

本節では、e² studio がサポートする典型的なデバッグビューを説明します。

- 標準的な GDB デバッグ (Eclipse フレームワークによってサポートされている): ブレークポイント、式、レジスタ、メモリ、逆アセンブル、変数
- 標準的な GDB デバッグの拡張: イベントポイント、IO レジスタ、トレース

[デバッグ・ツールバー] を開くには、プルダウンメニューボタンをクリックして [デバッグ・ツールバーを表示] を選択します。以下に示すのは、[デバッグ] ビューの便利なツールバーです。

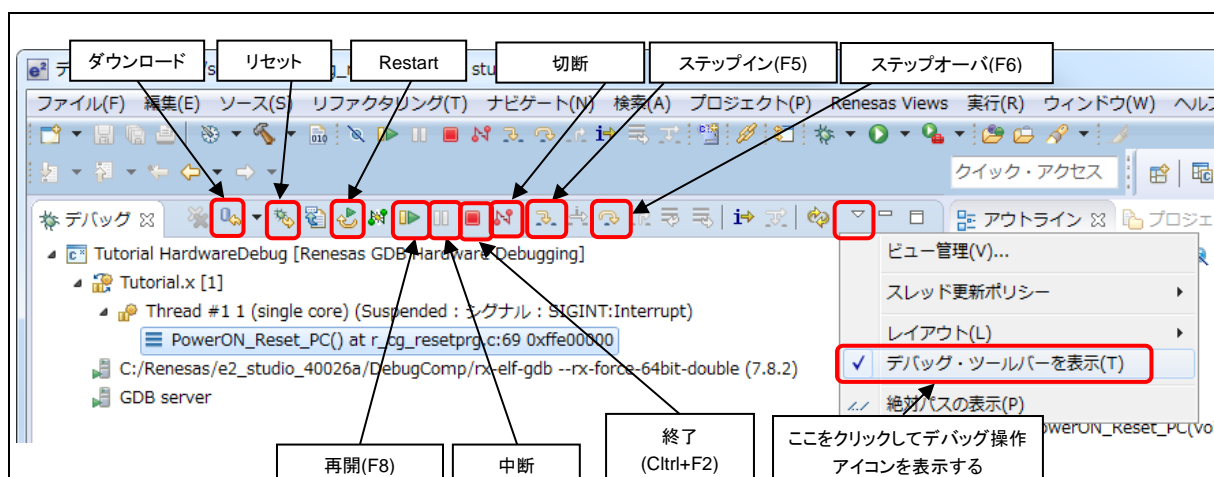


図 5-10 デバッグビューの便利なツールバー

プログラムを実行するには ボタンをクリックするか [F8] キーを入力します。

プログラムは、ブレークポイントで、あるいは ボタンをクリックすることで一時停止します。一時停止中は以下の操作が可能です。

- ボタンまたは [F5] キーは、現在実行中のプログラム行にある次の関数呼び出しへステップイン実行します (関数内に入って1ステップ実行)。
- ボタンまたは [F6] キーは、現在実行中のプログラム行にある次の関数呼び出しをステップオーバ実行します (1行実行するが関数内には入らない)。
- ボタンで、実行を再開します。

デバッグセッションの停止は、選択したデバッグセッション／プロセスを ボタンで終了するか、選択したプロセスとデバッグを ボタンで切断します。

他に以下のような操作が可能です。

- ボタンは、エントリポイントからプログラムを再実行します (→ の順にクリックするのと同じ)。
- ボタンは、プログラムをパワーオンリセットのエントリポイントにリセットします。
- ボタンは、ターゲットシステムにバイナリファイルを再びダウンロードします。

5.3.1. ブレークポイントの使い方

デバッグ接続中は、ソースコードまたは特定のアドレスにブレークポイントを設定することができます。エディタや逆アセンブルビューには、下図に示すようにブレークポイントマーカを置くことができます。プログラムカウンタがブレークポイントの設定された行に対応する実行コードのアドレスに達すると、その位置の命令が実行される前に中断(break)します。e² studio ではソフトウェアブレークポイントとハードウェアブレークポイントを区別して設定できます。エディタや逆アセンブルビューの左端でダブルクリックすると、デフォルトのブレークポイント型でブレークポイントが配置されます。ハードウェアブレークポイントのリソースが残っていない場合はデバッグコンソールにブレークポイント型を「ソフトウェア」に変更した旨を警告するメッセージが表示されます。

ブレークポイントの設定方法は 2 通りあります。

方法 1:

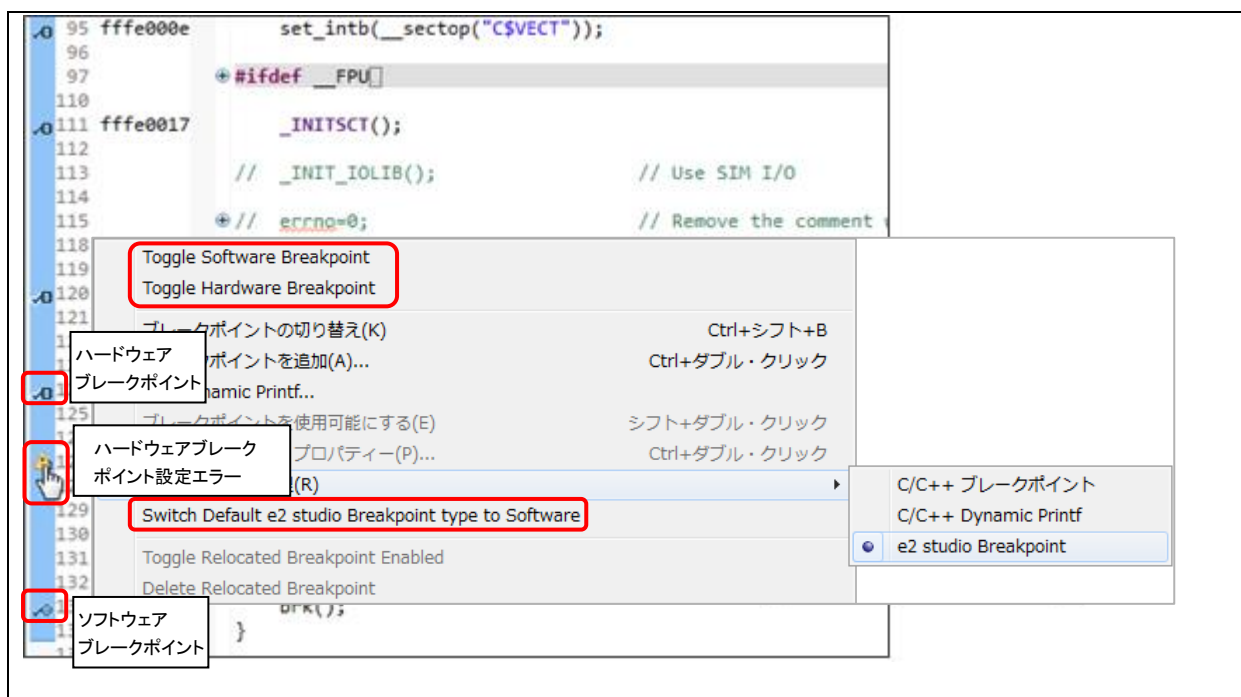




図 5-11 [ブレークポイント] ビュー - ブレークポイント設定方法 1

ブレークポイント設定方法 1 では、


- (1) ソースコードの左余白を右クリックして [Toggle Software Breakpoint] または [Toggle Hardware Breakpoint] を選択してハードウェアブレークポイント  またはソフトウェアブレークポイント  を設定します。

方法 2:




図 5-12 [ブレークポイント] ビュー - ブレークポイント設定方法 2

ブレークポイント設定方法 2 では、

- (1) 右クリックしてコンテキスト・メニューを開き、[ブレークポイント型] → [e2 studio Breakpoint](デフォルトのハードウェアブレークポイント)または [ブレークポイント型] → [C/C++ ブレークポイント](ソフトウェアブレークポイント)を選択します。
- (2) ソースコードの左余白をダブルクリックして、ハードウェアブレークポイントまたはソフトウェアブレークポイントを設定します。
- (3) [ビューの表示] → [ブレークポイント] またはアイコン  をクリックし(あるいはショートカットキー [ALT]+[Shift]+[Q], [B] を使い)、[ブレークポイント] ビューを開いて、設定したハードウェアブレークポイントまたはソフトウェアブレークポイントを表示します。

ブレークポイントを無効にする際は、選択したブレークポイントのみを無効にするか、あるいはすべてのブレークポイントを無効にするか選択できます。

- (1) [ブレークポイント] ビューで、ブレークポイントを有効または無効に切り替えることができます。無効になったソフトウェアブレークポイントは白い丸が表示されます。無効になったハードウェアブレークポイントは白い丸が表示されます。
- (2) すべてのブレークポイントを無効にするには、[ブレークポイント] ビューにあるアイコン  をクリックします。斜線を重ねた青い丸が [ブレークポイント] ビューとエディタ画面の両方に表示されます。

5.3.2. 式ビュー

式ビューでは、デバッグ中のグローバル変数、静的変数、ローカル変数の値を監視できます。ブレークした時に表示が更新され、前回表示した時と異なる値は背景が黄色になります。また「リアルタイム・リフレッシュ」を有効にした変数(“R”マークが表示されたもの)はデバッガが実行されている間、設定された周期で表示が更新されます。

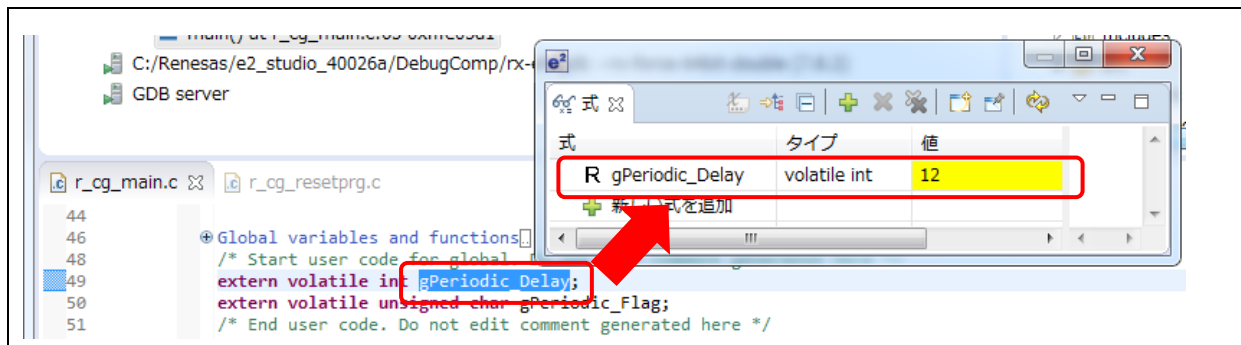



図 5-13 [式] ビュー

グローバル変数を見るには、

- (1) [ビューの表示] → [式] あるいはアイコン  をクリックし、[式] ビューを開きます。
- (2) “main.c” の 49 行目で、グローバル変数 (例えば “gPeriodic_Delay”) を [式] ビューヘドラッグ&ドロップします。(または、グローバル変数を右クリックして “監視式を追加(A)...” メニューアイテムを選択し、[式] ビューに追加します。)
- (3) [式] ビューで、“リアルタイム・リフレッシュ” メニューアイテムを右クリックして選択します。これで、プログラムの動作中、表示した値はリアルタイムに更新されます。“R” は、このグローバル変数がリアルタイムで更新されることを意味します。
- (4) “リアルタイム・リフレッシュ” を無効にするには、“リアルタイム・リフレッシュを無効にする” メニュー項目を右クリックして選択します。

5.3.3. レジスタービュー

レジスタービューは汎用レジスタについての情報を表示します。プログラムを停止すると、変化のあった値を強調表示します。

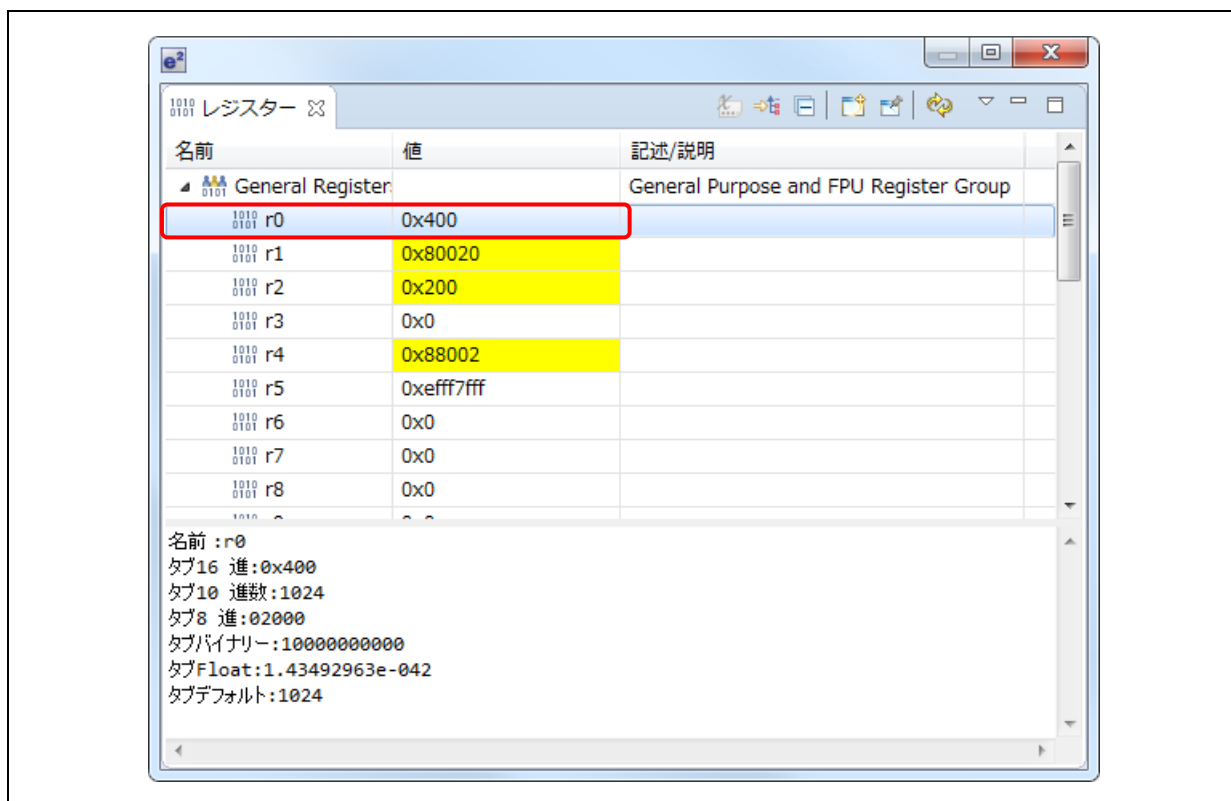



図 5-14 [レジスター] ビュー

汎用レジスタ “r0” を表示するには、

- (1) [ビューの表示] → [レジスター] あるいはアイコン  をクリックし、[レジスター] ビューを開きます。
- (2) “r0” をクリックすると、他の基数フォーマットで値を表示します。

プログラムを停止すると、変化のあった値は [レジスター] ビューの中で強調表示(黄色など)されます。

5.3.4. メモリービュー

メモリービューでは、ユーザは“メモリーモニター”でメモリーを表示し編集することができます。各モニターは“ベースアドレス”と呼ばれる格納位置によって特定される記憶場所を表します。各メモリーモニターの中のメモリーデータは異なる“メモリーレンダリング”で表示することができます。メモリーレンダリングはあらかじめ設定したデータフォーマット（例えば、16進数、符号付き整数、符号なし整数、ASCII、イメージなど）です。

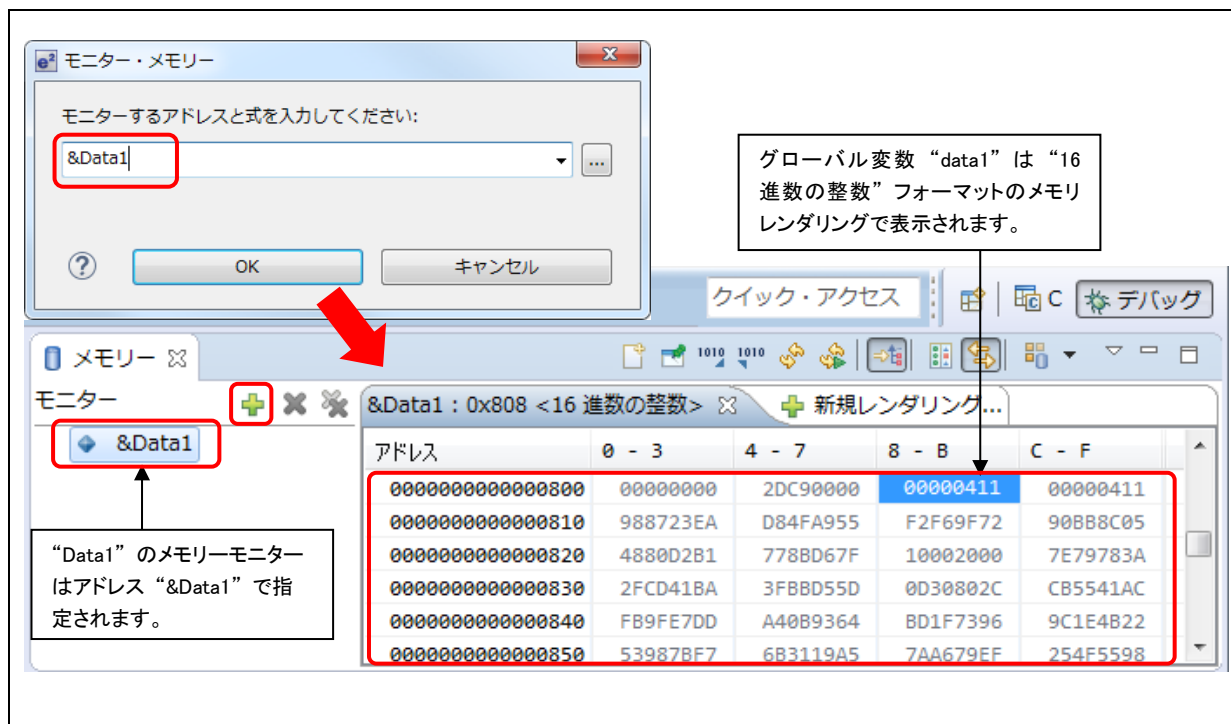




図 5-15 [メモリー] ビュー (1/2)

変数（例えば“Data1”）のメモリーを表示するには、

- (1) [ビューの表示] → [メモリー] あるいはアイコン  をクリックし、[メモリー] ビューを開きます。
- (2) アイコン  をクリックし、[モニター・メモリー] ダイアログボックスを開きます。変数“Data1”のアドレスを入力します。

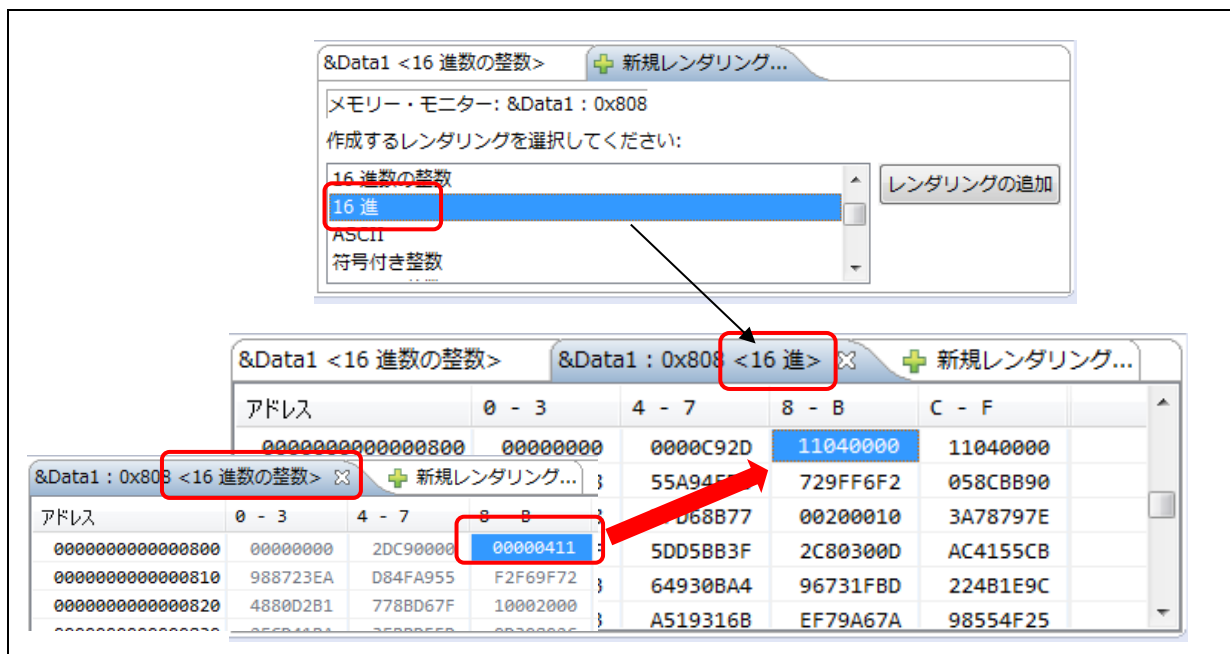


図 5-16 [メモリー] ビュー (2/2)

変数“Data1”用の新しいレンダリングフォーマット(例えば 16 進)を追加するには、

- (1) **+ 新規レンダリング...** タブをクリックし、“16 進” を選択してレンダリングを追加します。

これで、“&Data1<16 進数の整数>” タブの横に新しい “&Data1 <16 進>” タブが作成されます。

5.3.5. 逆アセンブルビュー

逆アセンブルビューは、ロードしたプログラムのソースコードとアセンブラ命令を混在して表示します。現在実行中の行は画面上で矢印のマーカで強調表示されます。逆アセンブルビューでは、アセンブラ命令へのブレークポイントの設定、ブレークポイントの有効化/無効化、逆アセンブル命令のステップ実行、プログラムの特定の命令へのジャンプが可能です。

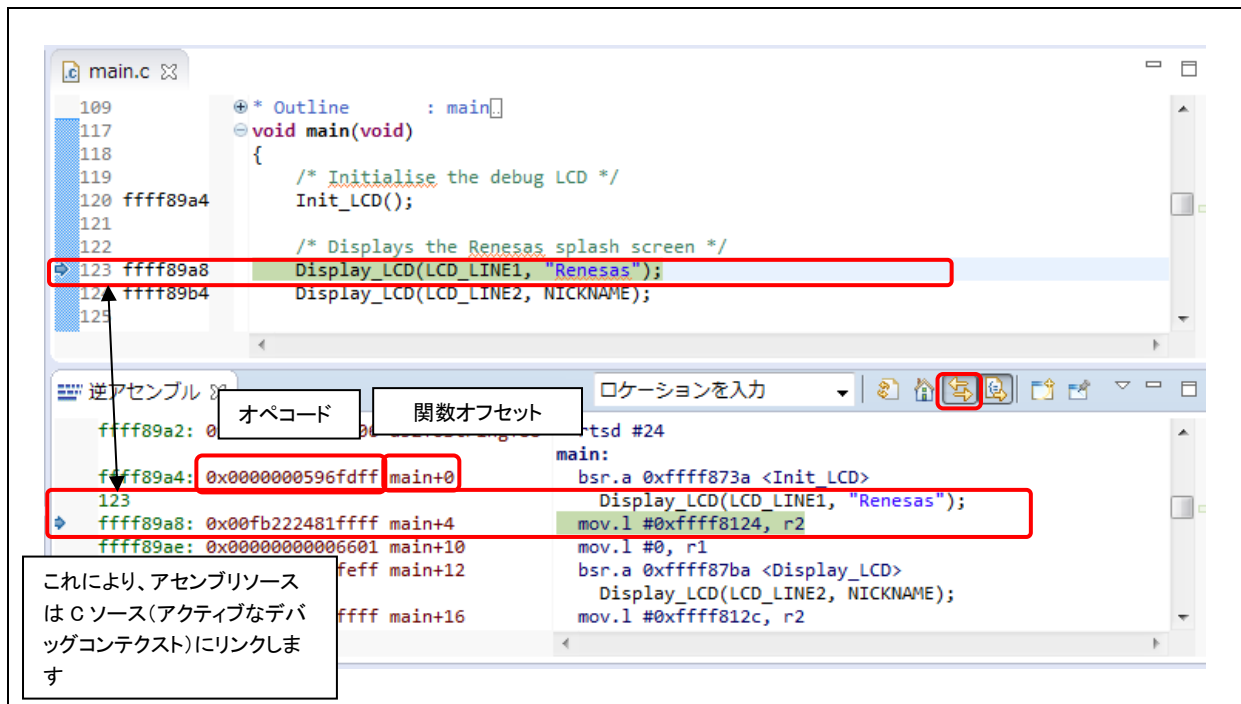




図 5-17 【逆アセンブル】ビュー

混合モードで C コードとアセンブリコードの両方を表示するには、

- (1) [ビューの表示] → [逆アセンブル] あるいはアイコン  をクリックし、[逆アセンブル] ビューを開きます。
- (2) アイコン  をクリックして、アセンブリソースと C ソース(アクティブなデバッグコンテキスト)のリンクを有効にします。
- (3) [逆アセンブル] ビューのアドレス列で右クリックし、“オペコードを表示する” と “関数オフセットを表示する” を選択します。
- (4) コンテキスト・メニューを使用すればエディタ内で Source Addresses を有効にすることができます。

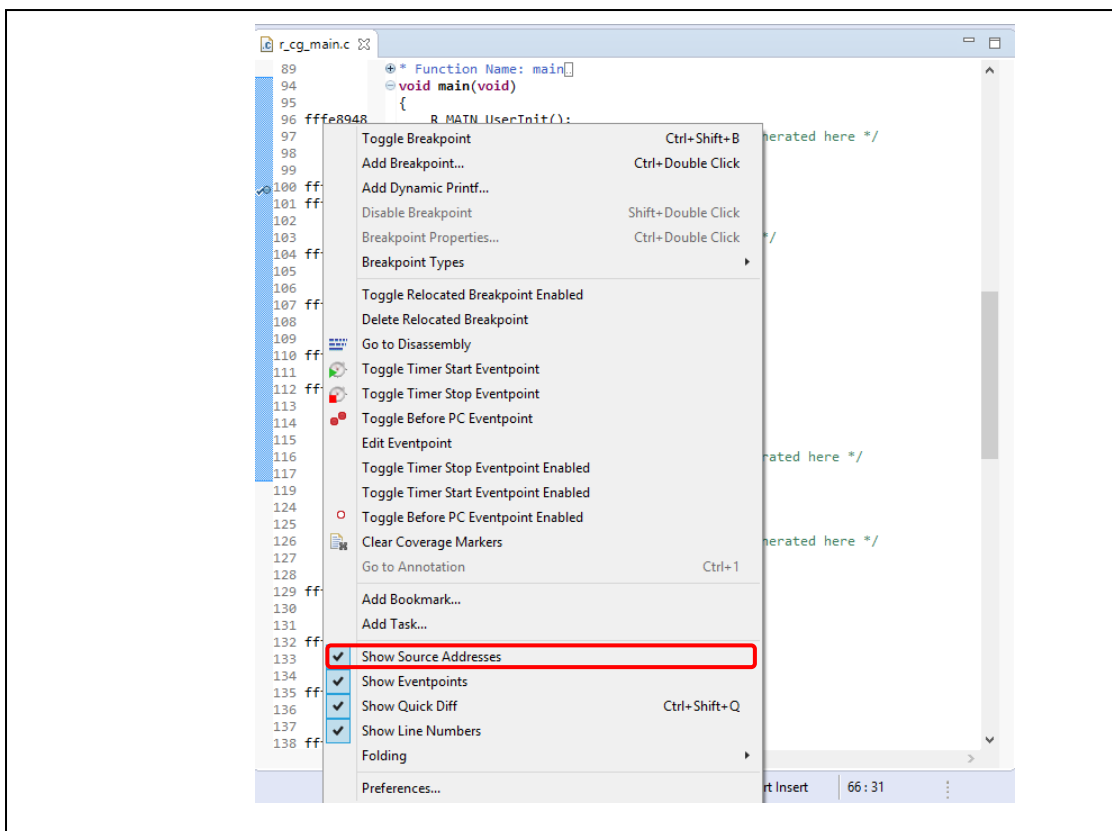


図 5-18 Source Addresses メニュー

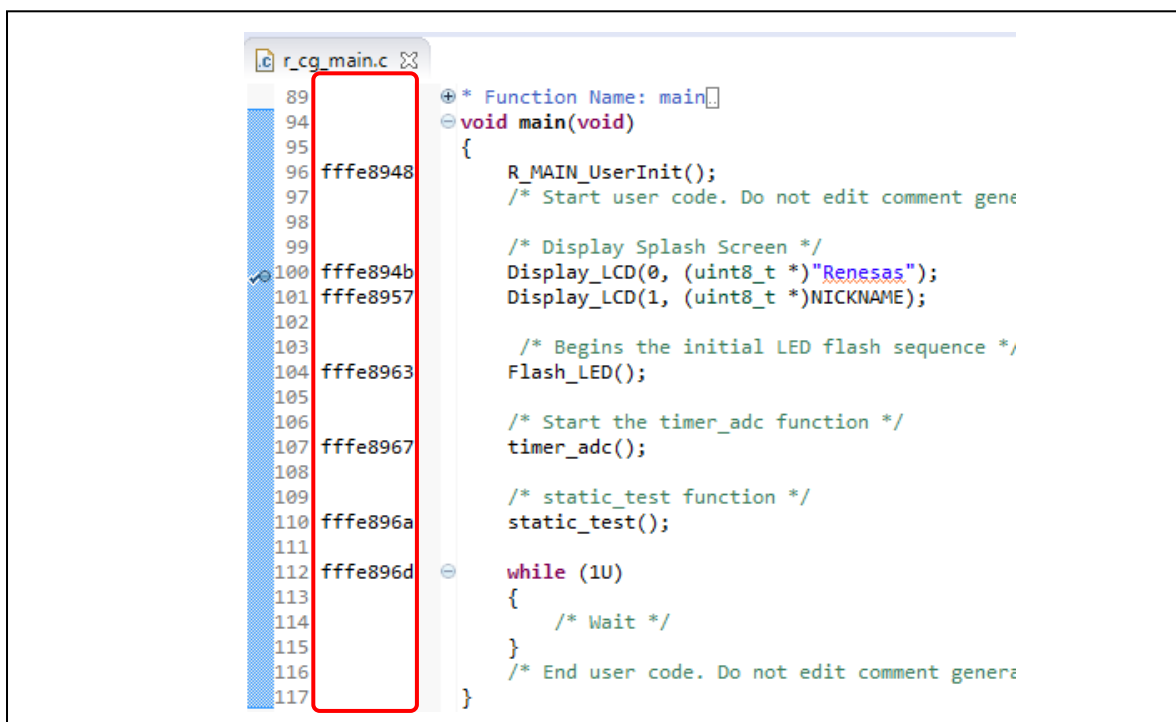


図 5-19 エディタに表示される Source Addresses

5.3.6. 変数ビュー

変数ビューは、現在実行中のスコープ内で表示可能な全てのローカル変数を表示します。

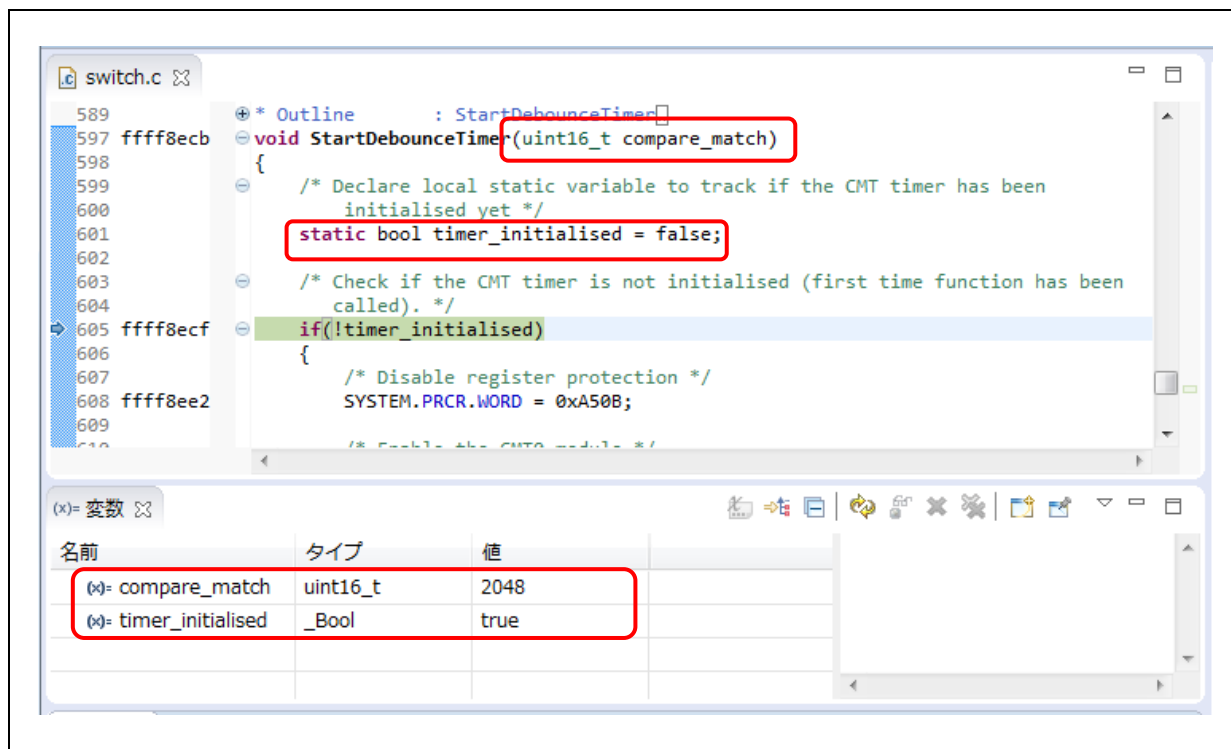



図 5-20 [変数] ビュー

ローカル変数(例えば、関数 “StartDebounceTimer()” の “compare_match” や “timer_initialised”)を見るには、

- (1) [ビューの表示] → [変数] またはアイコン  をクリックし、[変数] ビューを開きます。
- (2) ステップ実行で関数 “StartDebounceTimer()” の中に入ると、[変数]ビュー内にローカル変数の値が表示されます(上記の例では 2048 および true)。

5.3.7. イベントポイントビュー

イベントは、プログラム実行中にブレークあるいはトレース機能を実行するために設定された条件の組み合わせです。ユーザはイベントポイントビューで、異なる種類の定義されたイベント、たとえば、トレース開始、トレース終了、トレース・レコード、イベント・ブレーク、実行前 PC ブレーク、タイマー開始、およびタイマー終了、などを設定、表示することができます。

設定できるイベント数や設定条件は MCU によって異なります。以下に挙げる 2 種類のイベントがあります。

- 実行アドレス: エミュレータは CPU が特定のアドレスの命令を実行しようとしたことを検出します。これが“実行前 PC”ブレーク(例えば、イベントにより、条件は指定アドレスで命令の実行直前に成立する)、あるいは他のイベント(例えば、イベントにより、条件は指定アドレスで命令の実行直後に成立する)となります。
- データ・アクセス: エミュレータは指定された条件での指定アドレスあるいは指定アドレス範囲へのアクセスを検出します。これにより、アドレスとデータを組み合わせた条件を設定することができます。

イベントの組み合わせは(OR, AND(およびその組み合わせ), およびシーケンシャル)は 2 つ以上のイベントに使用できます。

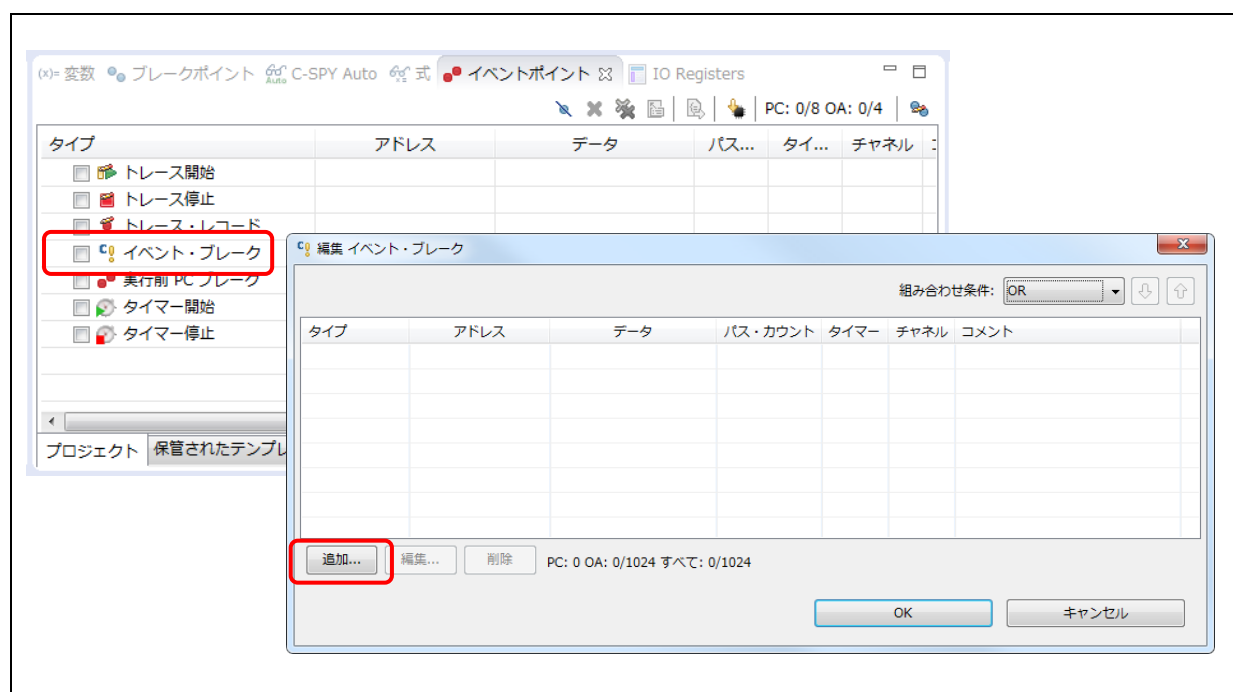



図 5-21 [イベントポイント] ビュー (1/2)

アドレスまたはデータが一致する条件(例えば、gFlashCount = “0xB0” のとき)で、グローバル変数にイベント・ブレークを設定するには、

- (1) [ビューの表示] → [イベントポイント] あるいはアイコン  をクリックし、[イベントポイント] ビューを開きます。
- (2) “イベント・ブレーク” オプションをダブルクリックし、[編集イベント・ブレーク] ダイアログボックスを開きます。

(3) [追加...] ボタンをクリックし、以下の操作を行います。

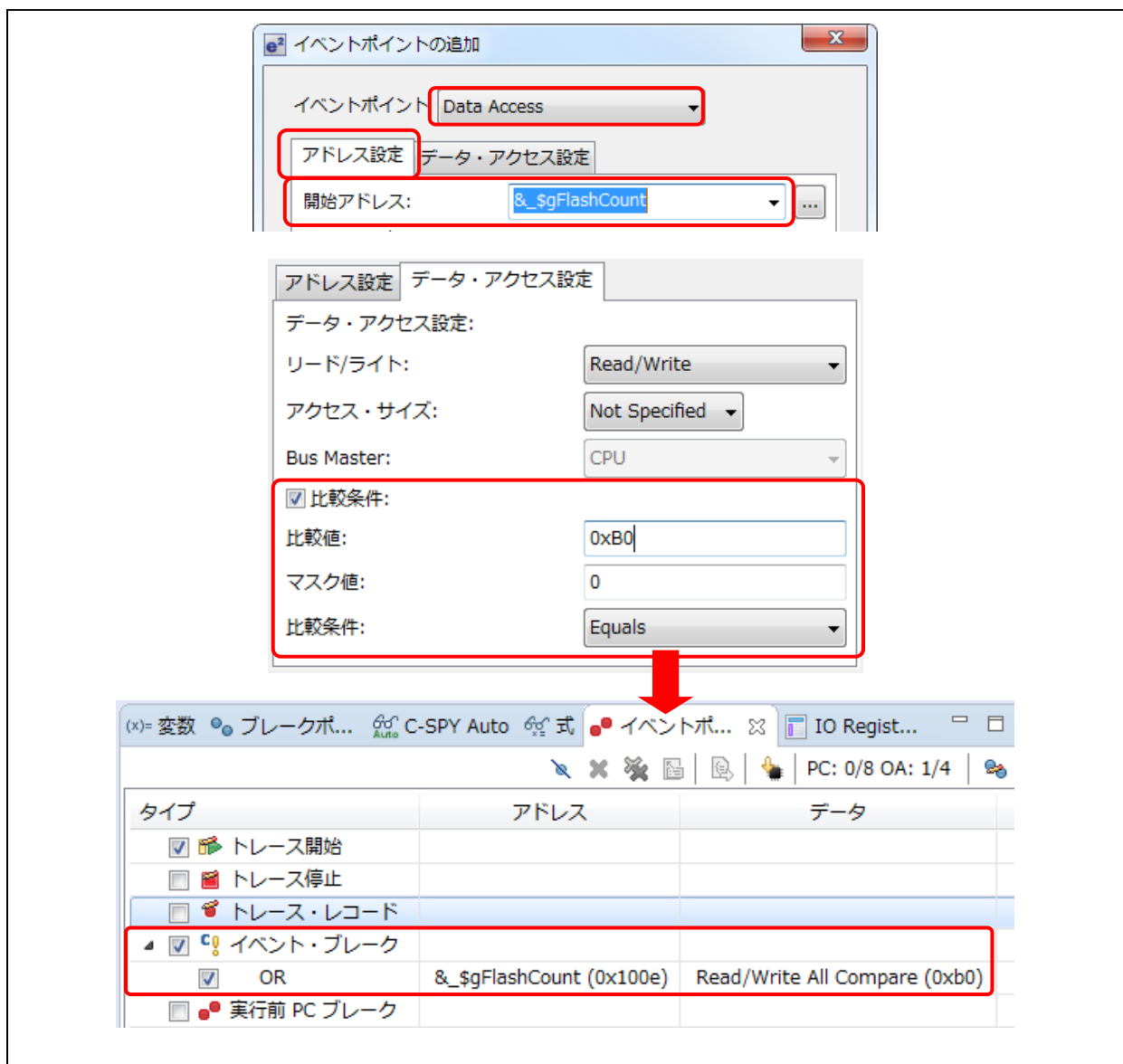



図 5-22 [イベントポイント] ビュー (2/2)

- (4) イベントポイントの種類に “Data Access” を選択します。
- (5) [アドレス設定] タブに進み、アイコン  をクリックしてシンボル “&_gFlashCount” を検索します。
(このグローバル変数のアドレスは “&_gFlashCount” です。)
- (6) 次に、[データ・アクセス設定] タブに切り替え、[比較条件] チェックボックスを有効にして比較値を “0xB0” に設定します。[OK] をクリックしてください。
- (7) [イベントポイント] ビューで、イベント・ブレイクが “gFlashCount = 0xB0” に設定されて有効になっていることを確認してください。プログラムを最初から実行するためにリセットします。

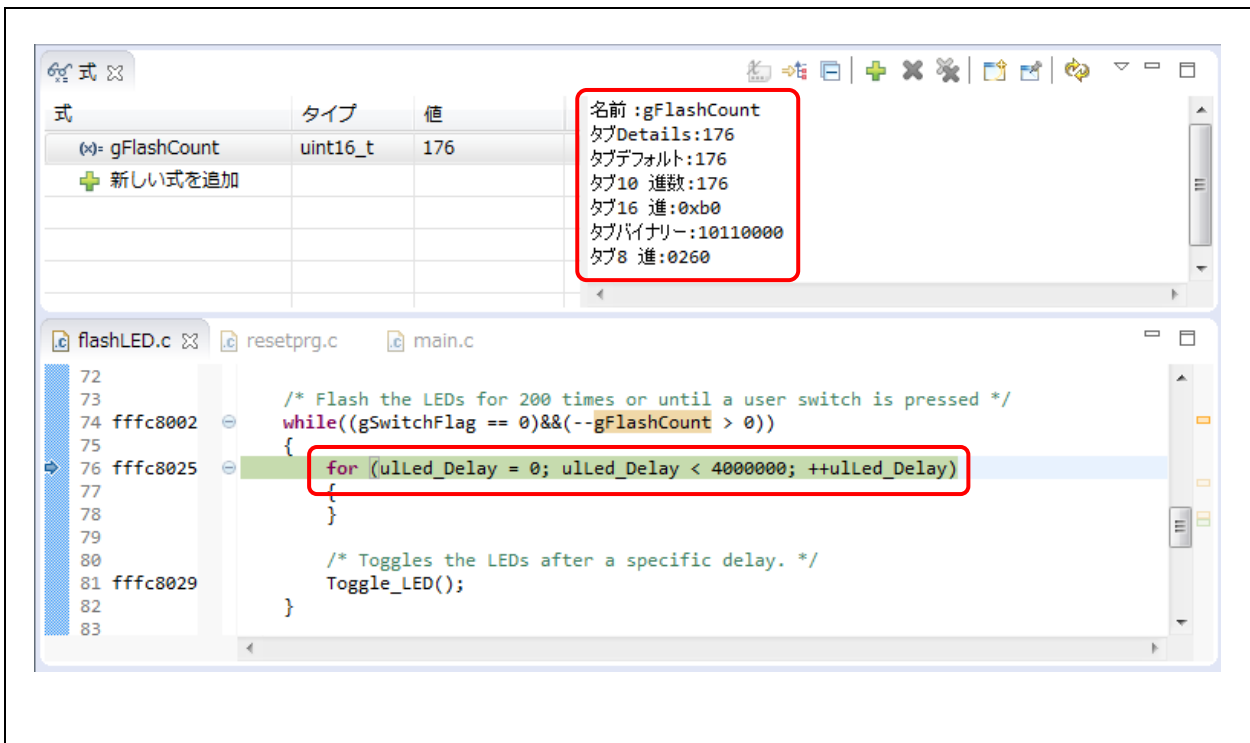


図 5-23 イベント・ブレークの実行

図 5-23 は、gFlashCount が 176(0xB0)になったときにプログラムの 76 行目で実行が停止する様子を示します。

5.3.8. IOレジスタビュー

IOレジスタは Special Function Register (SFR) のことです。IOレジスタビューは、ターゲット専用の IO ファイルで定義された全レジスタセットの名前、アドレス、値 (16 進数および 2 進数) を表示します。ユーザは、[選択されたレジスタ] に必要な IO レジスタを選択して追加することによって、IO レジスタビューをカスタマイズすることができます。

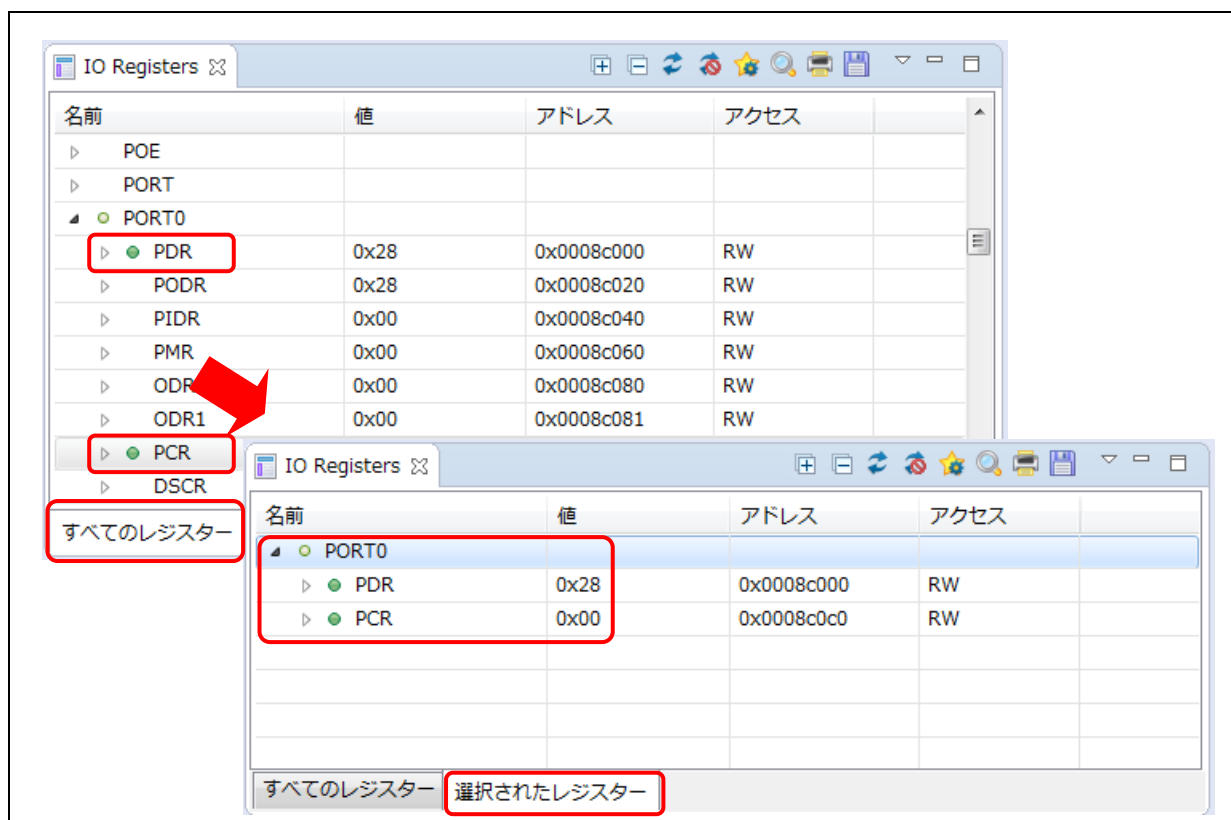


図 5-24 [IO Registers] ビュー

IOレジスタを選択して (例えばポート0のPDRやPCR) を表示するには、

- (1) [ウィンドウ] → [ビューの表示] → [その他...] の順でクリックします。[ビューの表示] ダイアログボックスで、[デバッグ] の下にある [IO Registers] をクリックするか、あるいはアイコン  をクリックして [IO Registers] ビューを開きます。
- (2) [すべてのレジスタ] タブ上で、[IO Registers] ビューで [PORT0] を探し、PORT0のIOレジスタリストを拡張します。
- (3) “PDR” および “PCR” を [選択されたレジスタ] タブヘドラッグ & ドロップします。IOレジスタ横の緑の丸いマーク ● は、選択されたレジスタであることを意味しています。
- (4) [選択されたレジスタ] タブに切り替え、“PORT0” IOレジスタの “PDR” および “PCR” を表示します。

拡張した IO レジスタリストを [すべてのレジスタ] にロードするには時間がかかる場合があるため、必要な IO レジスタを選択して [選択されたレジスタ] に表示することを推奨します。

5.3.9. トレースビュー

トレースとは、ユーザプログラム実行中、1 サイクルごとのバス情報をトレースメモリから取得することを意味します。取得されたトレース情報はトレースビューに表示されます。それによりユーザはプログラムの実行を追跡し、問題が発生した箇所を探することができます。

トレースバッファは有限(1~32M バイトのサイズ)なため、バッファが一杯になると、最も古いトレースデータを新しいデータで上書きします。

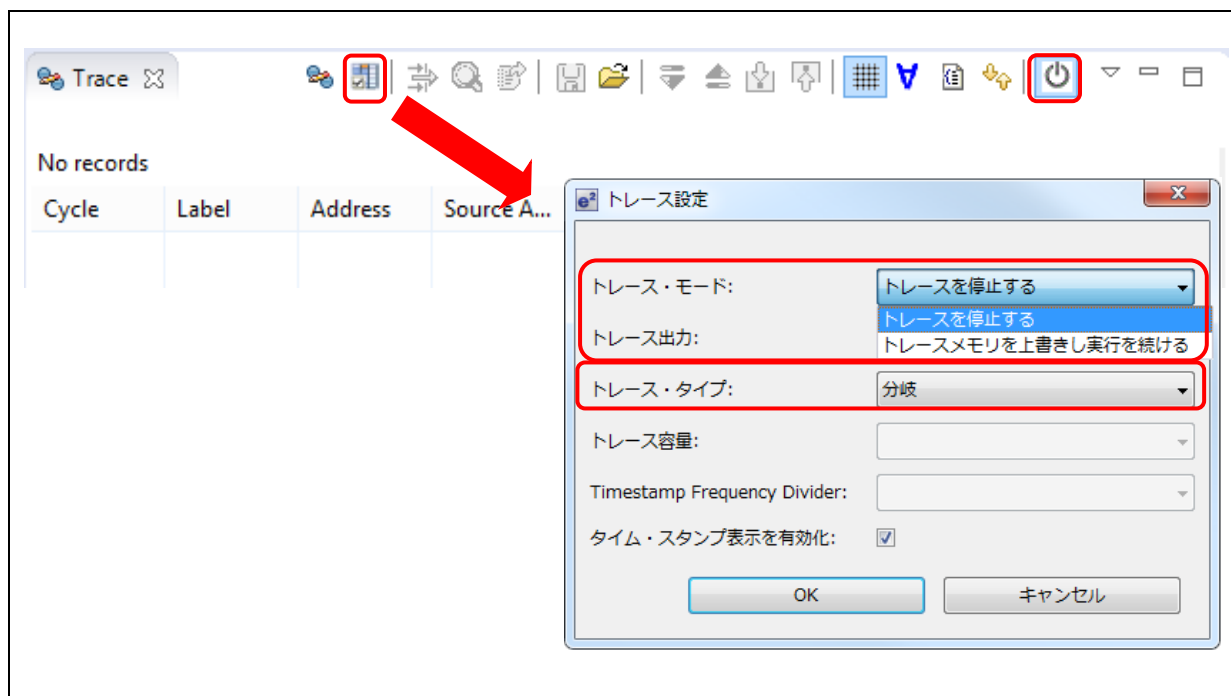





図 5-25 [Trace] ビュー (1/2)

2つの関数の間 (例えば、“main()” と “sort()” 間のトレース) の point-to-point トレースを設定するには、

- (1) [ウィンドウ] → [ビューの表示] → [その他...] の順にクリックします。[ビューの表示] ダイアログボックスで、[デバッグ] の下にある [Trace] あるいはアイコン  をクリックし、[Trace] ビューを開きます。
- (2)  アイコンを選択して[Trace] ビューをオンにします。
- (3) アイコン  (取得) をクリックし、以下を設定します。
 - トレース・モード: “トレースを停止する”
 - トレース・タイプ: “分岐”
- (4) [OK] をクリックします。

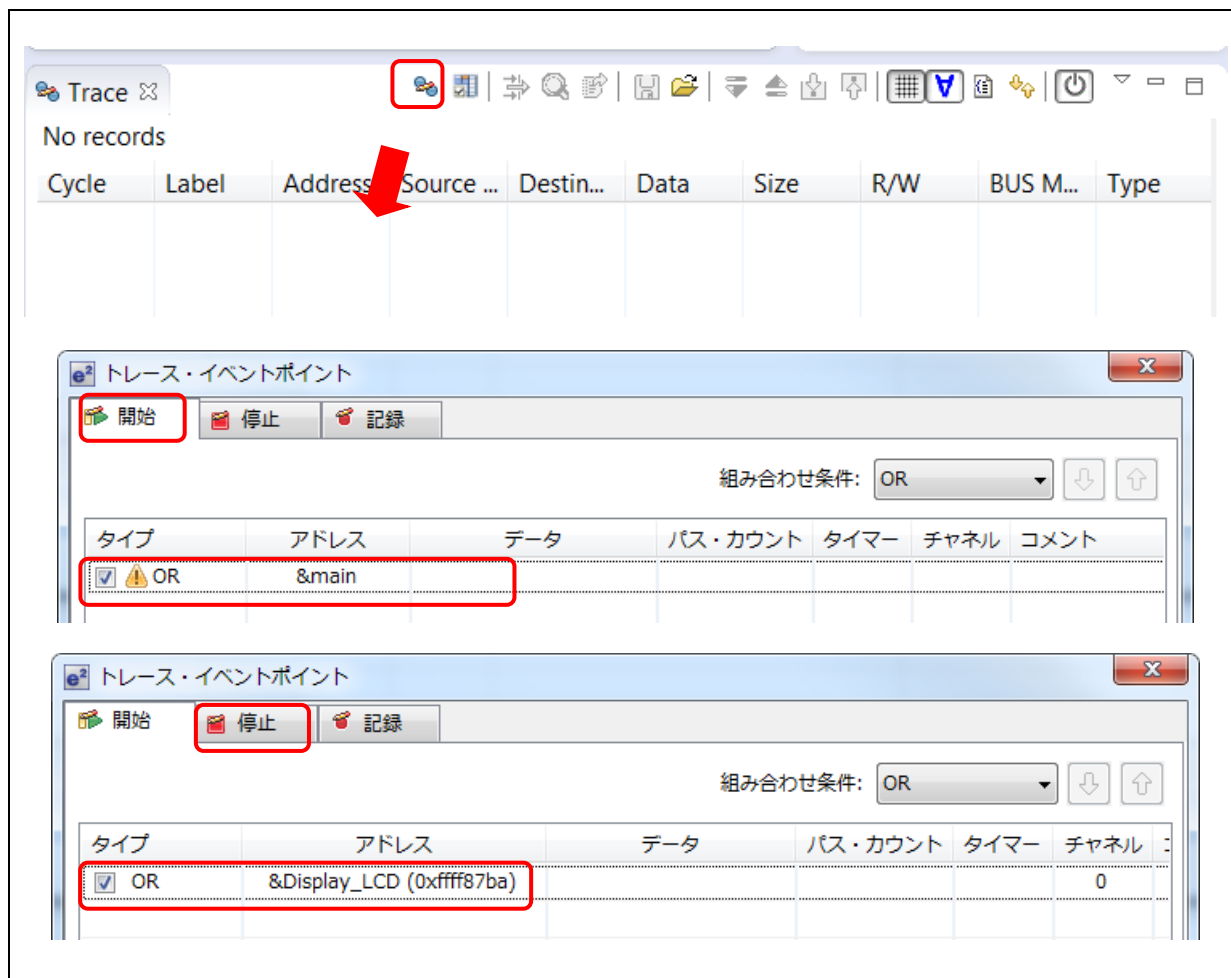



図 5-26 [Trace] ビュー (2/2)

- (1)  (トレース・イベントポイントを編集) をクリックし、[トレース・イベントポイント] ダイアログボックスを開きます。
- (2) [開始] タブの下の “main()” 関数に、最初のイベントポイント(実行アドレス “&main”、あるいは 0xFFFFC8228)を追加します。
- (3) 次に [停止] タブに切り替え、“Display_LCD()” 関数に、2 番目のイベントポイント(実行アドレス “&Display_LCD”, あるいは 0xFFFFC816F)を追加します。
- (4) リセット後にプログラムを実行します。

The image displays the e² studio interface. At the top, a code editor shows the 'main' function in 'main.c' with the following code:

```

127
128 void main(void)
129 {
130
131     /* Initialise the debug LCD */
132     fffc8268 Init_LCD();
133
134     /* Displays the Renesas splash screen */
135     fffc826c Display_LCD(LCD_LINE1, "Renesas");
136     fffc8278 Display_LCD(LCD_LINE2, NICKNAME);

```

Below the code editor is a 'Trace' window showing a listing of records. The relevant records are:

Cycle	Label	Address	Source Address	Destinatio...	Data	Size	R/W	BUS ...	Type	BCN	Branch ...	Channel
0		1718	-----	-----	FFFF89B4	LONG	W	CPU	MEMORY	-----	-----	-
-1	_Display_LCD	FFFF87BA	-----	FFFF87BA	-----	----	-	CPU	DESTINATION	-----	-----	-
-2		1718	-----	-----	FFFF89A8	LONG	R	CPU	MEMORY	-----	-----	-
-3		FFFF89A8	-----	FFFF89A8	-----	----	-	CPU	DESTINATION	-----	-----	-
	main.c	123	Display_LCD(LCD_LINE1, "Renesas");									
-4		1714	-----	-----	FFFF87B9	LONG	R	CPU	MEMORY	-----	-----	-
-5		FFFF87B9	-----	FFFF87B9	-----	----	-	CPU	DESTINATION	-----	-----	-
	lcd.c	100	Delay_us(10000u);									
-6		170C	-----	-----	FFFFFFF	LONG	W	CPU	MEMORY	-----	-----	-
-7			-----	-----	-----	----	-	CPU	BCND	000000...	-----	-

Below the trace window is a toolbar with several icons. Japanese labels with arrows point to the following icons:

- フィルタ (Filter)
- 保存 (Save)
- 逆アセンブル (Disassemble)
- ソース (Source)
- 逆順 (Reverse)
- バス (Bus)
- トレース オン/オフ (Trace On/Off)

図 5-27 2つの関数間の point-to-point トレース

上に示した図は、関数 “main()” から “Display_LCD ()” 間のトレース結果を示しています。トレース結果はトレースパラメータ(例えば、分岐タイプ、アドレス範囲)でフィルタし、.xml フォーマットで(バス、アセンブリ、ソース情報を含む)保存することができます。

第6章 ヘルプ

ヘルプシステムによって、ユーザはワークベンチ内の各ヘルプウィンドウやヘルプ画面から、ヘルプドキュメントのブラウズ、検索、ブックマーク、印刷が可能です。また、ヘルプメニューから e² studio 専用のオンラインフォーラムにアクセスできます。

[ヘルプ] をクリックしてヘルプメニューをプルダウンしてください。

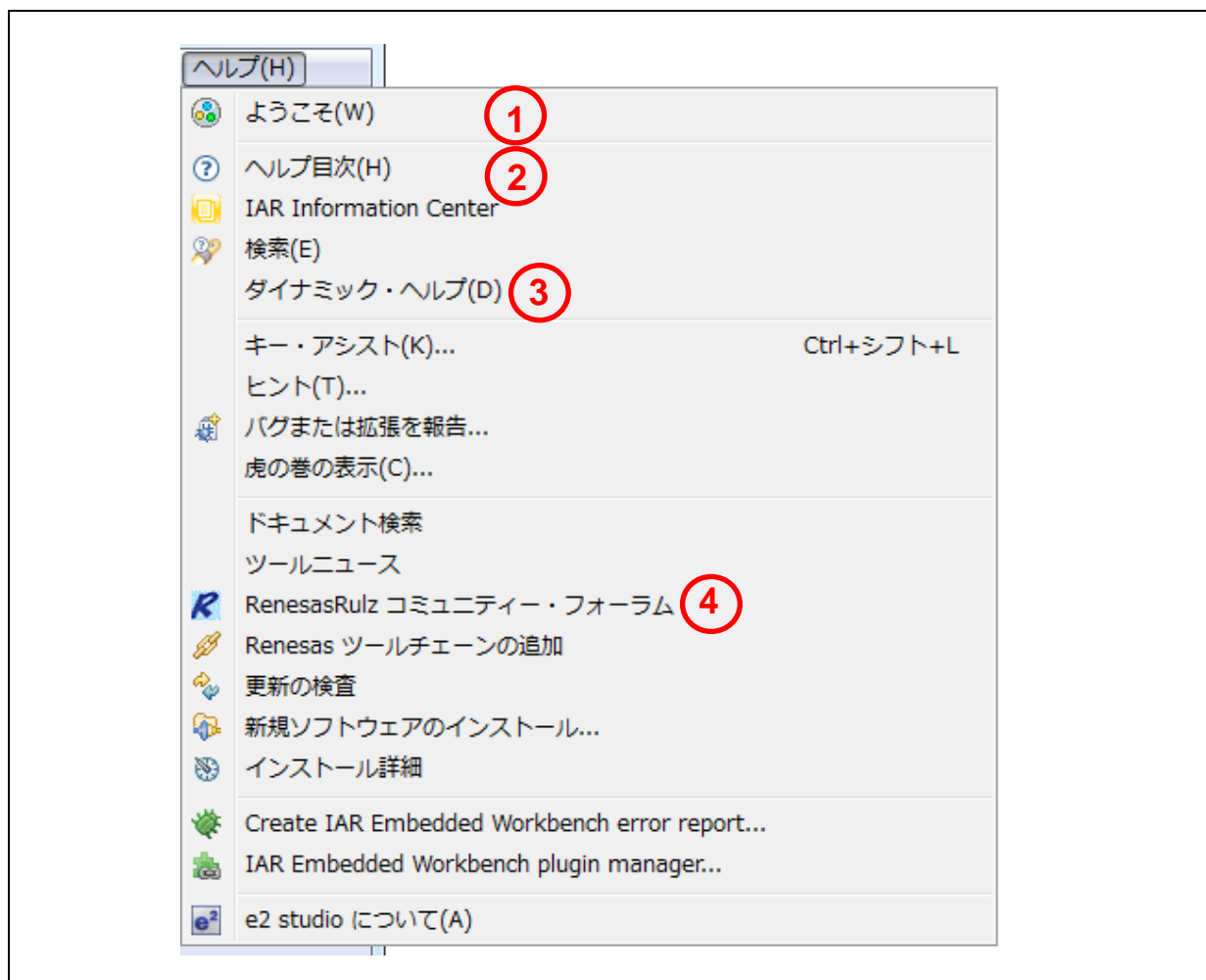


図 6-1 ヘルプメニュー

ヘルプメニューの使い方

- ① [ようこそ] をクリックすると、e² studio の概要、e² studio チュートリアルとサンプルプログラムへのリンク、リリースノートを表示します。
- ② [ヘルプ目次] をクリックすると、新たにヘルプウィンドウが開きヘルプを検索できます。
- ③ [ダイナミック・ヘルプ] をクリックすると、ワークベンチにヘルプ画面を開きます。
- ④ [RenesasRulz コミュニティ・フォーラム] をクリックすると、e² studio 関連のディスカッション参加型オンラインフォーラムにアクセスします。インターネット接続が必要です。

[ヘルプ] ウィンドウには、有用な情報が豊富にあります。例えば “e2 studio デバッグ・ヘルプ” では、デバッグ構成、設定できるブレークポイント数、エミュレータの使用法などの有用な情報を提供します。“e2 studio デバッグ・ヘルプ” にアクセスするには、[ヘルプ] メニュー → [ヘルプ目次] → “e2 studio デバッグ・ヘルプ” の順にクリックしてください。

改訂記録

Rev.	発行日	改訂内容	
		ページ	概要
1.00	2013.11.16	-	初版発行
2.00	2014.04.15	-	e ² studio v3.0.0.022をサポート
2.10	2014.12.08	-	e ² studio v3.1.2.009をサポート
3.00	2015.06.16	-	e ² studio v4.0.0.023をサポート
4.00	2016.05.27	-	e ² studio v5.0.0.043をサポート

e² studio ユーザーズマニュアル
入門ガイド

発行年月日	2013年11月16日	Rev.1.00
	2014年 6月 2日	Rev.2.00
	2015年 6月16日	Rev.3.00
	2016年 5月27日	Rev.4.00

発行 ルネサス エレクトロニクス株式会社



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 エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24 (豊洲フォレシア)

■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/>

e² studio

RENESAS

ルネサスエレクトロニクス株式会社

R20UT2771EJ0300