User's Manual

統合開発環境
e² studio 2020-04、
e² studio v7.8
ユーザーズマニュアル 入門ガイド

ルネサスマイクロコントローラ
RX, RL78, RH850 ファミリ

ルネサス エレクトロニクス
www.renesas.com

Rev.1.00 2020.05

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生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、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またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 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、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング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）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等
   当社製品は、データシート等により高信頼性、Harsh environment向け製品と定義しているものを除き、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（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合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、データシート等において高信頼性、Harsh environment向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他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。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. 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。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
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。
12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

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直接的、間接的に支配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。

(Rev.4.0-1　2017.11)

本社所在地

〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。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。

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策
   CMOS製品の取り扱いの際は静電気防止を心がけてください。CMOS製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS製品を実装したボードについても同様の扱いをしてください。
2. 電源投入時の処置
   電源投入時は、製品の状態は不定です。電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。
3. 電源オフ時における入力信号
   当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。
4. 未使用端子の処理
   未使用端子は、「未使用端子の処理」に従って処理してください。CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。
5. クロックについて
   リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。
6. 入力端子の印加波形
   入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS製品の入力がノイズなどに起因して、V_{IL}（Max.）からV_{IH}（Min.）までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、V_{IL}（Max.）からV_{IH}（Min.）までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。
7. リザーブアドレス（予約領域）のアクセス禁止
   リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。
8. 製品間の相違について
   型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

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e^2 studioについて説明します。

e^2 studioはルネサスエレクトロニクス製デバイスファミリ(RX、RL78、RA、 RE、 RZ)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。

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、e^2 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e^2 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、PC、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

1. 概説

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統合開発環境です。e² studioは、オープンソースEclipse IDEとCDT (C/C++ 開発ツール)をベースに作られており、ビルド（エディタ、コンパイラ、リンカ）から、デバッグまでをカバーします。デバッグはGDB(GNU Debugger)の拡張インタフェースにより実現されます。

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。

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。

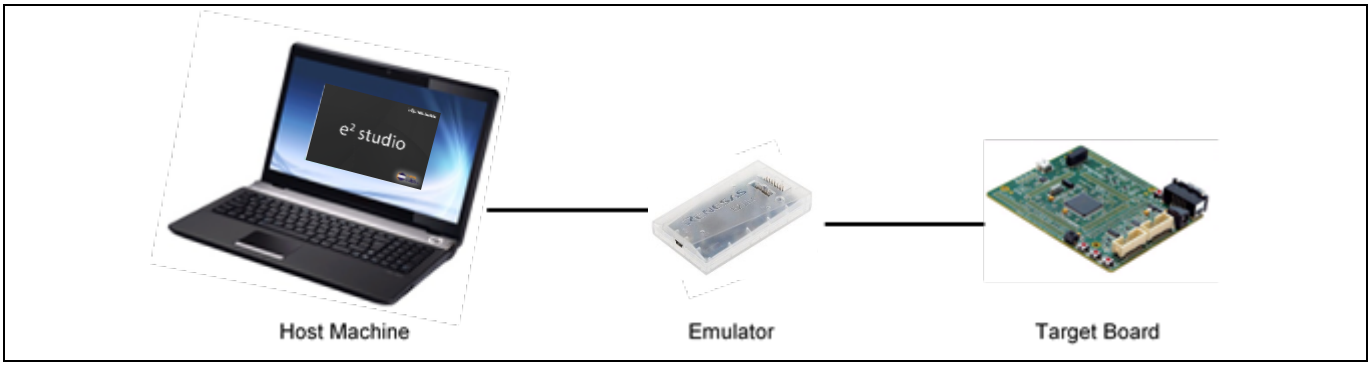

図 1-1 システム構成

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: 2GB以上の空きメモリ
ハードディスク: 2GB以上の空き領域
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

動作環境:
Windows 8.1 (32/64ビット OS) and Windows 10 (32/64ビット OS).
e² studio 2020-04 またはそれ以降には64ビット OSが必要です。

1.2.2 サポートするツールチェーン

ルネサス RXファミリ用 C/C++ コンパイラパッケージ
ルネサス RL78ファミリ用 C コンパイラパッケージ
Gnu ARM Embedded Toolchian
GNURX Windows Toolchain (ELF)
GNURL78 Windows Toolchain (ELF)
注1: ルネサス製コンパイラパッケージにはそれぞれ「統合開発環境付き」(CS+に同梱)、「統合開発環境なし」(コンパイラ単体)がありますが、e² studioにはいずれも適用可能です。

注2: e² studioはRH850用に作られたロードモジュールをデバッグに利用できますが、ELF/DWARF形式で作られている必要があります。ビルド環境はIARシステムズ社製IAR Embedded WorkbenchまたはGreen Hills Software社製MULTIをお使いください。

1.3 サポートするエミュレータ

E2 emulator Lite (RX, RL78)
E1 (RX, RL78, RH850)
E2 (RX, RL78, RH850)
E20 (RX)

注： E1エミュレータは生産を終了しております。

1.4 サポートするシミュレータ

Renesas Simulator (RX, RL78)
GDB Simulator (RH850)

2. インストール

ルネサスウェブサイトから最新のe² studio統合開発環境インストーラパッケージを無償でダウンロードできます。e² studioには32ビット版と64ビット版があります。詳しくはhttps://www.renesas.com/e2studioをご覧ください。なお、ソフトウェアをダウンロードするには無償のユーザ登録が必要です(MyRenesasページでのログインを求められます)。

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。

e² studioのインストーラは新規のインストールおよびアップデートに使用できますが、メジャーバージョンをまたいだ更新(V5.*からV6.*、V6.*からV7.*など)には対応しません。旧バージョンをアンインストールするか、インストール先のフォルダを別に作成して新規にインストールを行ってください。

詳しいインストール方法を以下に示します。

2.1 e² studio (64ビット版)のインストール

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。
[インストール] をクリックします。

注：既にe² studioがインストールされている場合は、[インストール]の他に[変更](インストール済e² studioの変更)や[削除] (アンインストール)の選択肢が表示されます。

図 2-1 64 ビット版 e² studio のインストールウィザード

1. ようこそページ

[変更...]をクリックすると、インストール先のフォルダを変更できます。[Next]で次に進みます。

注 1： 複数バージョンの e² studio を利用したい場合は新しいフォルダを指定してください。
注 2： インストール先のフォルダ名にマルチバイト(全角)文字を使用しないでください。

図 2-2 64 ビット版 e² studio のインストール - ようこそページ

1. デバイスファミリ

   お使いのデバイスファミリを選択してください。 [Next] で次に進みます。

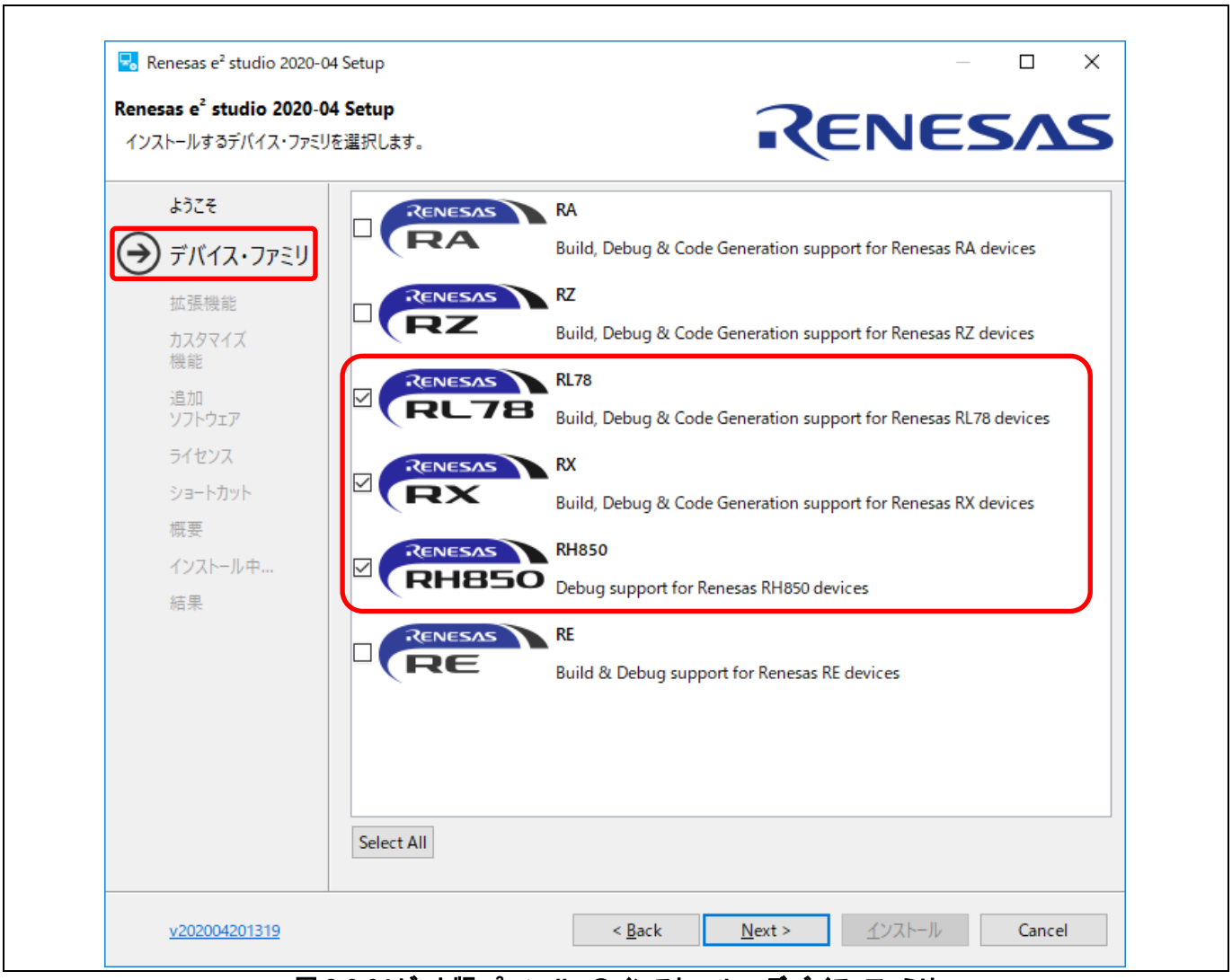

図 2-3 64 ビット版 e² studio のインストール – デバイス・ファミリ

拡張機能

インストールする拡張機能（言語パック、SVN や Git、RTOS のサポートプラグイン）を選択してください。
英語以外の言語を利用する場合はここで言語パックを選択しておく必要があります。

[Next]で次に進みます。

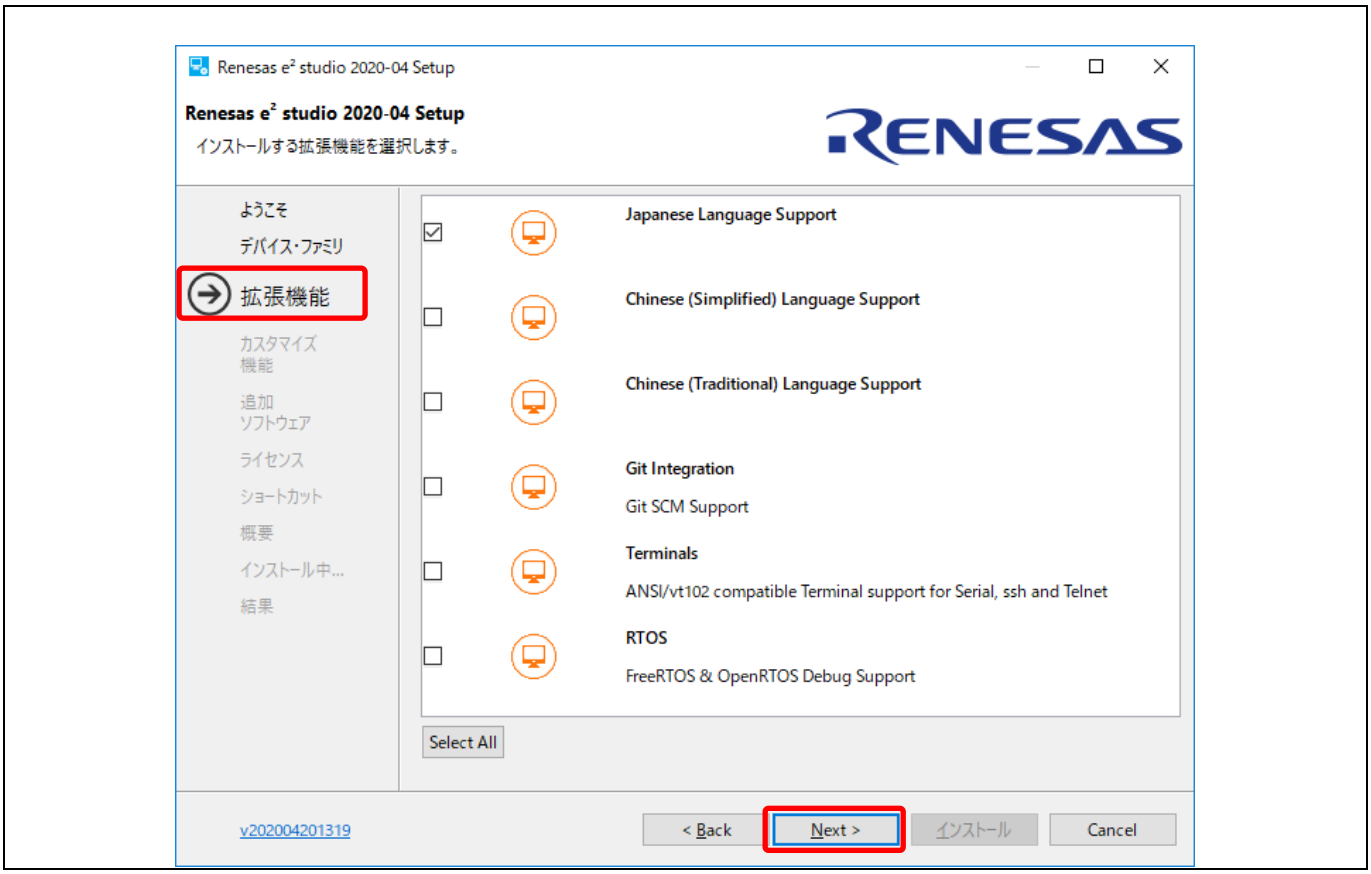

図 2-4 64 ビット版 e² studio のインストール – 拡張機能

1. カスタマイズ機能

   インストールするコンポーネントを選択する画面です。 [Next] で次に進みます。

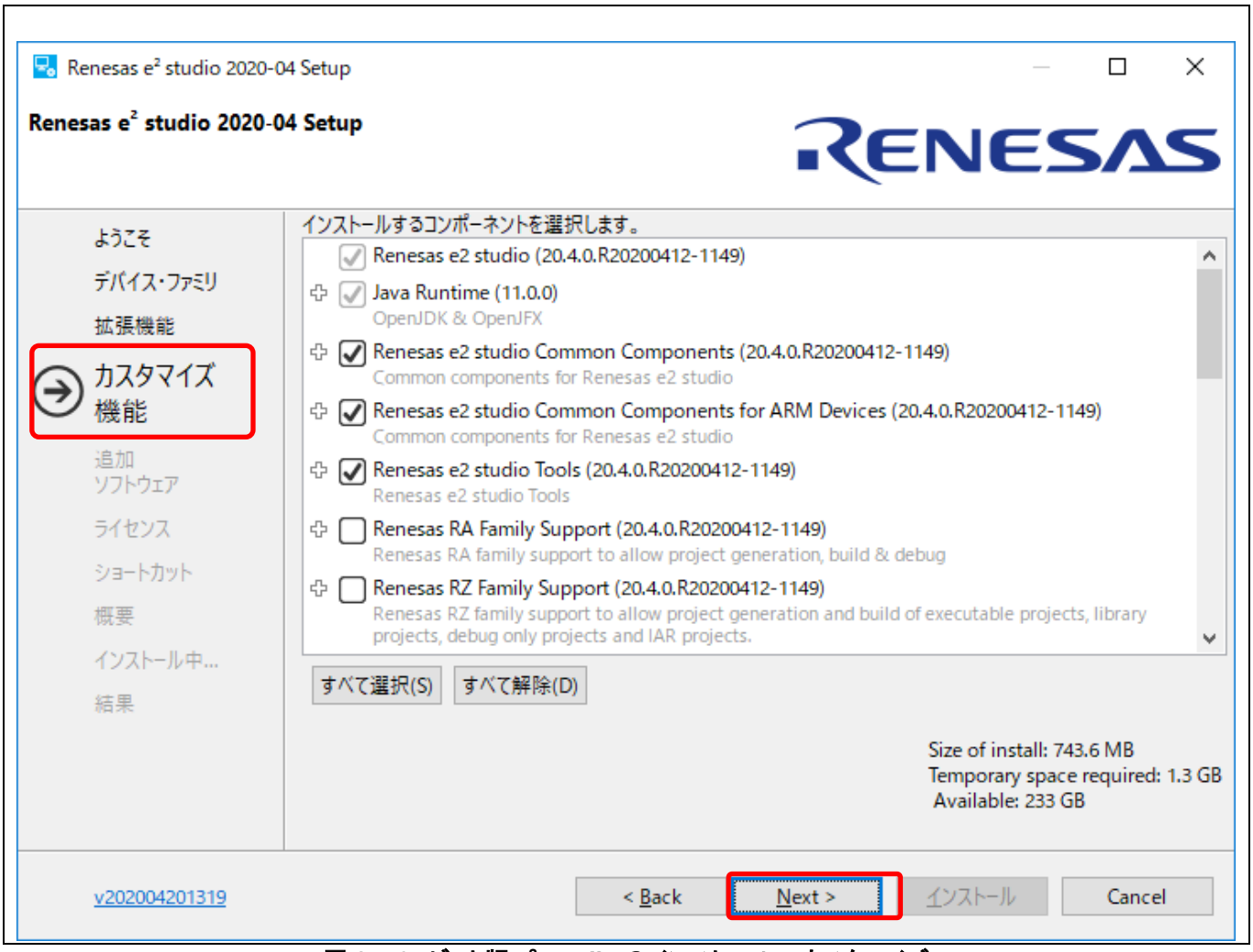

図 2-5 64 ビット版 e² studio のインストール – カスタマイズ

追加ソフトウェア

追加でインストールするソフトウェア（コンパイラ、ユーティリティ、QE など）を選択してください。[Next] で次に進みます。

注：インターネット接続のない環境ではソフトウェアカタログがダウンロードできないとの警告が表示されますが、インストールは続けることができます。必要なソフトウェアは別途インストールしてください。

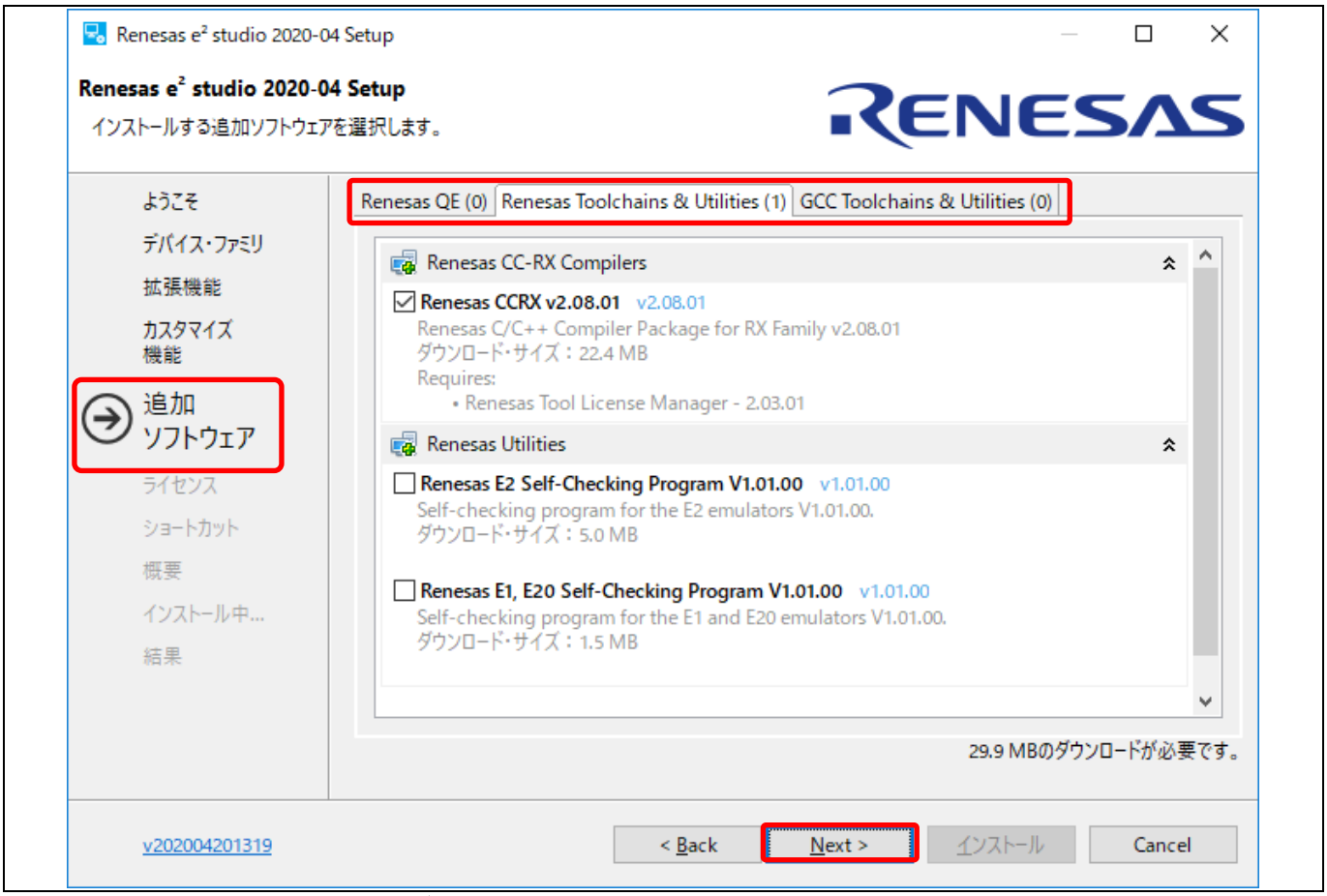

図 2-6 64 ビット版 e² studio のインストール – 追加ソフトウェア

1. ライセンス

   ソフトウェア契約の条項を確認したのち、同意戴けましたらチェックを入れて［Next］ボタンを押してください。同意戴けない場合はインストールを継続できません。

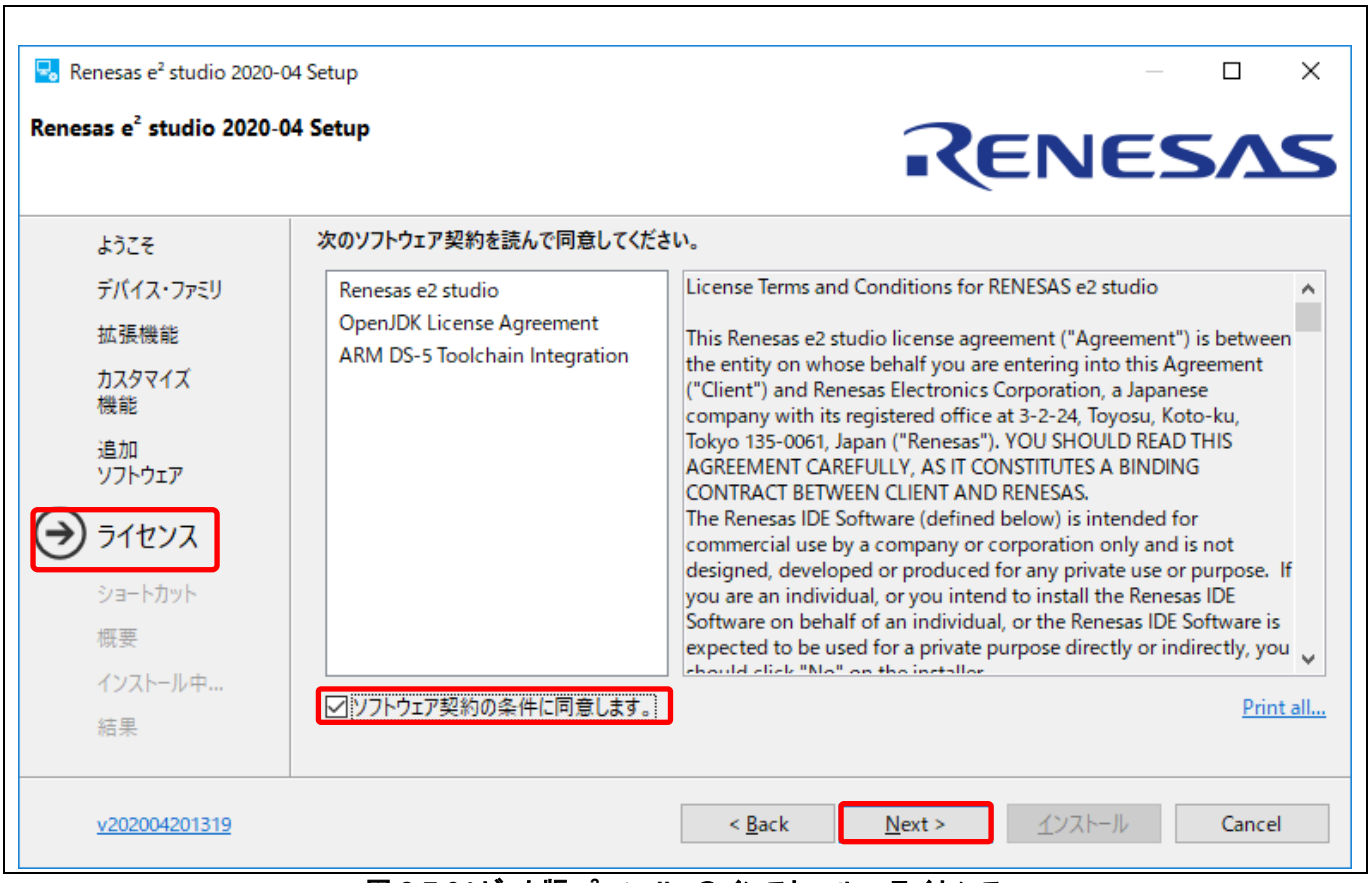

図 2-7 64 ビット版 e² studio のインストール – ライセンス

1. ショートカット

スタートメニューに登録するショートカット名を入力します。 [Next]で次に進みます。

注： 既に別のバージョンの e² studio がインストールされている場合はそれと区別が付くような名前に書き換えておくことをお勧めします。

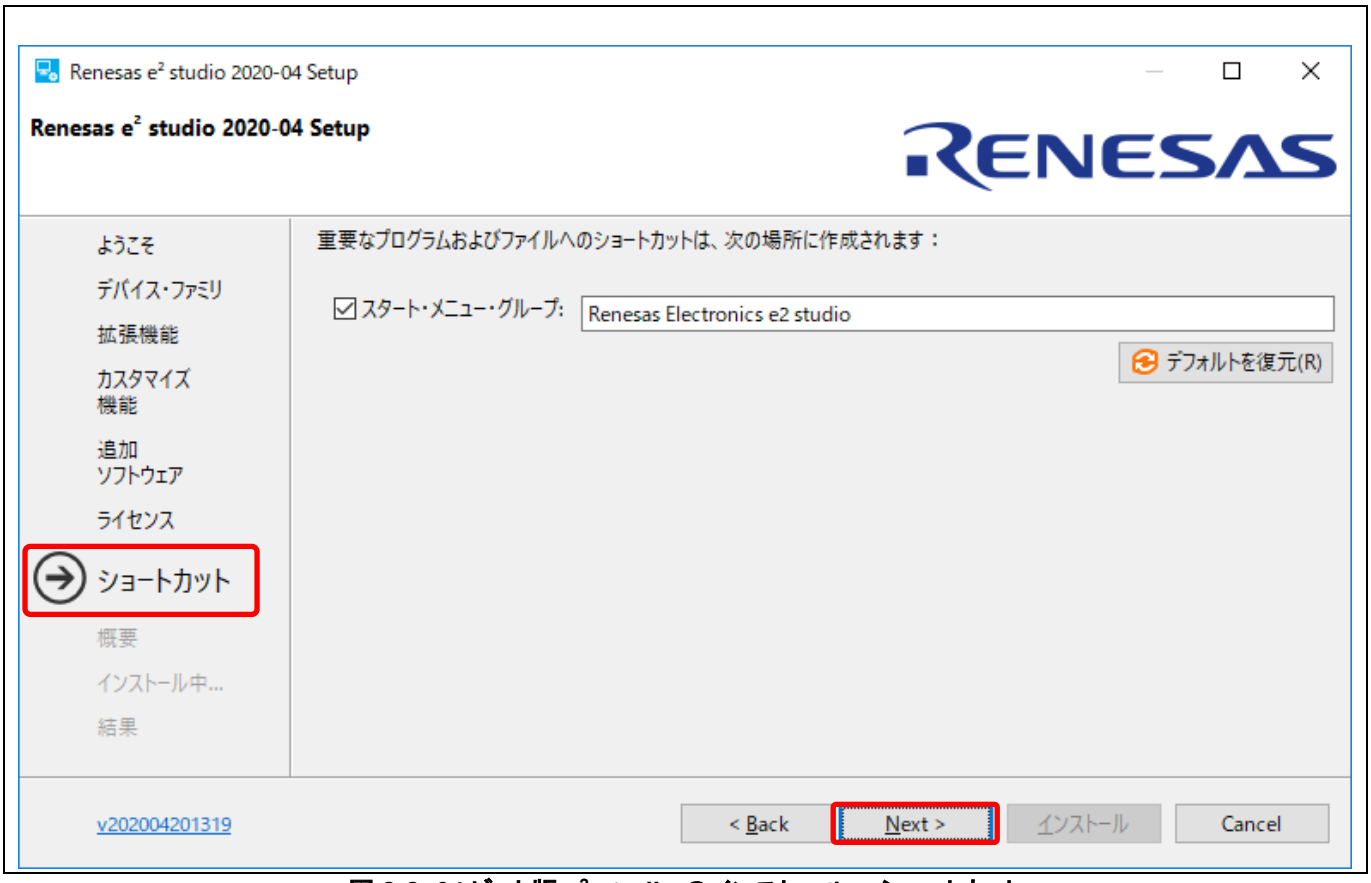

図 2-8 64 ビット版 e² studio のインストール – ショートカット

概要

これからインストールされるコンポーネントが一覧で表示されます。内容を確認してから[インストール]ボタンを押すと e² studio のインストールが始まります。

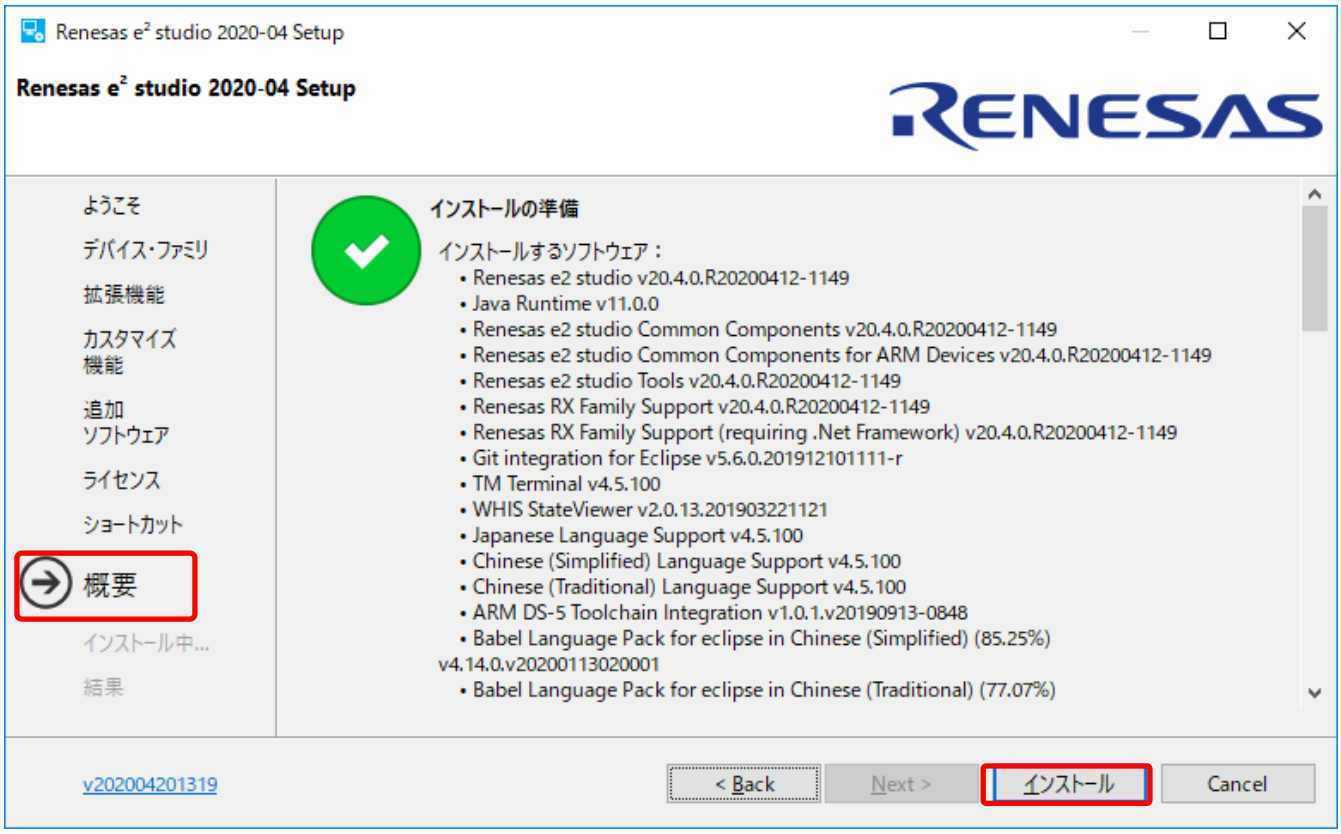

図 2-9 64 ビット版 e² studio – 概要

1. インストール中. . .

   インストールが進行中です。選択した追加ソフトウェアによっては、それらのインストーラが途中で起動しますので画面に従って操作を行ってください。

2. 結果

   インストールを行った結果が表示されます。エラーが表示されていないかを御確認ください。

   [OK]を押すとインストーラが終了します。

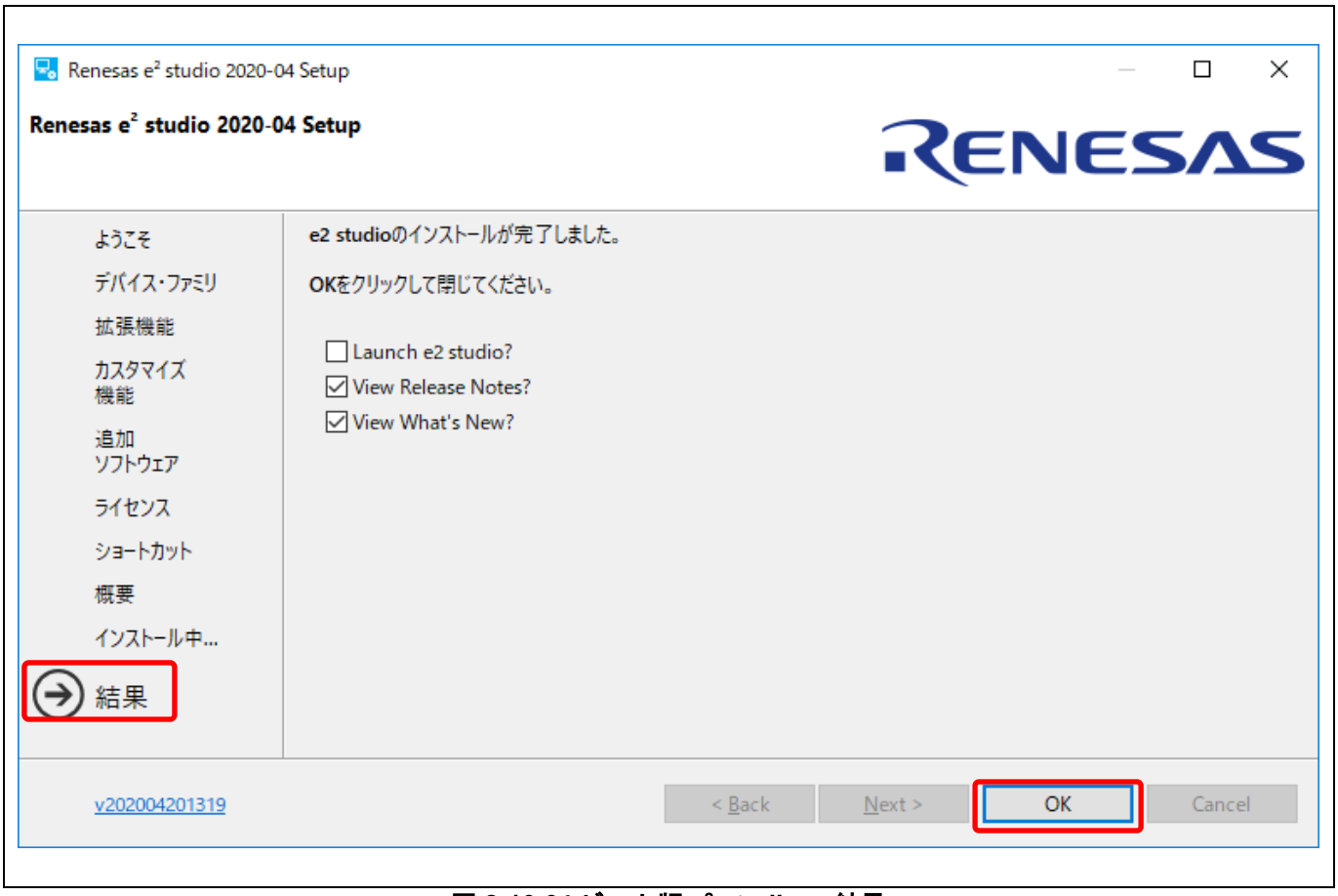

図 2-10 64 ビット版 e² studio – 結果

2.2 e² studio (32ビット版)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。［インストール］をクリックします。

注：既に e² studio がインストールされている場合は、［インストール］の他に［変更］（インストール済 e² studio の変更）や［削除］（アンインストール）の選択肢が表示されます。複数バージョンの e² studio を利用したい場合は新しいフォルダを指定してください。

図 2-11 32 ビット版 e² studio インストールウィザード

ようこそページ

[Next]で次に進みます。

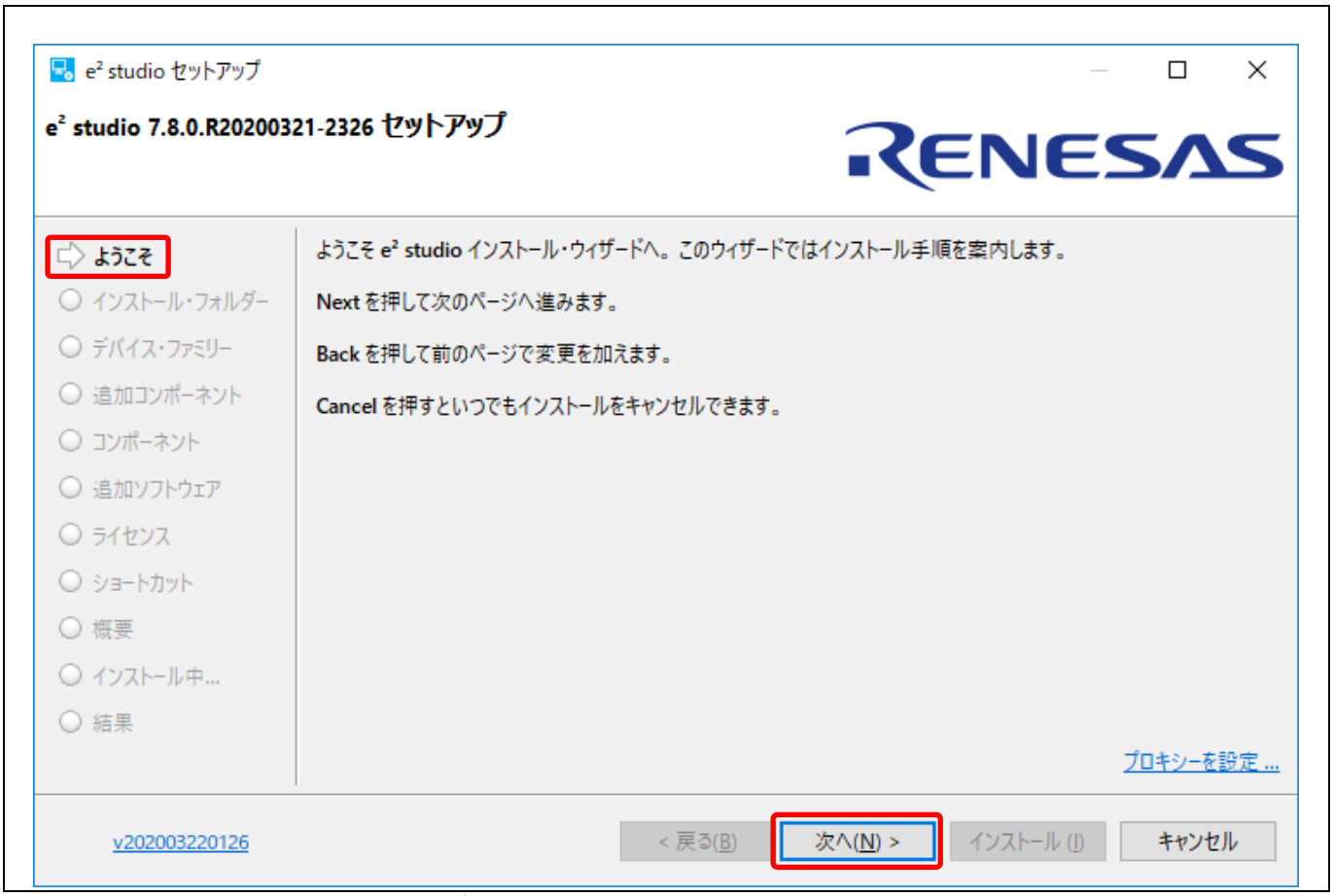

図 2-12 32 ビット版 e² studio のインストール – ようこそ ページ

2. インストールフォルダー

デフォルトのインストール先は “C:\Renesas\e2_studio”になっています。テキストボックスにフォルダ名を直接入力するか、[参照. . .]ボタンからフォルダを選択してください。 [Next]で次に進みます。

注 1： 複数バージョンの e² studio を利用したい場合は新しいフォルダを指定してください。
注 2： インストール先のフォルダ名にマルチバイト(全角)文字を使用しないでください。

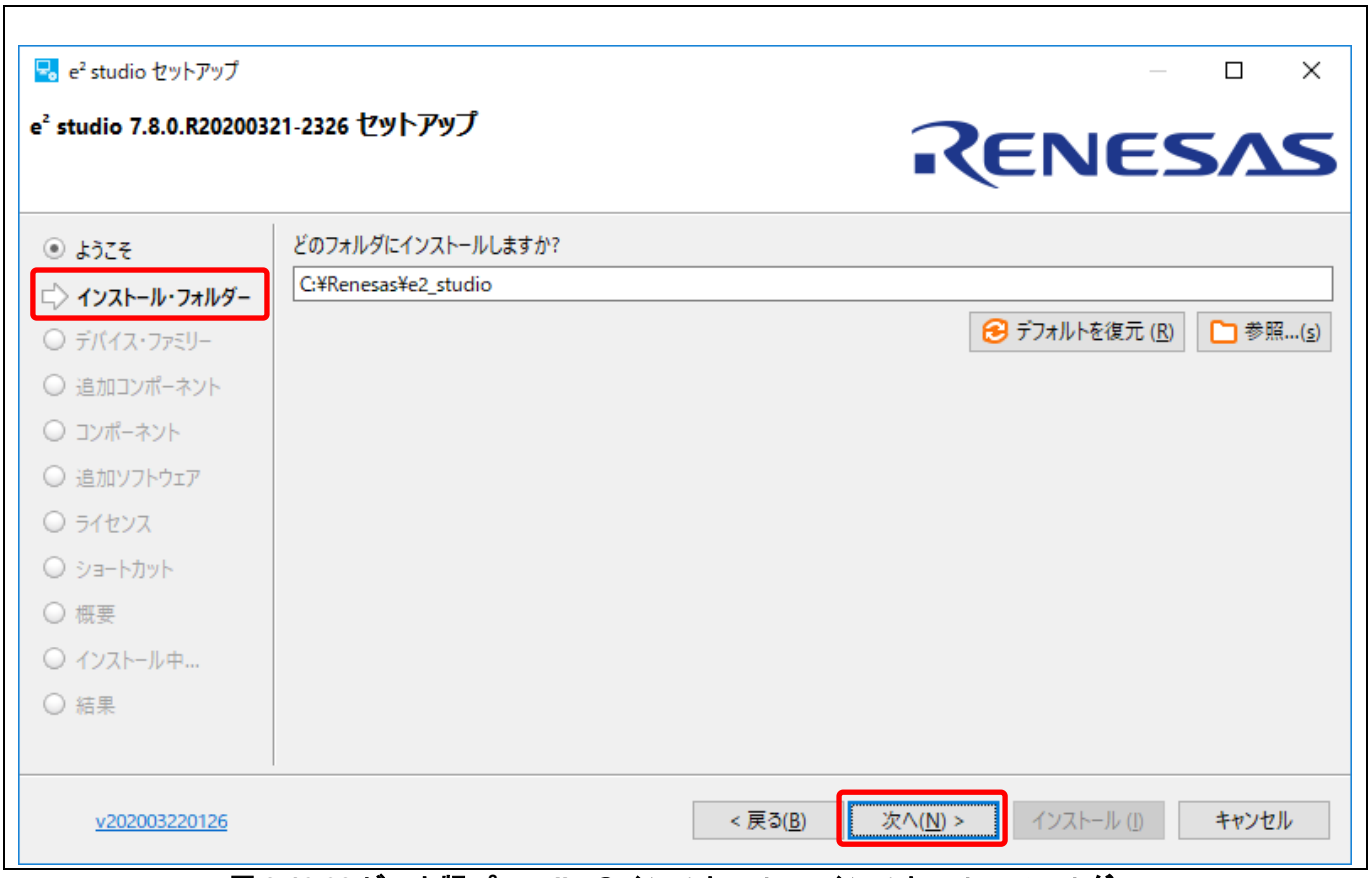

図 2-13 32 ビット版 e² studio のインストール - インストール・フォルダー

3. デバイス・ファミリー

お使いのデバイスファミリを選択してください。 [Next] で次に進みます。

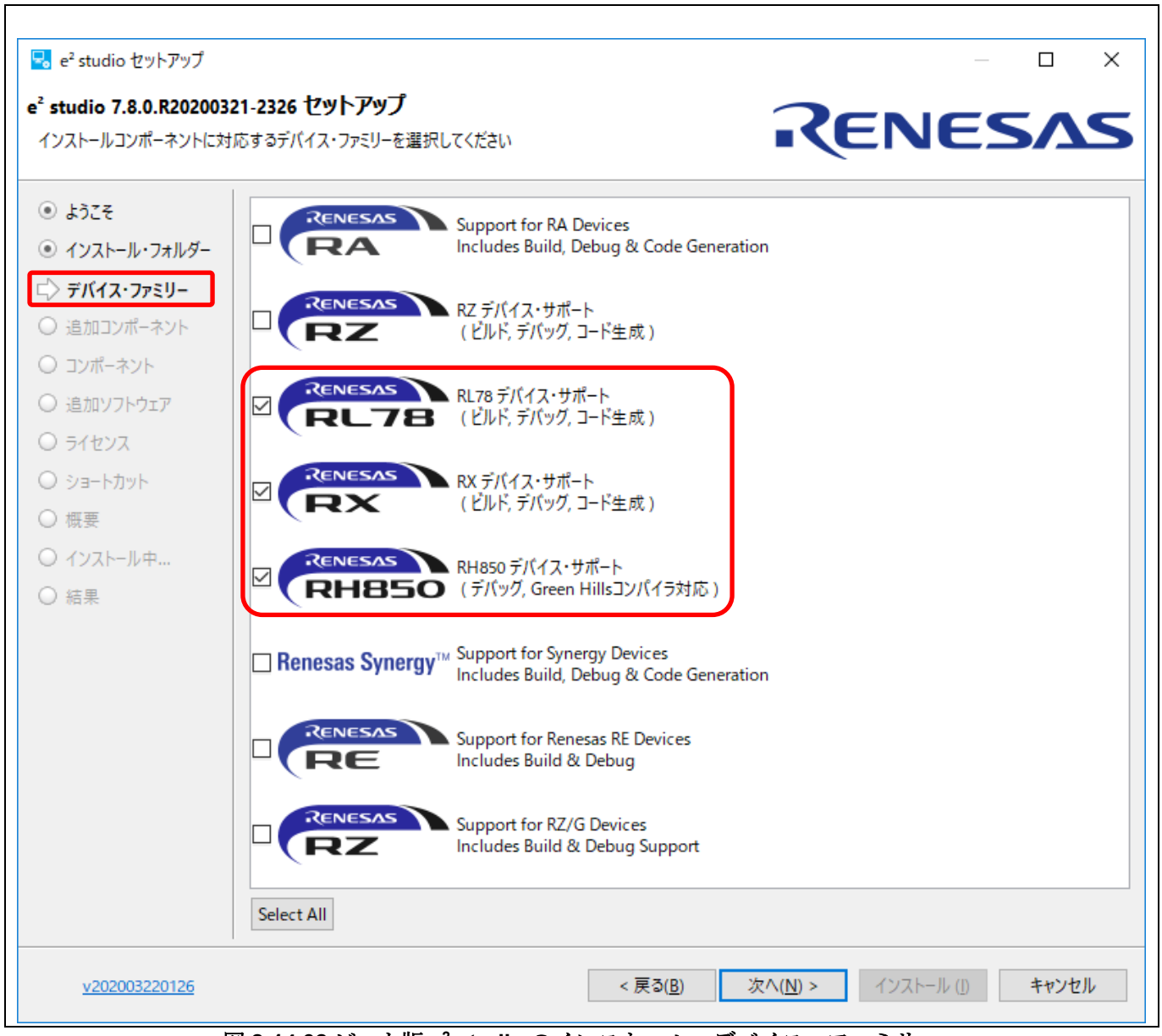

図 2-14 32 ビット版 e² studio のインストール – デバイス・ファミリー

4. Extra Components

インストールする拡張機能（言語パック、SVN や Git、RTOS のサポートプラグイン）を選択してください。
英語以外の言語を利用する場合はここで言語パックを選択しておく必要があります。

[Next]で次に進みます。

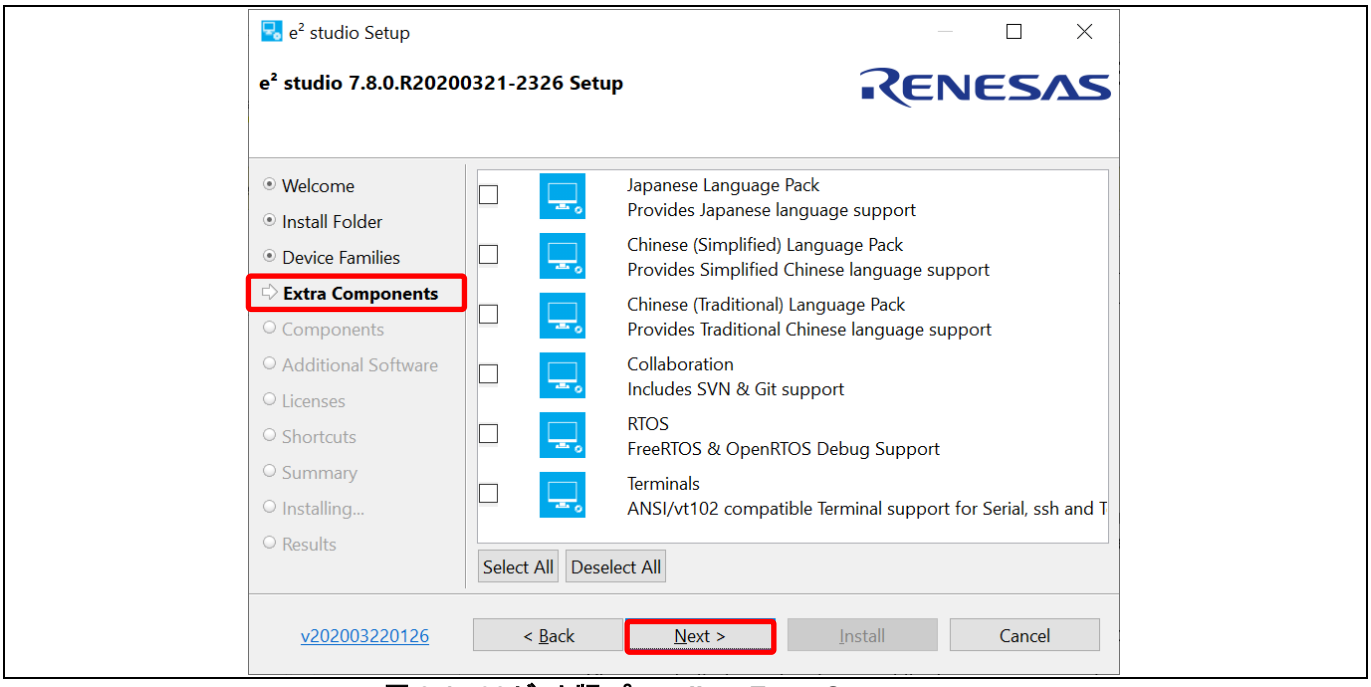

図 2-15 32 ビット版 e² studio – Extra Components

5. Components

インストールするコンポーネントを選択する画面です。 [Next] で次に進みます。

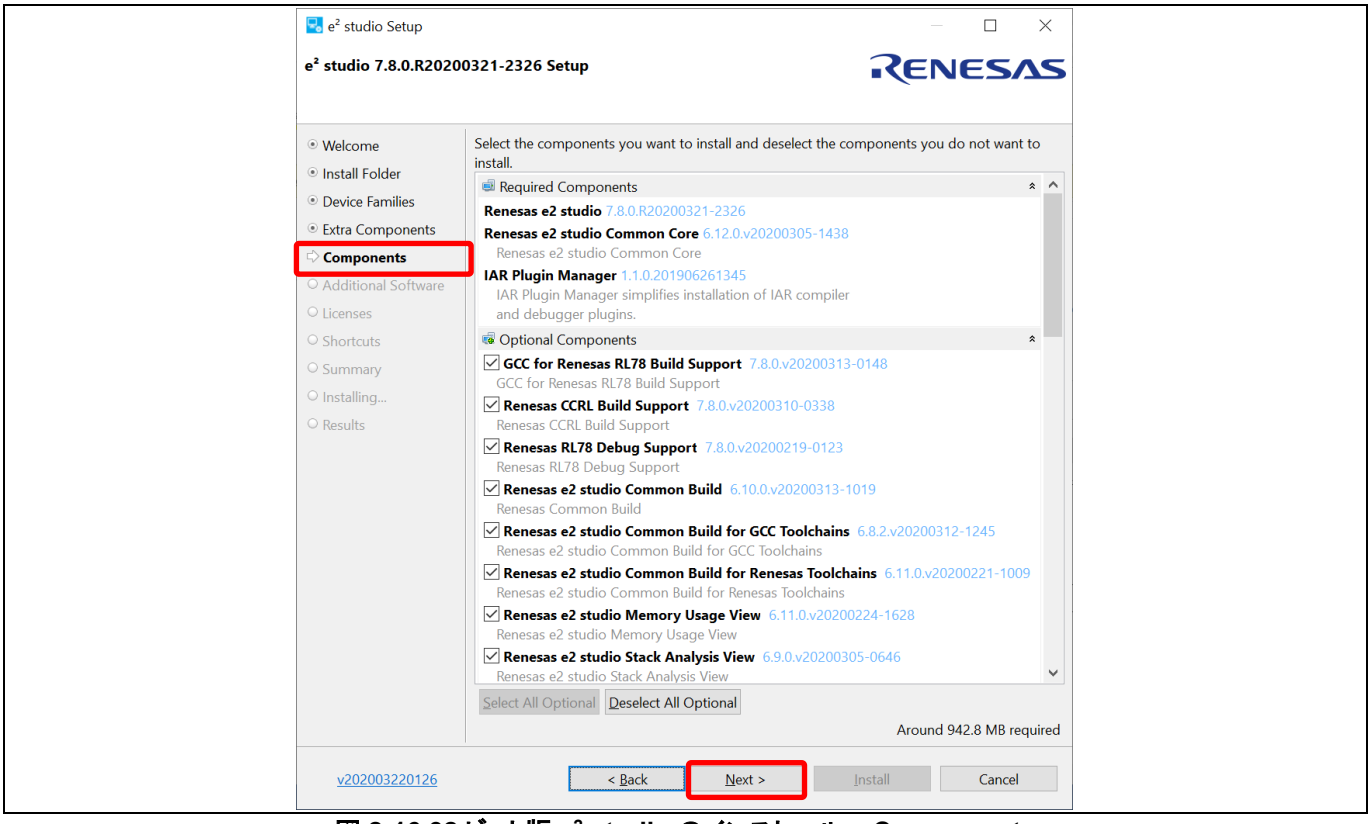

図 2-16 32 ビット版 e² studio のインストール – Components

6. Additional Software

追加でインストールするソフトウェア（コンパイラ、ユーティリティ、QE など）を選択してください。
[Next] で次に進みます。

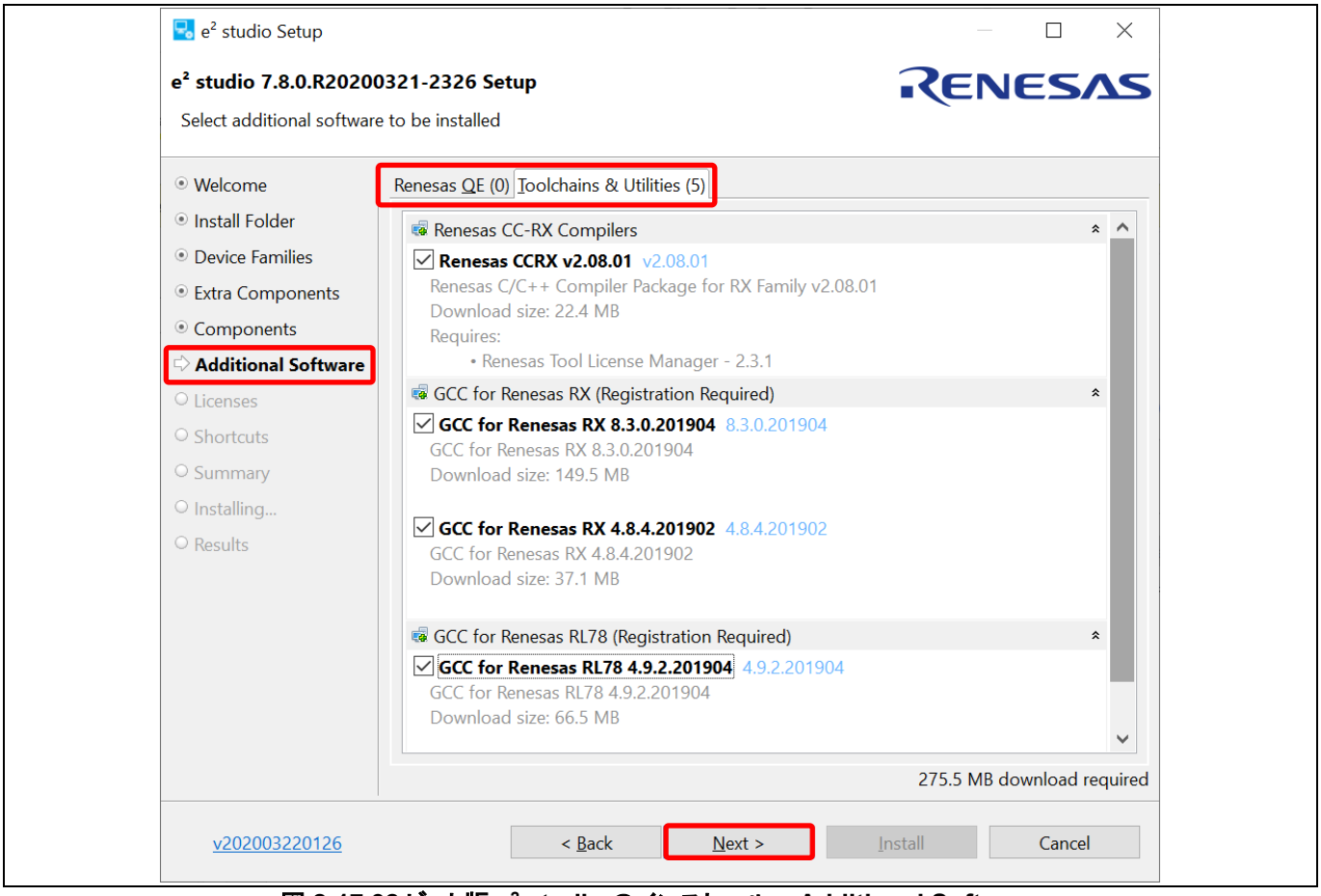

図 2-17 32 ビット版 e² studio のインストール – Additional Software

License Agreement、Shortcuts、Summary、インストール中画面、結果表示画面

これらのページは e² studio 64 ビット版のインストーラと同様です。2. 1 章の 7. 以降を参照してください。

2.3 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。

注： コントロールパネルに該当するe² studioが表示されないか、アンインストールができない場合は、
以下のフォルダにあるアンインストーラを直接実行してください。

{e² studioのインストールフォルダ}¥uninstall¥uninstall.exe

2.4 e² studioプラグインの更新

以下に示す方法は、インストール済のプラグインをインターネット経由で更新します。ただしこの方法では新しい機能をインストールすることはできませんのでe² studioのバージョンアップはインストーラにより行ってください。

2.4.1 マイナーバージョン間の更新

この章ではオンラインでマイナーバージョン間の更新を示します。

1. [ヘルプ] メニューから [更新の検査] をクリックして、[利用できる更新] パネルを表示してください。

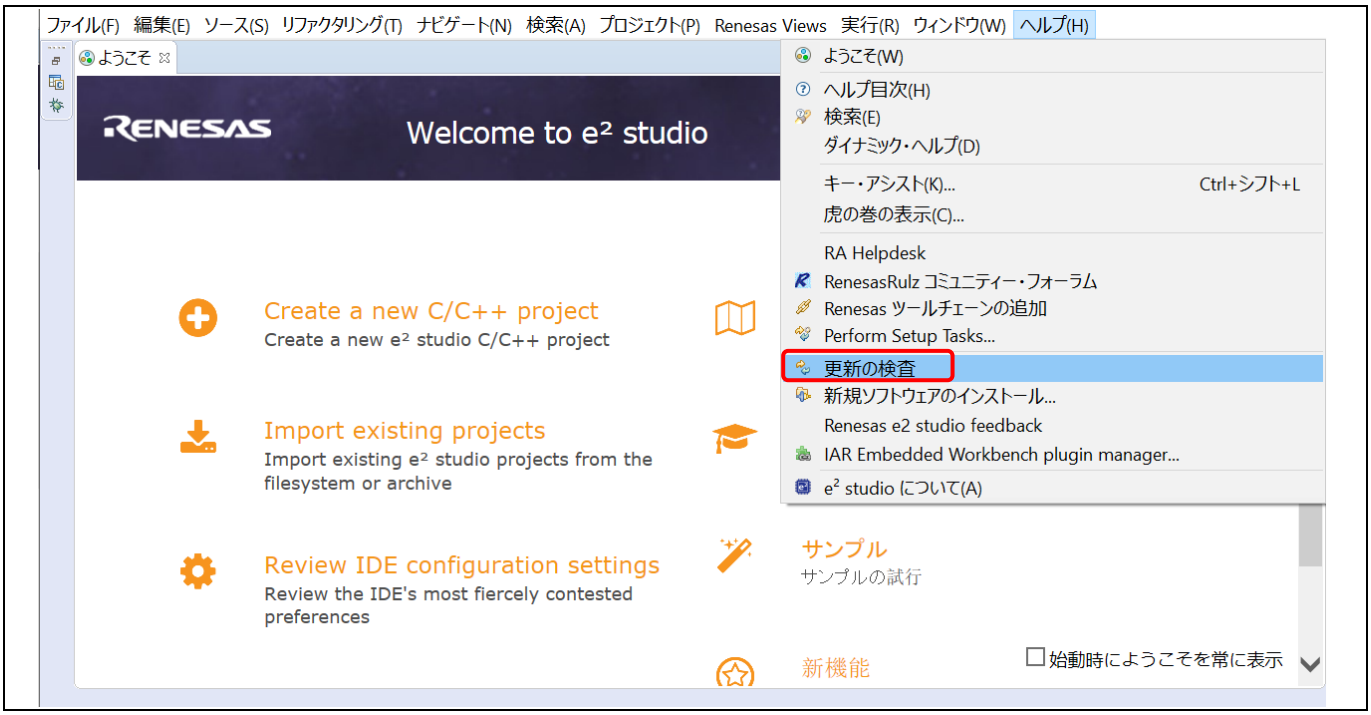

図 2-18 [更新の検査] メニュー

[利用できる更新] パネルでは、デフォルトですべてのソフトウェアコンポーネントが選択されています。これにより、すべてのソフトウェアを最新のものにアップデートすることができます。(下図参照)
[次へ(N)>] ボタンをクリックしてください。

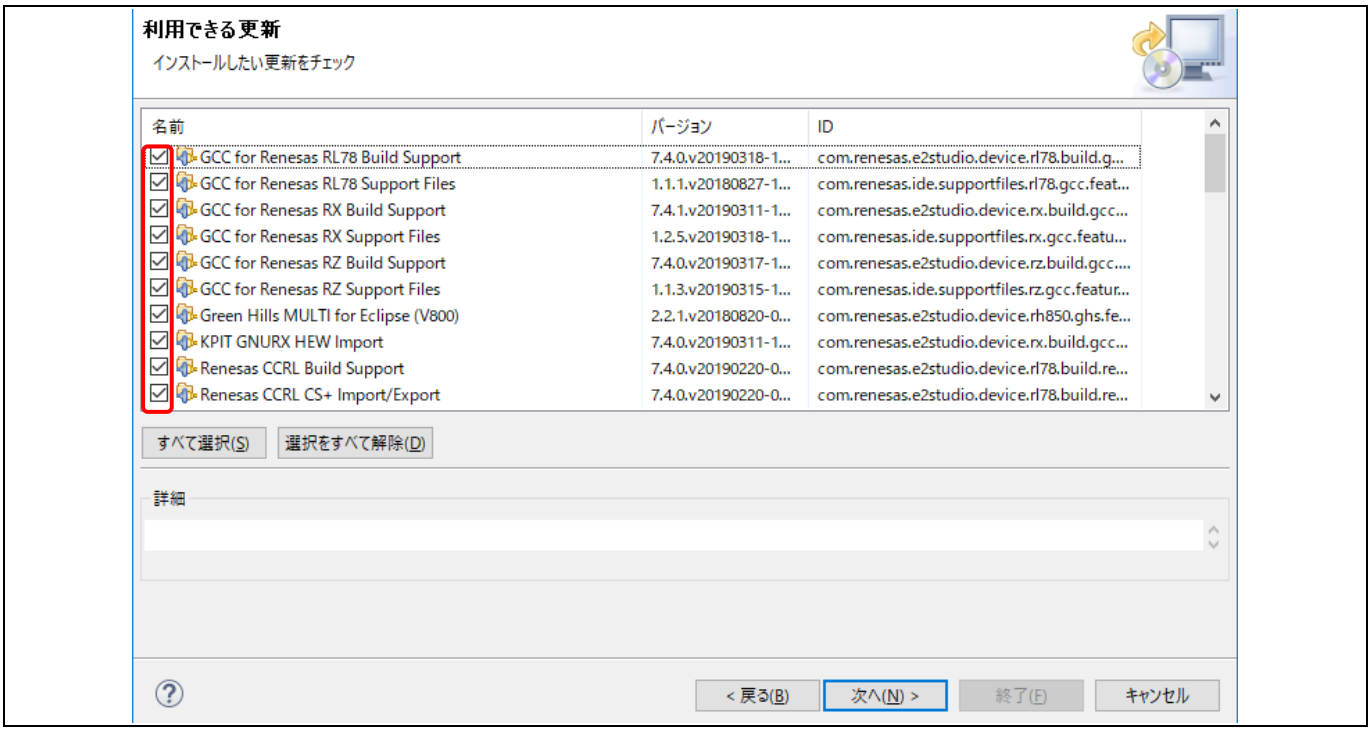

図 2-19 e² studio – [利用できる更新]パネル(1/3)

[次へ(N)>]ボタンでアップデートを続行します。

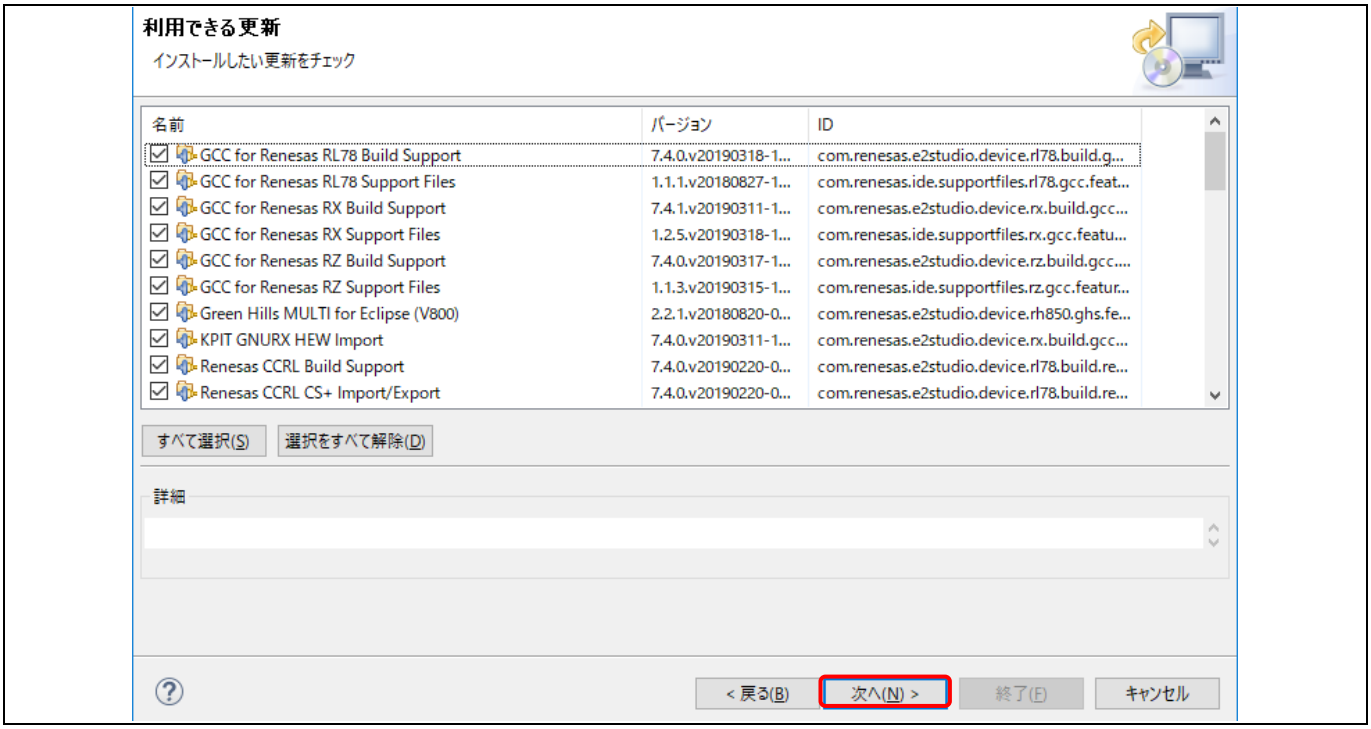

図 2-20 e² studio – [利用できる更新]パネル(2/3)

2. ソフトウェアライセンス契約を確認してください。[終了(F)]ボタンを押すとアップデートが完了します。

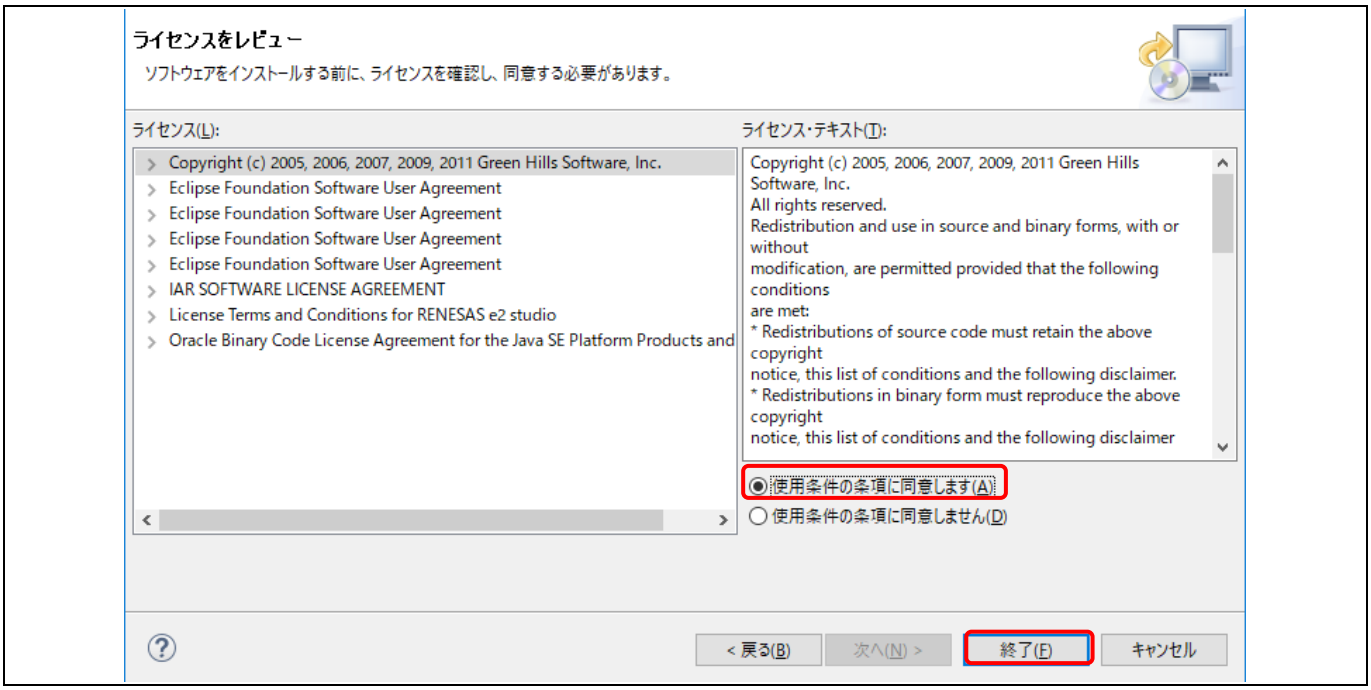

図 2-21 e² studio – [利用できる更新]パネル(3/3)

3. [ヘルプ] メニューの [e2 studio について] をクリックし、アップデートされたバージョンを確認してください。

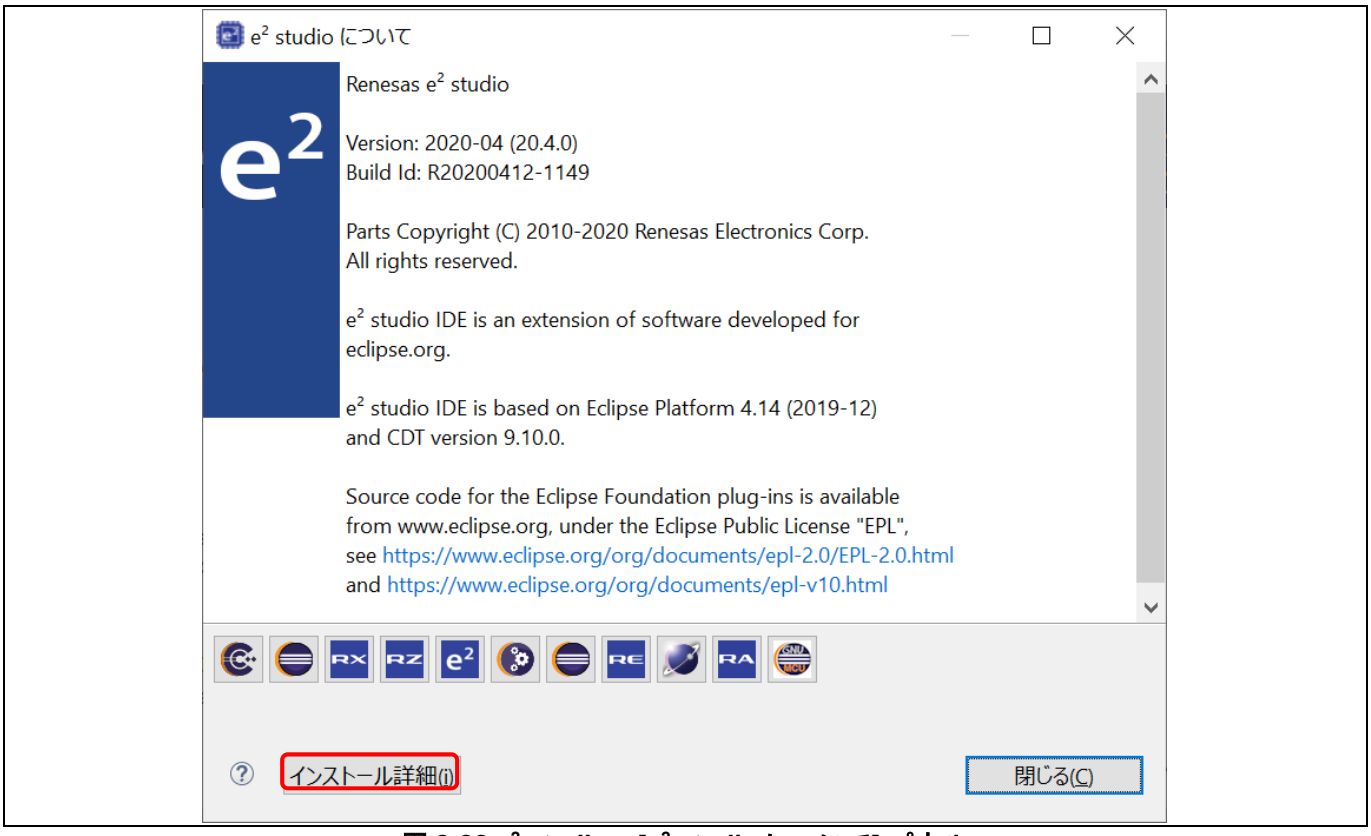

図 2-22 e² studio – [e² studio について] パネル

2.4.2 インストーラによるマイナーバージョンの更新方法

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更新するための手順を説明します。

(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(オフライン用)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更新方法は、バージョン 4.0 以降では利用できません。

(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。既にインストール済みのバージョンがあれば「インストール」以外の選択肢が現われます。

[アップグレード] ― 同一メジャーバージョン間で更新を行う (例: V6.2.0→V6.3.0)
[修正] ― 同一バージョンのインストーラを使ってプラグインを追加インストールする
[削除] ― インストール済の e² studio をアンインストールします
[インストール] ― 新しい場所に e² studio をインストールします

[次へ(N)>] ボタンで次に進みます。

図 2-23 e² studio インストーラの起動画面 (インストール済のものがある場合)

注意: インストーラとメジャーバージョンが異なるe² studioには「アップグレード」や「修正」を適用しないでください。「インストール」を選択して別なフォルダにインストールするか、旧版を一旦アンインストールしてください。

以降の操作は 2.1 章または 2.2 章と同様です。[インストール]以外を選択するとインストール先フォルダの入力はスキップされ、インストール済の e² studio に上書きされます。

2.5 Installation of Compiler Package

2.1章で示したように、インターネット接続時には「追加ソフトウェア」の画面でコンパイラパッケージをインストールできますが、インストール時にコンパイラを選択しなかった場合、あるいはインターネット接続のない環境ではコンパイラパッケージのインストールは別途行ってください。

コンパイラパッケージ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。

RXファミリ用: http://www.renesas.com/rx_c

RL78ファミリ用: http://www.renesas.com/rl78_c

GNUツールチェーン: https://gcc-renesas.com/ (GCC for Renesas RXおよびRL78)

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、[Renesasツールチェーン管理] (下図)を開きます。e² studioに登録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。

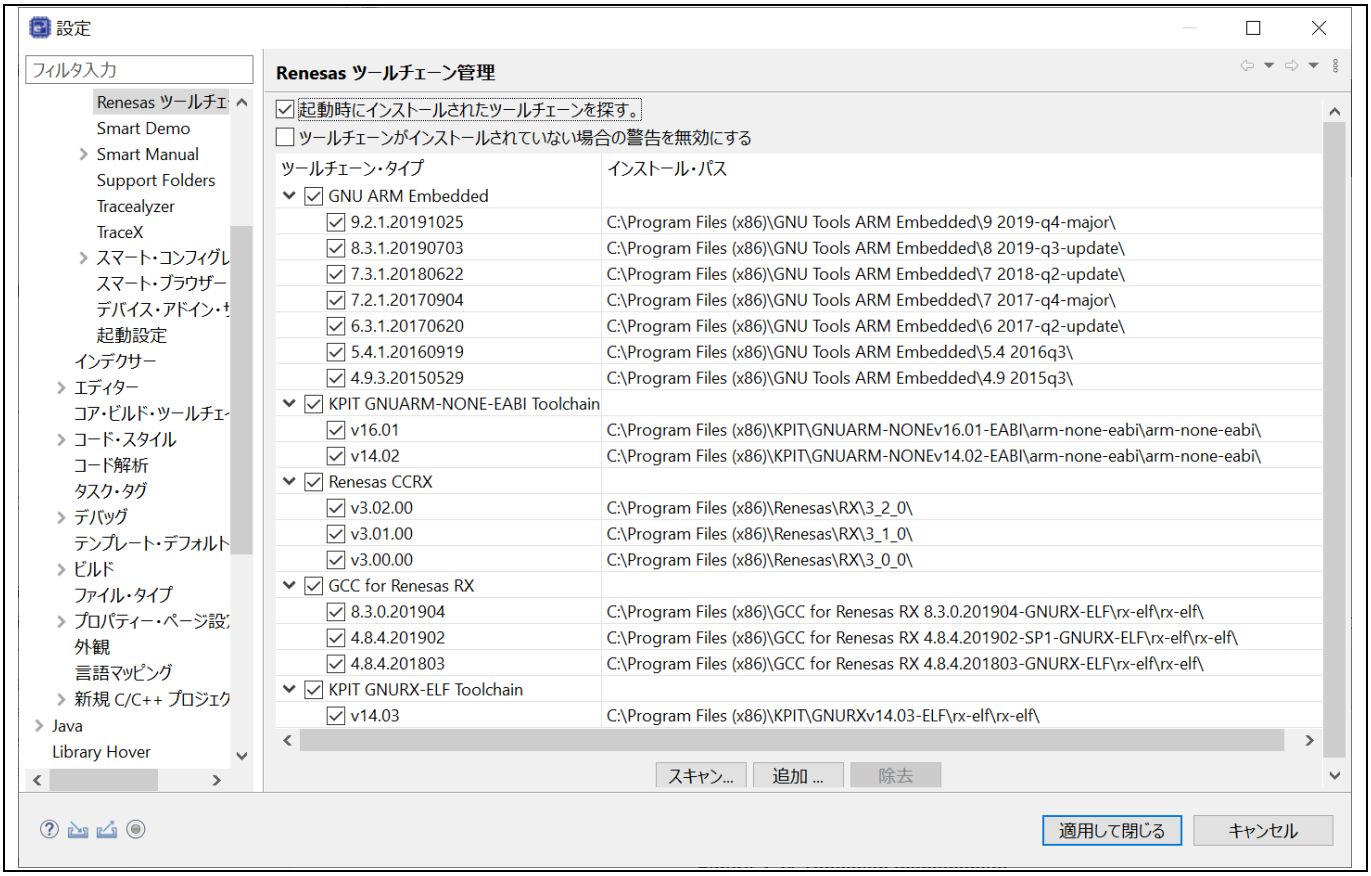

図 2-24 ツールチェーン管理

3. プロジェクトの作成

e² studioでは、「プロジェクト」がビルドやデバッグ操作の基本単位です。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. e² studioを使用するには、e² studioだけでなくコンパイラパッケージのインストールも必要です。

2. e² studioをインストールするフォルダや、プロジェクト名、ワークスペースフォルダのパスには空白文字やマルチバイト文字が混じらないようにしてください。

3.1 新規プロジェクトの作成

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。このワークスペースに作成したプロジェクトが追加されます。
新規にプロジェクトを作成するには、以下の手順を実行してください。

(1) [ファイル] → [新規] → [C/C++ Project] の順にクリックすると、新規プロジェクト作成ウィザードが起動します(下図)。

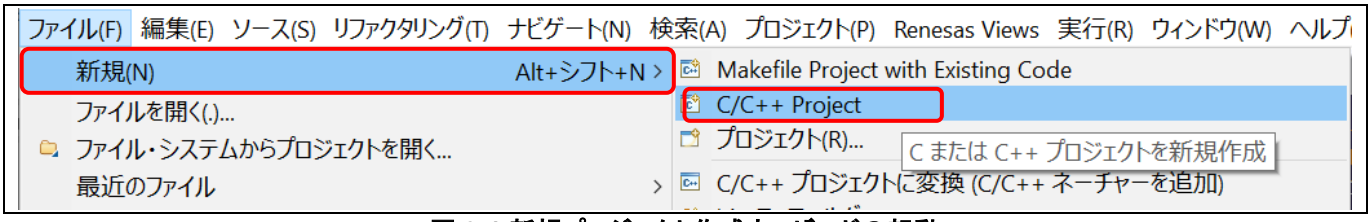

図 3-1 新規プロジェクト作成ウィザードの起動

(2) デバイスとプロジェクトの種類を選択します(例えば RX デバイスの場合は「Renesas CC-RX Toolchain」)。もし使用したいデバイスファミリが表示されない場合はそのデバイス用のビルド・デバッグサポートプラグインがインストールされていないかもしれません。その場合はインストーラで「修正」を選択し、プラグインを追加するか再インストールしてください。
[次へ(N)>] をクリックすると次の画面に進みます。

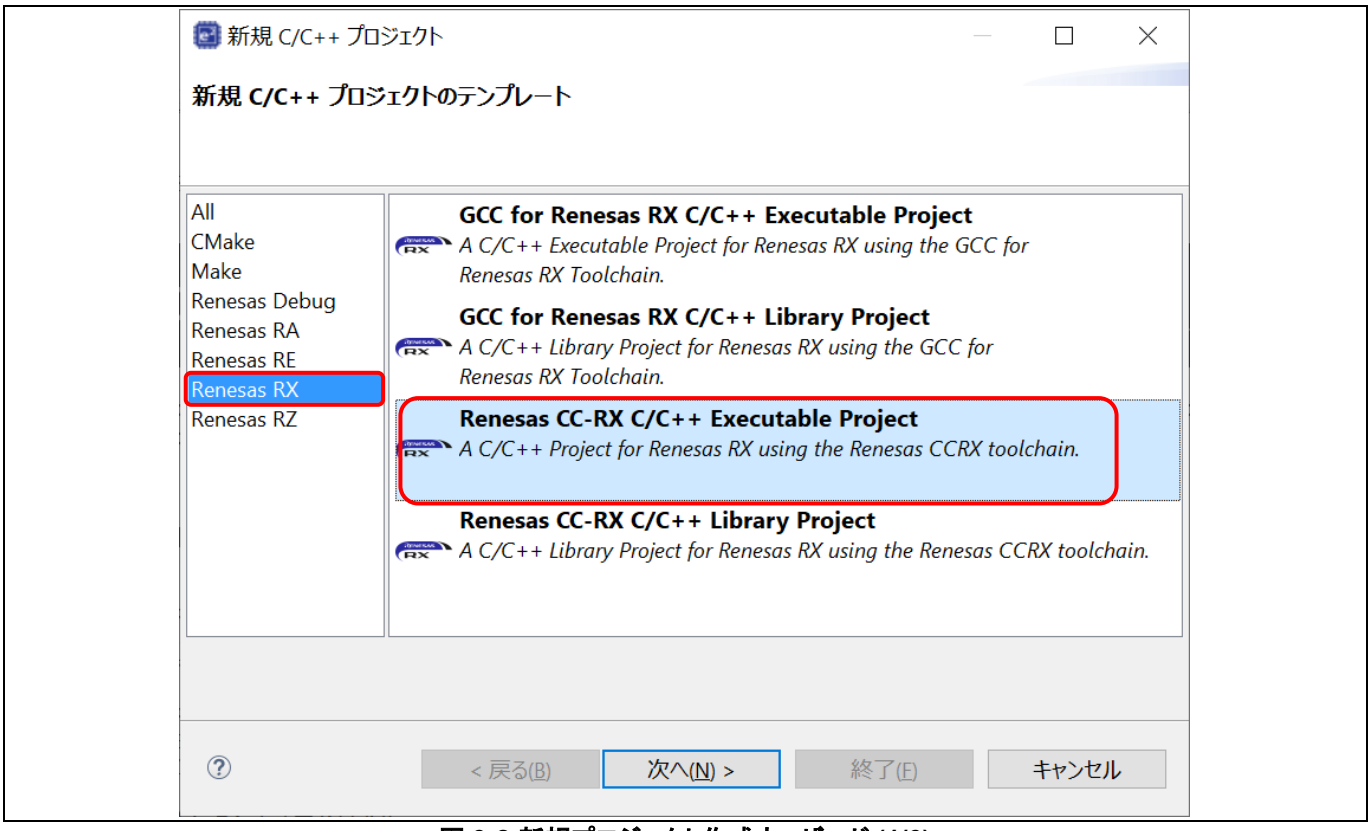

図 3-2 新規プロジェクト作成ウィザード (1/6)

プロジェクト名を入力し、[次へ(N)>] で進めます。

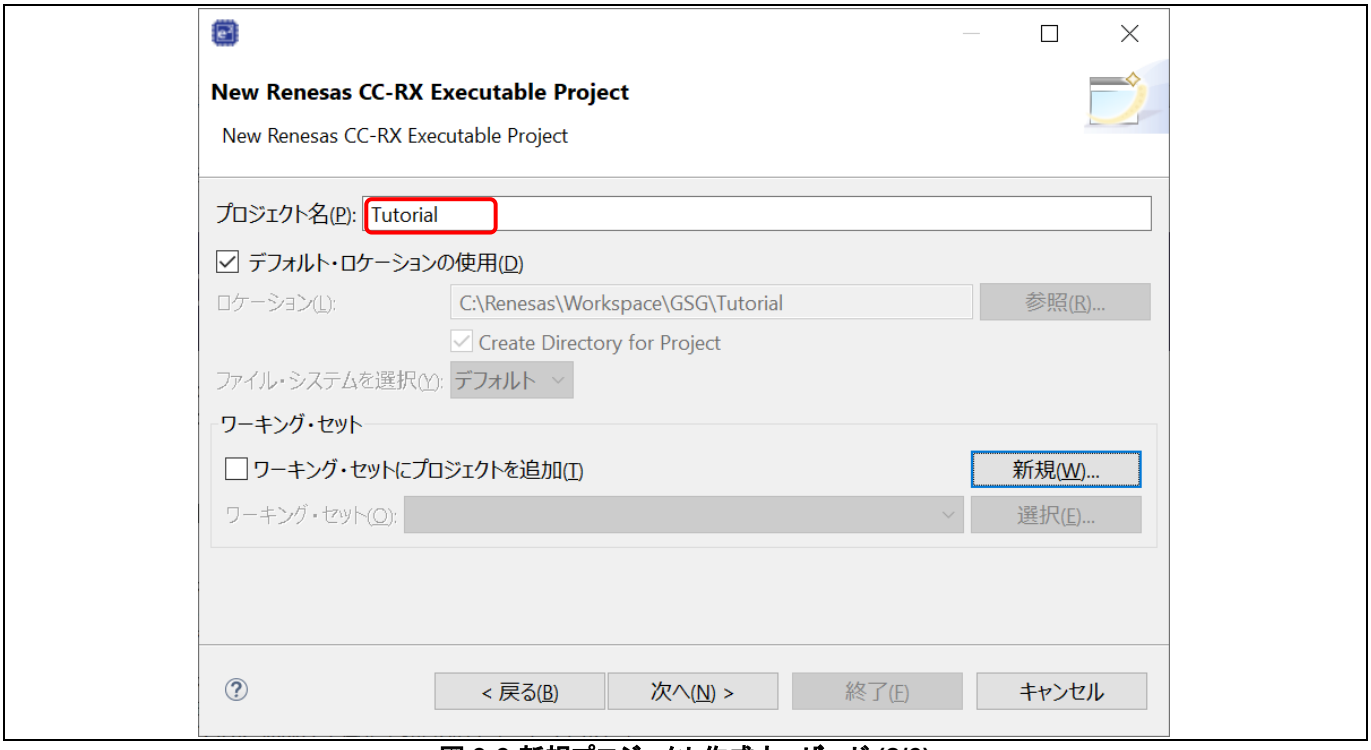

図 3-3 新規プロジェクト作成ウィザード (2/6)

[言語]、[ツールチェーン] [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。
[次へ(N)>] ボタンで次の画面に進みます。

注 1: E1 と同様に”E2”や“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す。

注 2:「Configurations」欄は必要なものだけを選択してください。複数のビルド構成を作成するとビルドオプションの設定も複数組作成されます。

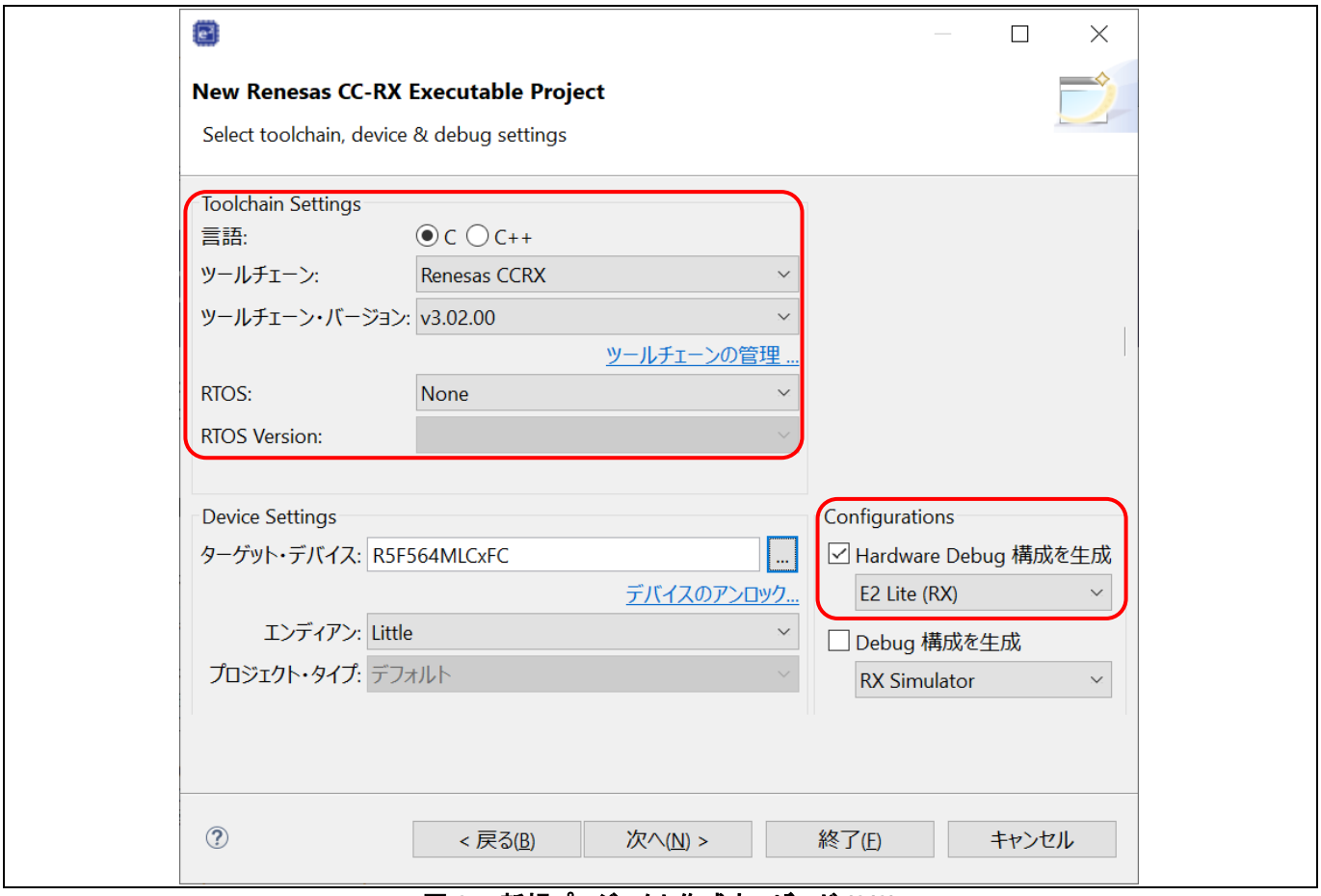

図 3-4 新規プロジェクト作成ウィザード (3/6)

コーディング支援機能を必要に応じて選択できます。[次へ(N)>] ボタンで次の画面に進みます。

[コード生成] ― ドライバや周辺機能のコードを GUI 画面から API 関数として生成する機能です。レジスタの初期化コードや割り込みハンドラが生成されます。

[FIT] ― プロトコル制御や周辺機能を使ったサンプルアプリケーションなど、コード生成機能よりも上位層のコードを出力します。FIT モジュールは共通のインタフェースを持つので自由に入れ替えることができます。

[スマート・コンフィグレータ] ― コード生成と FIT コンフィグレータを1画面で統一的に扱えるようにしたインタフェースです。クロック設定、割り込み設定、ピン設定が全て含まれます。

注: コード生成、スマート・コンフィグレータのいずれが利用可能かはデバイスによって異なります。

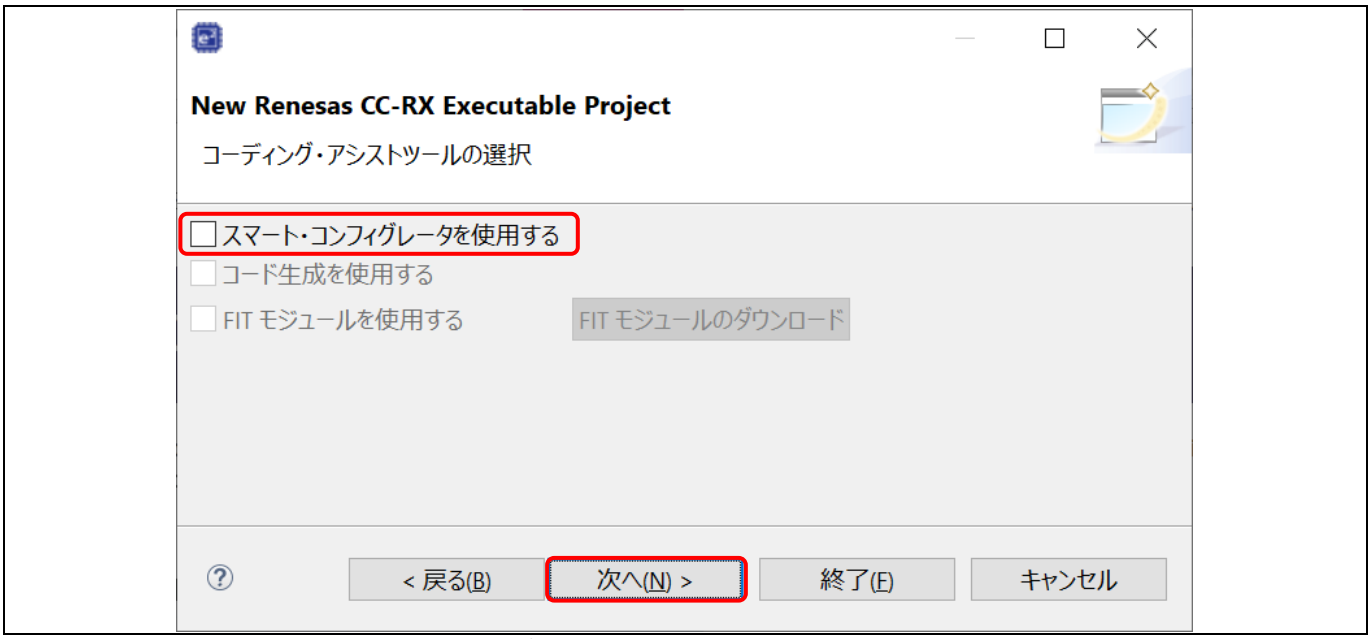

図 3-5 新規プロジェクト作成ウィザード(4/6)

(3) ここでは “Renesas デバッグ仮想コンソールを使用する” にチェックを入れないまま [次へ]で進めてください。

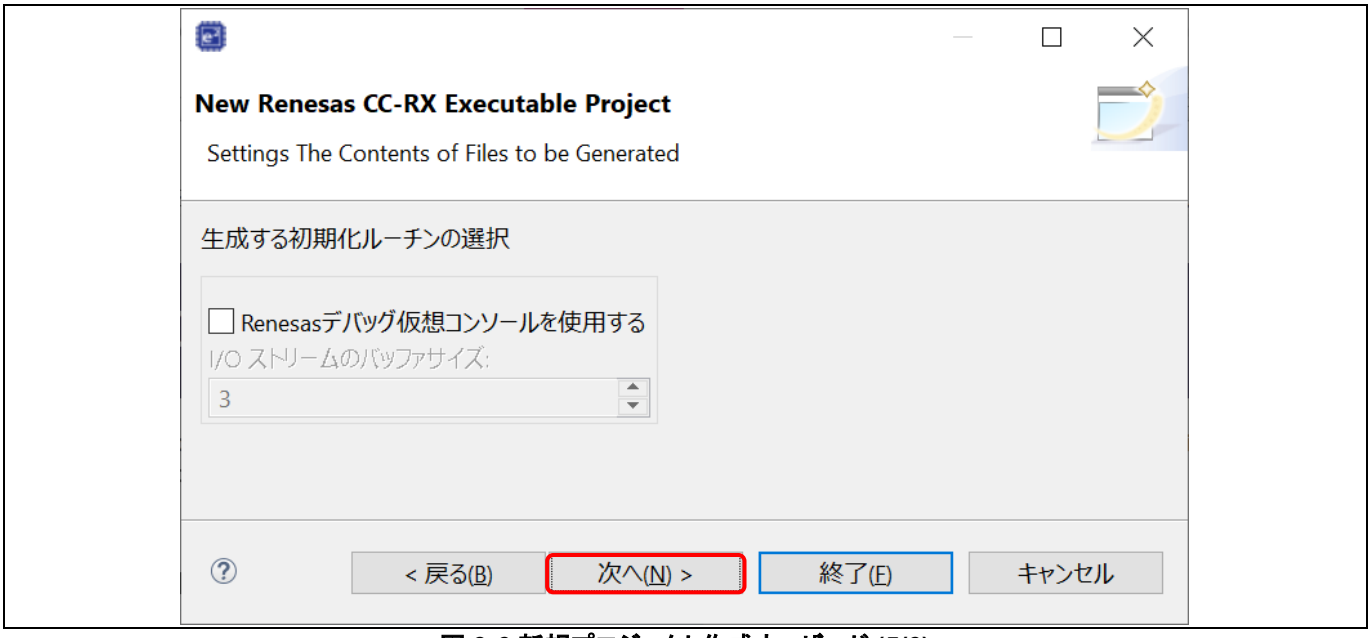

図 3-6 新規プロジェクト作成ウィザード (5/6)

(4) プロジェクトの概要が表示されます。最後に[終了(F)]ボタンを押すとプロジェクトの作成が完了します。

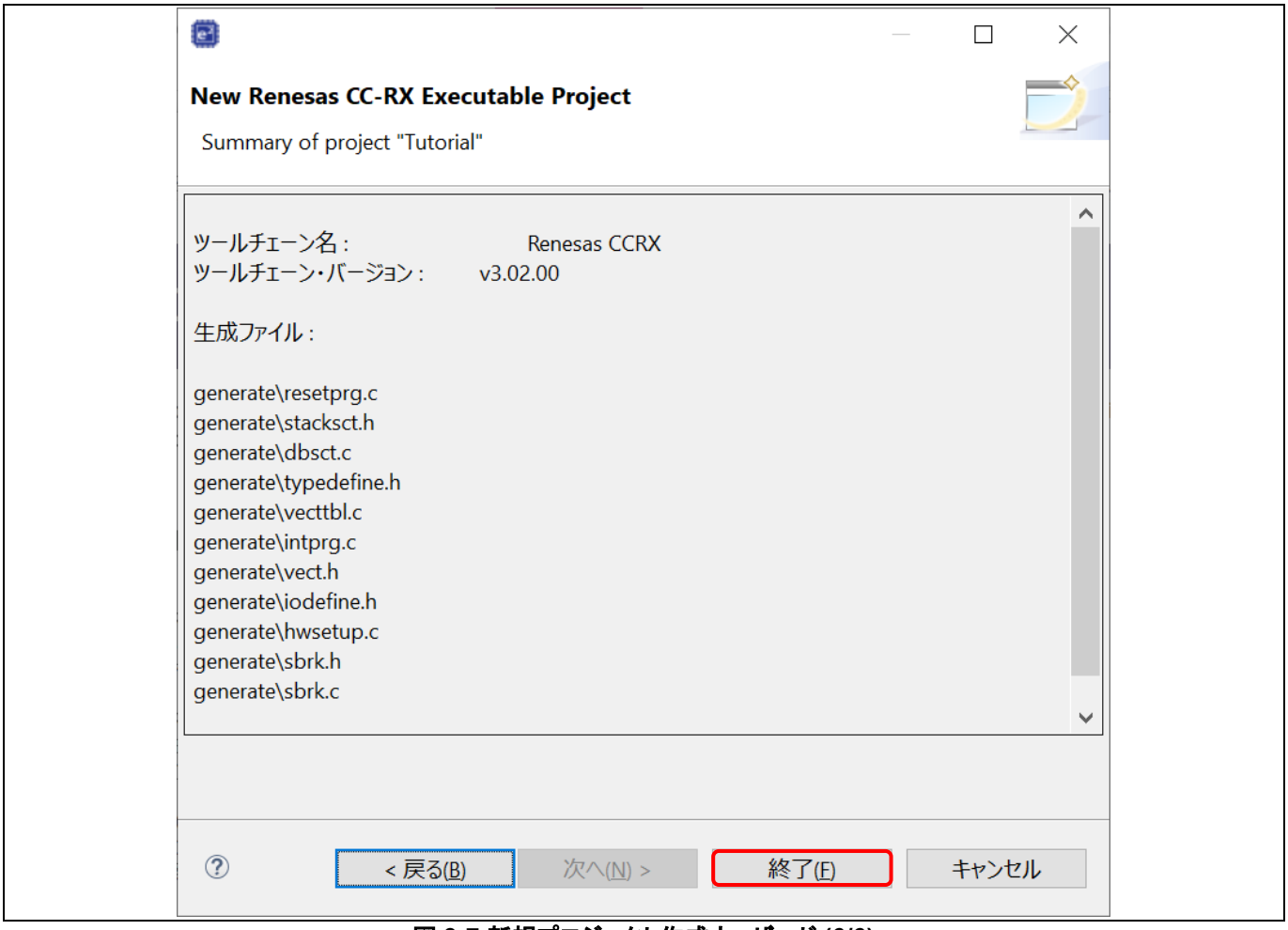

図 3-7 新規プロジェクト作成ウィザード (6/6)

(5) “Tutorial”の名前でプロジェクトが作成されました。

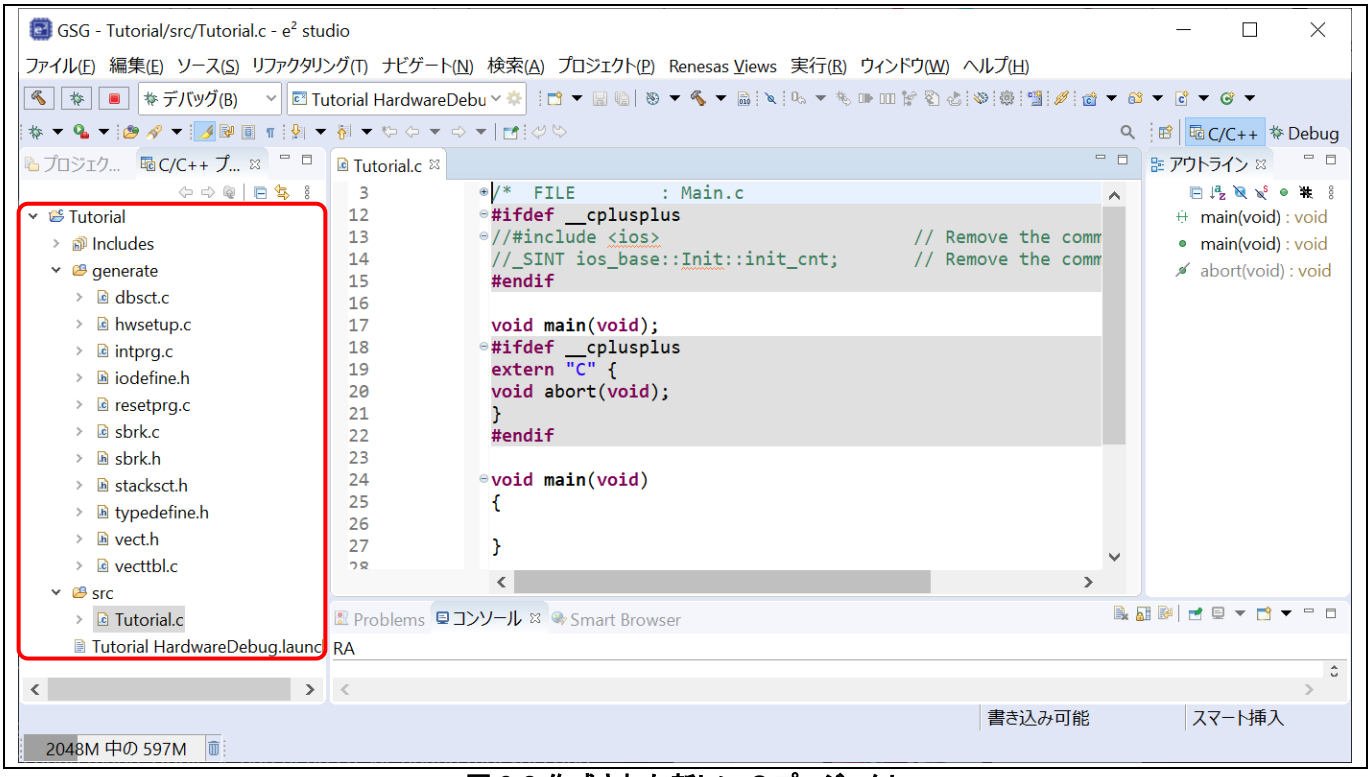

図 3-8 作成された新しい C プロジェクト

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, intprg.c, sbrk.cなど) から構成されています。[プロジェクト・エクスプローラー] 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エクスプローラー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(.project や.cproject)にはプロジェクトの設定情報が含まれますので、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も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 デバッグ専用プロジェクトの作成方法

デバッグ専用のプロジェクトを作成すると、他の環境で作成されたロードモジュールをe² studio上でデバッグすることができます。この機能によりデバッガ設定入りのプロジェクトが作成されます。

注: e² studioではIAR Systems社のIAR Embedded Workbench やGreen Hills SoftwareのMULTIで作成したELF/DWARF形式のロードモジュールを利用できます。

以下の手順でデバッグ専用プロジェクトが作成できます。

(1) 「ファイル」メニューの「新規(N)」→「C/C++ Project」を選択し、プロジェクト作成ウィザードを起動します。

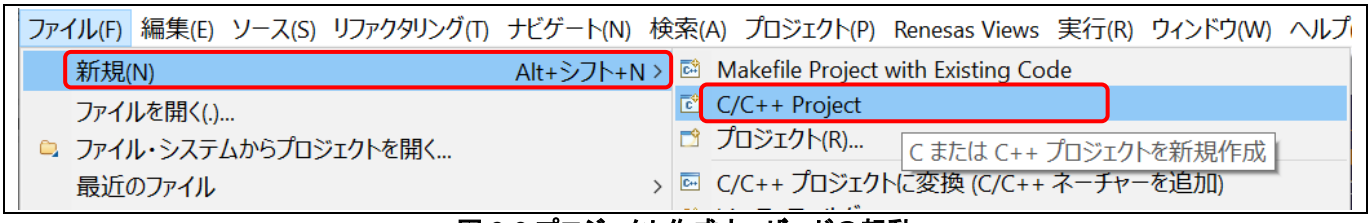

図 3-9 プロジェクト作成ウィザードの起動

(2) 新たに作成するプロジェクトのテンプレートとして、[Renesas Debug]カテゴリ内の"Renesas Debug Only Project"を選択して[次へ(N)>]ボタンで進めます。

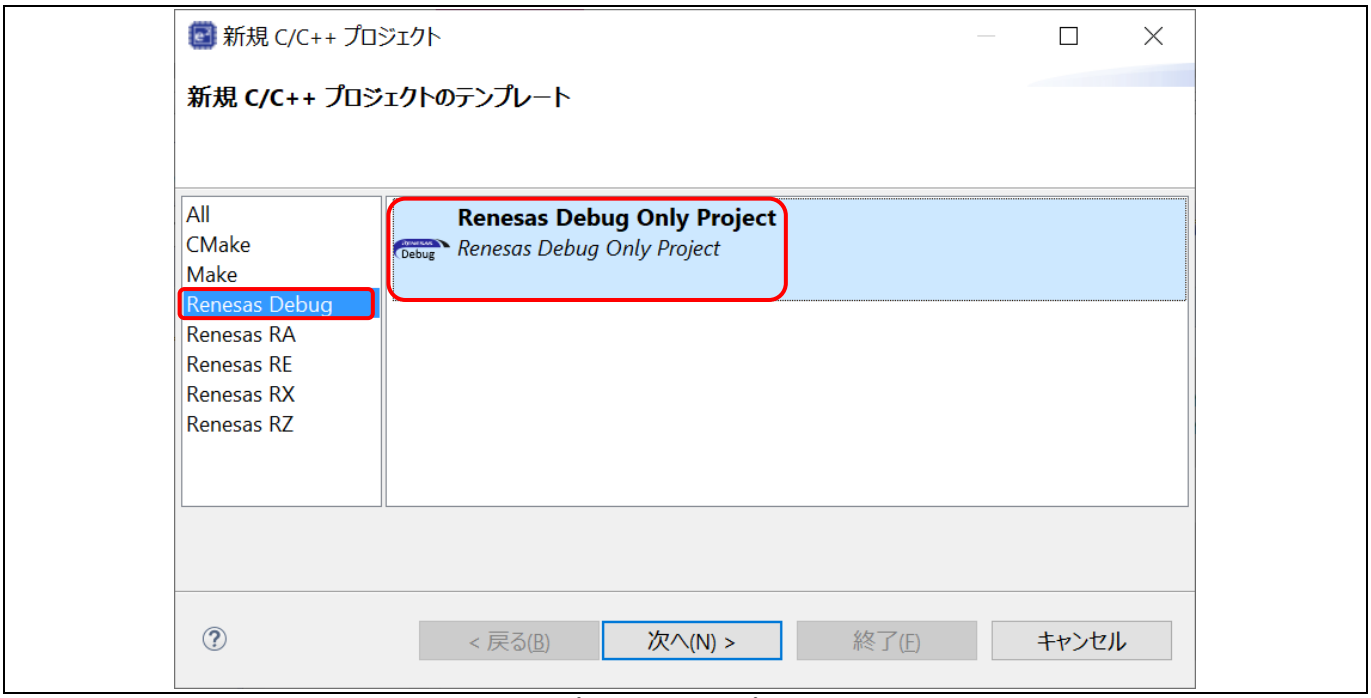

図 3-10 プロジェクトテンプレートの選択

(3) プロジェクト名を入力し、[次へ(N)>]ボタンで進めます。

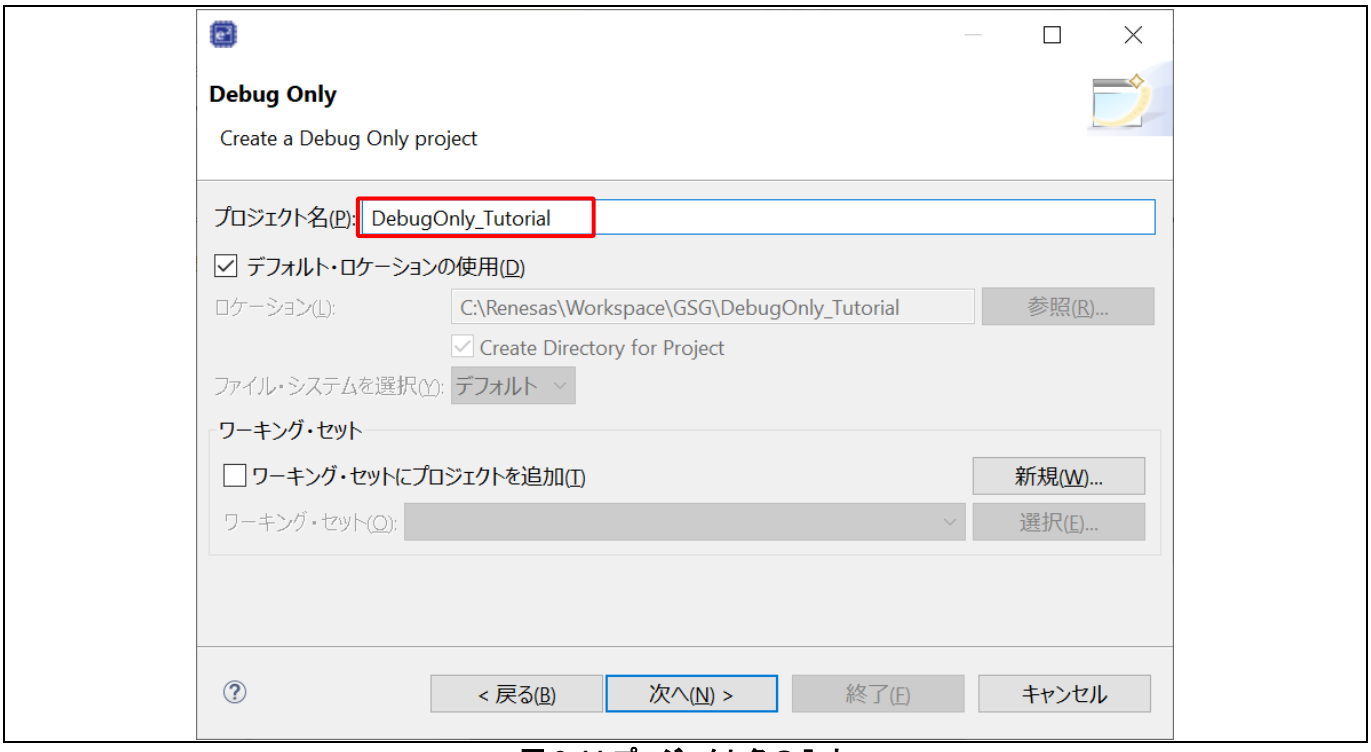

図 3-11 プロジェクト名の入力

(4) エミュレータ等のデバッグハードウェア、デバッグ対象のデバイス名を指定します(下図の例では"E1 (RH850)"と"R7F701002xAFP")。ここで指定するデバイス名はロードモジュールの作成時に指定したデバイスと矛盾しないものを選択してください。Executable Path 欄右の [...]ボタンでロードモジュール(他の開発環境で作成されたELF/DWARF 形式の実行ファイル)の置かれたパスを指定します。[終了(F)]を押すとプロジェクトが作成されます。

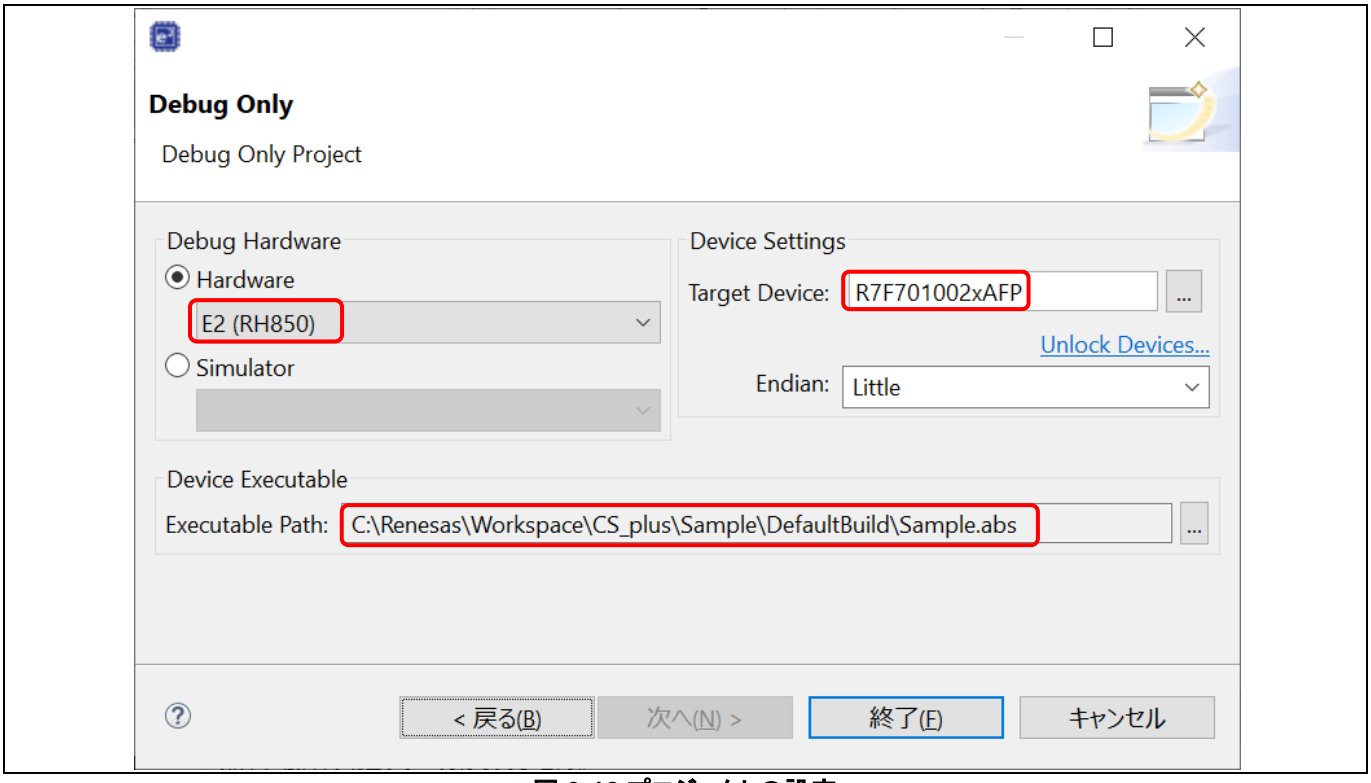

図 3-12 プロジェクトの設定

(5) 以下のようにプロジェクト名 "DebugOnly_Tutorial" が作成されました。デバッグ専用プロジェクトはデバッガの設定変更とデバッガの実行のみができるようになっています。

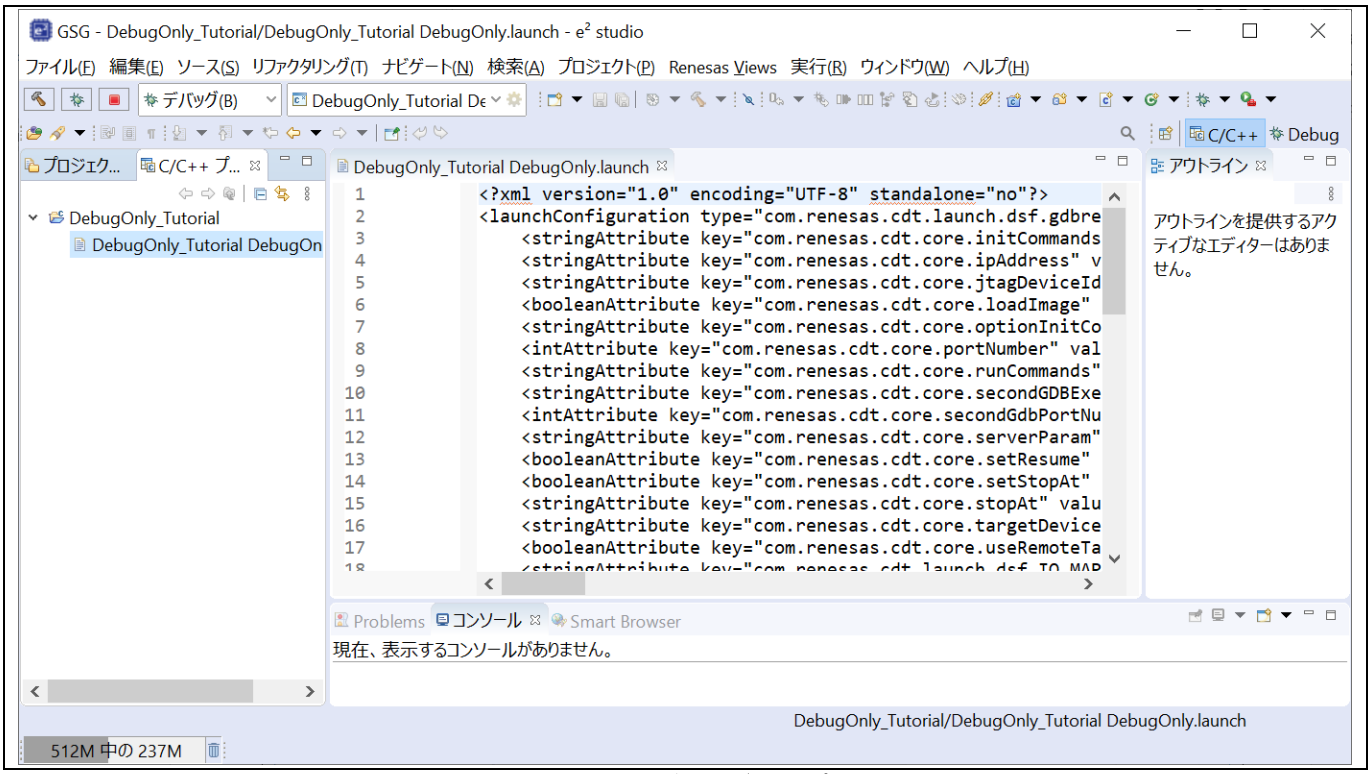

図 3-13 作成されたデバッグ専用プロジェクト

3.3 ワークスペースへの既存プロジェクトのインポート

各 IDE 間のプロジェクト移行方法(CS+/HEW からのインポートや CS+へのエクスポート)は以下のページで詳しく解説されています。
統合開発環境の移行方法:
https://www.renesas.com/products/software-tools/tools/migration-tools/migration-tools-ide.html

作成済の e²studio プロジェクトを現在のワークスペースにインポートできます。ここでは Renesas Web サイトからダウンロードしたサンプルプロジェクトをインポートしてデバッグ(5.4 章参照)するまでの手順を示します。

(1) Renesas Web サイト(https://www.renesas.com/search/keyword-search.html)でサンプルプロジェクトを検索します。(下図の例では “RX65N Starter Kit Sample Code”をキーワード検索した例です)

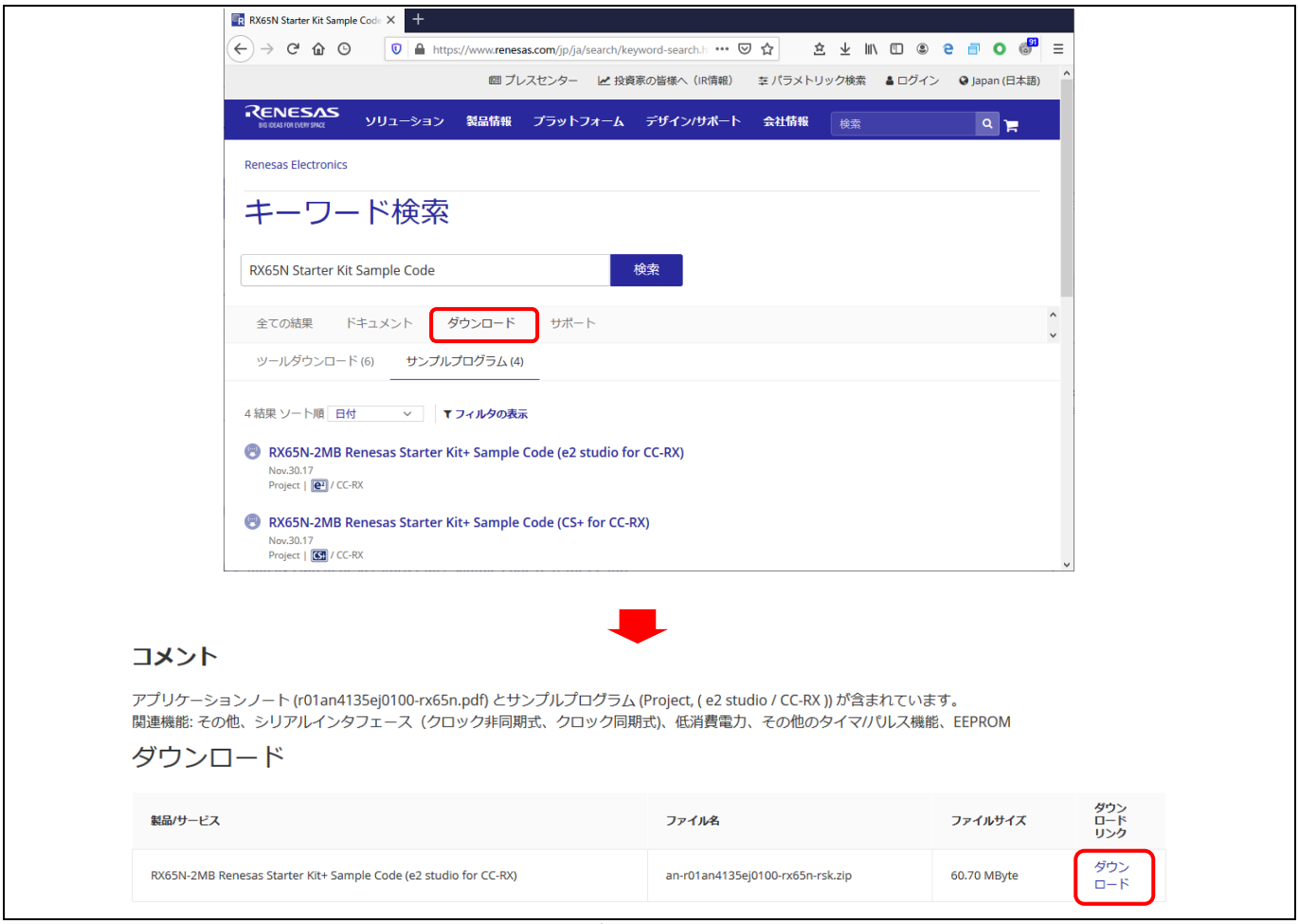

図 3-14 サンプルコードのダウンロード

(2) 以下はダウンロードした圧縮ファイルを解凍した例です。“Tutorial”プロジェクトが見えます。

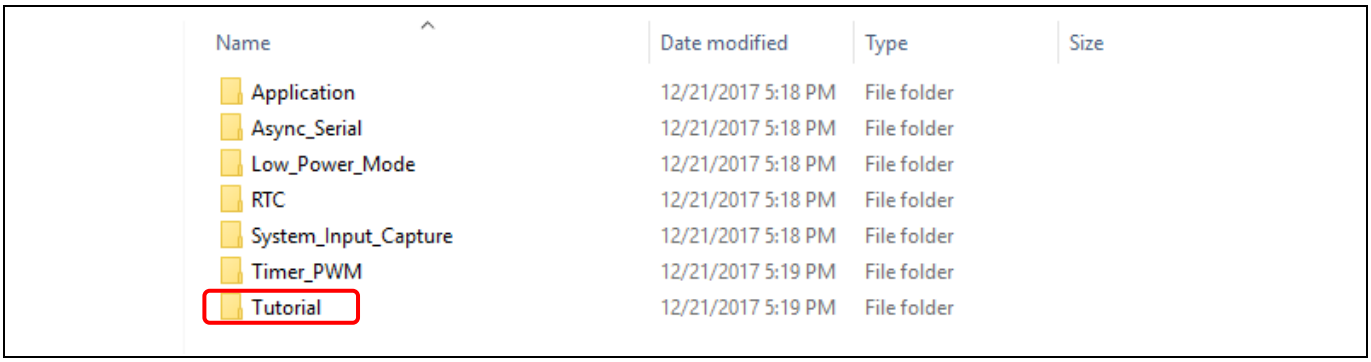

Figure 3-15 The Sample project

(3) e² studio の [ファイル(F)] → [インポート(I)...]を選びます。

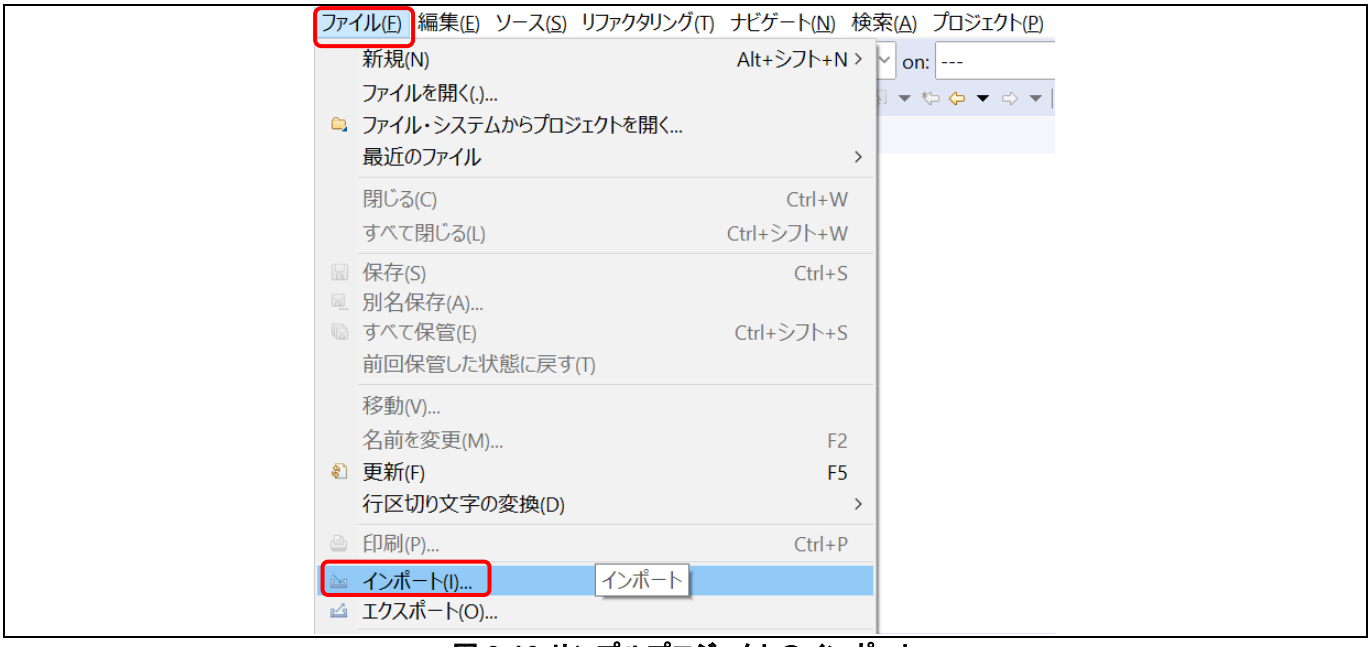

図 3-16 サンプルプロジェクトのインポート

(4) [インポート] ダイアログの [一般] → [既存プロジェクトをワークスペースへ]を選んで [次へ(N)]で進めます。

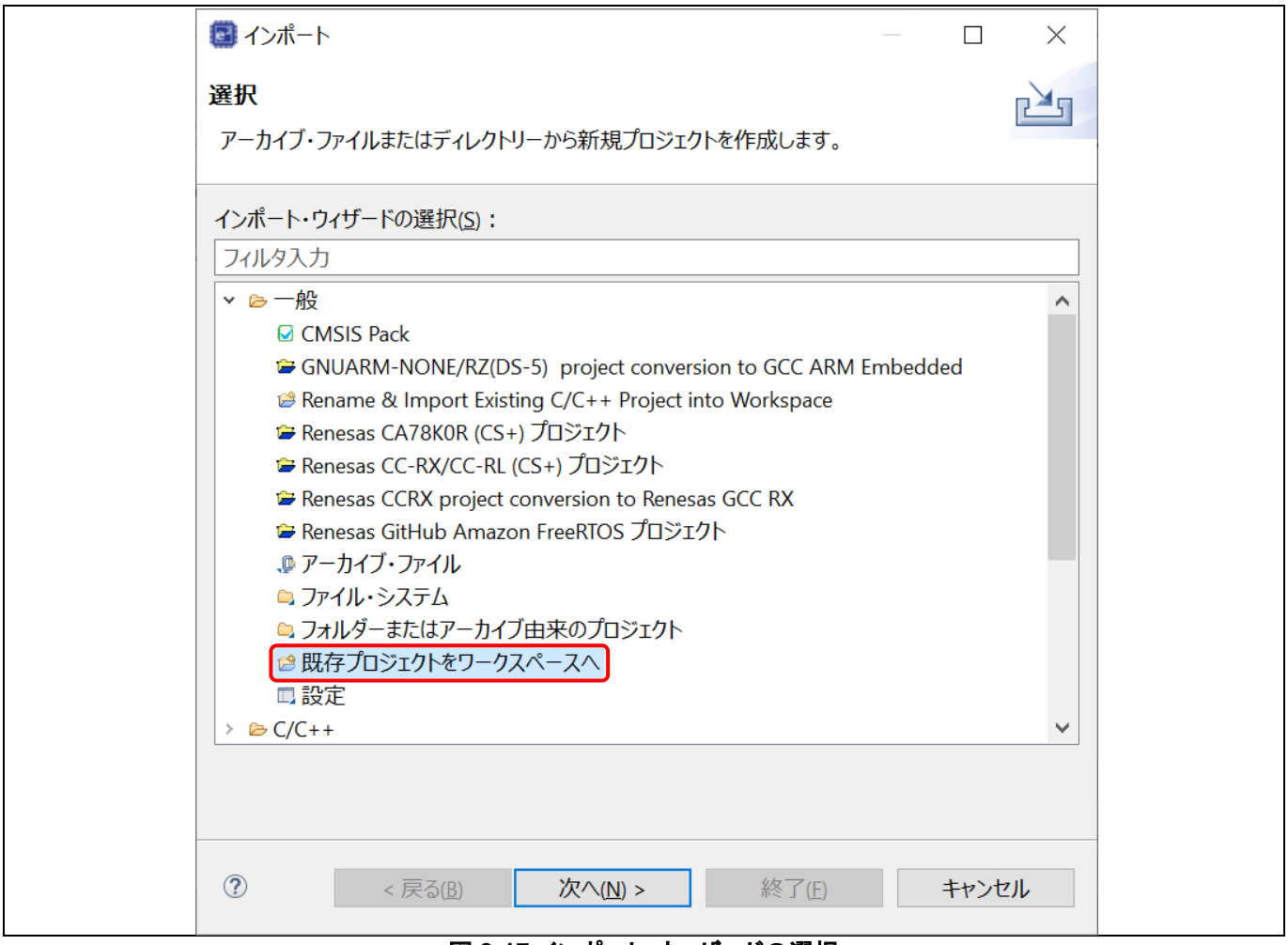

図 3-17 インポート・ウィザードの選択

(5) [プロジェクトのインポート]画面で “ルート・ディレクトリーの選択”を選び、右の [参照...]ボタンでサンプルプロジェクトを解凍したフォルダを選択します。

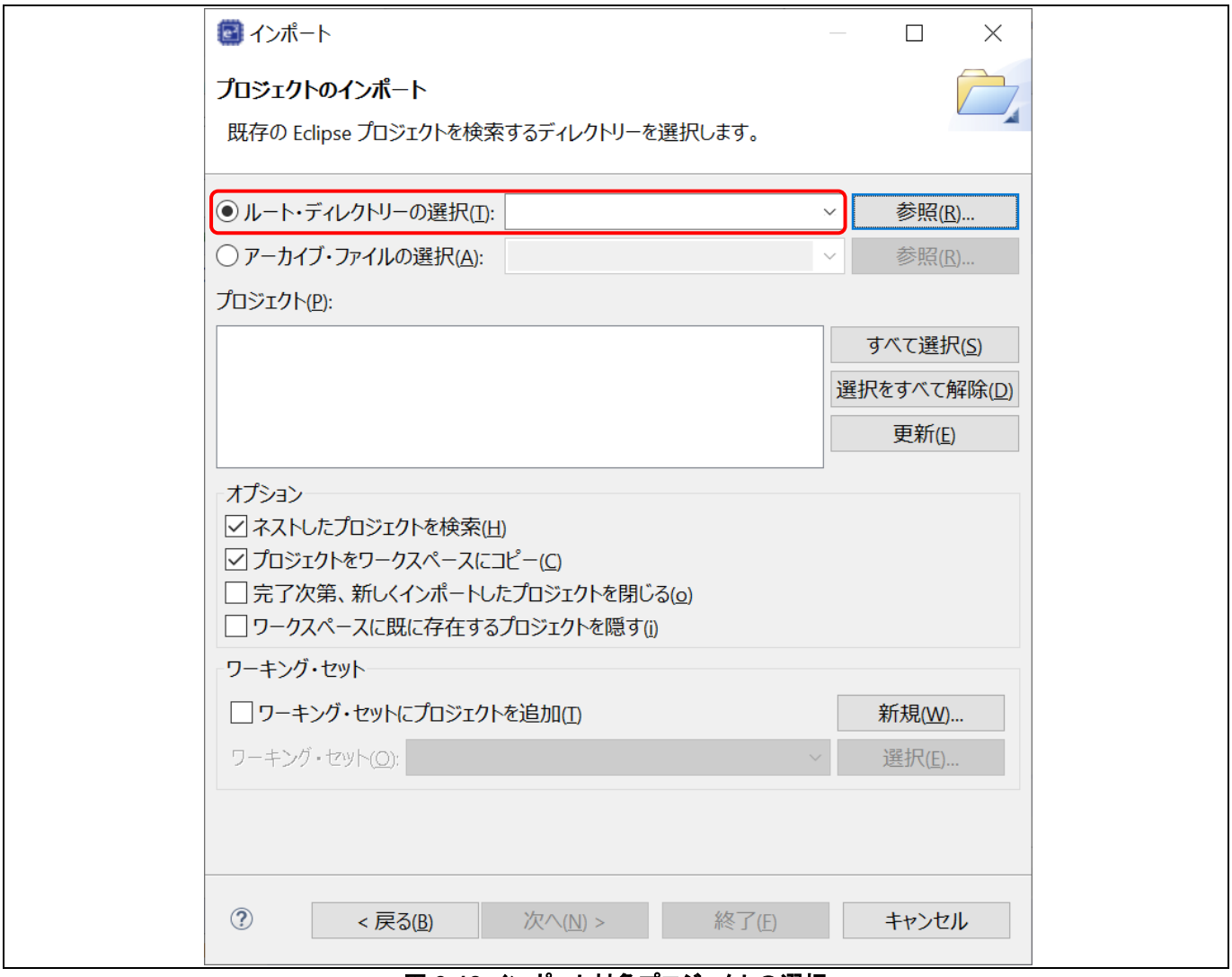

図 3-18 インポート対象プロジェクトの選択

(6) 表示された中から “Tutorial” プロジェクトを選択し、更に「プロジェクトをワークスペースにコピー」にチェックを入れてから [終了(F)]でプロジェクトがインポートされます。

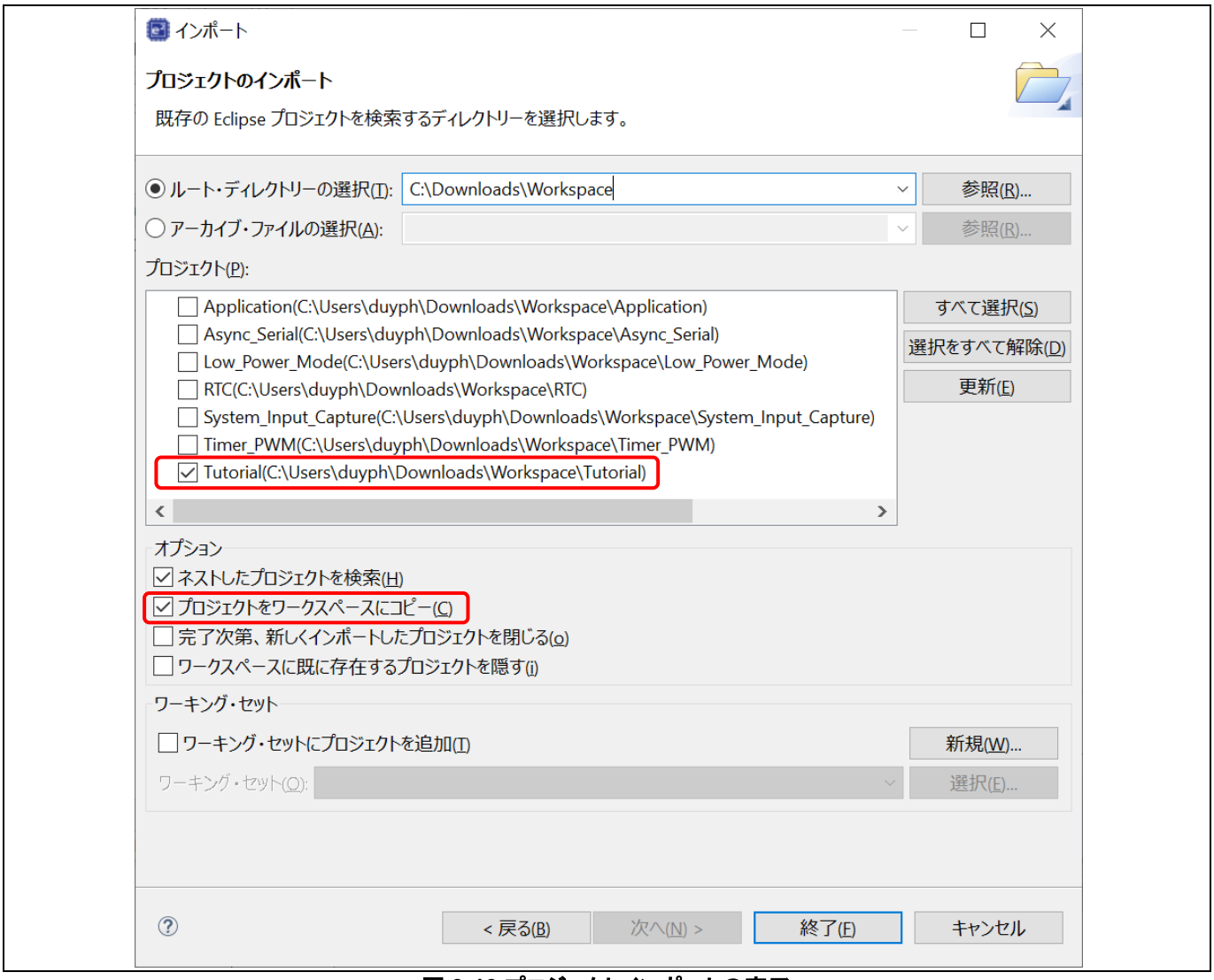

図 3-19 プロジェクトインポートの完了

(7) 旧い形式のプロジェクトであればプロジェクトを右クリックして、「古い e2studio プロジェクトを更新...」のメニューが表示されていますのでこれを実行します。表示されていなければステップ 9 に進んでください。

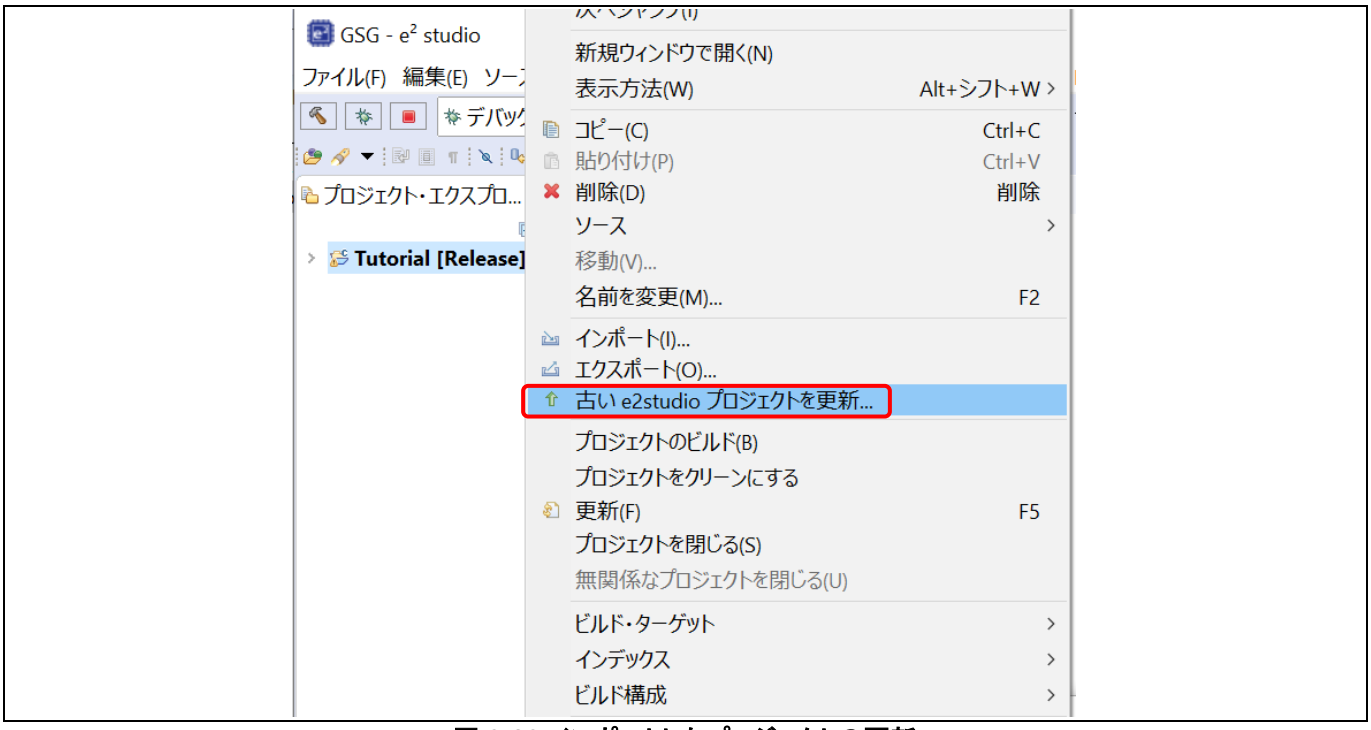

図 3-20 インポートしたプロジェクトの更新

(8) 更新するプロジェクト(ここでは“Tutorial”)を選択して [終了(F)]でプロジェクトが更新されます。

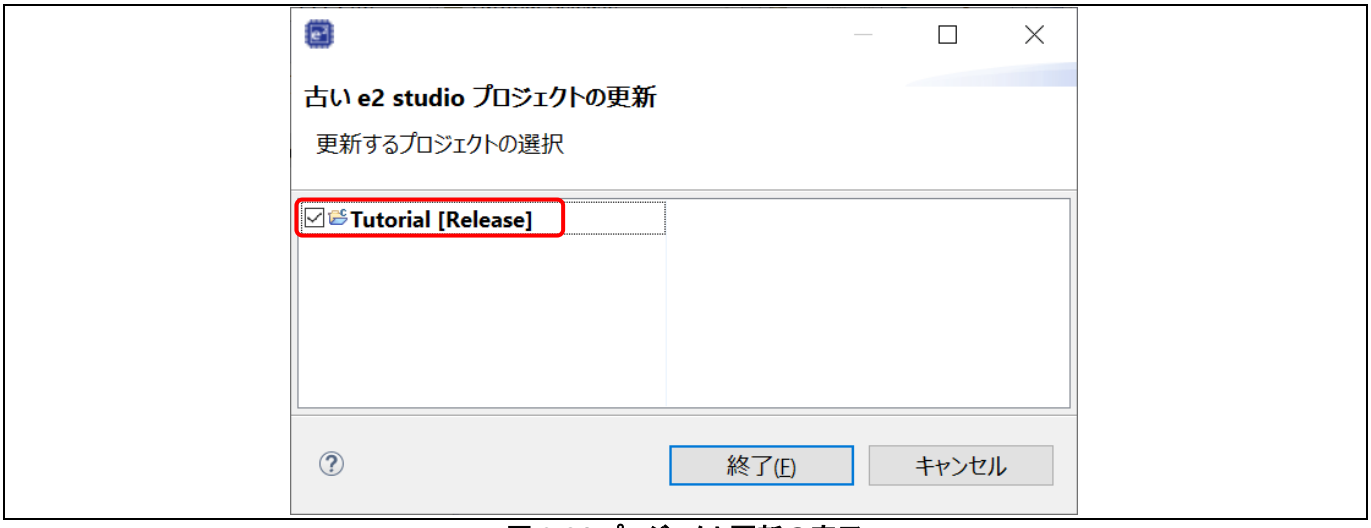

図 3-21 プロジェクト更新の完了

(9) プロジェクトのプロパティ(プロジェクト名の上で右クリック→[プロパティ(P)])ダイアログを開き、左ペインの[C/C++ ビルド] → [設定]を選択します。[Toolchain]タブでツールチェーンの種類とバージョンを確認してください。もし空欄であればいずれかを選択してください。[適用して閉じる]で設定を終わります。

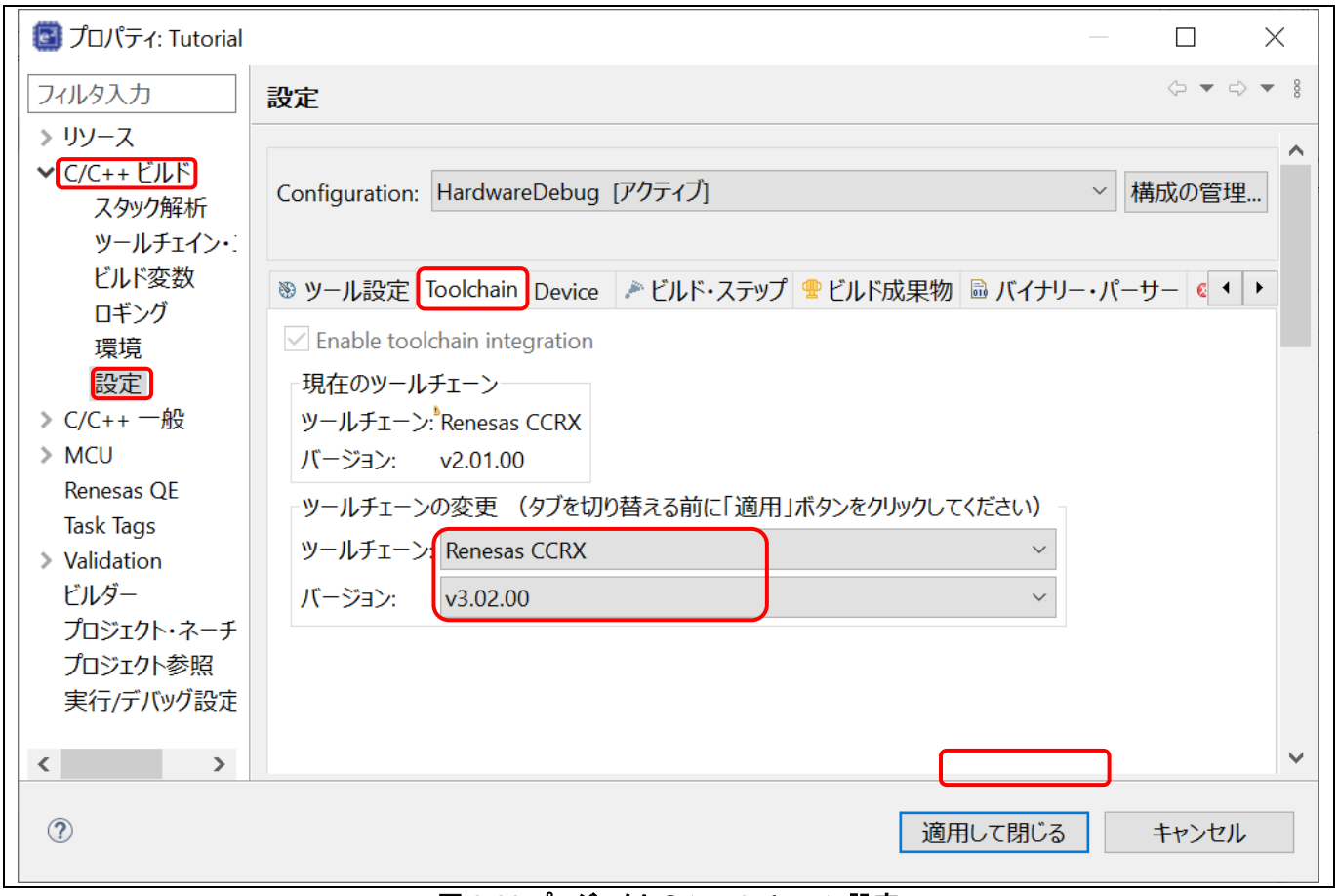

図 3-22 プロジェクトのツールチェーン設定

(10)プロジェクトのビルドが正常に終わるかを確認してください。

4. ビルド

この章ではe² studioのビルドに関する設定や主なビルド機能について解説します。

4.1 ビルドオプションの設定

デフォルトのオプションで新規に作成されたプロジェクトは正常にビルドできます。ビルドオプション(toolchainのバージョンや、最適化オプション)を変更したい場合はビルドする前に以下の手順を実行してください。

(1) プロジェクト・エクスプローラーのプロジェクト名を右クリックし[プロパティ(R)]を選択するか、  ボタンでプロパティ画面を開きます。

ワークスペースのプロパティ画面はプロジェクトおよびソースファイルの単位で設定できます。プロジェクト単位のプロパティ画面ではそのプロジェクト内の全てのファイルに対して共通の設定を持つことができます。

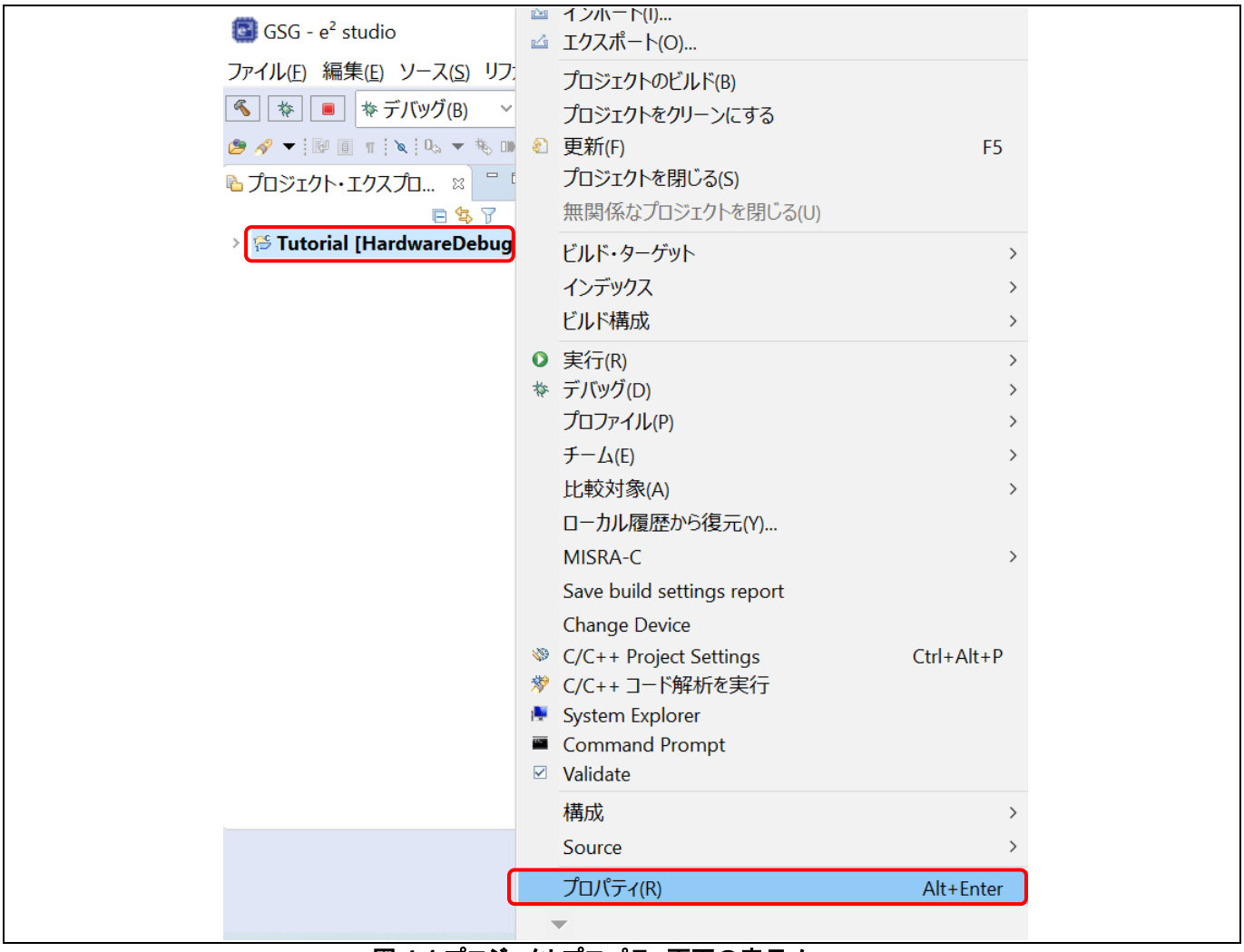

図 4-1 プロジェクトプロパティ画面の表示メニュー

(2) 左カラムの[C/C++ ビルド]→[設定]を選択し、[Toolchain]タブをクリックするとツールチェーンバージョンの確認と変更ができます。

(2つ以上のバージョンが利用可能であれば)[バージョン]欄でツールチェーンバージョンが選択できます。

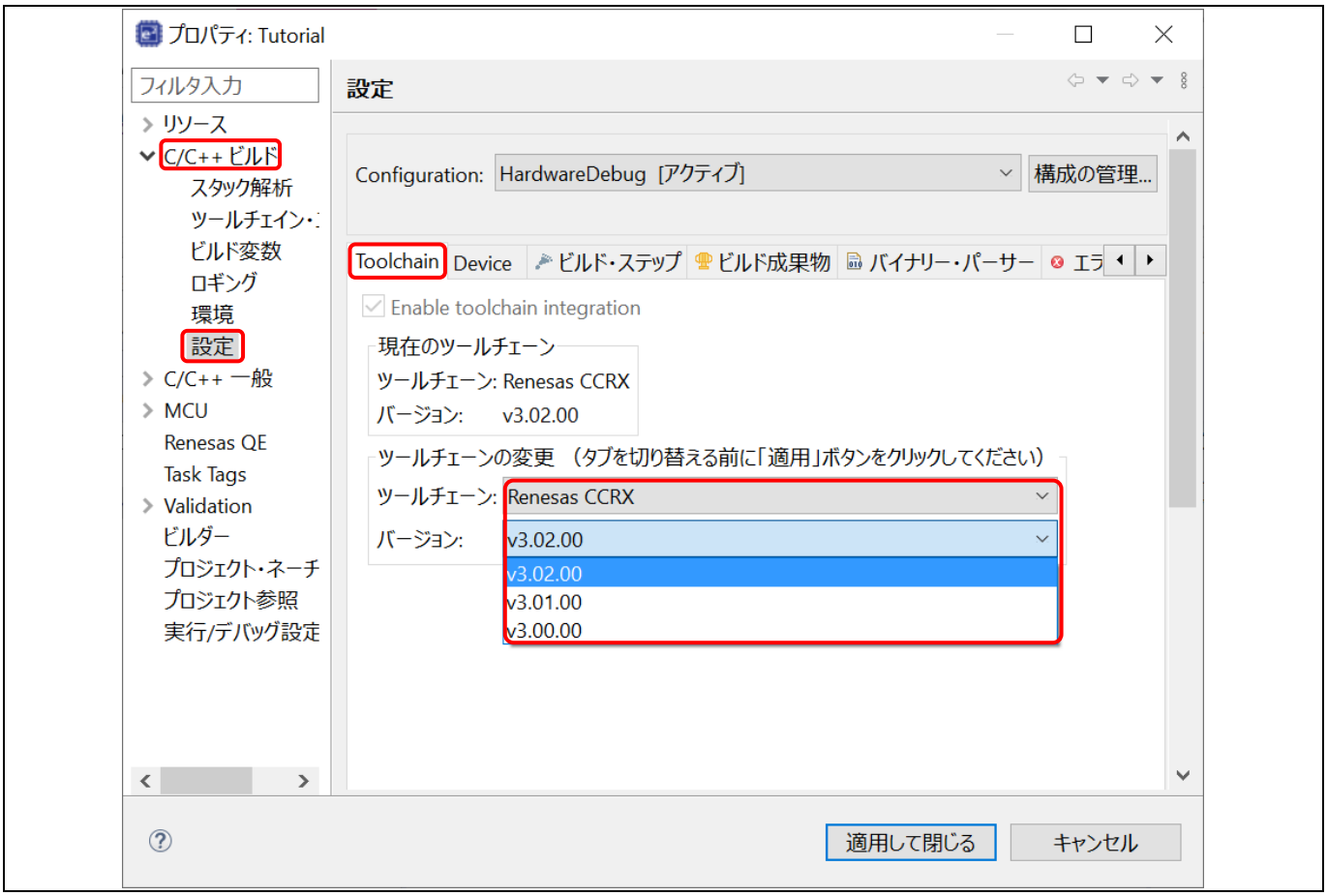

図 4-2 ツールチェーンバージョンの変更

(3) [C/C++ ビルド] → [環境] でビルドに使われる環境変数の追加や変更ができます。

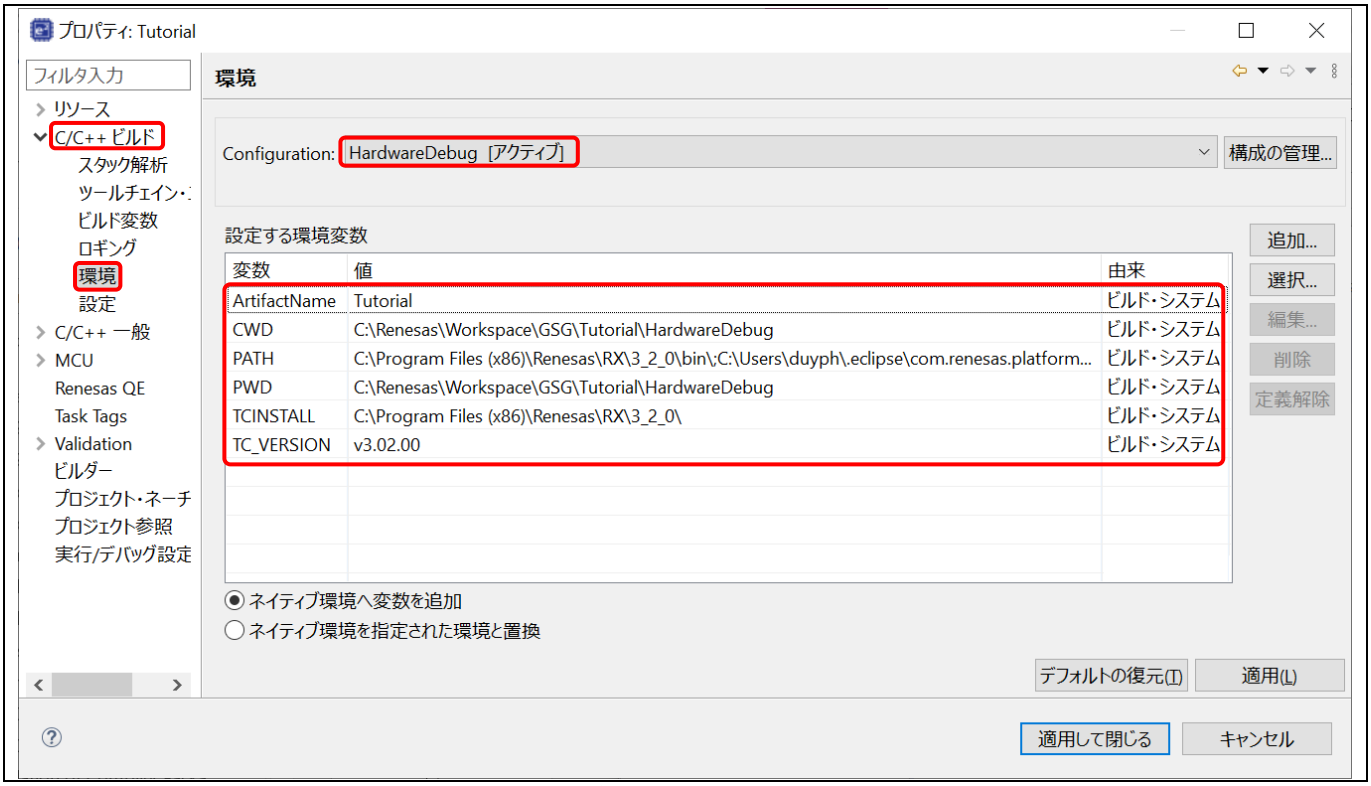

図 4-3 ビルド用環境変数の設定

(4) ビルドオプションの設定

プロジェクト・エクスプローラーのプロジェクト名を右クリック→プロパティか、ツールバーの ボタンでプロパティ画面が表示されます。

コンパイラ、リンカなどのビルドオプションは[C/C++ ビルド]→[設定]の[ツール設定]タブで設定できます。
設定可能なビルドオプションはすべてこのタブ内にあります。

画面上部のドロップダウンリスト「Configuration:」で「ビルド構成」を切り替えられます。ビルド構成とは、ビルドオプション一式をまとめて管理する単位です。

[適用して閉じる]で変更した設定内容が保存されます。

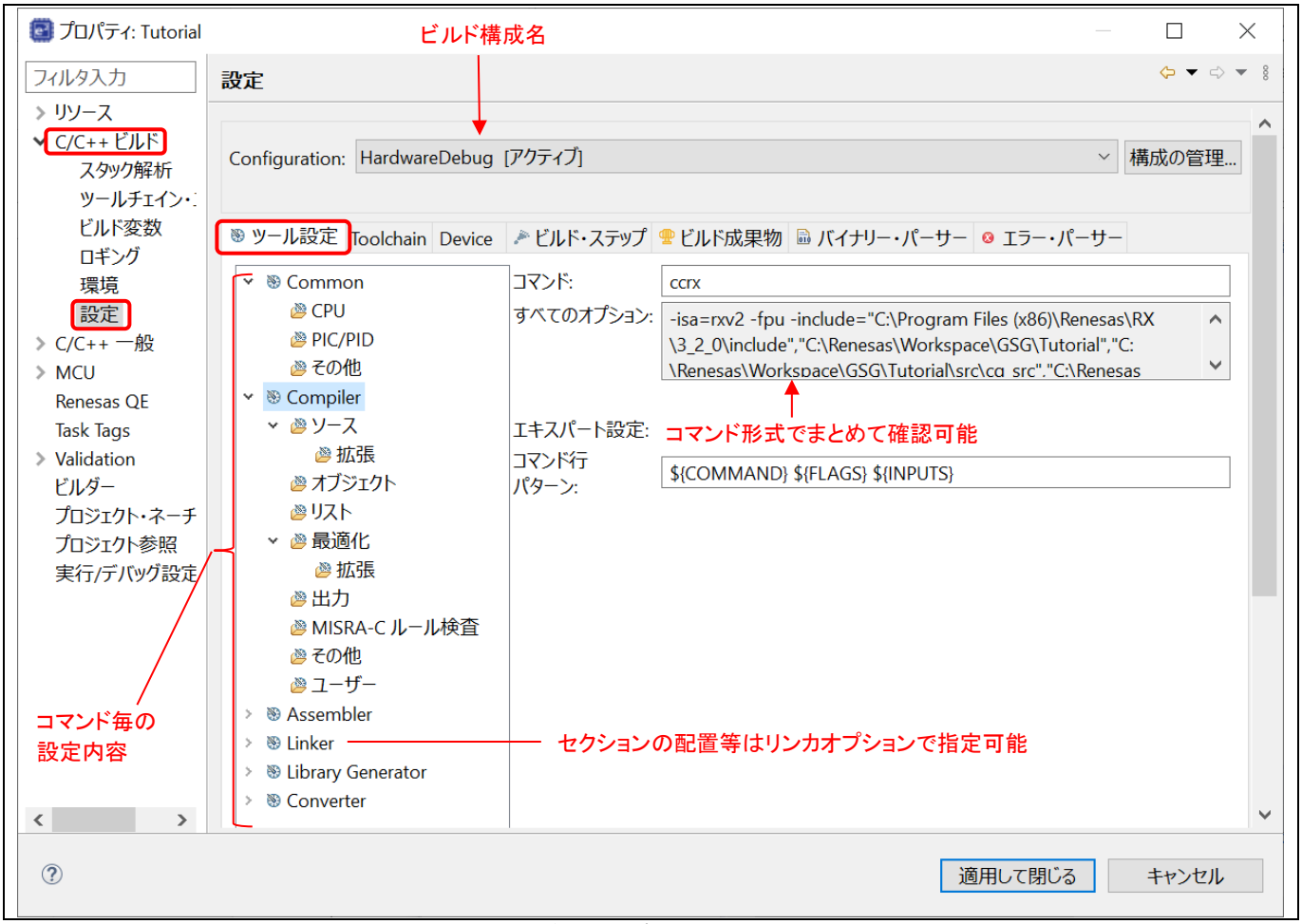

図 4-4 ビルドオプションの設定画面

ビルドオプションの詳細は、各コンパイラ製品のユーザーズマニュアル(ヘルプ)を参照してください。
CC-RX/CC-RLのIDEなし版の場合は、インストールディレクトリのdocフォルダ内にあります。
(例: `C:\Program Files (x86)\Renesas\RX\3_2_0\doc\`)

注: “C/C++ ビルド”→”**Tool chain エディター**”の設定は変更しないでください。この設定はRenesasのビルドサポートプラグインがサポートしていないツールチェーンで使われるべきものです。

4.2 サンプルプロジェクトのビルド

以下のような手順でプロジェクトをビルドできます。

(1) プロジェクトを右クリックして[プロジェクトのビルド(B)]を選択

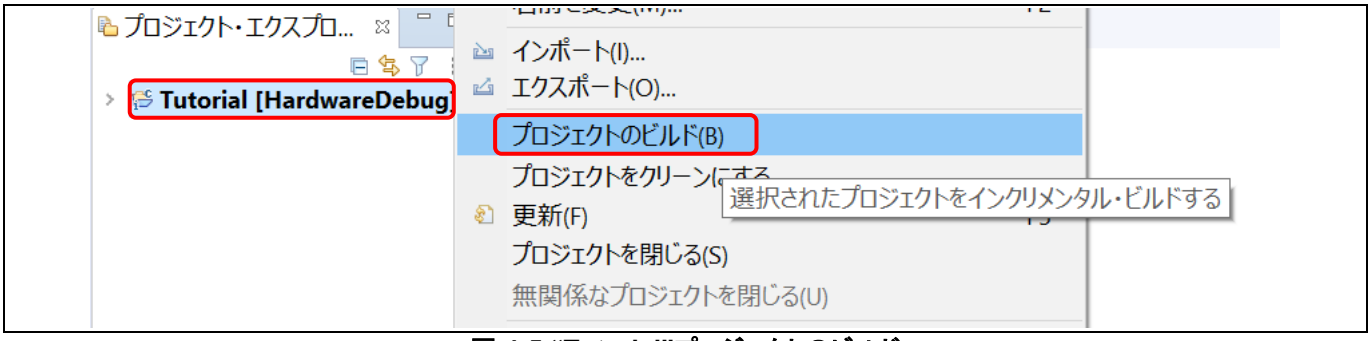

図 4-5 “Tutorial”プロジェクトのビルド

(2) 「コンソール」タブを確認してください。ビルドが完了していれば下記のようなメッセージが表示されます。

ビルドを実行した後には${CONFIGDIR} (ビルド構成名；上図では“HardwareDebug”)フォルダに “makefile”, “Tutorial.abs”, “Tutorial.map”, “Tutorial.mot”, “Tutorial.x” 等が出力されています。

“Tutorial.abs” (拡張子 *.abs) はRenesas標準のELF/DWARF形式のファイルですが、e² studio ではGDB (GNU Debugger) に対応するELF/DWARFロードモジュール形式(拡張子 *.xまたは*.elf)に変換したもの(ここでは “Tutorial.x”)をデバッグに使用します。

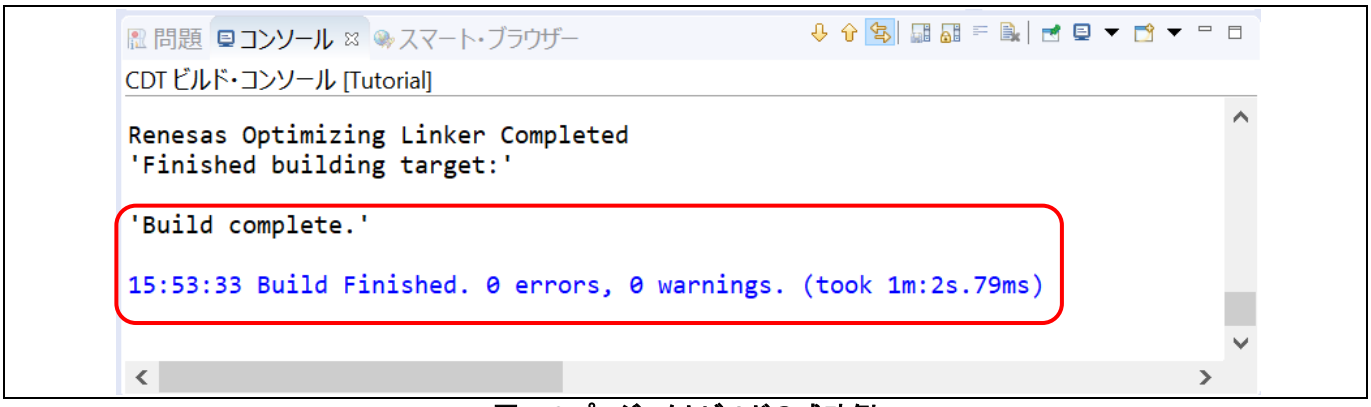

図 4-6 プロジェクトビルドの成功例

(3) ビルドに失敗する場合もあります。コンソールに表示されるエラーを確認し、ソースコードや設定内容を見直してからビルドをやり直してください。

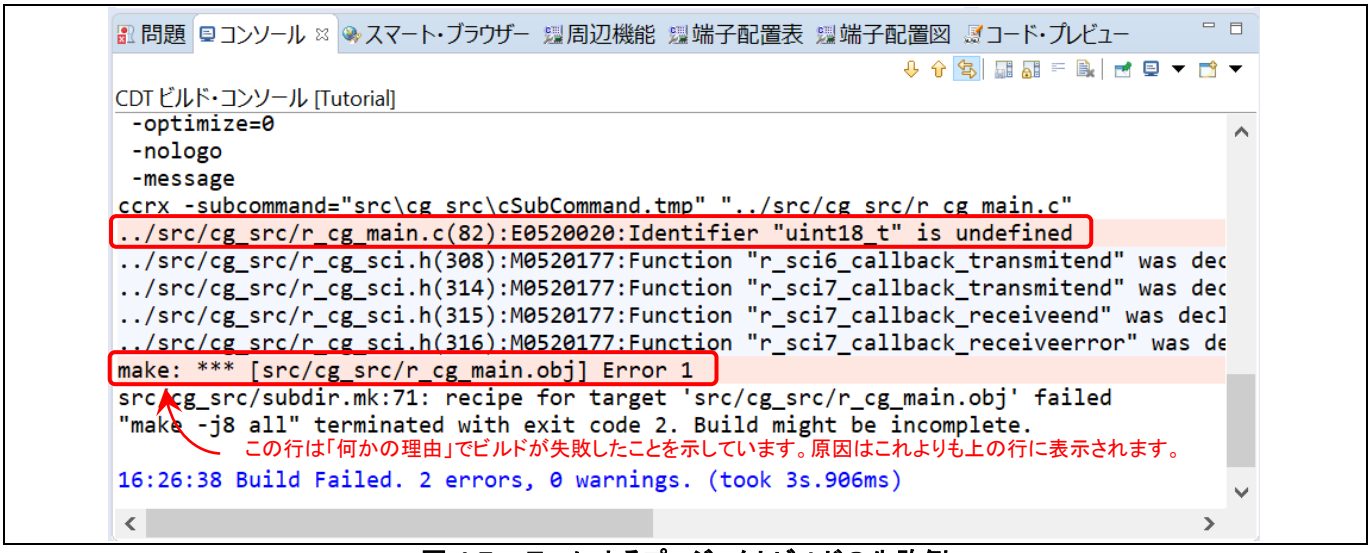

図 4-7 エラーによるプロジェクトビルドの失敗例

4.3 プロジェクトレポート機能 (ビルドオプション一覧の出力)

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。

(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クスト・メニューを開きます。

(2) [Save build setting report] を選択してファイル名を入力すると、レポートがファイルに保存されます。

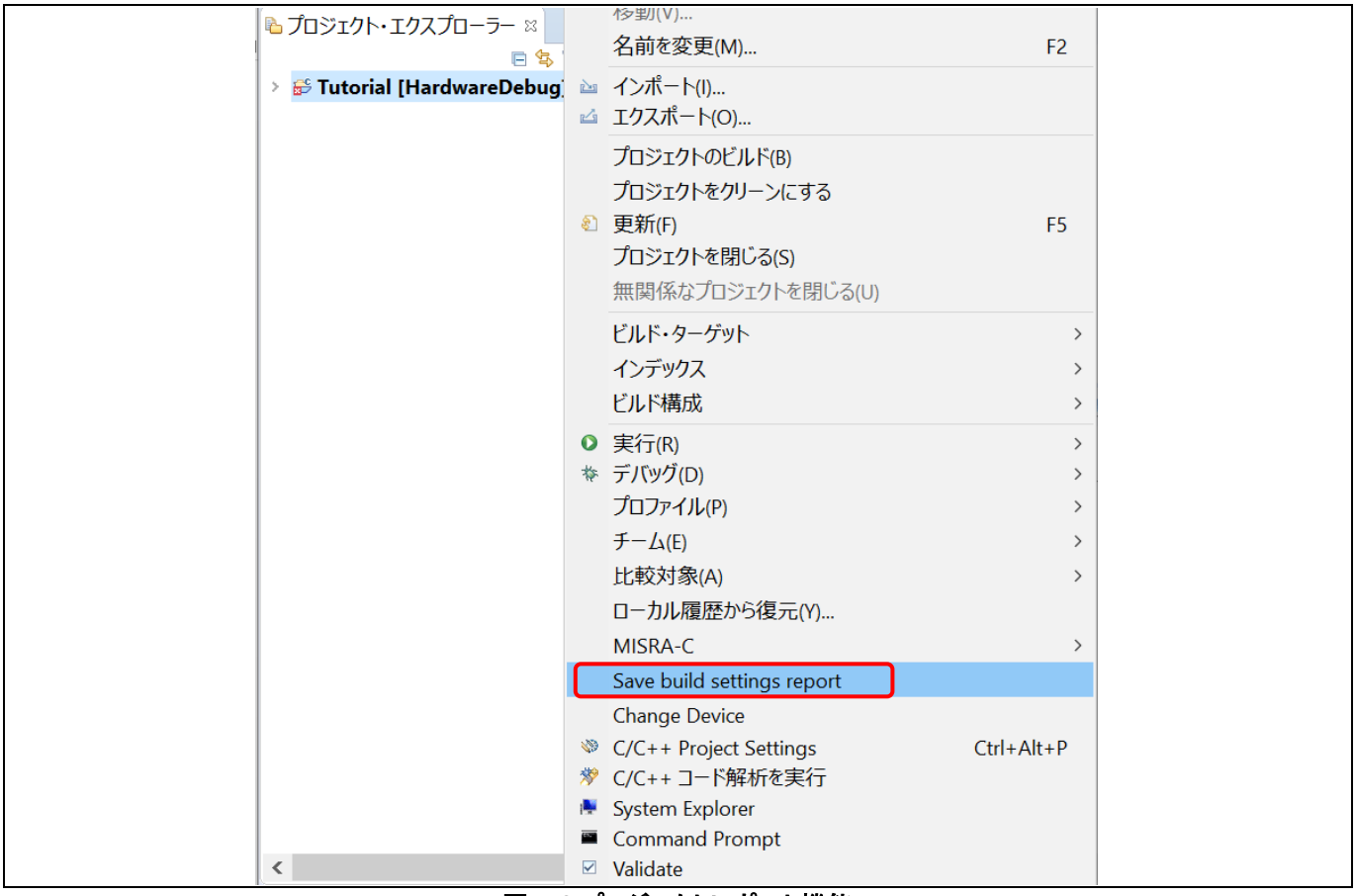

図 4-8 プロジェクトレポート機能

5. デバッグ

この章では、e² studio統合開発環境のデバッグ構成(Debugger Launch Configuration)と主要なデバッグ機能の使い方について説明します。以下では、EシリーズエミュレータとRSK RX64Mボードを動作環境とする“Tutorial” プロジェクト(4章参照)を例に説明します。

e² studioで “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。

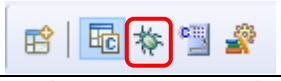

図 5-1 [デバッグ] パースペクティブへの切り替え

パースペクティブでワークベンチのウィンドウのレイアウト表示(開発ツール関連)を定義します。それぞれのパースペクティブは、特定の用途に向いたビュー、メニュー、ツールバーの組み合わせで構成されます。

例えば、[C/C++] パースペクティブは、ユーザがC/C++プログラム開発を行うために必要なビューを持ち、[デバッグ] パースペクティブは、プログラムのデバッグに必要なビュー表示をすることができます。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。

一つのワークベンチのウィンドウのなかに、一つまたは複数のパースペクティブを表示することができます。ユーザはそれらをカスタマイズし、新しいパースペクティブを追加することができます。

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、一度だけデバッグ構成を設定する必要があります。既存のデバッグ構成は以下のように変更できます。

(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。

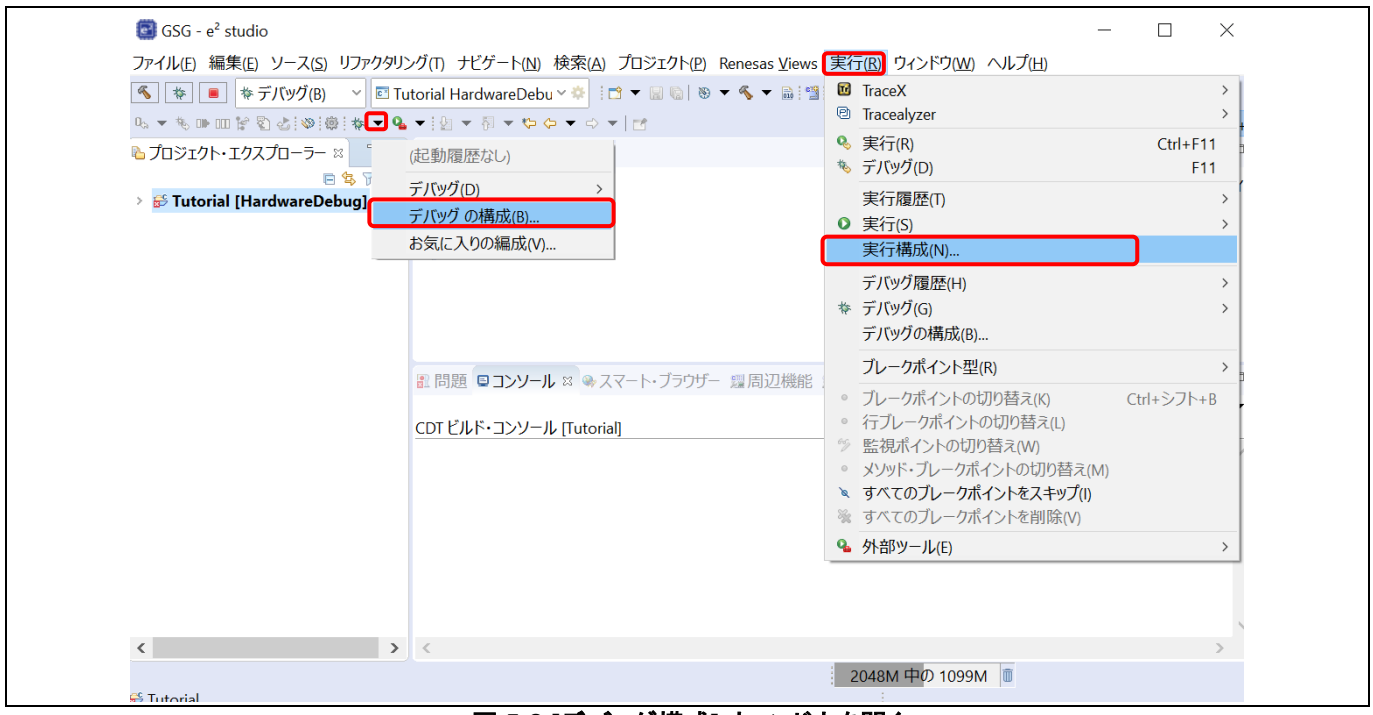

図 5-2 [デバッグ構成] ウィンドウを開く

(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。[メイン] タブを選択し、デバッグ対象のロードモジュール(*.x または*.elf)が指定されているかを確認してください。

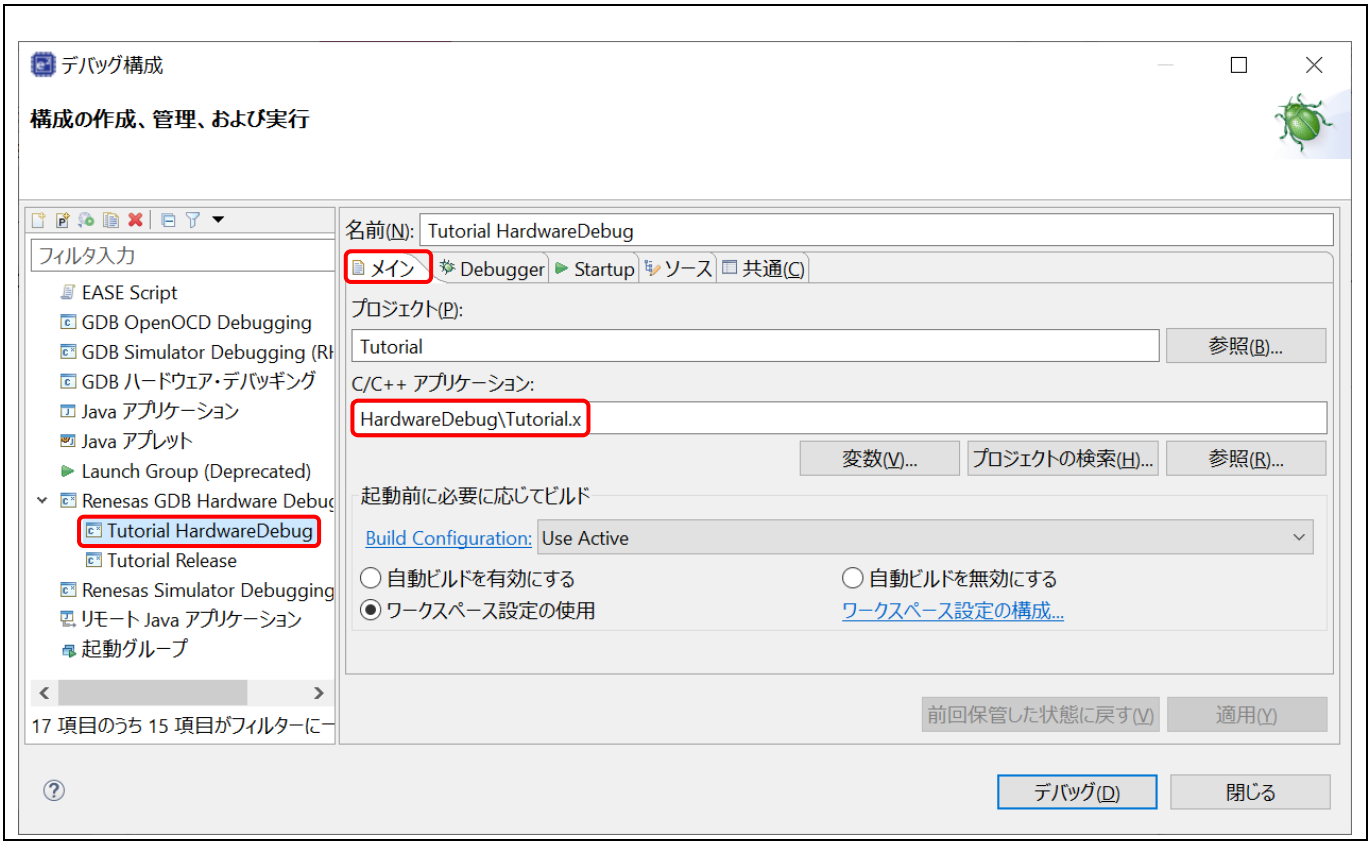

図 5-3 ロードモジュールの選択

(3) [Debugger] タブに切り替え、エミュレータの種類とデバイス名を選択してください。
下記の例では Debug Hardware に“E2 Lite”、Target Device に “R5F564ML”を設定しています。

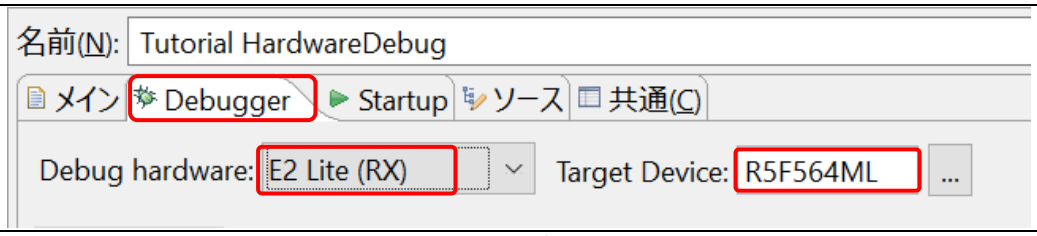

図 5-4 ターゲットデバイスの選択

(4) [Debugger] タブの中の [Connection Settings] サブタブでは、エミュレータの接続に関する設定を行います。
E1 エミュレータと RSK RX64M ボードを例に取ると、以下のような設定内容になります。

- クロック
  - メイン・クロック・ソース= “EXTAL”
  - Extal 周波数[MHz] = “24.0000”

  注意:この設定はクロックの逓倍率を考慮せず、発信子の周波数をそのまま与えてください。

- 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”
  - ホット・プラグ = “いいえ”

ホット・プラグイン接続可能なデバイスでのみ "はい" を選択できます。詳しくはデバイスのハードウェアマニュアルおよびオンチップデバッガ機能一覧をご参照ください。

- 電源
  - エミュレータから電源を供給する (MAX 200mA) = “いいえ”

    ここを “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)。外部電源を接続している場合は"いいえ"を選択してください。

- 通信モード
  - モード= “デバッグ・モード”

通信モード "内蔵フラッシュメモリーへの書き込み" を選択するとIDコード領域を含めて書込むことができます。ただし書込み終了後にデバッガは終了します。

注意: この章での設定内容は例として示したものです。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、接続の前に設定値がボードの仕様と合っているかを慎重にご確認ください。

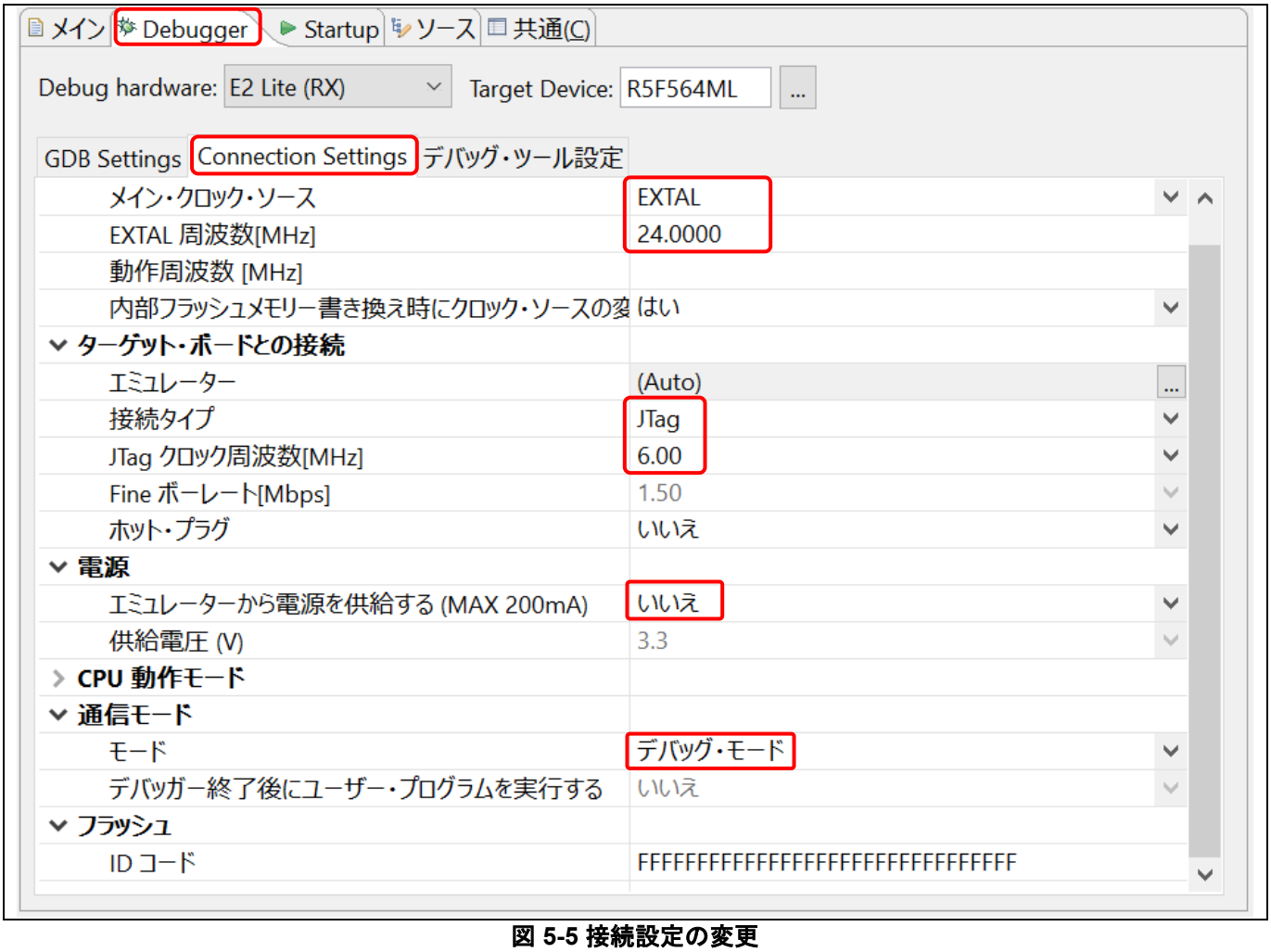

図 5-5 接続設定の変更

(5) サブタブ [デバッグ・ツール設定] では、デバッガが起動した後の動作に関する設定を行います。
詳しい設定項目の内容は、ヘルプの「e2 studio ユーザガイド」→「デバッグに関する機能」をご覧ください。

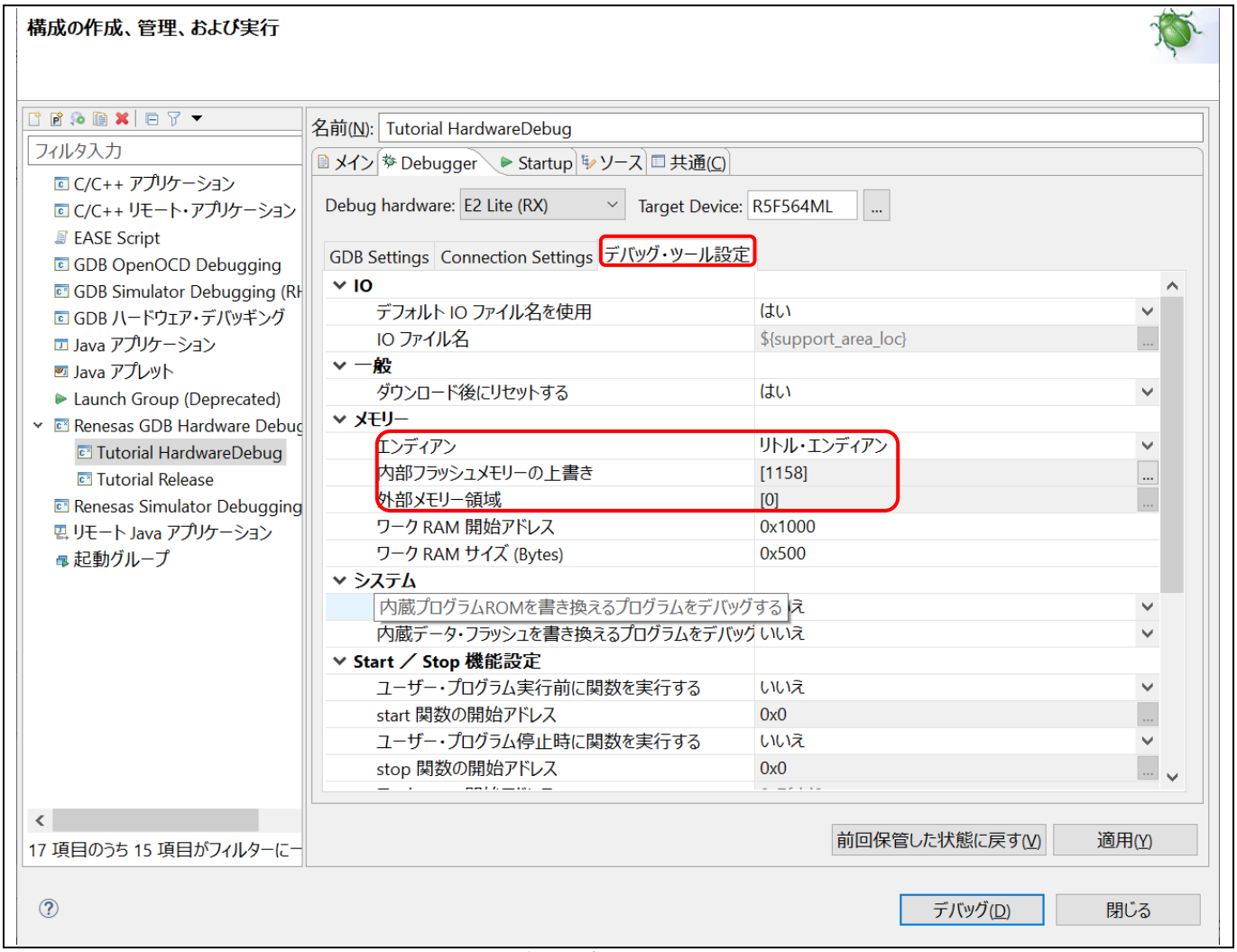

図 5-6 デバッグ・ツール設定の変更

- メモリー
  - エンディアン= “リトル・エンディアン”

    デバッガでメモリ参照する際のエンディアンを指定します。プログラムの動作内容を変更するものではありません。
  - 内部フラッシュメモリーの上書き、外部メモリー領域

    ロードモジュールのダウンロード時に書込みを許可/禁止する領域を指定します。デフォルトでは全てのブロックが書込み対象となっています。メモリの内容を保持したい領域があれば書込み対象から外してください。

(6) [適用(Y)] ボタンを押すと設定が保存されます。[デバッグ(D)] ボタンを押すと、デバッガが起動しボードへの接続、ロードモジュールのダウンロードが開始されます。

(7) 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。接続後はエントリポイント (この例では“r_cg_resetprg.c” の “PowerON_Reset_PC()”)でプログラムの実行が一旦中断されます。

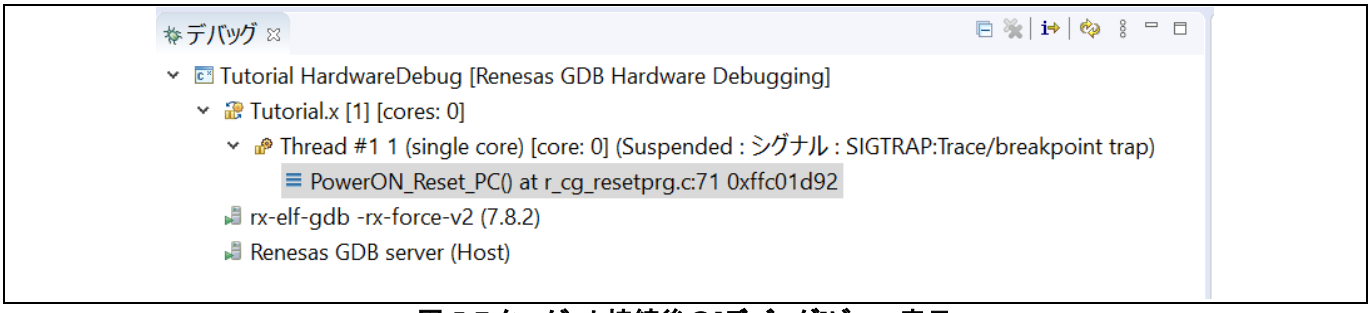

図 5-7 ターゲット接続後の[デバッグ]ビュー表示

5.2 新規デバッグ構成の作成

別な種類のエミュレータを使用したいなどで、デバッグ構成を追加する場合に、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。

(1) 5.1 章の(1)と同様の手順で[デバッグ構成]ダイアログを表示します。

(2) 左ペインに表示されたデバッグ構成 (例: “Tutorial HardwareDebug”) を選択し アイコンをクリック(現在選択している起動構成をコピー)すると、複製されたデバッグ構成 (例: “Tutorial HardwareDebug (1)”) が作成されます。デバッグ構成名(「名前:」欄)や設定を変更して[適用]ボタンを押すと保存されます。

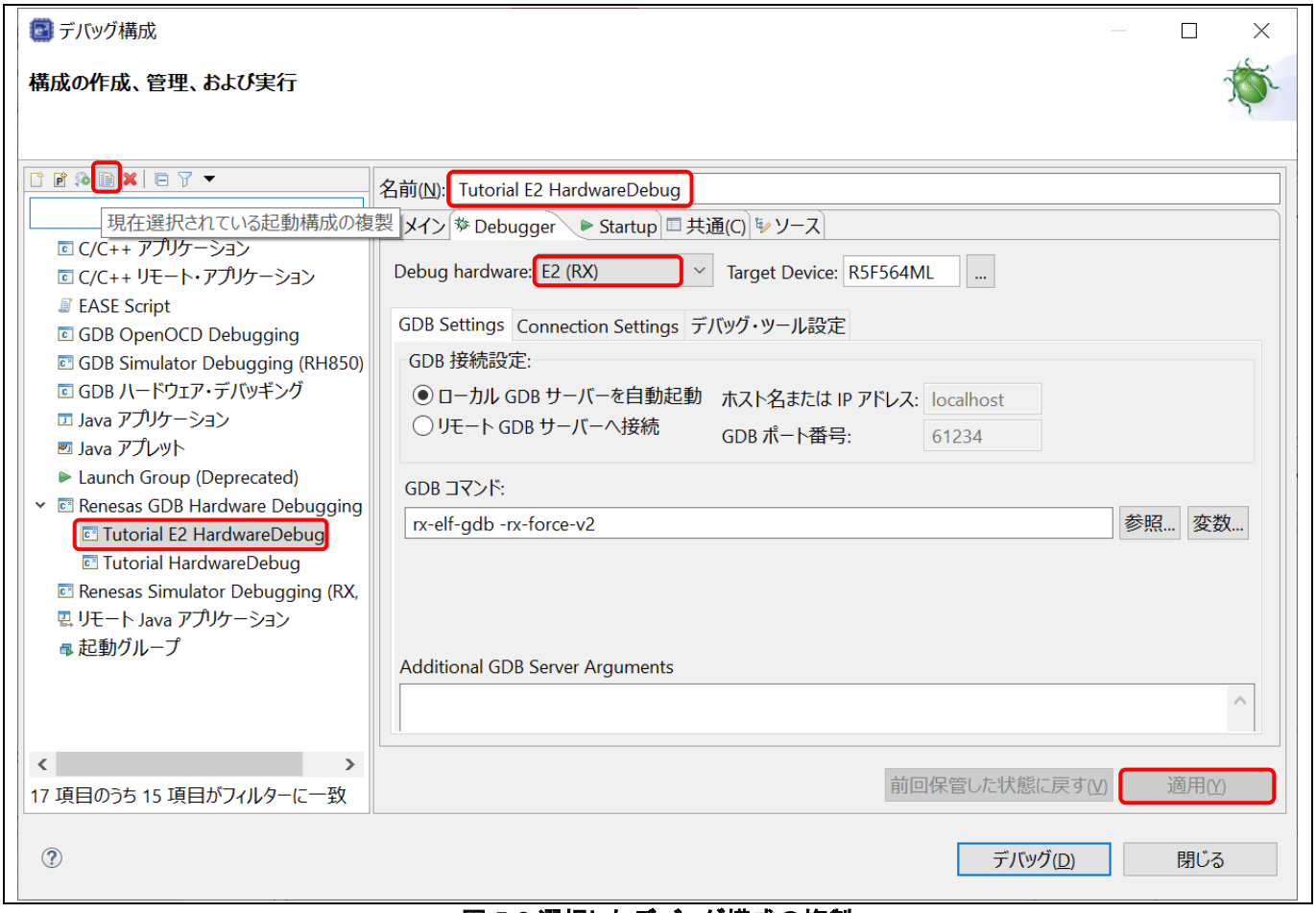

図 5-8 選択したデバッグ構成の複製

(3) デバッグ構成は 5.1 章の方法で設定できます。この例ではエミュレータを “E2 (RX)” に変更しています。

(4) デバッグ構成を追加した時、デバッグ構成のツリーに*印と[local]との警告が表示されますが、その状態ではプロジェクトに関連付けられていませんので、下図のように「共通」タブで保存先をプロジェクトフォルダに指定してください。

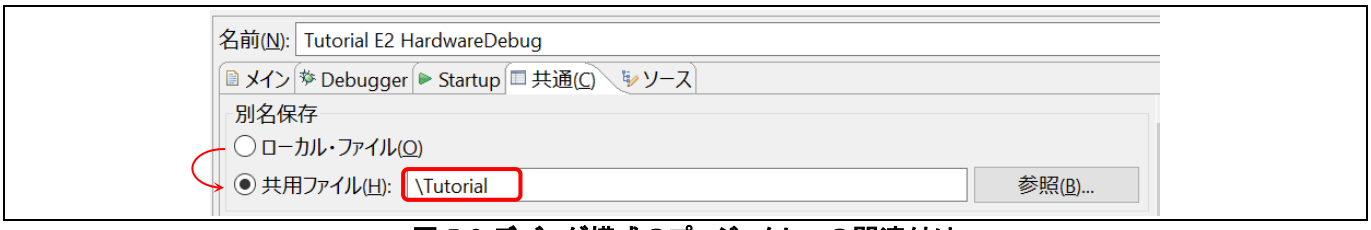

図 5-9 デバッグ構成のプロジェクトへの関連付け

5.3 Launch Bar

本節では、V6.0.0以降で追加されたLaunch Bar(下図のツールバー)の使い方を説明します。Launch Barはe² studioメインwindowのツールバーエリア内に表示されます。

以下に示すようにこのインタフェースはデバッガで起動対象とするターゲットに対してビルドやデバッグを行います。

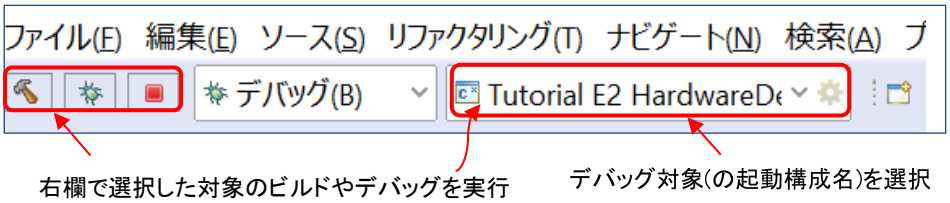

図 5-10 Launch Bar

Launch Barの各ボタンは以下の動作を行います:

-  ボタンを押すと、右欄で選択されたデバッグ対象のロードモジュールをビルドします。

  注意: 「ファイル」ツールバーにも同様のボタン  がありますが、そちらはプロジェクト・エクスプローラでアクティブ状態にしたビルド構成のビルドを行うものです。Launch Barのビルドボタンを押した時とはビルドされるものが異なる場合があります。

-  ボタンでデバッグの開始／終了(ブレークではなくデバッグセッションの終了)を行います。

Launch Barの機能が不要な場合は、「ウィンドウ」メニュー →「設定」→「実行/デバッグ」→「起動中」の「Launch Bar」の設定でツールバーやボタンを非表示にすることができます。

5.4 基本的なデバッグ機能

本節では、e² studioがサポートする以下の典型的なデバッグビューについて説明します。

- Eclipse フレームワークに標準的な GDB デバッガの機能のビュー(ウィンドウ): ブレークポイント、式、レジスタ、メモリ、逆アセンブル、変数
- GDB デバッガの Renesas による拡張機能のビュー: イベントポイント、IO レジスタ、トレース

[デバッグ] ビューには、下記のような便利なツールバーがあります。

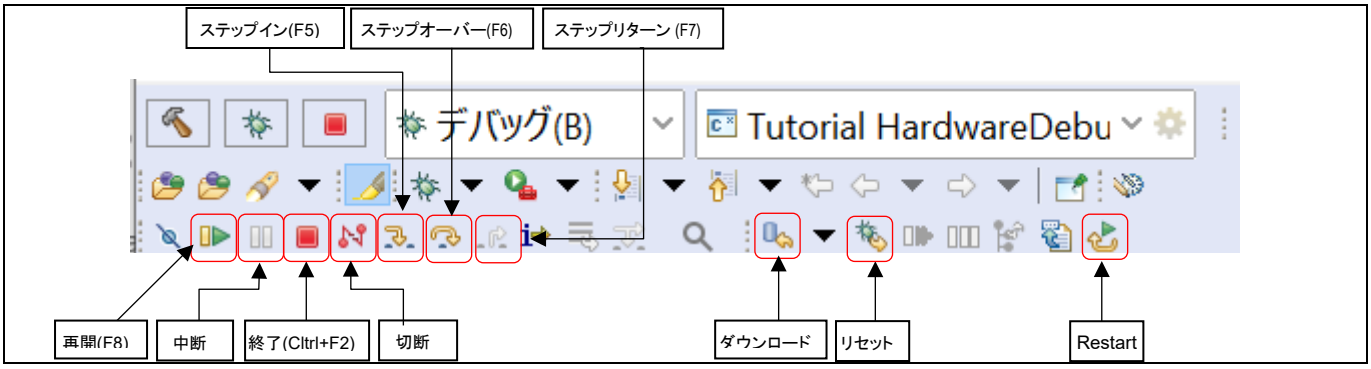

図 5-11 デバッグビューの便利なツールバー

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ー実行します(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 ボタンで停止するか、選択したプロセスとデバッガを 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。

以下の章の操作説明には、3.3章で示したサンプルプロジェクトを使用します。.

5.4.1 ブレークポイントビュー

ブレークポイントビュー(Breakpoints view)には、プログラムの実行可能な行に設定されたブレークポイントが表示されます。デバッガでプログラムを実行中、有効なブレークポイントの行またはアドレスに達すると実行が中断されます。e² studioではソフトウェアブレークポイント(のマーカー)とハードウェアブレークポイント(のマーカー)を区別して設定できます。ダブルクリックによりブレークポイントを設定すると、デフォルトのブレークポイント型が適用されます。ハードウェアブレークポイントの場合、ブレークポイント用のハードウェアリソースが残っていないときは設定できません。その場合は、ソフトウェアブレークポイントに置き換えるとのメッセージが表示されます。

ブレークポイントを設定するには、デバッガが起動した状態で以下の操作を行います。

(1) エディタ画面上でブレークポイントを設定する

ソースファイルエディタの左余白をダブルクリックすると、デフォルトの種別でブレークポイントが設定され、ブレークポイントマーカーが表示されます。マーカーがならハードウェアブレークポイント、ならソフトウェアブレークポイントです。

ダブルクリックではなく、ソースコードの左余白を右クリックすると [Toggle Software Breakpoint] または [Toggle Hardware Breakpoint] で直接ブレークポイント種別を選択できます。

(2) 「ブレークポイント」ビューでブレークポイントプロパティを設定する

[ウィンドウ]メニューの[ビューの表示](または[ALT] + [Shift] + [Q]ショートカットに続けて[B]を押す)で[ブレークポイント]ビュー(のアイコンのビュー)を開くとブレークポイントの有効・無効、ブレークポイント種別やブレークポイントを設定する行番号やアドレス等の変更が行えます。

以下の方法でブレークポイントを無効(ブレークポイントで止まらない)にすることができます。

- エディタの左余白に表示されたブレークポイントマーカー(または)を右クリックし、コンテクストメニューの「ブレークポイントを使用不可にする」を選択すると無効に切り替わります。無効になるとブレークポイントマーカーが白くなります(または)。
  再度コンテクストメニューの「ブレークポイントを使用可能にする」で有効に戻せます。あるいは、シフトを押しながらダブルクリックすると無効・有効が切り替わります。

- 全てのブレークポイントを一括して無効に切り替えるには、ツールバーまたはブレークポイントビューの ボタンをクリックします。一括して無効になっている間は、全てのブレークポイントマーカーに斜線(＼)が重ねて表示され、どのブレークポイントでも止まらなくなります。

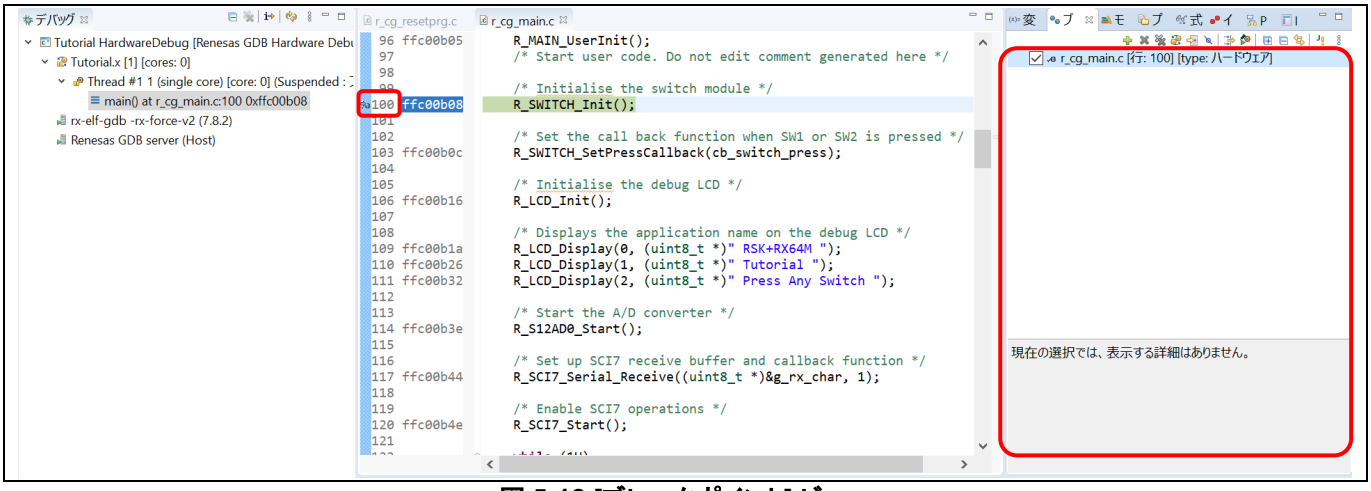

図 5-12 [ブレークポイント] ビュー

5.4.2 式ビュー

式ビュー(Expressions view)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モニタできます。「リアルタイムリフレッシュ」を有効にすると、デバッガ実行中に設定された周期で変数の値が更新されます。
変数名で登録する他に、計算式(例：“Aval+Bval*2”)や型キャスト(例：“(struct mystr *)&buf[1]”)を使って表示させることができます。

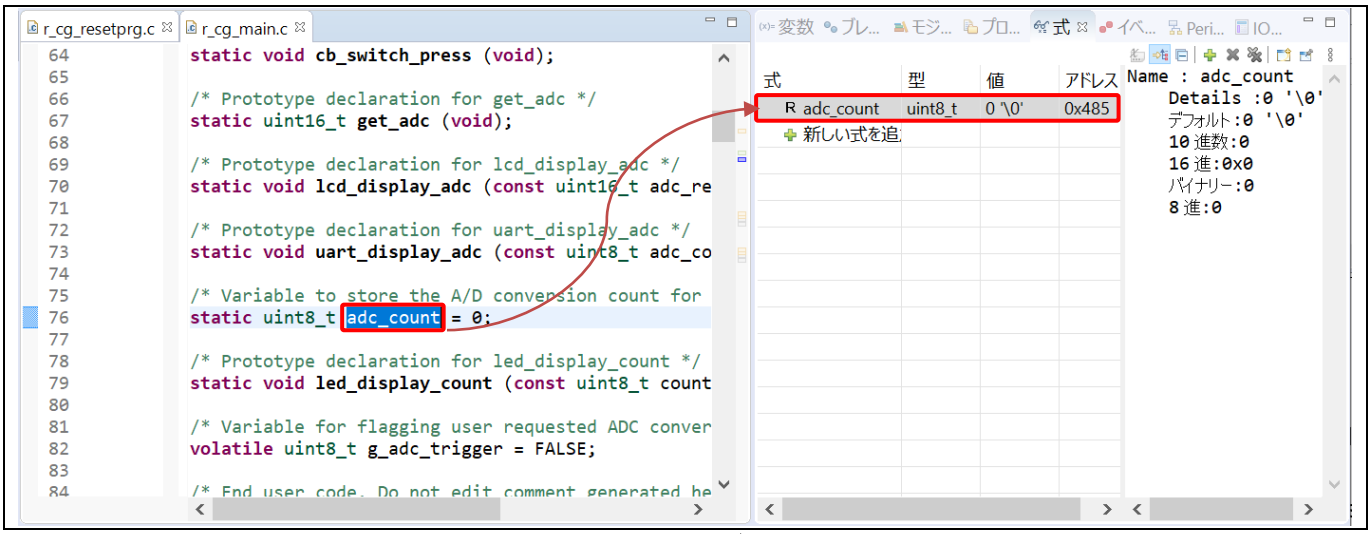

図 5-13 [式] ビュー

変数を見るには、

(1) [Window]メニュー [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) エディタ上で変数名を選択状態にしてから[式] ビューへドラッグ&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] ビュー内で変数の行を右クリックして“リアルタイム・リフレッシュ”を選択すると、先頭に“R”が表示されプログラムの動作中は一定周期で値が更新されるようになります。

(4) 再度右クリックして”リアルタイム・リフレッシュを無効にする”を選択すると、リアルタイム・リフレッシュが無効になります。

式ビューにはローカル変数も同様にして表示することができますが、スコープ(その変数が属する関数またはブロック)外の行を実行している時には値が表示されません。

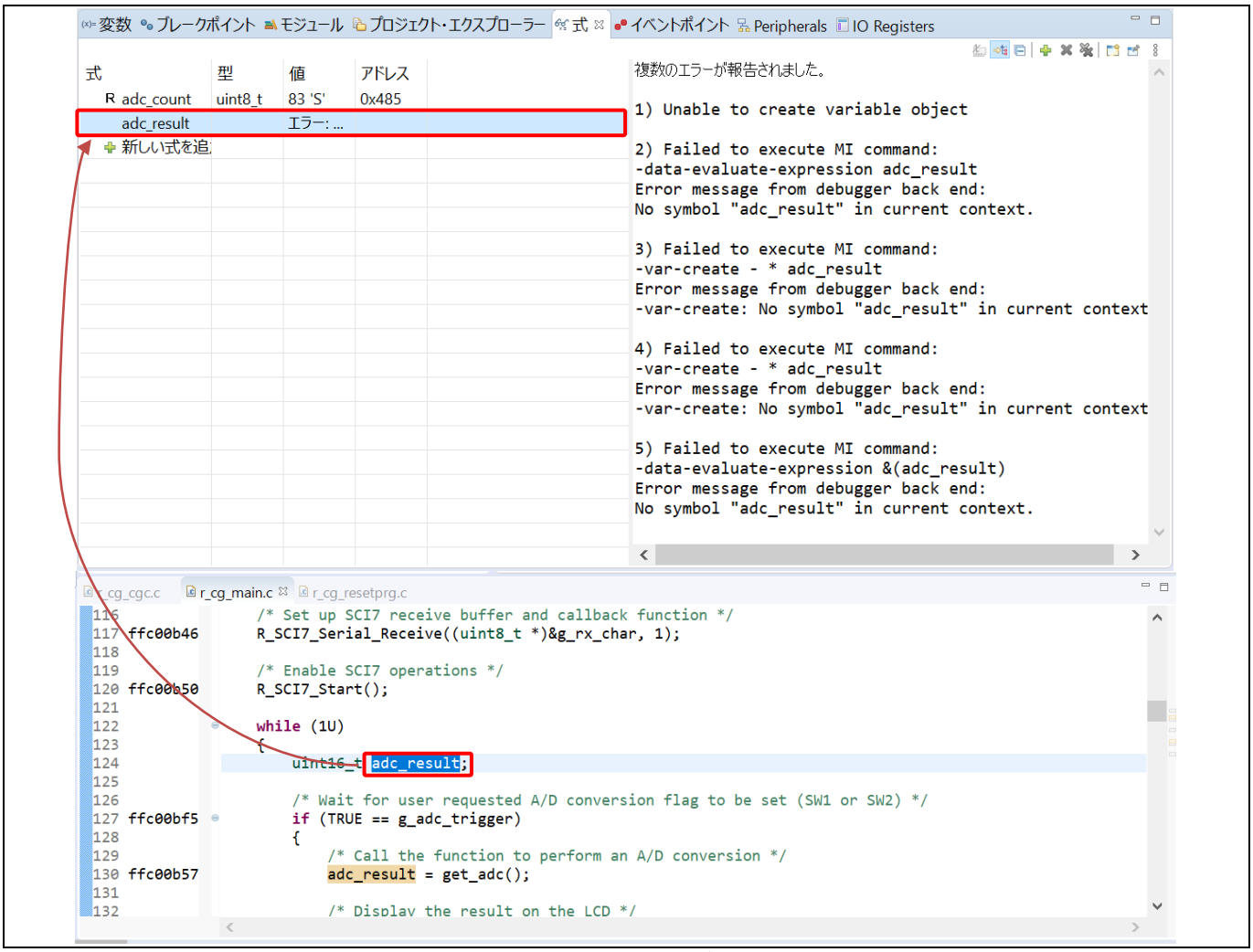

図 5-14 ローカル変数を式ビューで表示

ただし、アドレスが一意に決まる変数(static変数)に対しては、明示的にスコープを指定することで変数のスコープ外を実行している時でも値の参照が可能になります。
例えば関数 myfunc( ) のスコープ内の変数 ”myval” を参照する場合は、式ビューに「myfunc::myval」(間にコロン二つ)の書式で登録します。

5.4.3 レジスター ビュー

レジスタービュー(Registers view)は汎用レジスタについての情報を表示します。

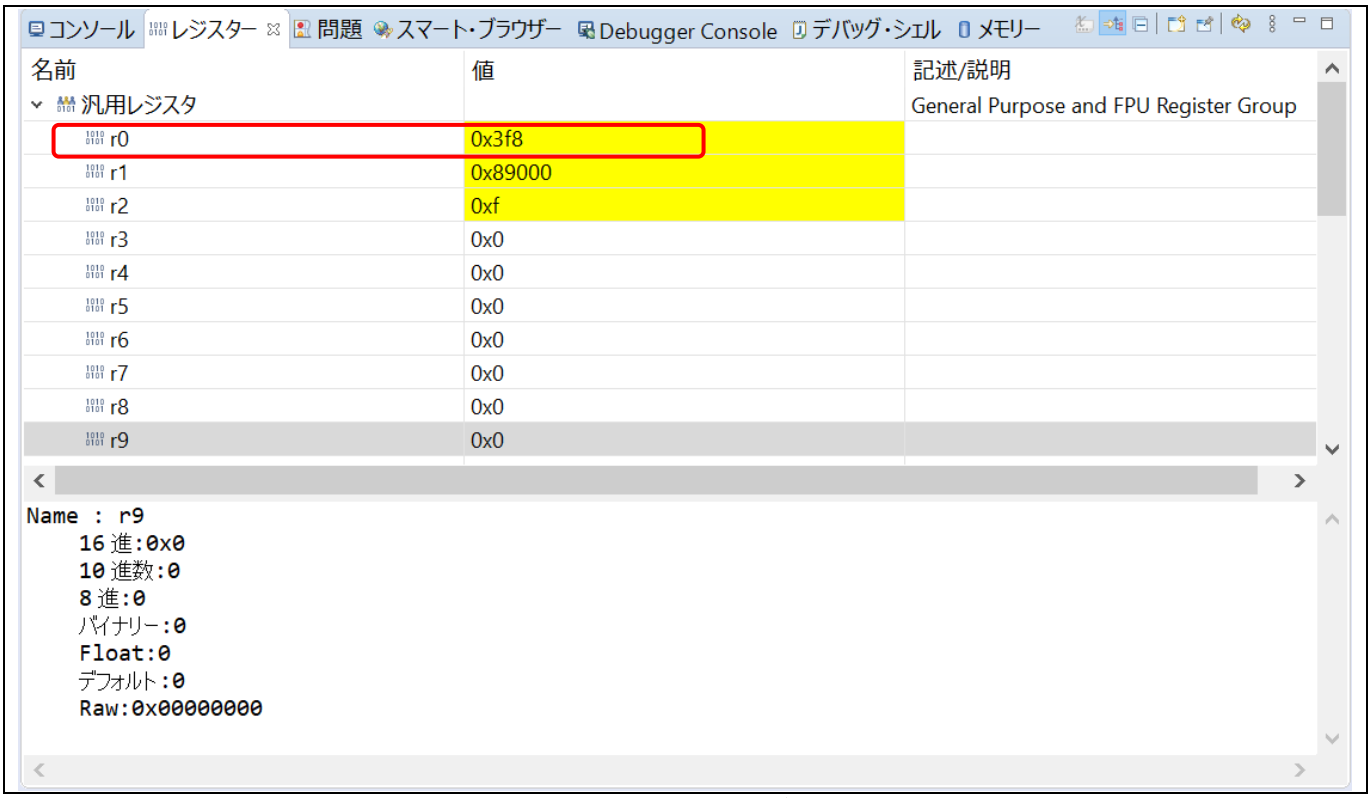

図 5-15 [レジスター] ビュー

汎用レジスタの値を見るには、

(1) 1010 0101 のアイコンが付いた[レジスター]ビューを選択するか、表示されていなければ[ウィンドウ]メニューの[ビューの表示]→ [レジスター]を選択して表示させます。

(2) レジスタ名を選択すると詳細欄には各基数のフォーマットで値が表示されます。

前回ブレーク時以降に変化のあった値は強調表示(背景色が黄色)になっています。

5.4.4 メモリー ビュー

メモリービュー(Memory view)では、ユーザは “メモリーモニター” (表示開始アドレスと表示形式の組み合わせ)を指定してメモリ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データは異なる “メモリレンダリング” で表示することができます。メモリ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。

変数(例: adc_count)をメモリービューで確認するには、

(1) メモリー ビュー(のアイコン)が表示されていればそれを選択、表示されていなければ [ウィンドウ] → [ビューの表示] → [メモリー] でビューを開いてください。

(2) ビュー内の をクリックして[モニター・メモリー]のダイアログを開き、メモリーの表示開始アドレスを指定してください。下記の例では変数のアドレス&adc_count を指定しています。

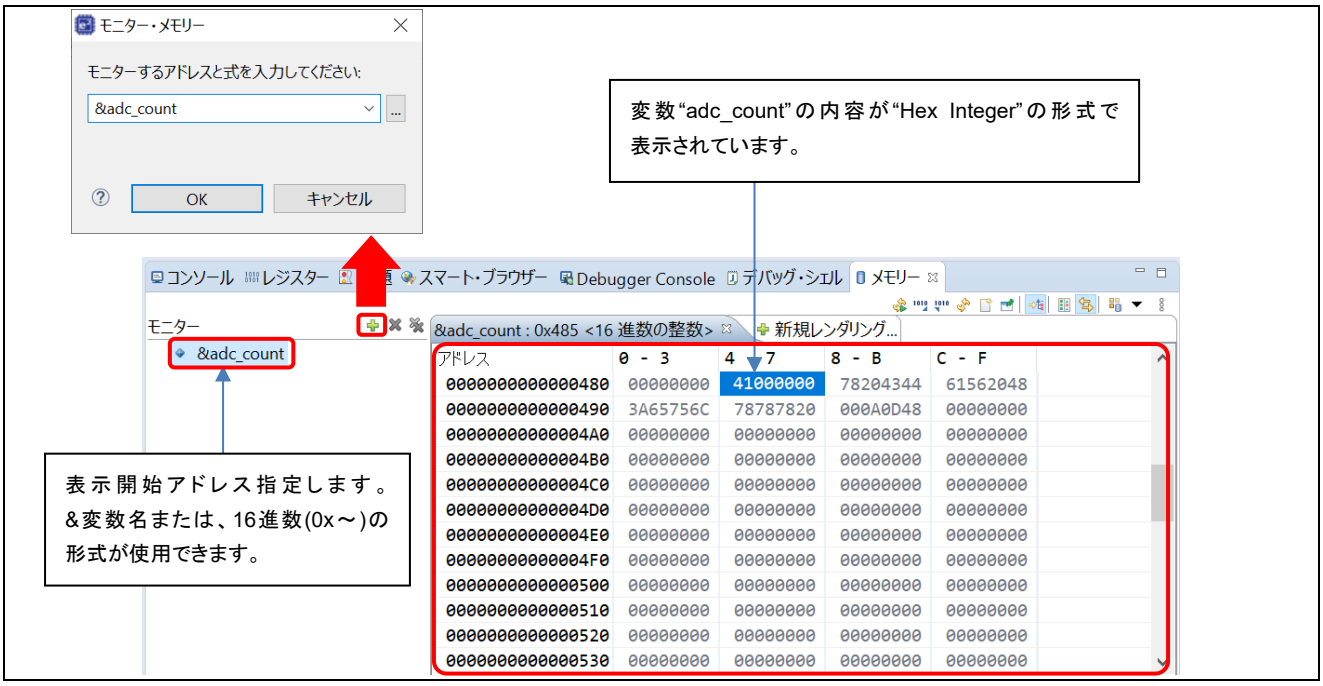

図 5-16 [メモリー] ビュー(1/2)

指定したメモリー・モニターに対するレンダリング(表示形式)を追加するには、

(1) 新規レンダリング... のタブをクリックし、レンダリングを選択してから[レンダリングの追加]ボタンをクリックすると指定したレンダリングで表示するタブが追加されます。

下記の例では変数 adc_count を「16進数の整数」と[ASCII]の二種類で表示を切り替えられるようにしています。

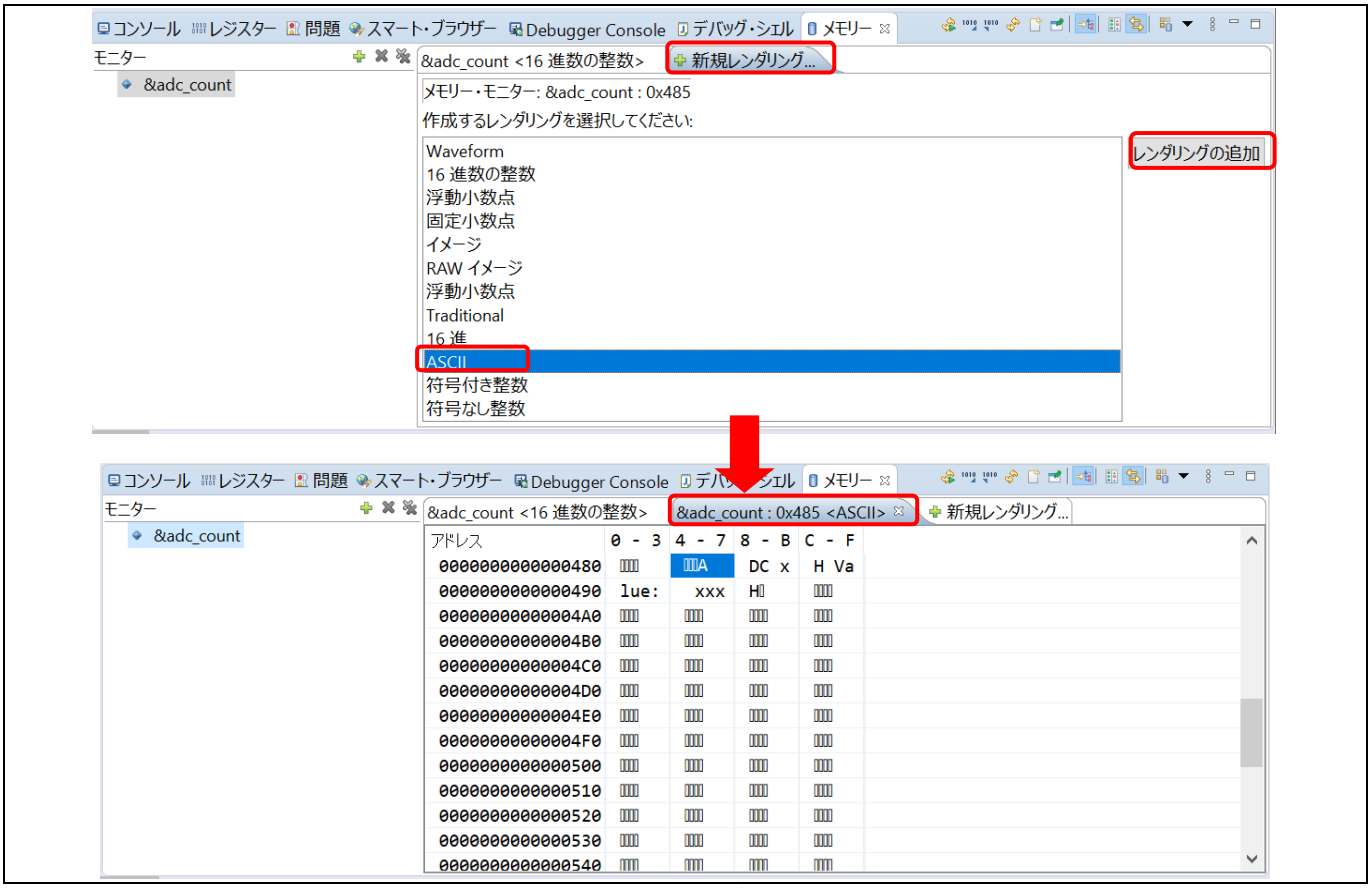

図 5-17 [メモリー] ビュー (2/2)

5.4.5 逆アセンブル ビュー

逆アセンブル ビュー(Disassembly view)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。

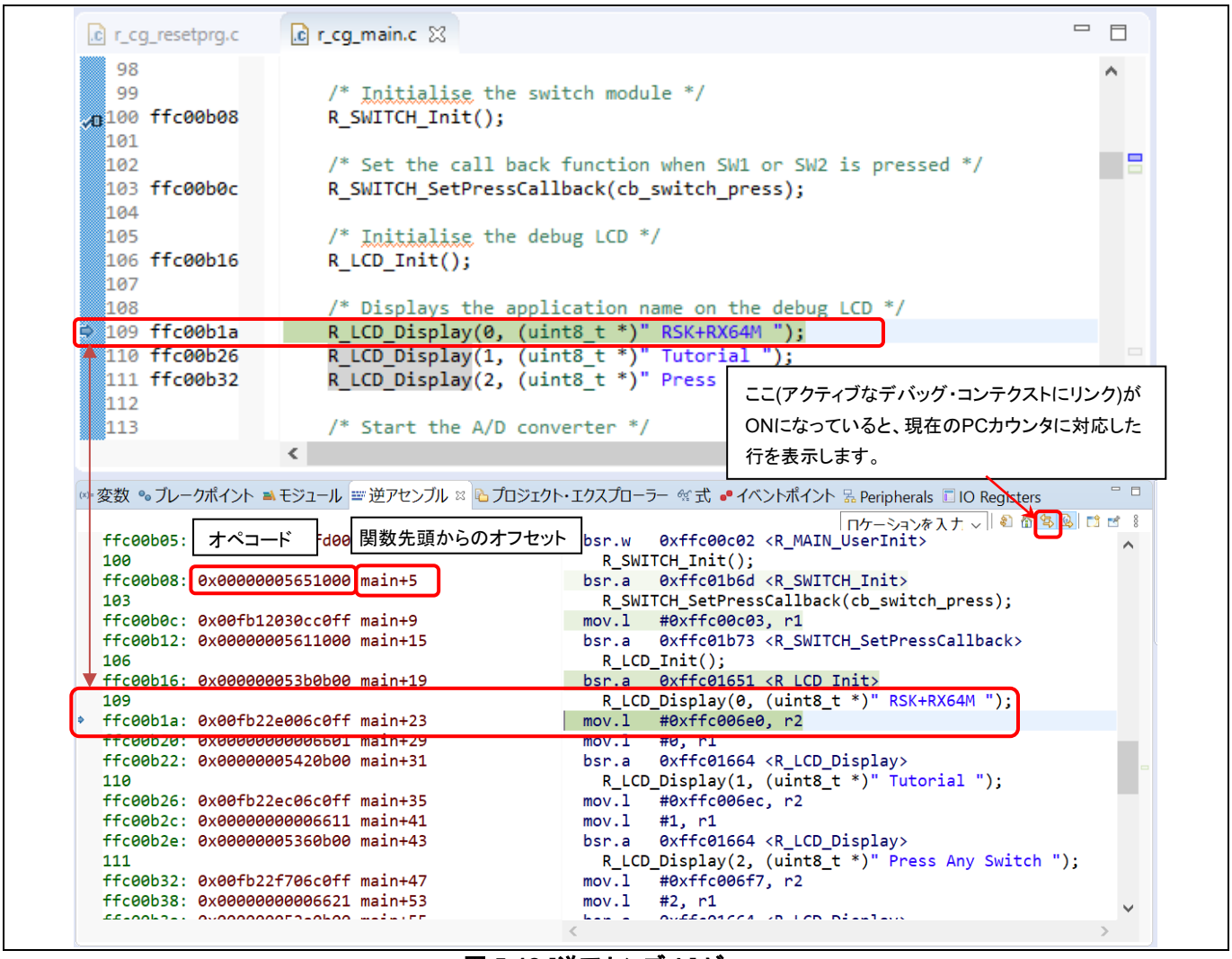

図 5-18 [逆アセンブル] ビュー

逆アセンブル ビューを使うには、

(1) デバッガを起動し、エディタ左端のアドレスが表示されている場所を右クリックし[逆アセンブリーへジャンプ]を選ぶか、[ウィンドウ] → [ビューの表示] → [逆アセンブル]を選択します。既に[逆アセンブル]ビュー (アイコンのついたタブ)が表示されていればそれをクリックします。

(2) (アクティブなデバッグ・コンテクストにリンク)ボタンが有効になっていれば逆アセンブルビューは PC カウンタのアドレスに対応した行に自動的にスクロールします。

(3) [逆アセンブル]ビューの左端、アドレス欄を右クリックし[オペコードを表示]、[関数オフセットを表示]でオペコードと関数先頭からのオフセットの表示を切り替えられます。

同様に、エディタ ビューで左端のコンテクストメニューを使ってアドレスの表示が切り替えられます。

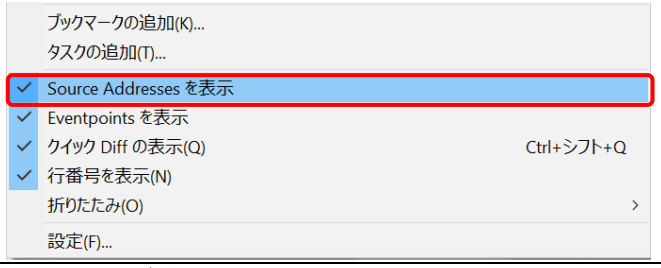

図 5-19 エディタのコンテクストメニュー(ソース行アドレス)

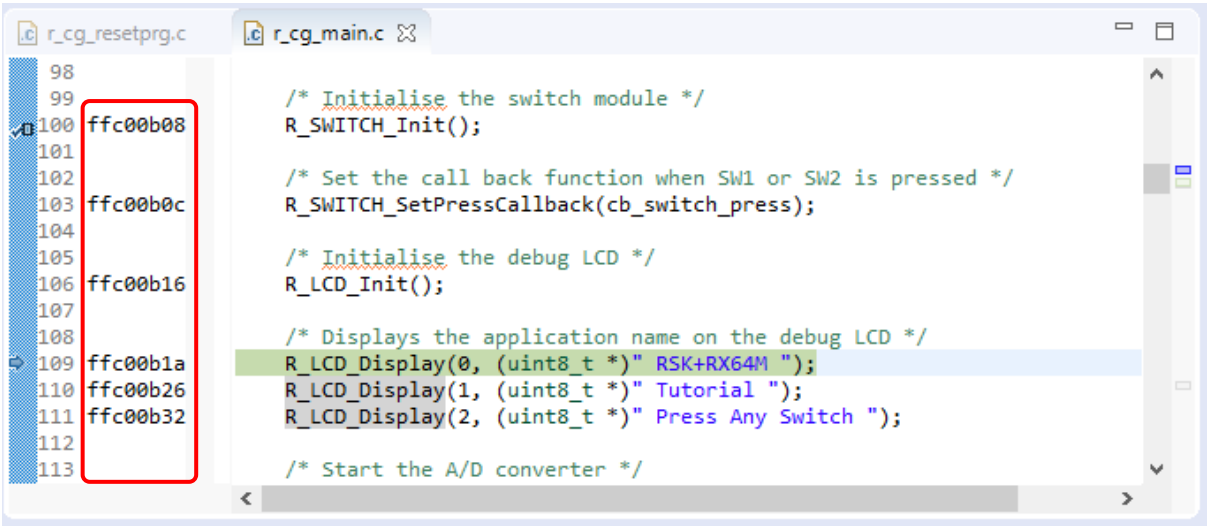

図 5-20 エディタにアドレスが表示されている様子

5.4.6 変数 ビュー

変数ビュー(Variables view)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。

グローバル変数やスコープ外でスタティック変数を参照したい場合は[式]ビュー(**5.4.2** 参照)をお使いください。

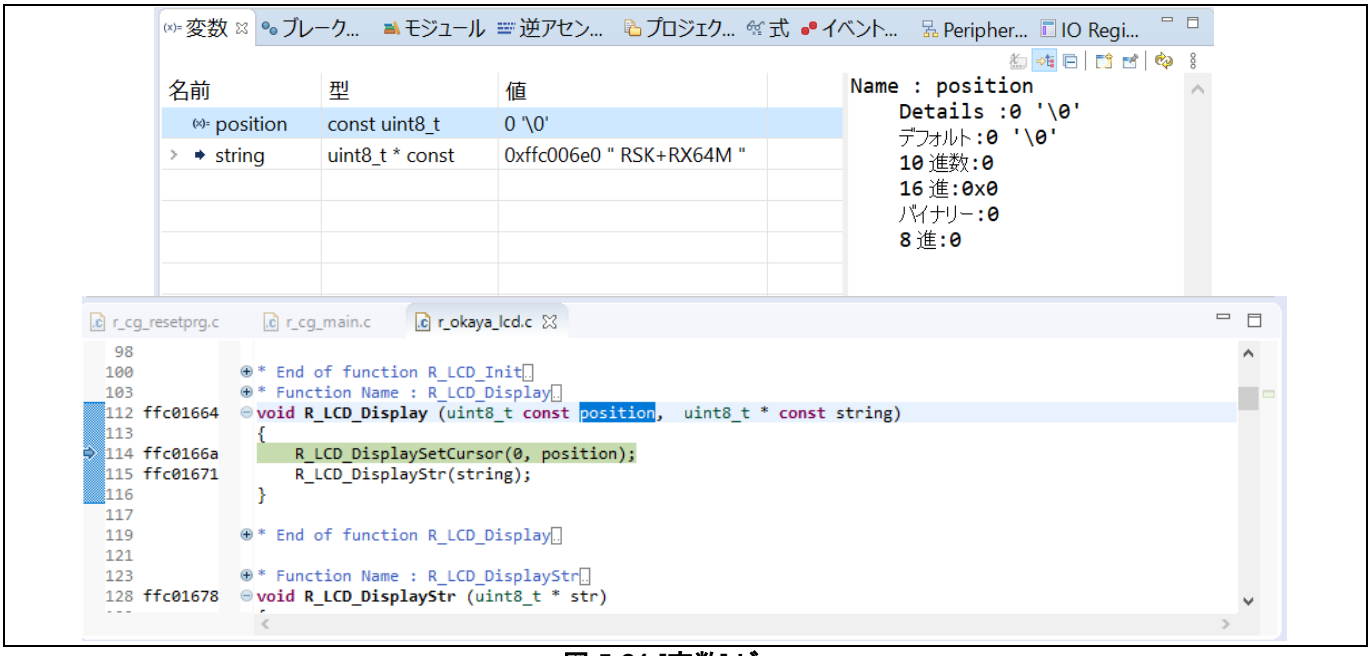

図 5-21 [変数] ビュー

ローカル変数を見るには、

(1) [変数] ビュー(のアイコン)を選択するか、[ウィンドウ] → [ビューの表示] → [変数]でビューを開きます。

(2) ステップ実行で関数内に入ると、変数ビューにローカル変数(上記の例では position と string) とその値が表示されます。

注: コンパイラやリンカの最適化処理により、変数の実体がなくなる(変数に対応するメモリを保持する必要がない場合や、変数や変数の演算に使用するレジスタが他の用途にも使われる場合など)と、[変数]ビューに値が表示されないことがあります。その際は[逆アセンブリ]ビューで変数に対応するレジスタやメモリがどれかを確認し、[レジスター]ビュー等でその値を直接見てください。

最適化を無効とすれば多くの場合デバッグ時に変数値の確認は容易になります。ただし処理速度の低下やメモリ消費量の増加等のデメリットを伴いますので、最適化を無効にするかどうかはそれらも勘案の上でご検討ください。

5.4.7 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(Eventpoints view)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。

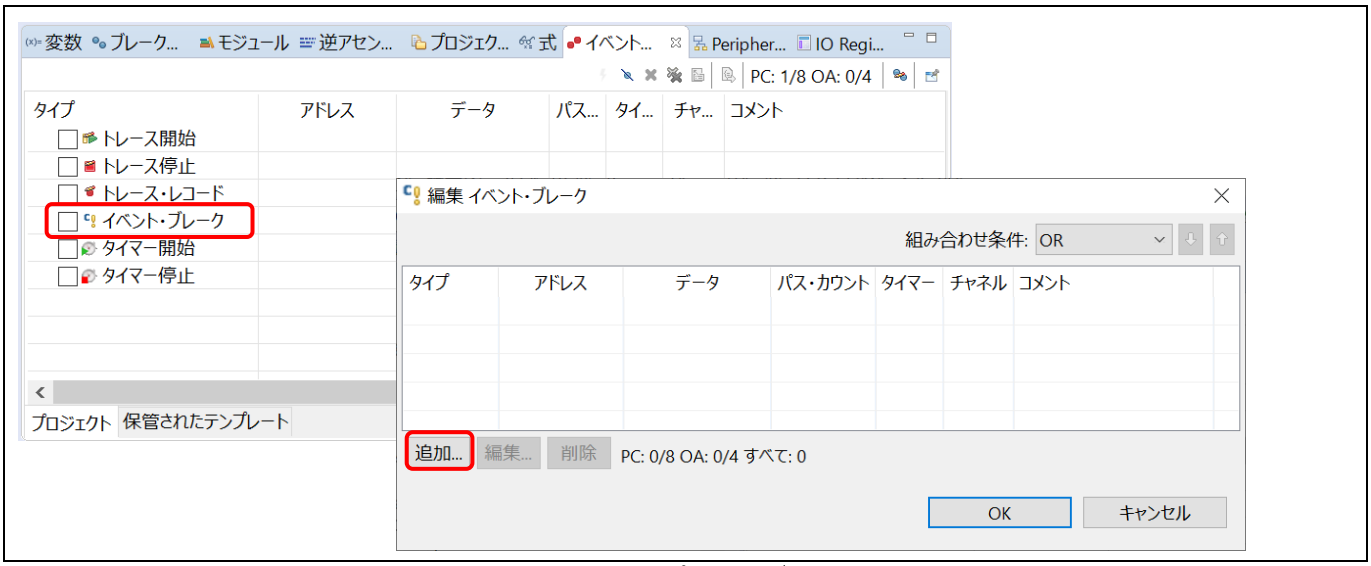

図 5-22 [イベントポイント] ビュー (1/2)

アドレスまたはデータが一致する条件(例えば、adc_count = “0x6” 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) [イベントポイント] ビュー(のアイコン)を選択するか、[ウィンドウ] → [ビューの表示] → [イベントポイント] でビューを開きます。

(2) タイプ欄の「イベント・ブレーク」をダブルクリックし、[編集 イベント・ブレーク]のダイアログを開きます。

(3) [追加...]ボタンをクリックして以下の操作を行います。

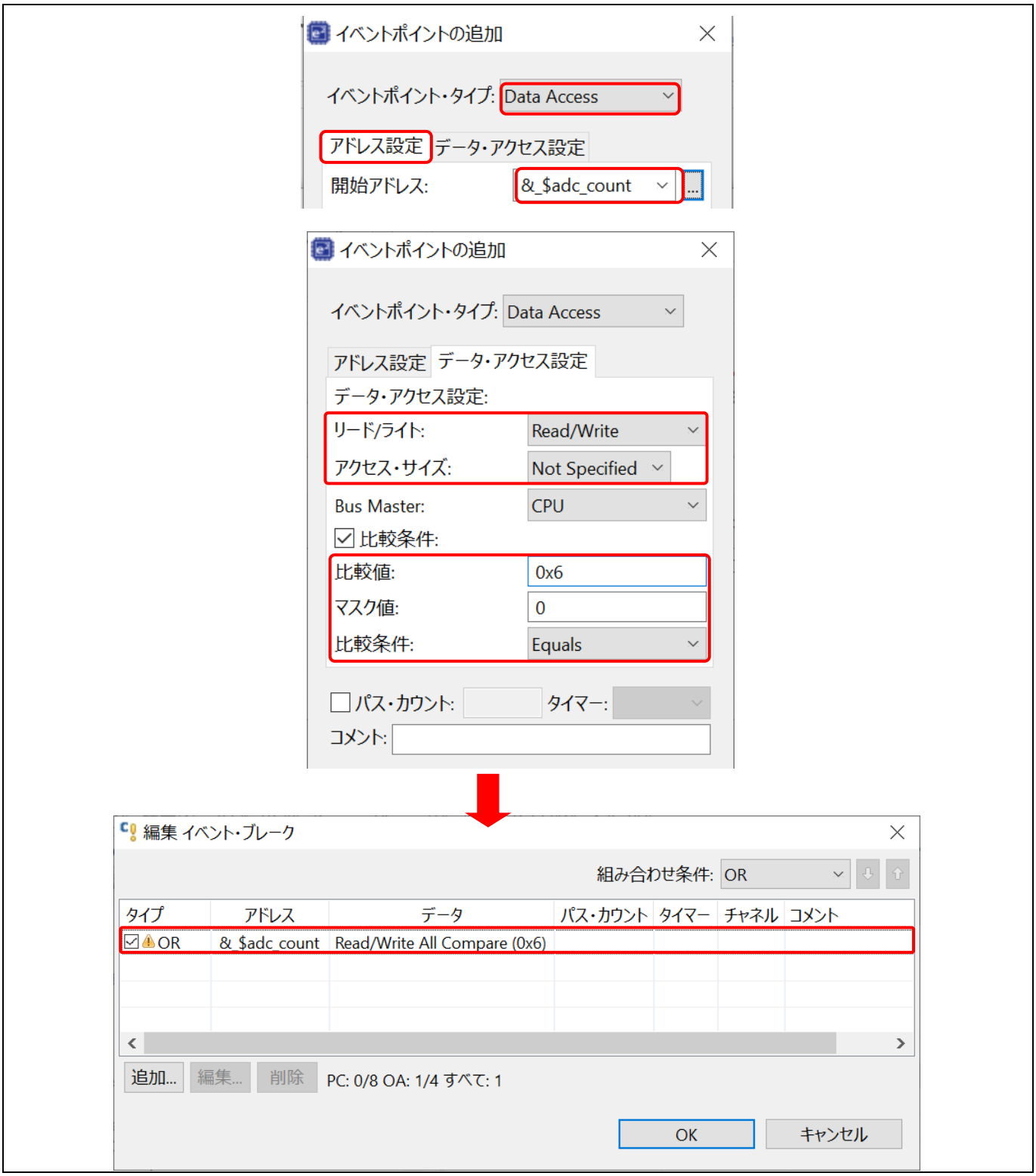

図 5-23 [イベントポイント] ビュー(2/2)

(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 “_$adc_count” を検索します。
（スタティック変数には"_$"が付加され、アドレスは “&_$adc_count” で示されます）

(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 “0x6” 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・ブレークが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た後、設定した条件が成立するまで実行してください。

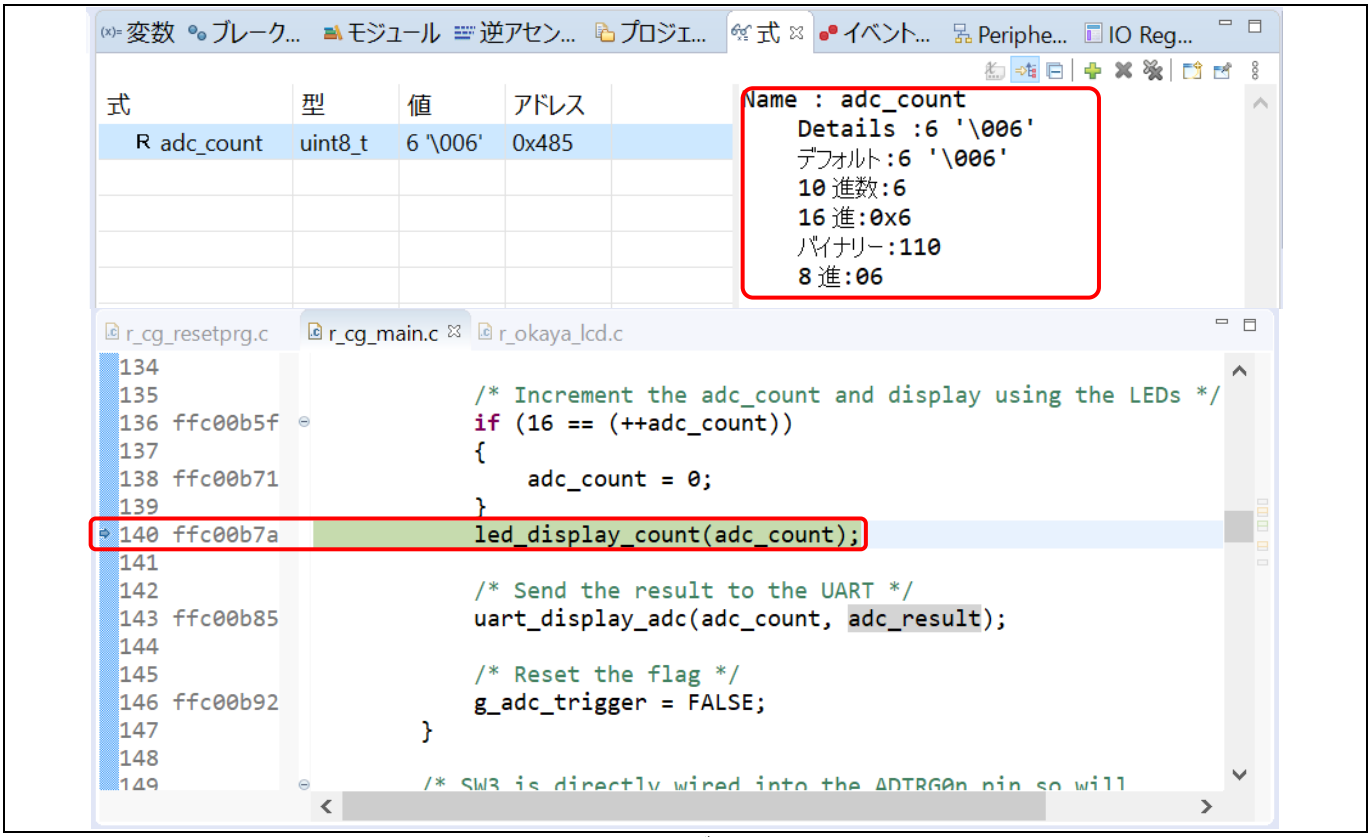

図 5-24 イベントブレークの実行

図5-24 は、条件が成立してイベント・ブレークが掛かった様子を示します。adc_countが6になった次の行でブレークしています。

5.4.8 IO Registers ビュー

IO Registersビュー(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ー] 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。

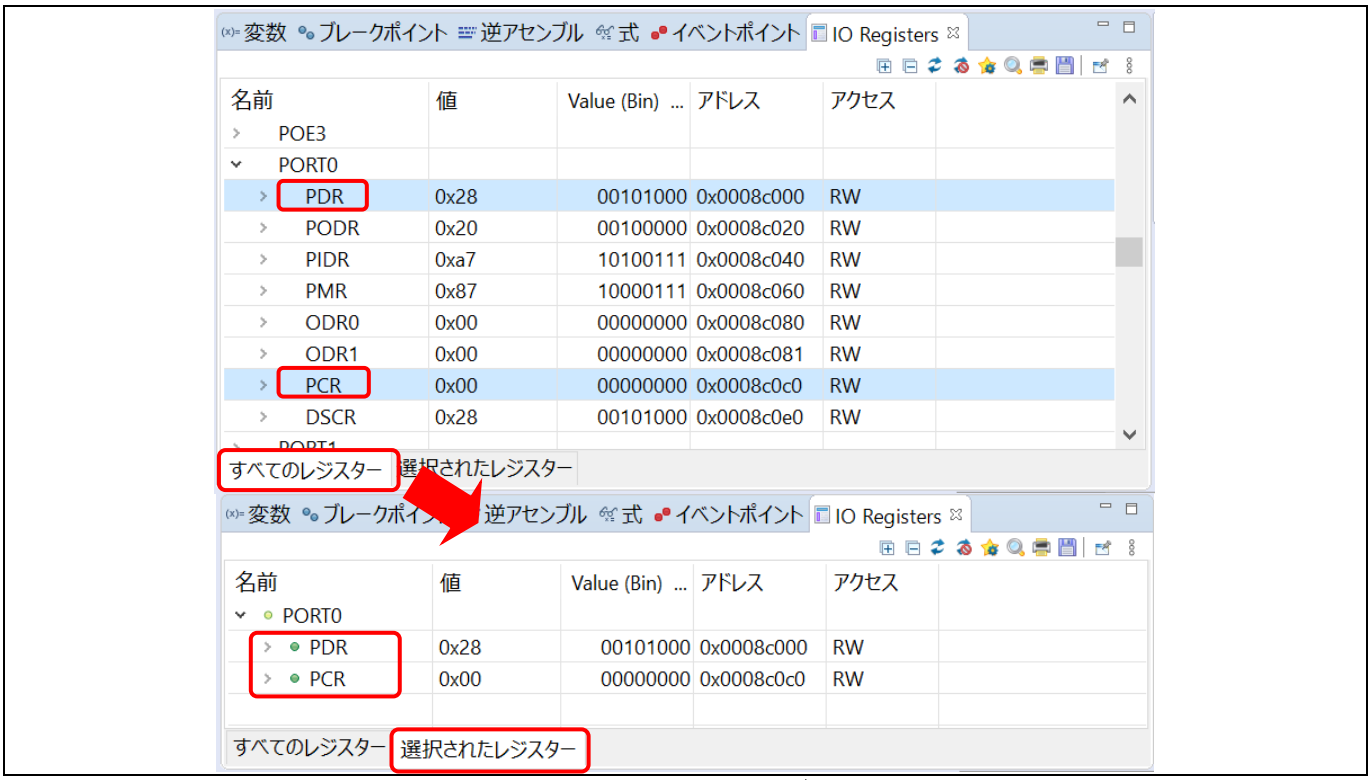

図 5-25 [IO Registers] ビュー

選択したIOレジスタの値(この例ではPORT0のPDRとPCR)を見るには、

(1) [IO Registers] ビュー(のアイコン)を選択するか、[ウィンドウ] → [ビューの表示] → [その他…]で[デバッグ]から[IO Registers]を選択してビューを開いてください。

(2) 「すべてのレジスター」タブの中から名称を探すか、検索ボタン を押して文字列検索を行ってください。例えば上図の ”PDR” レジスタを検索すると、PORT0 から一致するものが順次表示されます。

(3) “PDR”と“PCR”を[選択されたレジスター]ペインにドラッグ &ドロップします。レジスタ名左側の は、選択されたレジスタであることを示します。

(4) [選択されたレジスター] タブを選んで“PORT0” IO レジスタの“PDR”と“PCR”が表示されることを確認してください。

[すべてのレジスター] タブだと表示に時間が掛かるので、複数のレジスタを見るには[選択されたレジスター] を使うことを勧めます。

5.4.9 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。

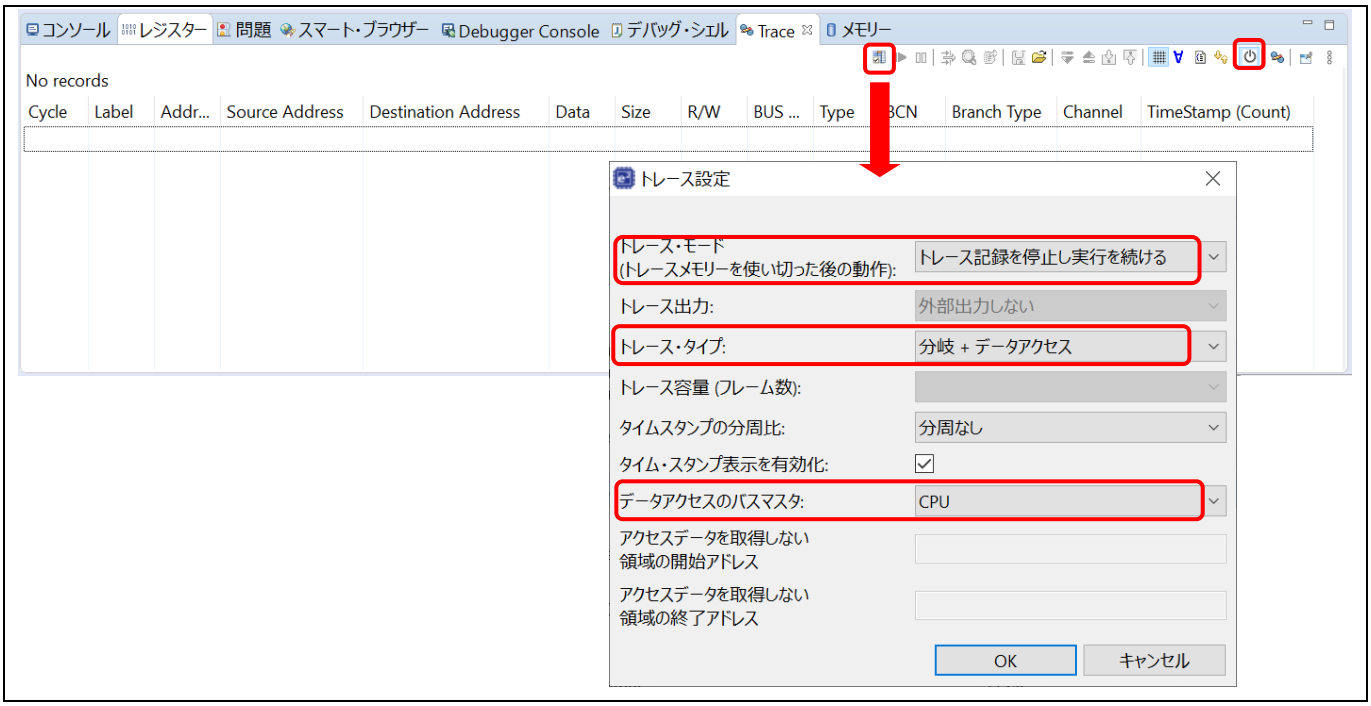

Figure 5-26 [Trace] View (1/2)

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

(1) [Trace] ビュー(のアイコン)を選択するか、[ウインドウ] → [ビューの表示] → [その他…]で「デバッグ」カテゴリ内の[Trace]を選択してビューを開いてください。

(2) トレース ビューの アイコンをクリックしてトレースの収集を有効にしてください。

(3) (Acquisition) のアイコンをクリックして、以下を設定します。

- トレース・モード: "トレース記録を停止し実行を続ける”
- トレース・タイプ: “分岐”
- データアクセスのバスマスタ: “CPU”

(4) [OK]をクリックします。

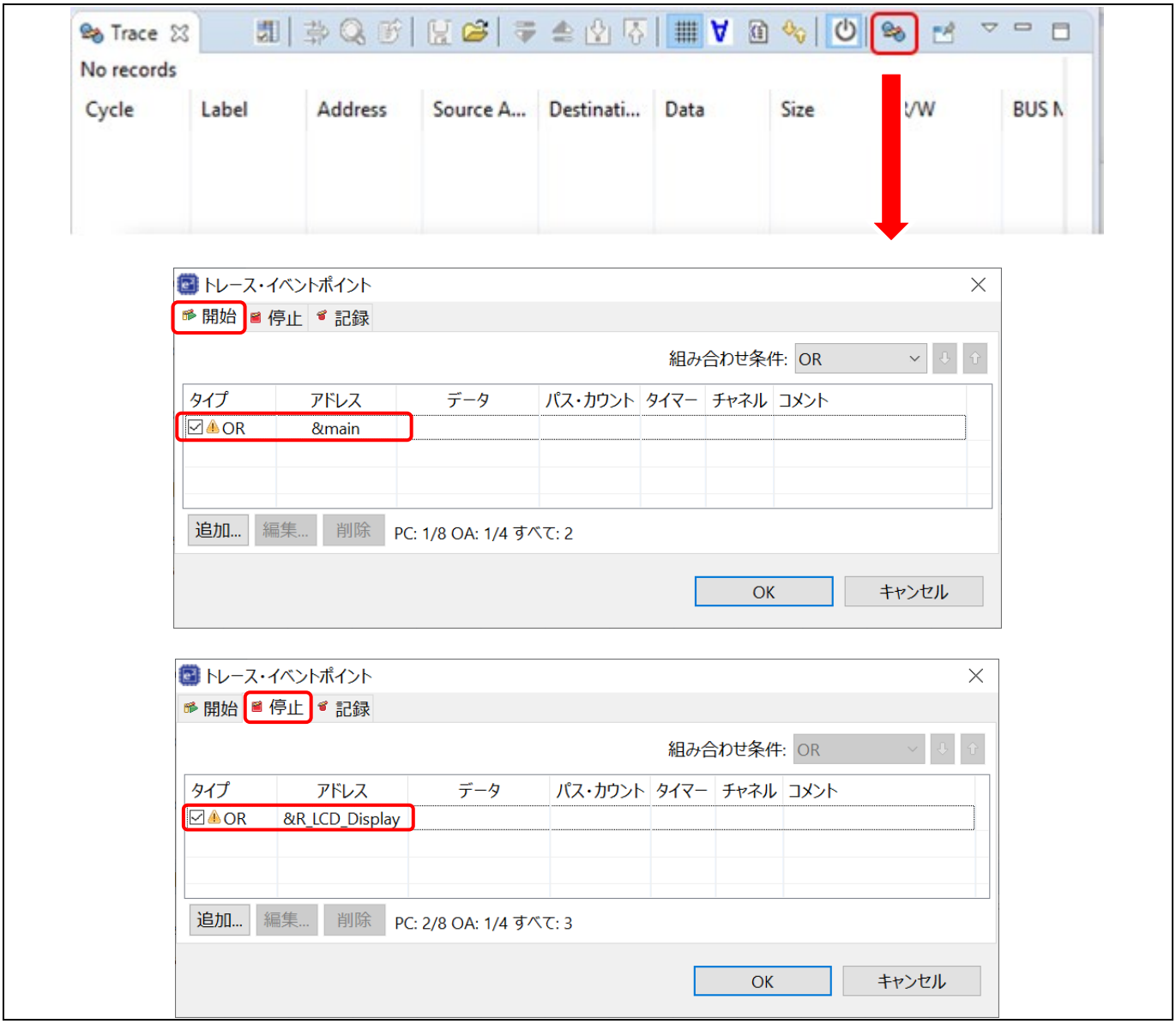

Figure 5-27 [Trace] View (2/2)

(5) ボタン (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]ダイアログを開きます。

(6) [開始]タブ内に 1 番目のイベントポイントとして main( ) 関数 (実行開始アドレスとして “&main”)を追加します。

(7) 同様に、[停止]タブ内に 2 番目のイベントポイントとして R_LCD_Display( )関数のアドレス “&R_LCD_Display” を追加します。

(8) リセット後にプログラムを実行します。

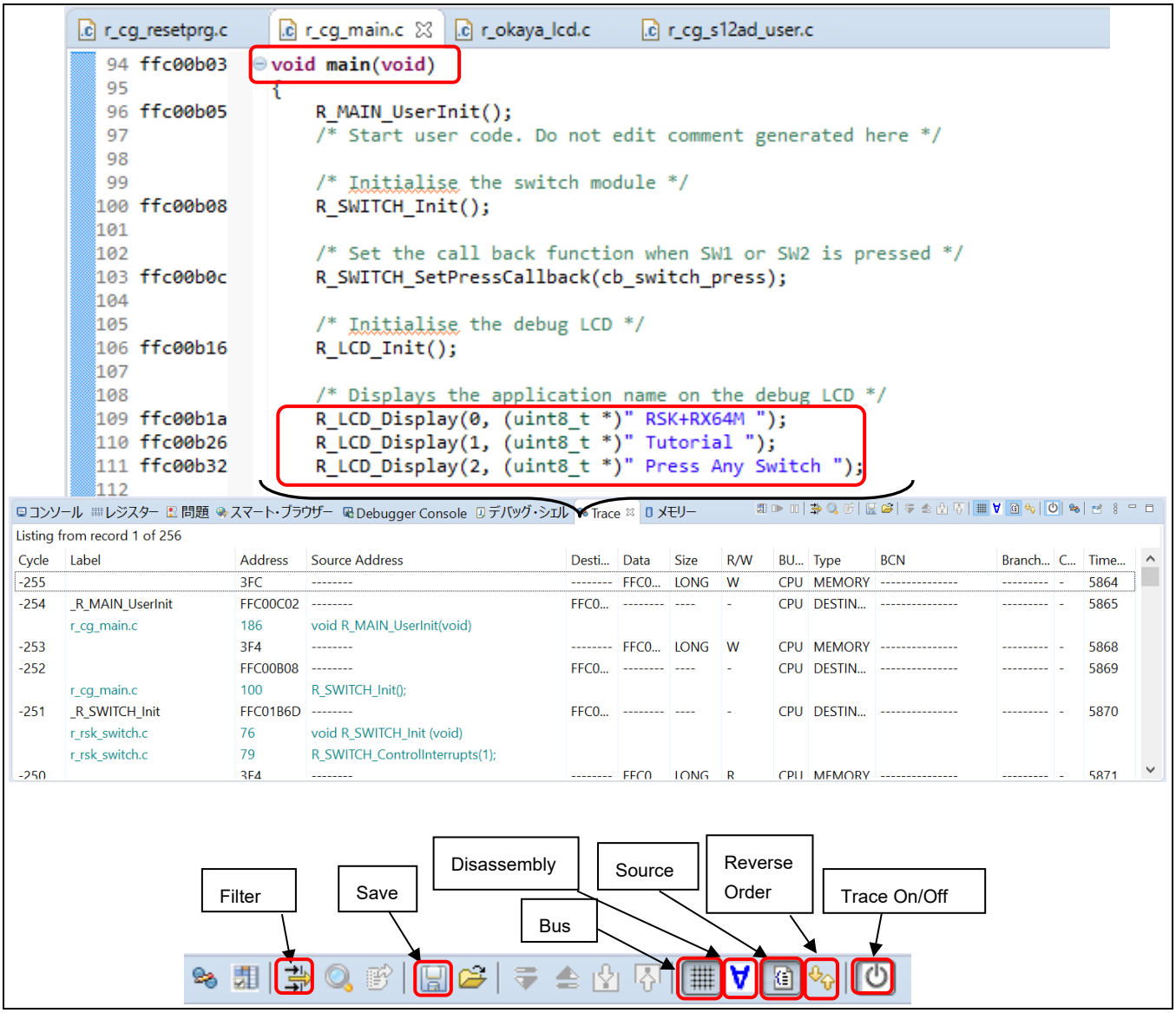

図 5-28 関数から関数までの2点間トレース

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。

注意:

RXデバイスでE2/E20エミュレータにより外部トレース機能を使うには、Mictor-38pinインターフェースを経由する必要があります。E2/E20エミュレータを使用していても14pin JTAG/FINEコネクタで接続した場合には外部トレース機能を利用することができませんのでご注意ください。RXエミュレータの接続に関する詳細は以下のドキュメントをご参照ください。

E1/E20/E2エミュレータ, E2エミュレータLiteユーザーズマニュアル別冊(RX接続時の注意事項)
https://www.renesas.com/search/keyword-search.html#q=R20UT0399

5.4.10 メモリー使用量 ビュー

[メモリー使用量] ビューが表示するのは対象プロジェクトの合計のメモリサイズ、ROM/RAMの各使用率、セクション/オブジェクト/シンボル/モジュール単位の情報、ベクタテーブルやクロスリファレンスです。

メモリー使用量ビューを使うには、

1. [ウィンドウ] → [ビューの表示] → [その他...]の [デバッグ]カテゴリから [メモリー使用量]を選択してビューを開いてください。
2. プロジェクトによって最初に表示される内容は異なります。
   a. Renesas 製ツールチェーン向けの実行可能プロジェクト(Executable project)では、メモリー使用量ビューは (1) サイズ (2) RAM/ROM 使用量とアドレス空間使用量(使用されているアドレス範囲の表示)、(3) 詳細 が表示され、画面下にはマップファイルのパスが表示されます。

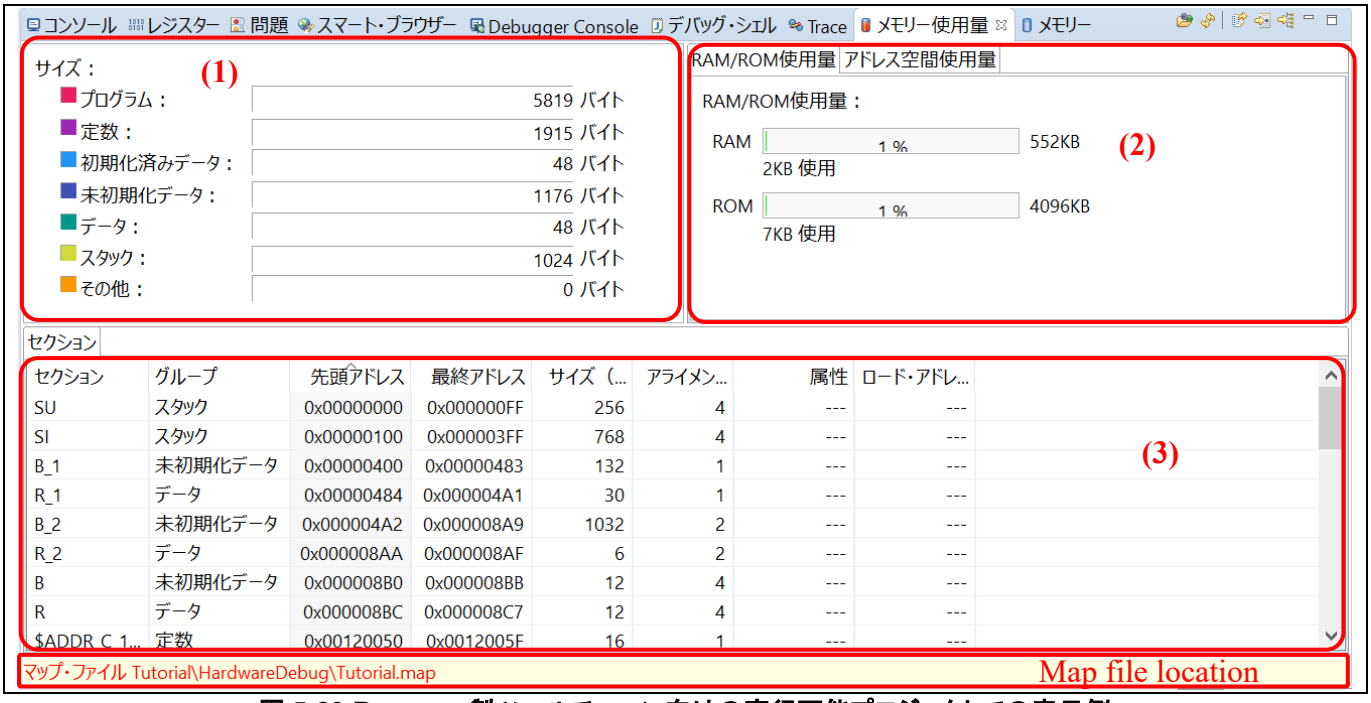

図 5-29 Renesas 製ツールチェーン向けの実行可能プロジェクトでの表示例

   b. GCC ツールチェーン向けの実行可能プロジェクトでは、(2)の RAM/ROM 使用量が”メモリ領域使用量”(Memory region 毎の使用量)に置き換わります。

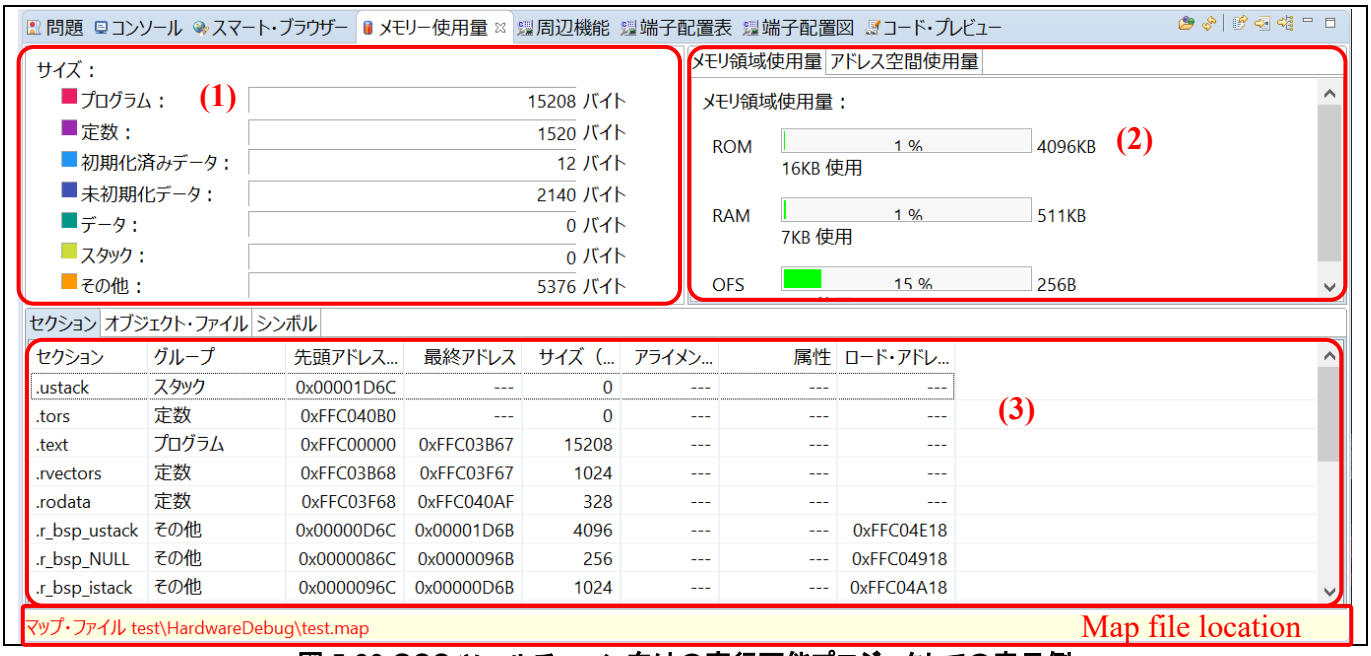

図 5-30 GCC ツールチェーン向けの実行可能プロジェクトでの表示例

c. Renesas ツールチェーン向けのライブラリプロジェクト[注]では(1)がライブラリ情報に置き換わり、画面下にはライブラリリストファイルのパスが表示されます。

注: CC-RXおよびCC-RL ツールチェーンをサポートしています。

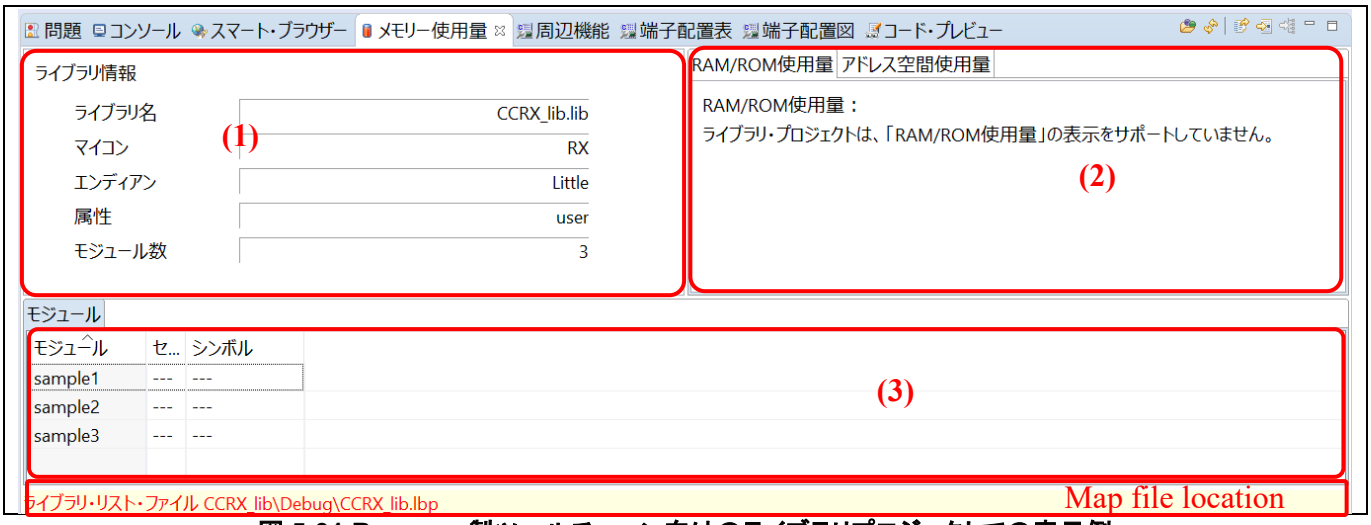

図 5-31 Renesas 製ツールチェーン向けのライブラリプロジェクトでの表示例

d. GCC ツールチェーン向けのライブラリプロジェクトでは、メモリ使用量ビューは未対応です。

メモリー使用量 ビューの各表示領域の機能を以下に示します。

サイズ:

グループ毎の合計サイズを表示します。グループはプログラム、定数、初期化済みデータ、未初期化データ、データ、スタック、その他に分類されます。

注: この機能はサポートされるツールチェインで作られた実行可能プロジェクトで利用可能です。

ライブラリ情報:

選択されたライブラリリストファイル(*.lbp)の情報を、以下の項目として表示します。

- ライブラリ名

- デバイス種別(デバイスファミリ)
- エンディアン
- 属性
- モジュール数(ライブラリを構成するオブジェクト数)

RAM/ROM 使用量:

RAM/ROM使用量をバイト数と使用率%、および使用率の棒グラフを以下の色で表示します。

- 75%未満 : 緑
- 75%以上 90%未満 : オレンジ
- 90%以上 : 赤

メモリ領域使用量:

GCCツールチェーンを使ったプロジェクトで、リンカスクリプトのMemory Region(MEMORYブロック)のアドレス範囲に対する使用率を表示します。それ以外はRAM/ROM使用量の機能と同様です。

アドレス空間使用量 :

プロジェクトが使用するデバイスのメモリ領域別の情報を表示します。各領域について名称、開始・終了アドレス、領域全体のサイズに対する使用量(バイト数と%)を表示します。

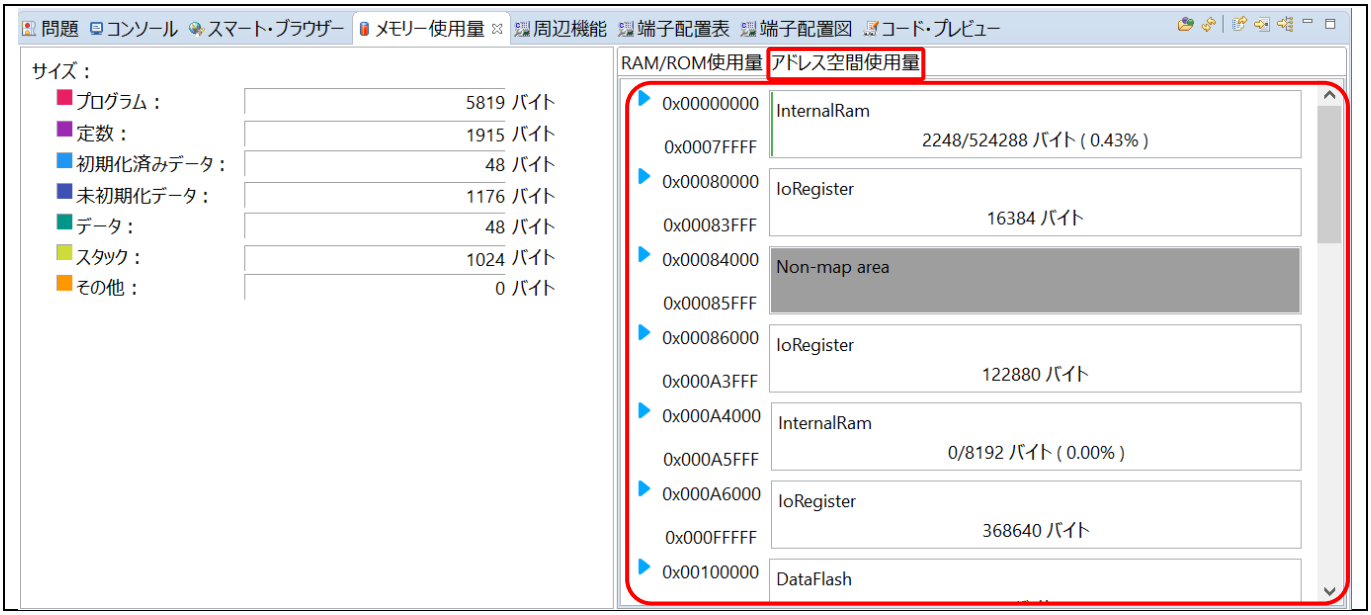

図 5-32 アドレス空間使用量の表示例

メモリ領域を ▶ で展開すると、そのメモリ領域に配置されるセクションが全て表示されます。使用量の棒グラフと同様、セクション毎の使用率で色分けされています。

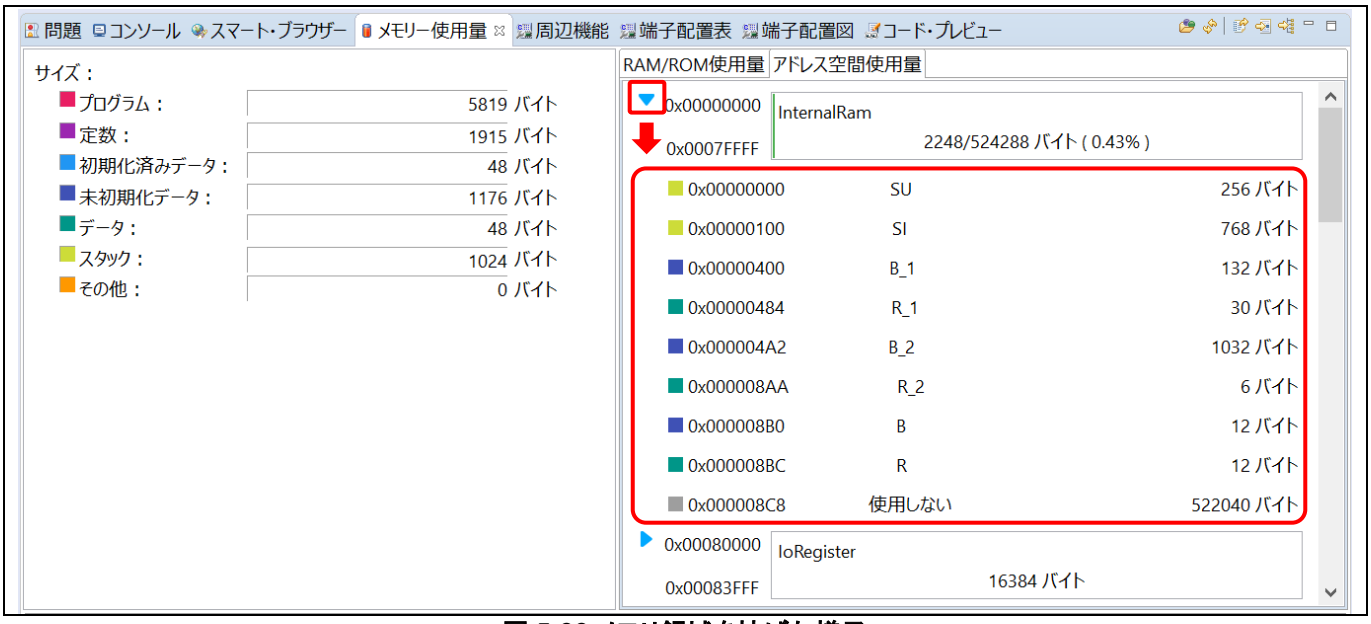

図 5-33 メモリ領域を拡げた様子

詳細表示(セクション、オブジェクトファイル、シンボルの一覧)：

アクティブなプロジェクトのマップファイルまたは指定されたマップファイルの情報を表示します。

- [セクション] タブ： マップファイルから得たセクション情報の詳細を一覧で表示します。
- [オブジェクト・ファイル] タブ ： マップファイルから得た オブジェクトファイル毎の情報を一覧表示します。
- [シンボル] タブ: マップファイルに出力されたシンボルの詳細を表示します。
- [ベクタ] タブ: マップファイルからベクタテーブルの情報を表示します。Renesas CC-RX または CC-RL のプロジェクトで、マップファイルへのベクタテーブル情報出力オプションが指定された時のみ利用できます。
- [クロスリファレンス] タブ: クロスリファレンスを表示します。実行可能プロジェクトでクロスリファレンス情報がマップファイルに出力されている場合に利用できます。
- [モジュール] タブ: ライブラリに含まれるオブジェクトの情報を表示します。Renesas CC-RX または CC-RL ツールチェインのライブラリプロジェクトでのみ利用できます。

マップファイルのパス:

メモリ使用量 ビューの下部に、マップファイル(*.map)またはライブラリリストファイル(*.lbp)のワークスペース相対パスを表示します。

6. 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。

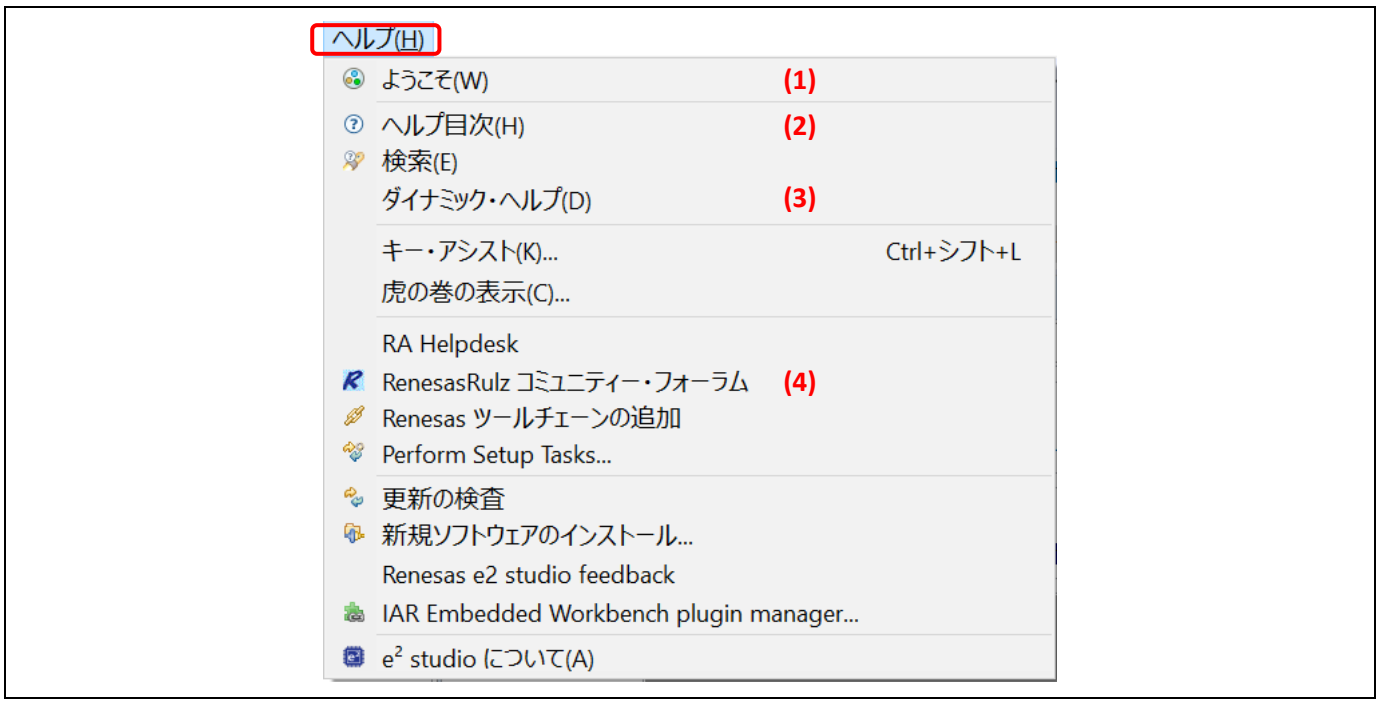

図 6-1 ヘルプ メニュー

ヘルプメニューの機能:

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。

ヘルプ目次内には多くの有用なコンテンツが入っています。例えば “e2 studioデバッグ・ヘルプ”にはデバッガの設定、ブレークポイントの使用可能数、などが書かれています。

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コミュニティー・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。

改訂記録

Rev.	発行日	改訂内容	
		ページ	概要
1.00	2020.05.30	-	RX, RL78, RH850 向けe² studio 2020-04, v7.8.0対応版として新規発行

e² studio ユーザーズマニュアル
入門ガイド

発行年月日　2020年 5月30日　Rev.1.00

発行　ルネサス エレクトロニクス株式会社

e²studio
RX/RL/RH ファミリ