User's Manual

統合開発環境
e² studio 2021-07以上
ユーザーズマニュアル クイックスタートガイド

ルネサスマイクロコントローラ
REファミリ

ルネサス エレクトロニクス
www.renesas.com

Rev.1.00 2021.09

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。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、お客様の責任において、お客様の機器・システムを設計ください。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. 当社製品または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. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 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。
6. 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。
7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因しまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 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。
9. 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。
10. 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。
11. 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。
12. 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。
13. 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。
14. 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.5.0-1　2020.10)

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。

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。

e² studio はソフトウェアパッケージ用のコンフィグレータを備えています。ソフトウェアパッケージは、組み込みシステムを開発するためのソフトウェアパッケージで、使いやすく、拡張性があり、高い品質を備えています。

RE コンフィグレータを含む e² studio は GUI によるさまざまなウィザードを備えており、コードの自動生成、ドライバの設定、ビルドやデバッグのオプション設定、作成したアプリケーションの実行などに使用できます。ドライバの情報はツールチップの形で提供され、コードエディタビュー上で参照できます。

この章では、e² studio を使用して RE ファミリ・マイクロコントローラ用アプリケーションを開発するためのシステム構成と動作環境について説明します。

本ドキュメントの概要を下記の表に示します。プロジェクトを簡単に作成し、インポートして実行するには、3章～5章を読むことをお勧めします。

本ドキュメントでは、Software Development Kit をSDKと称します。SDK は、e² studio 2021-07の [FSP Configuration] パースペクティブを使って設定できます。

表1-1 本ドキュメントの内容

章	ユーザが学習する内容	プロジェクトの作成と実行のガイド
1. 概説	開発環境の要件	△
2. インストール	IDE、特にRE開発環境とツールチェーンのインストール手順	△
3. プロジェクトの作成	ソフトウェアパッケージあり/なしの場合でのプロジェクトの作成手順およびREファミリ・マイクロコントローラ用のソフトウェアパッケージの設定手順	○（3.1、3.4節） △（上記以外の節）
4. ビルド	ビルドオプションの設定とプロジェクトのビルド手順	○（4.2節） △（上記以外の節）
5. デバッグ	e² studioの基本的なデバッグ機能の使用方法	○（5.1節） △（上記以外の節）
6. ヘルプ	ヘルプメニューの重要アイテムの使用方法	△

○：読むことをお勧めします。

△：読むことは任意です。

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。

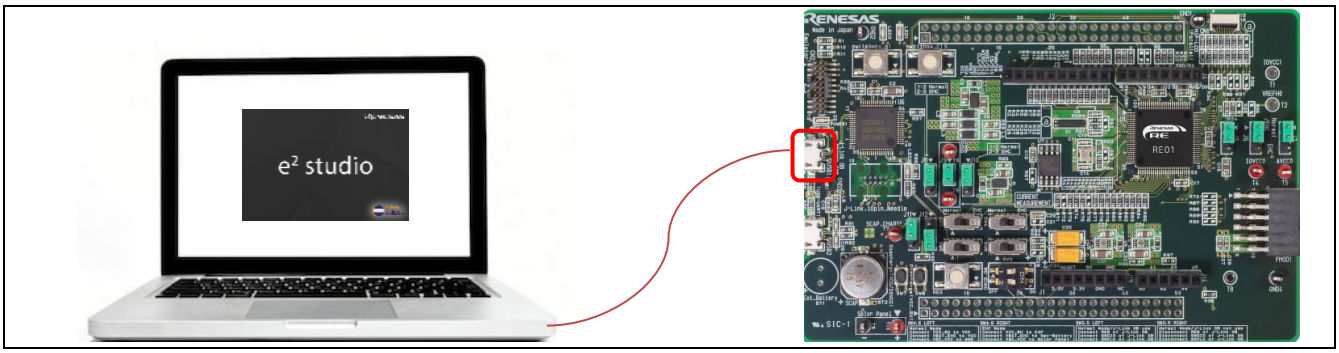


図1-1 システム構成

1.2 システム要件

1.2.1 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 以上の空きエリア
ハードディスク： 2GB 以上の空きエリア
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

1.2.2 動作環境

アーキテクチャ	Windows	e² studio
64ビットバージョン	Windows 8.1、Windows 10	2021-07

1.3 サポートするツールチェーン

GNU Arm Embedded Toolchain（バージョン：GCC V.6 GNU 6-2017-q2-update）

IAR C/C++ Compiler for ARM（バージョン：8.50.x以上）

1.4 サポートするエミュレータ

Segger J-Link、Segger J-Link OB、Renesas E2、Renesas E2 Lite

2. インストール

この章の概要を下記の表に示します。環境を設定するには、2.1節を読むことをお勧めします。環境の更新やアンインストールを行う必要がある場合は、他の節も読んでください。

表2-1 本章の概要

節	インストールのガイド
2.1 e² studioとソフトウェアパッケージのインストール	○
2.2 e² studioのアンインストール	△
2.3 e² studioの更新	△
2.4 ソフトウェアパッケージの更新	△

○：読むことをお勧めします。

△：読むことは任意です。

2.1 e² studioとソフトウェアパッケージのインストール

この章では、以下のコンポーネントをインストールする場合の手順を説明します。

- e² studio 2021-07
- GCC ARM embedded compiler
- RE用のソフトウェアパッケージ

2.1.1 e² studioのインストール

1. e²studio 2021-07オフラインインストーラを以下の場所からダウンロードしてください。
   https://www.renesas.com/e2studio
2. e² studioインストーラを起動するとe² studioインストールウィザードページが開きます。
3. 既に e² studio がインストールされている場合は、[変更]（インストール済 e² studio の変更）、[削除]（アンインストール）、[インストール]（別の場所にインストール）の選択肢が表示されます。複数のバージョンを別々の場所にインストールすることも可能です。[Next] で次に進みます。

図2-1 e2 studioのインストール ー インストール・タイプ

4. [ようこそ] ページ

デフォルトのフォルダを使用するか、あるいは [変更...] をクリックしてフォルダを変更できます。
[Next] で次に進みます。

注：e² studio のインストールフォルダ名、プロジェクト名とそのフォルダ名、およびソースファイル名にはマルチバイト文字を使用できません。

図2-2　e2 studioのインストール ― [ようこそ] ページ

5. [デバイス・ファミリ] ページ

“RE” を選択してください。必要に応じて他のデバイスファミリも選択可能です。[Next] で次に進みます。

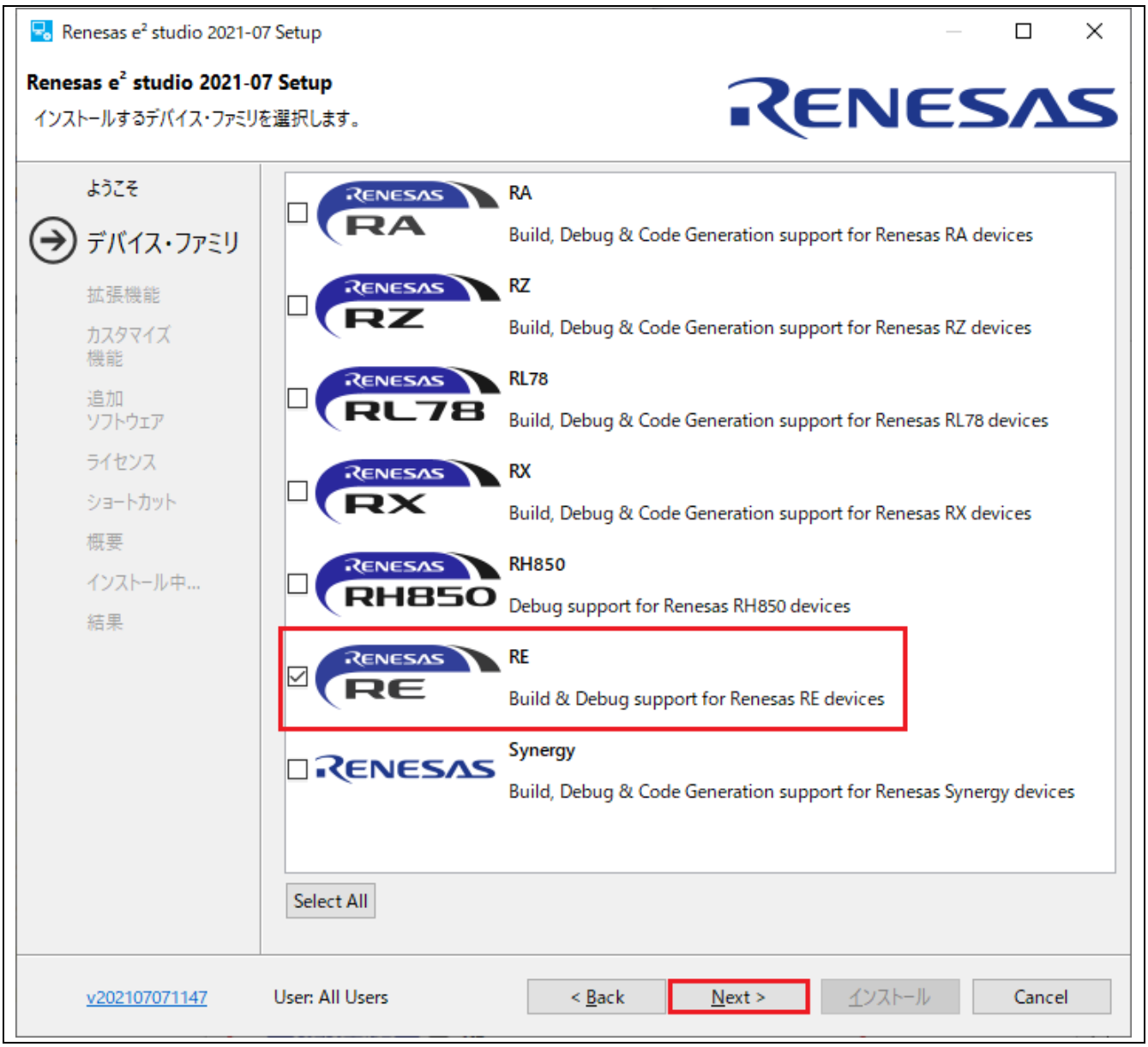


図2-3 e2 studioのインストール ― [デバイス・ファミリ] ページ

6. [拡張機能] ページ

インストールする拡張機能（言語パック、SVN & Git、RTOS のサポートプラグイン）を選択してください。
英語以外の言語のメニューを利用する場合は、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。

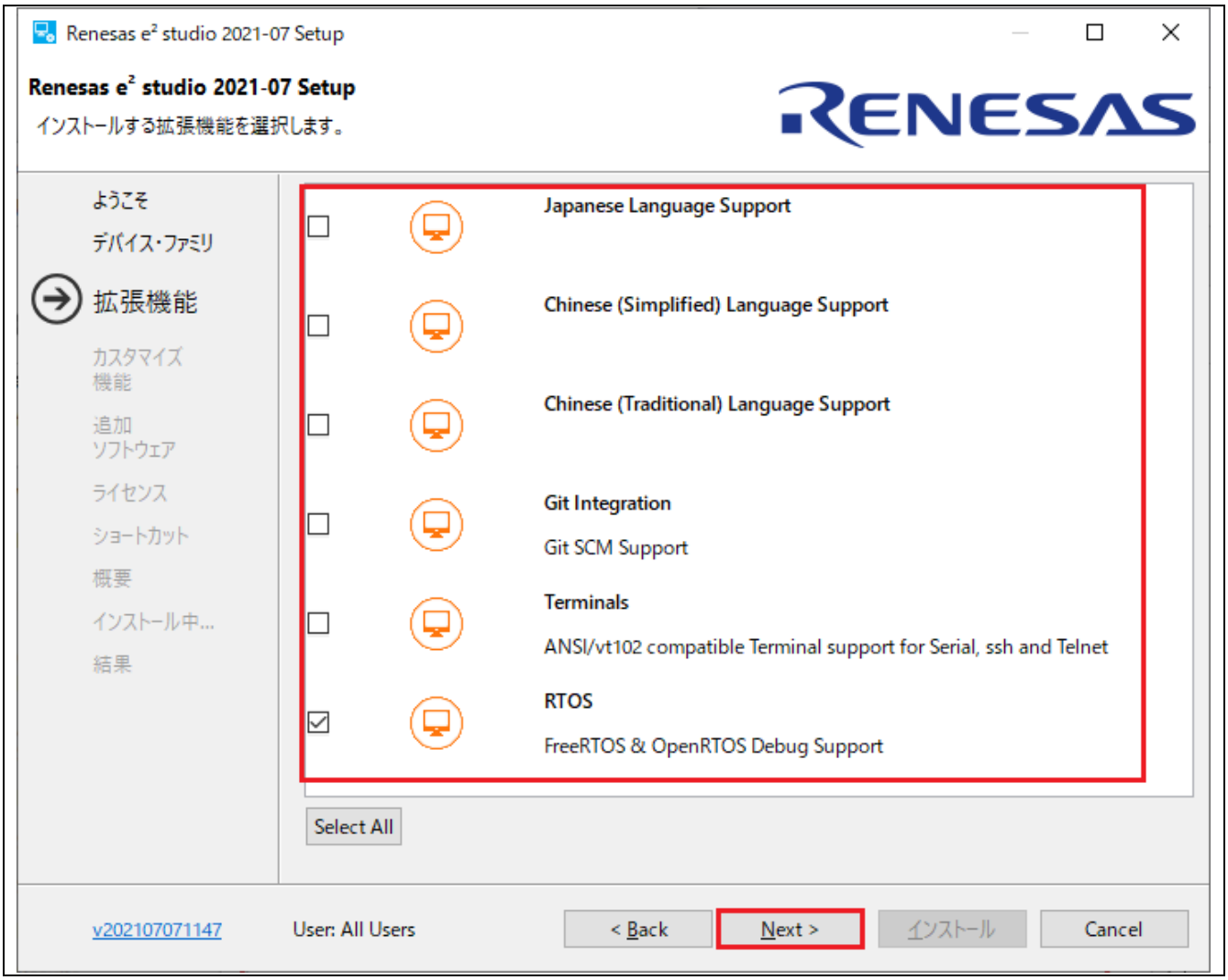


図2-4　e2 studioのインストール ― [拡張機能] ページ

7. [カスタマイズ機能] ページ

必ず“Renesas RE Family Support”と“Renesas FSP Smart Configurator”を選択してください。[Next] で次に進みます。

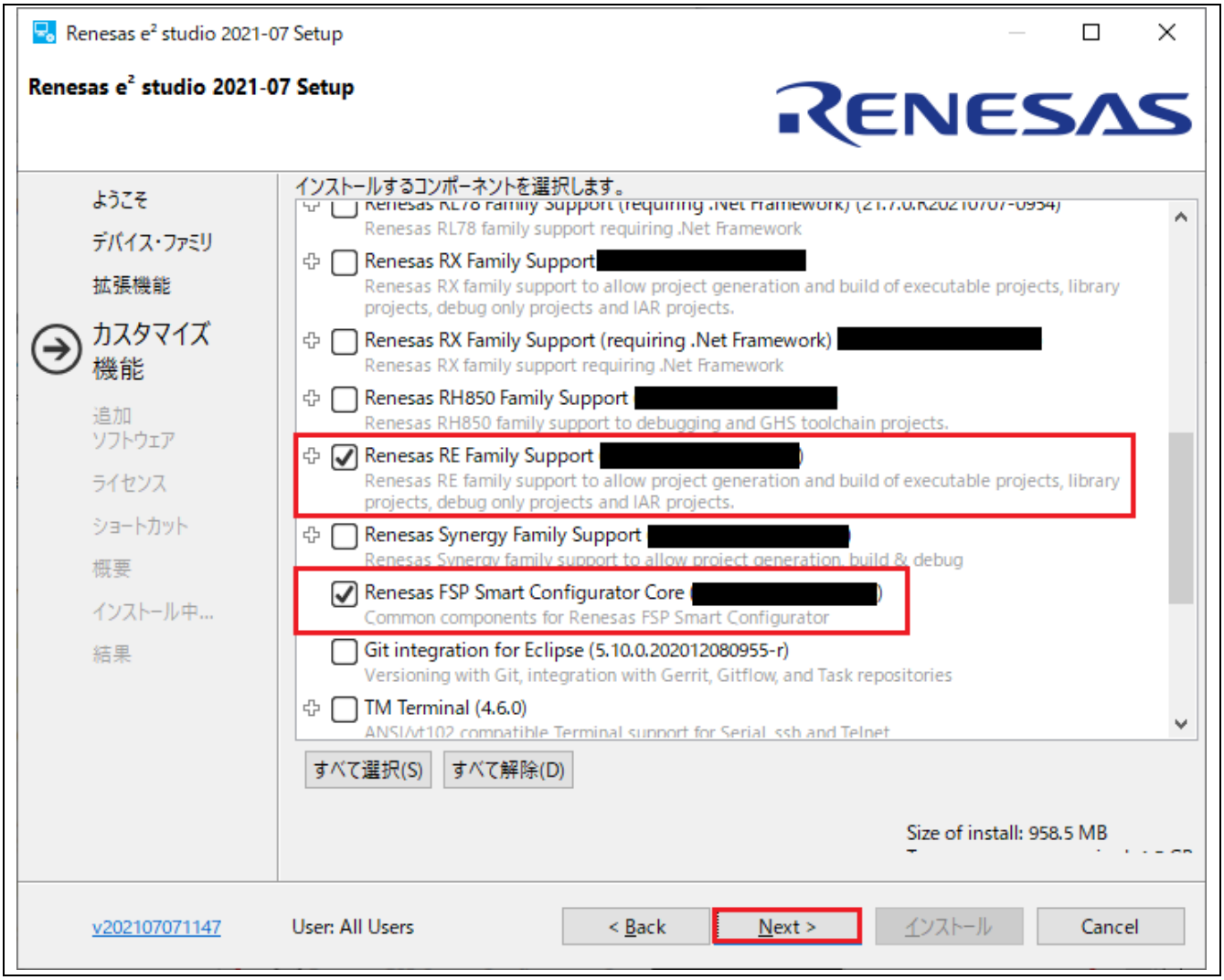


図2-5 e2 studioのインストール ― [カスタマイズ機能] ページ

8. [追加ソフトウェア] ページ

追加ソフトウェア（例：コンパイラ、ユーティリティ、QE）を選択して、[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追加ソフトウェアはインストールできません。追加ソフトウェアは後でインストールすることができます。

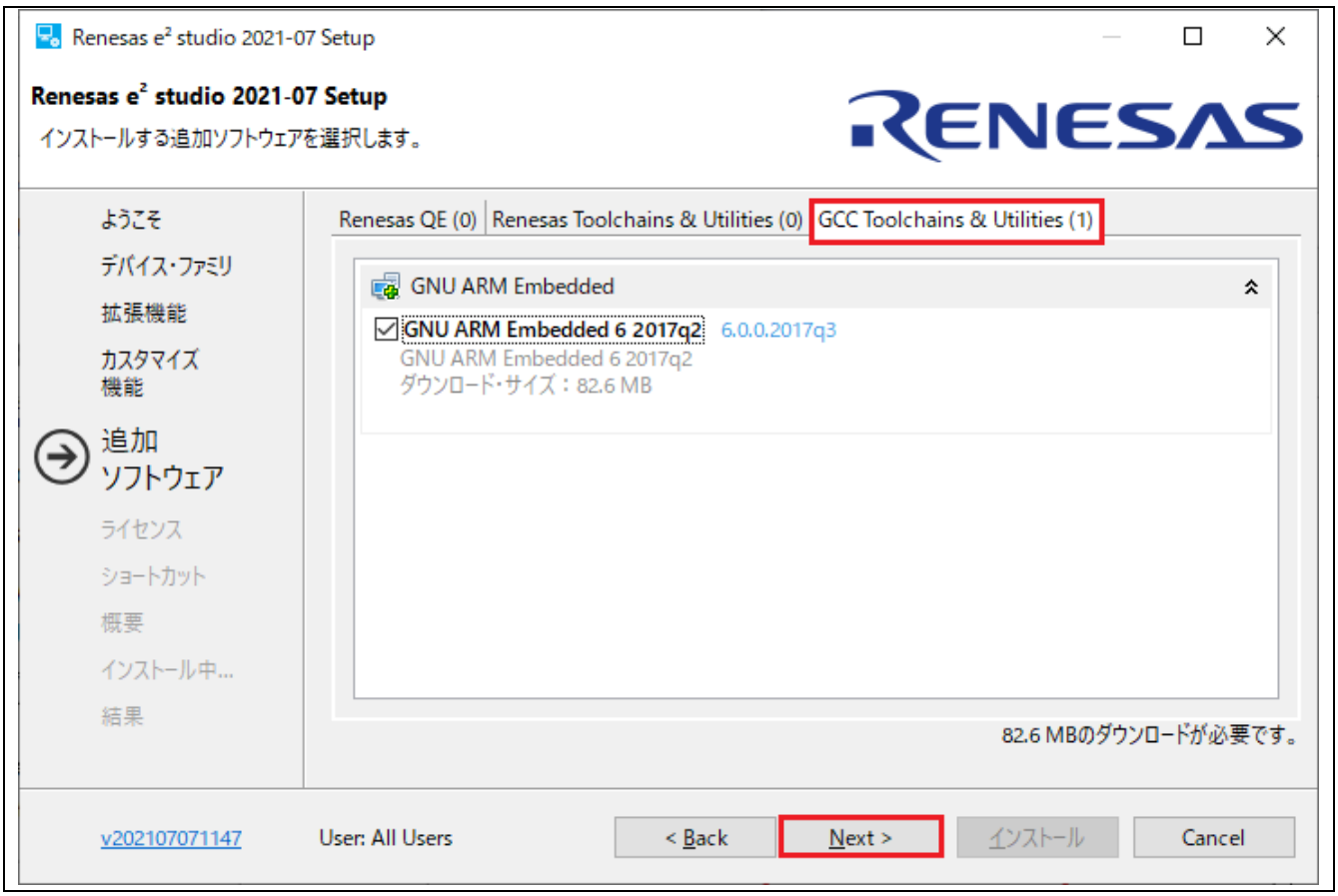


図2-6 e2 studioのインストール ― [追加ソフトウェア] ページ

9. [ライセンス] ページ

ソフトウェア契約の条項を確認したのち、同意いただけましたらチェックを入れて、[Next] で次に進みます。同意いただけない場合はインストールを継続できません。

図2-7 e2 studioのインストール ー [ライセンス] ページ

10. [ショートカット] ページ

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て、[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ることをお勧めします。

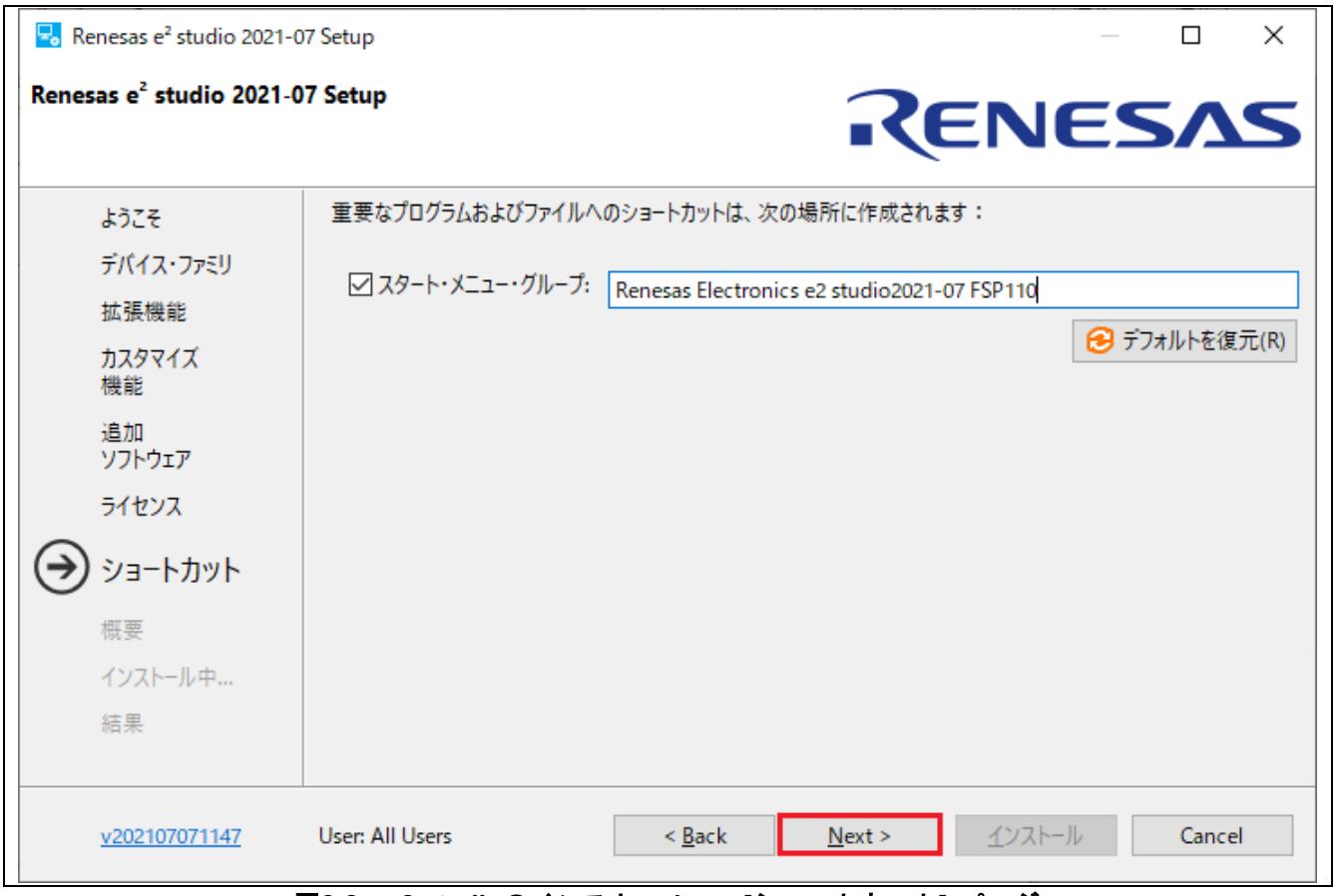


図2-8　e2 studioのインストール ― [ショートカット] ページ

11. [概要] ページ

インストールされるコンポーネント一覧を以下に示します。内容を確認した後、[インストール] ボタンを押して e² studio をインストールしてください。

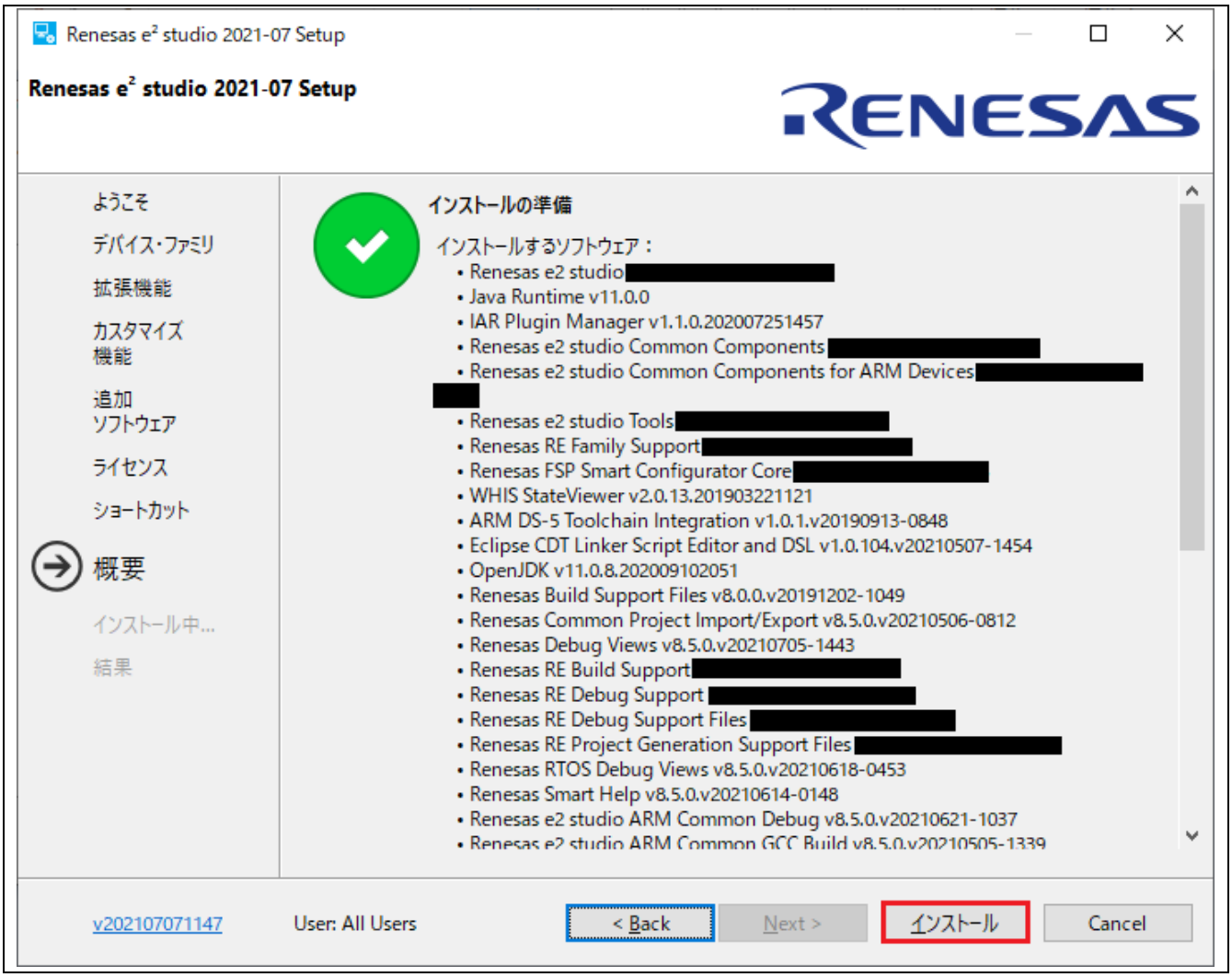


図2-9　e2 studioのインストール ― [概要] ページ

12. [インストール中...] ページ

インストールが実行されています。[追加ソフトウェア] ページで選択したソフトウェアに基づいて、それらをインストールするための新しいダイアログが開きます。

13. [結果] ページ

ここでインストール結果が表示されます。エラーが表示されてないことを確認してください。

[OK] をクリックして、インストールを完了します。

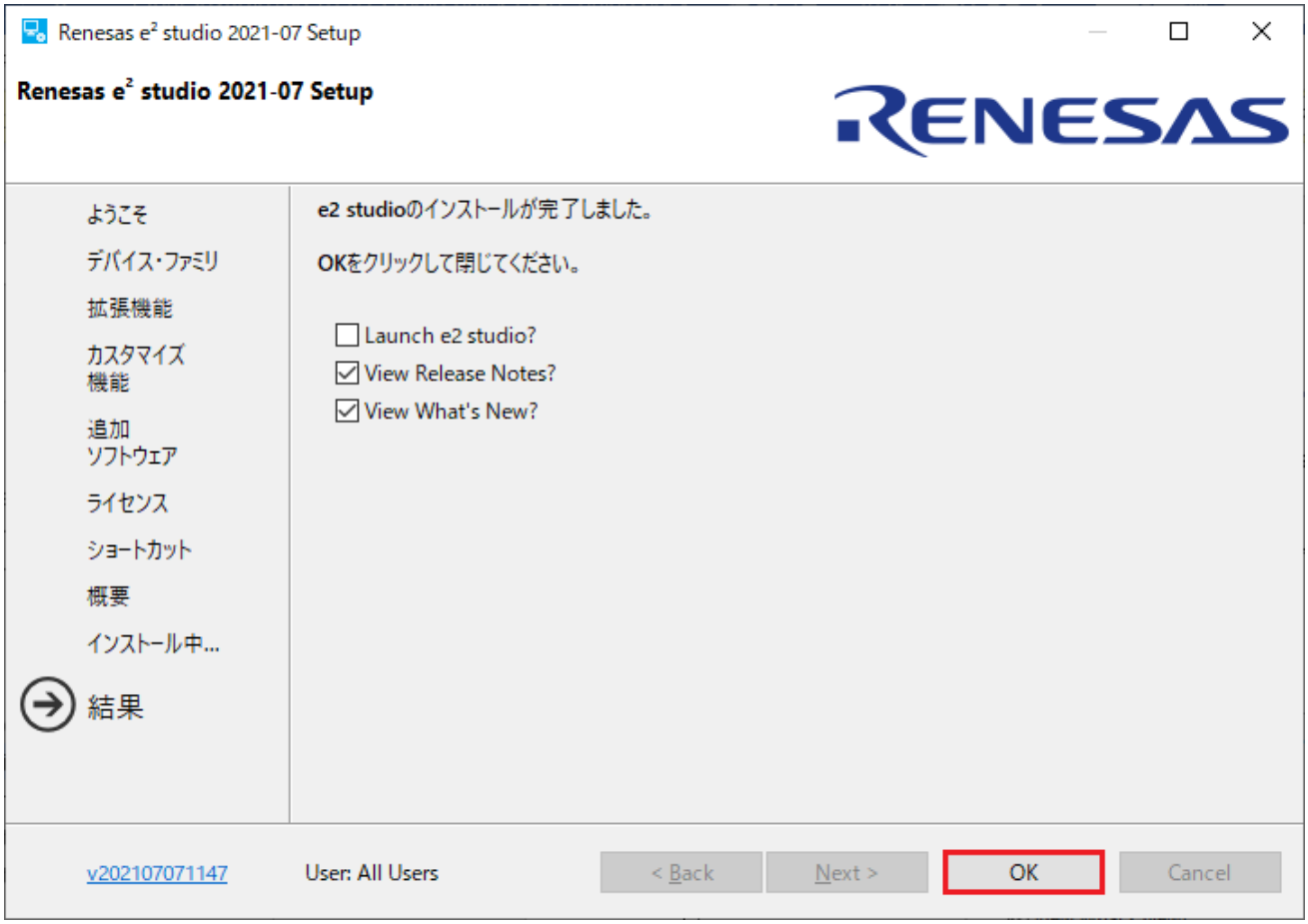


図2-10　e2 studioのインストール ― [結果] ページ

2.1.2 GNU Arm Embedded Toolchainのインストール

GNU Arm Embedded Toolchain は e² studio のインストールと同時にインストールできます。また、同時にインストールせずに、e² studio のインストール後に別途インストールすることもできます。

GNU Arm Embedded Toolchain を別途インストールするには、以下の手順を実行してください。

1. ルネサス RE ファミリがサポートする GNU Arm Embedded Toolchain の 6-2017-q2-update バージョン（gcc-arm-none-eabi-6-2017-q2-update-win32.exe）を次の場所からダウンロードします。
   https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads
2. ホスト PC で GNU Arm Embedded Toolchain のインストーラを実行します。
3. インストールする言語を選択します。インストール確認の画面で [Yes] を選択してください。
4. インストールウィザードでは、すべてデフォルト設定のままにします。
5. [Install wizard Complete]の画面が表示されたら、“Add path to environment variable”、“Add registry information”、および“Launch gccvar.bat”を選択し、[Finish] ボタンを押してインストールを完了します。

2.1.3 RE用ソフトウェアパッケージのインストール

ソフトウェアパッケージはe² studioのインストールと同時にインストールできます。また、同時にインストールせずに、e² studioのインストール後に別途ソフトウェアパッケージをインストールすることもできます。

ソフトウェアパッケージを別途インストールするには、以下の手順を実行してください。

(1) https://www.renesas.com/software-tool/re-software-development-kit にアクセスしてください。
(2) 最新の SDK（例：“r01an5970xx0110-re-sdk.zip ”）をダウンロードしてください。このファイルを解凍して、最新のソフトウェアパッケージインストーラ（例：“RE_SDK_Packs_<version>.exe”）を入手します。ソフトウェアパッケージインストーラには、ドライバと HTML 版のユーザーズマニュアルが含まれています。
(3) 互換性のある e² studio がインストール済みであること、e² studio が実行中でないことを確認してください。また、インストール中は e² studio を実行しないでください。
(4) ソフトウェアパッケージインストーラを実行してください。[Next] で次に進みます。
(5) [I Agree] をクリックして、契約条項に同意してください。
(6) e² studio をインストールしたフォルダ（例：C:\Renesas\e2_studio）を選択し、[Install] ボタンを押してください。
(7) [Finish] ボタンを押してインストールを終了します。

2.2 e² studioのアンインストール

e² studioプログラム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] → [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、“Renesas e² studio” を右クリックし、[アンインストール(U)] ボタンを押してください。
3. [アンインストール] ダイアログの [はい] ボタンを押して削除を実行してください。

アンインストール処理の最後に、e² studioはインストール先から削除され、Windowsの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の[プログラムと機能]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

2.3 e² studio の更新

e² studio を更新するには、新しいバージョンの e² studio インストーラ（標準の e² studio インストーラ）を実行します。インストーラのダウンロード手順は「2 章　インストール」を参照してください。

インストール済みのバージョンを上書きしないように注意してください。e² studio の更新前に、旧バージョンをアンインストールします。旧バージョンと新バージョンを両方使用したい場合は、新バージョンを別の場所にインストールしてください。

2.4 ソフトウェアパッケージの更新

ソフトウェアパッケージを更新するには、新しいバージョンのソフトウェアパッケージインストーラを実行します。インストーラのダウンロード手順は、「2.1.3　RE 用ソフトウェアパッケージのインストール」を参照してください。

3. プロジェクトの作成

この章では、新規 RE プロジェクトの作成について説明します。e² studio は、新規 RE プロジェクトをすぐに作成できるようなウィザードを用意しています。このウィザードは、使用する RE デバイスとユーザボードに適したプロジェクトを作成できます。

プロジェクトジェネレータでは、端子構成、割り込み、クロック構成を設定可能で、必要なドライバソフトウェアも設定できます。

プロジェクト作成の前に、「2 章　インストール」の説明にしたがってソフトウェアパッケージとツールチェーンがホスト PC にインストールされていることが必要です。

この章の概要を下記の表に示します。e² studio で RE プロジェクトを作成し、インポートするには、3.1 節、3.4 節を読むことをお勧めします。スマート・コンフィグレータを使用したプロジェクト作成、スマート・コンフィグレータを使用しないプロジェクト作成、およびスタティックライブラリプロジェクトの作成に関する情報が必要な場合、他の節も読んでください。

表3-1　本章の概要

節	ユーザが学習する内容	プロジェクト作成のガイド
3.1 スマート・コンフィグレータを使用した新規プロジェクトの作成	スマート・コンフィグレータを使用して新規プロジェクトを作成する手順	○
3.2 スマート・コンフィグレータを使用しない新規実行プロジェクトの作成	スマート・コンフィグレータを使用せずに新規プロジェクトを作成する手順	△
3.3 REスタティックライブラリの作成と使用	スマート・コンフィグレータを使用してスタティックライブラリプロジェクトを作成し、実行プロジェクトから使用する手順	△
3.4 既存プロジェクトのワークスペースへのインポート	既存プロジェクトをe² studioにインポートする方法	○
3.5 コンフィグレーションエディタ	スマート・コンフィグレータを使用してプロジェクトを設定する手順	△

○：読むことをお勧めします。

△：読むことは任意です。

3.1 スマート・コンフィグレータを使用した新規プロジェクトの作成

1. [ファイル] → [新規] → [Renesas C/C++ Project] → [Renesas RE] の順に選択して、新規プロジェクトを作成するためのウィザードを起動します。

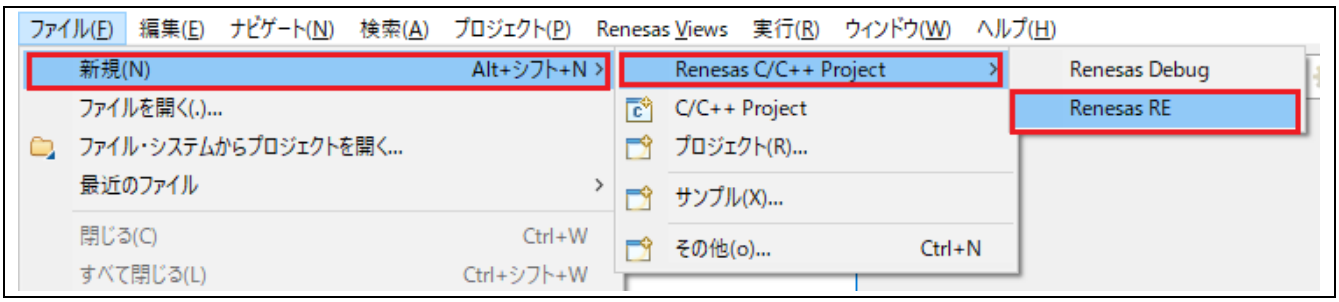


図3-1 新規プロジェクト作成用ウィザードの起動

2. [All] で [Renesas RE C/C++ SDK Project] テンプレートを選択してください。[次へ] で次に進みます。

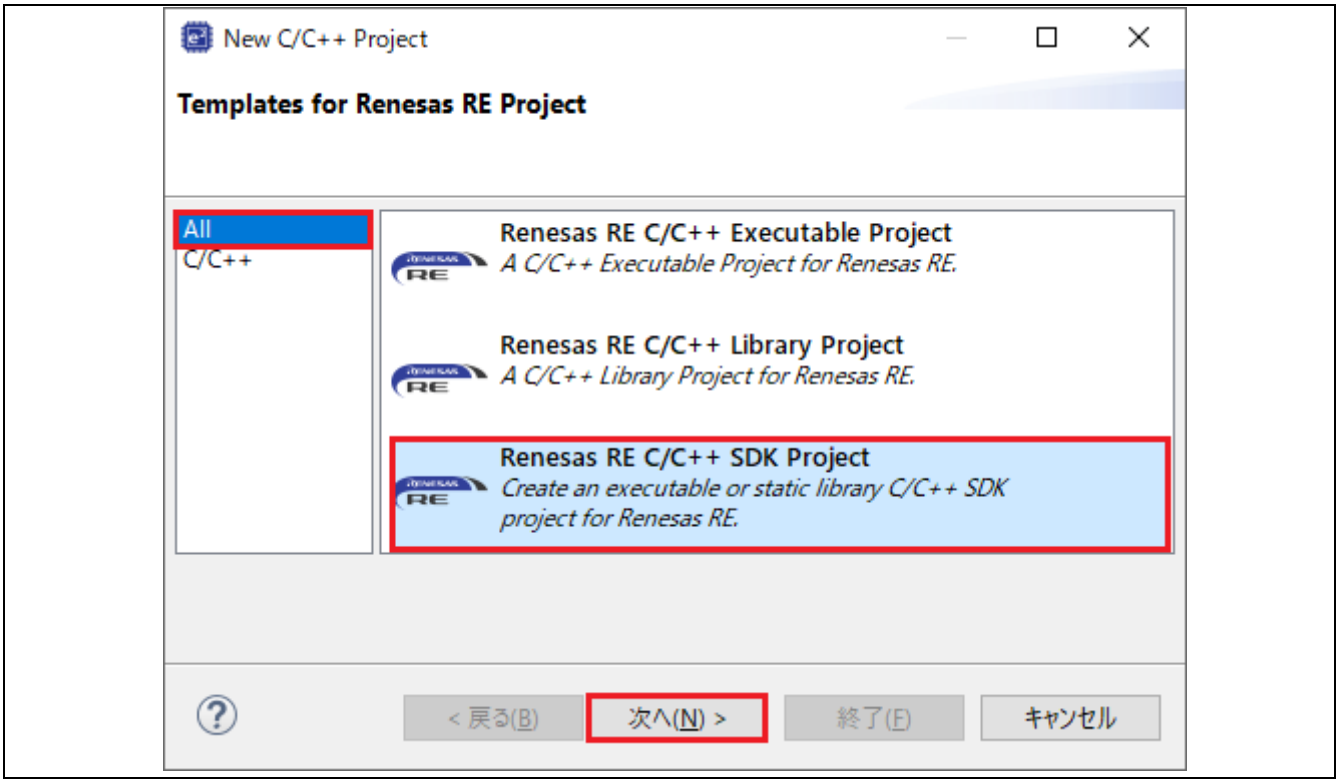


図3-2 プロジェクトの作成 ― テンプレートの選択

3. プロジェクト名を入力します。[次へ] で次に進みます。

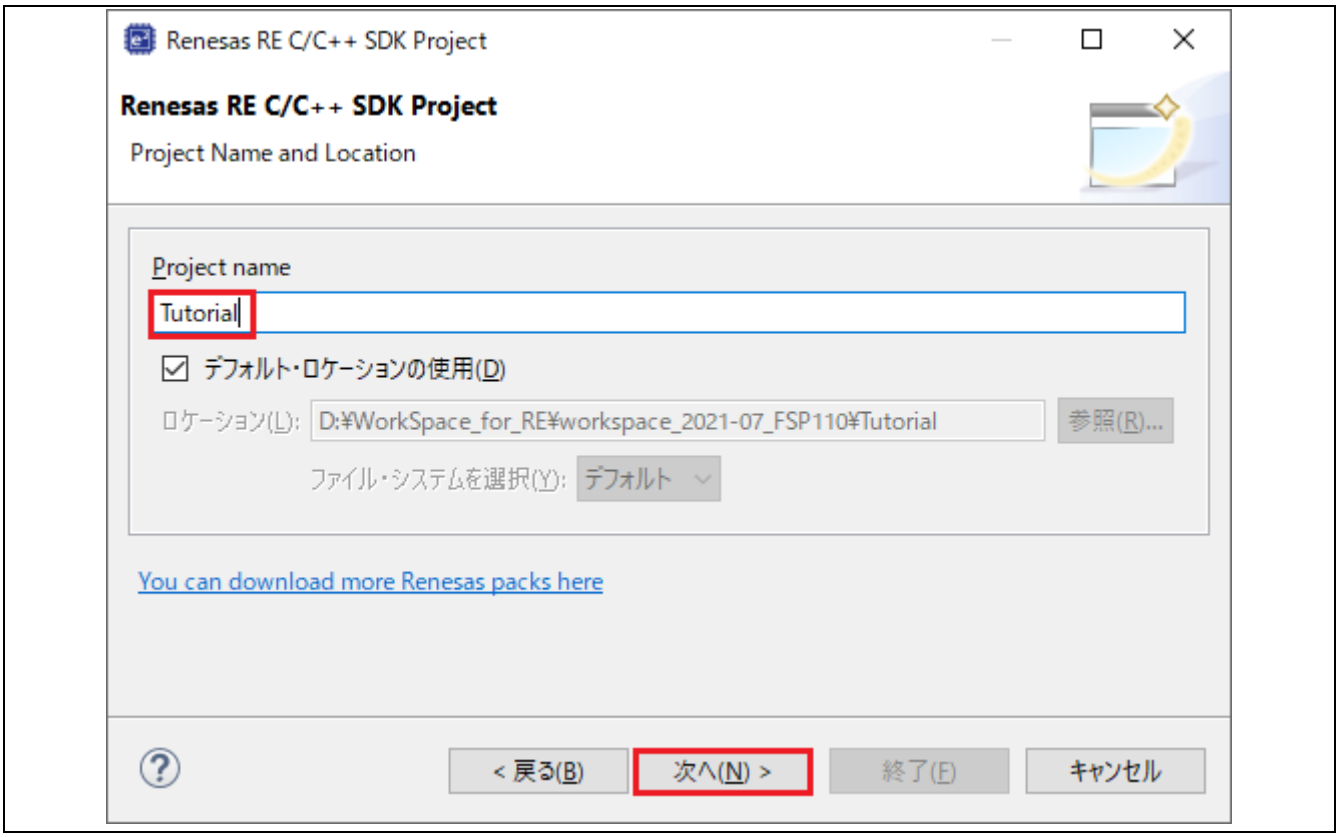


図3-3 プロジェクトの作成 ― プロジェクト名の指定

4. デバイス選択の画面で、デバイスとツールの情報を入力します。

   - Board：EK-RE01 1500KB
   - Toolchains：ルネサスREファミリ用に認定された最新のGNU Arm Embedded Toolchainを選択します（例：GNU ARM Embedded 6.3.1.20170620）。

   その他はデフォルト設定のままにします。[次へ] で次に進みます。

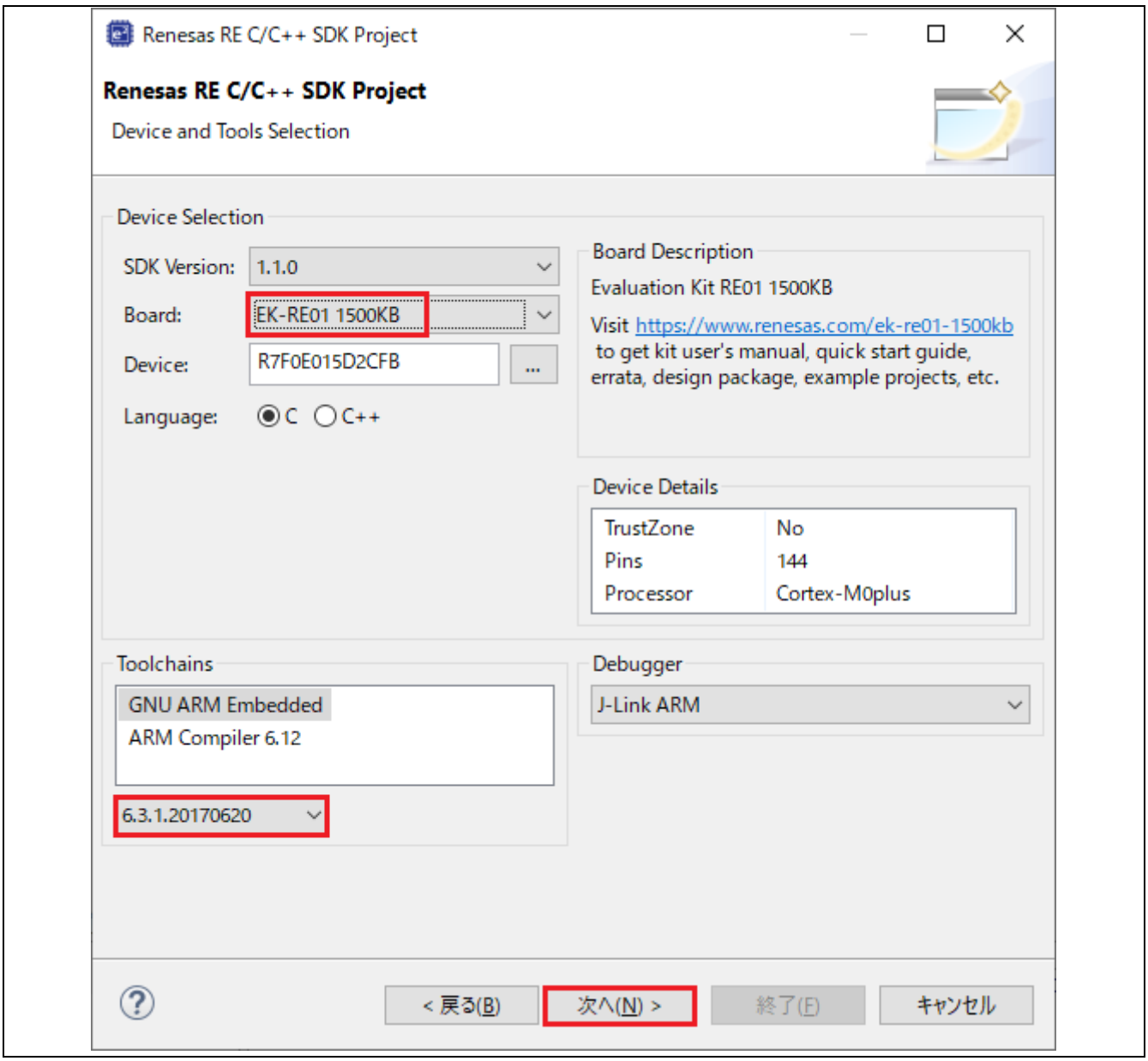


図3-4　プロジェクトの作成 ― デバイスとツールの選択

5. ビルドアーティファクト選択の画面で、[Executable]または[Static Library]のビルドアーティファクトを選択します。[次へ] で次に進みます。

注：REファミリ用のFreeRTOSは現在ありません。

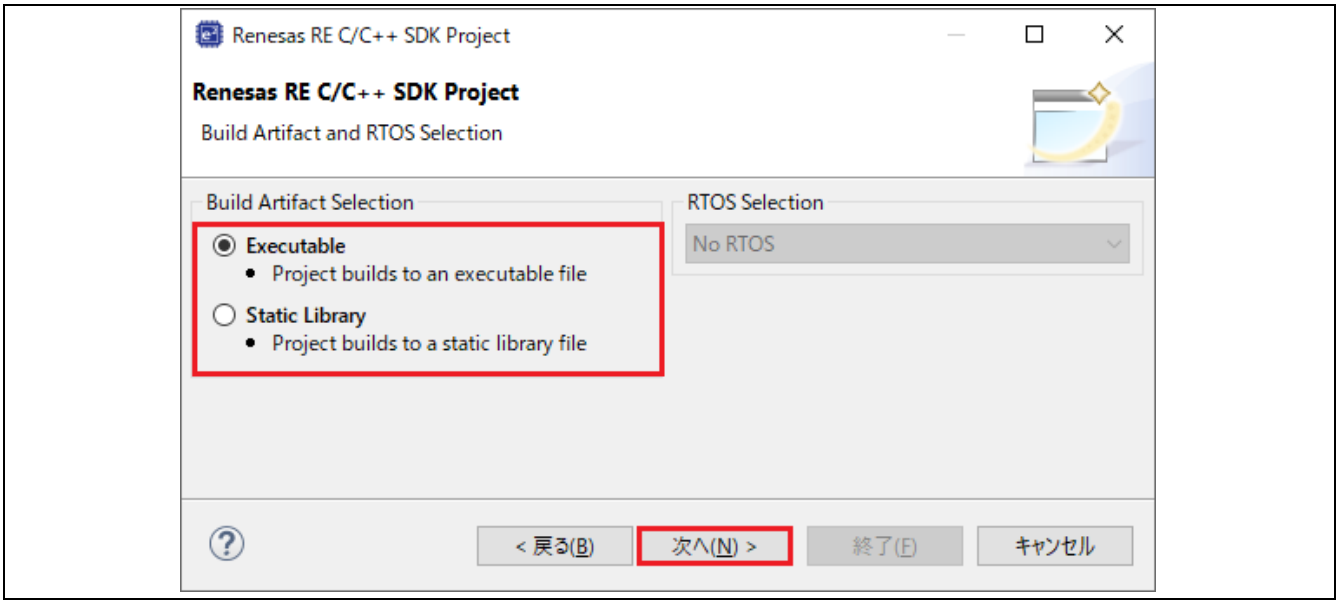


図3-5　プロジェクトの作成 ― ビルドアーティファクトの選択

6. プロジェクトテンプレート選択の画面で、作成されるプロジェクトテンプレートを選択します。

- Bare Metal – Blinky：BSPおよびLED点滅サンプルプログラムを含むREプロジェクト
- Bare Metal – Minimal：BSPを含むがサンプルプログラムを含まないREプロジェクト

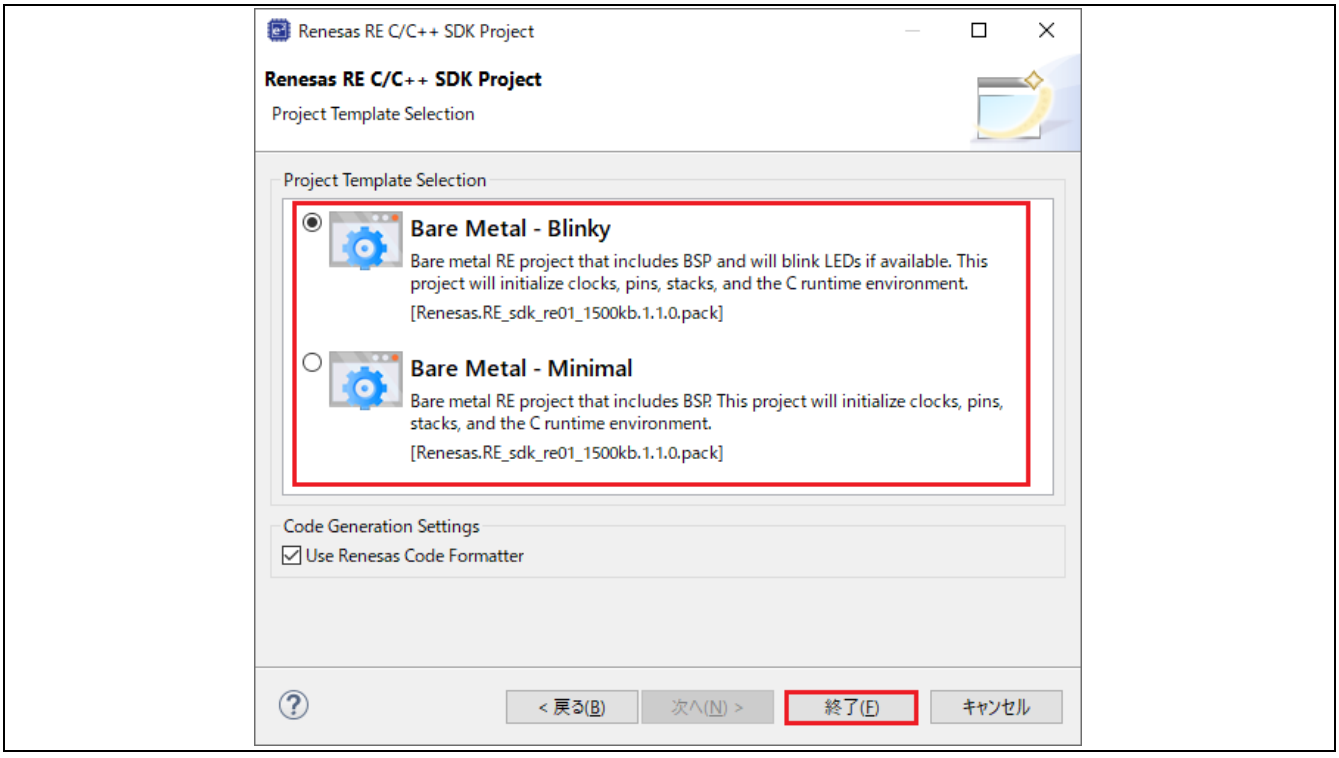


図3-6　プロジェクトの作成 ― プロジェクトテンプレートの選択

7. [終了] ボタンを押すとプロジェクトが作成されます。

[FSP Configuration] パースペクティブを開くようメッセージが表示される場合があります。[パースペクティブを開く] を選択して開いてください。

（Eclipse で「パースペクティブ」とは、あらかじめペインとビューを組み合わせてレイアウトを定義したものを指します。）

e² studio が作成する新規プロジェクトには、[プロジェクトエクスプローラー] ビュー、[コンフィグレーションエディタ] ビュー、[パッケージ] ビューなど各種ビューが用意されています。

注：[パッケージ] ビューを見るためには、コンフィグレーションエディタで [Pins] タブを選択する必要があります。

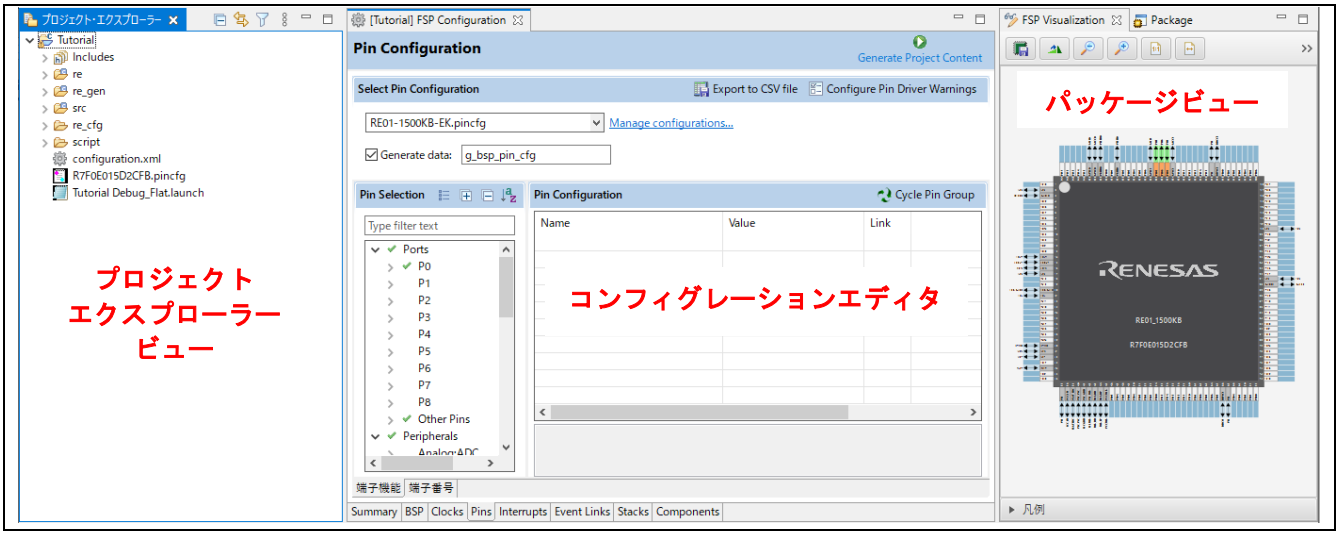


図3-7 プロジェクトの作成 ― 新規プロジェクト作成のビュー

3.2 スマート・コンフィグレータを使用しない新規実行プロジェクトの作成

注：REファミリ・マイクロコントローラを使用するプロジェクトにはソフトウェアパッケージの使用をお勧めします。スマート・コンフィグレータを使用することで、簡単にプロジェクトを設定できたり、デバイスのカスタマイズ設定や手動設定によって生じる特有の問題を回避できたりします。

e² studioを起動し、ワークスペースフォルダを選択します。以下の手順で新規REプロジェクトを作成してください。

1. [ファイル] → [新規] → [Renesas C/C++ Project] → [Renesas RE] の順に選択して、新規プロジェクトを作成するためのウィザードを起動します。

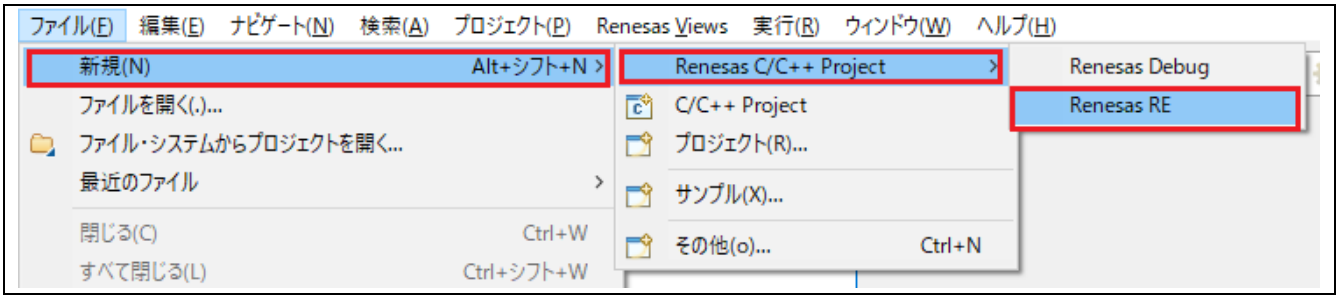


図3-8 新規プロジェクト作成用ウィザードの起動

2. [All] で [Renesas RE C/C++ Executable Project] テンプレートを選択してください。[次へ] で次に進みます。

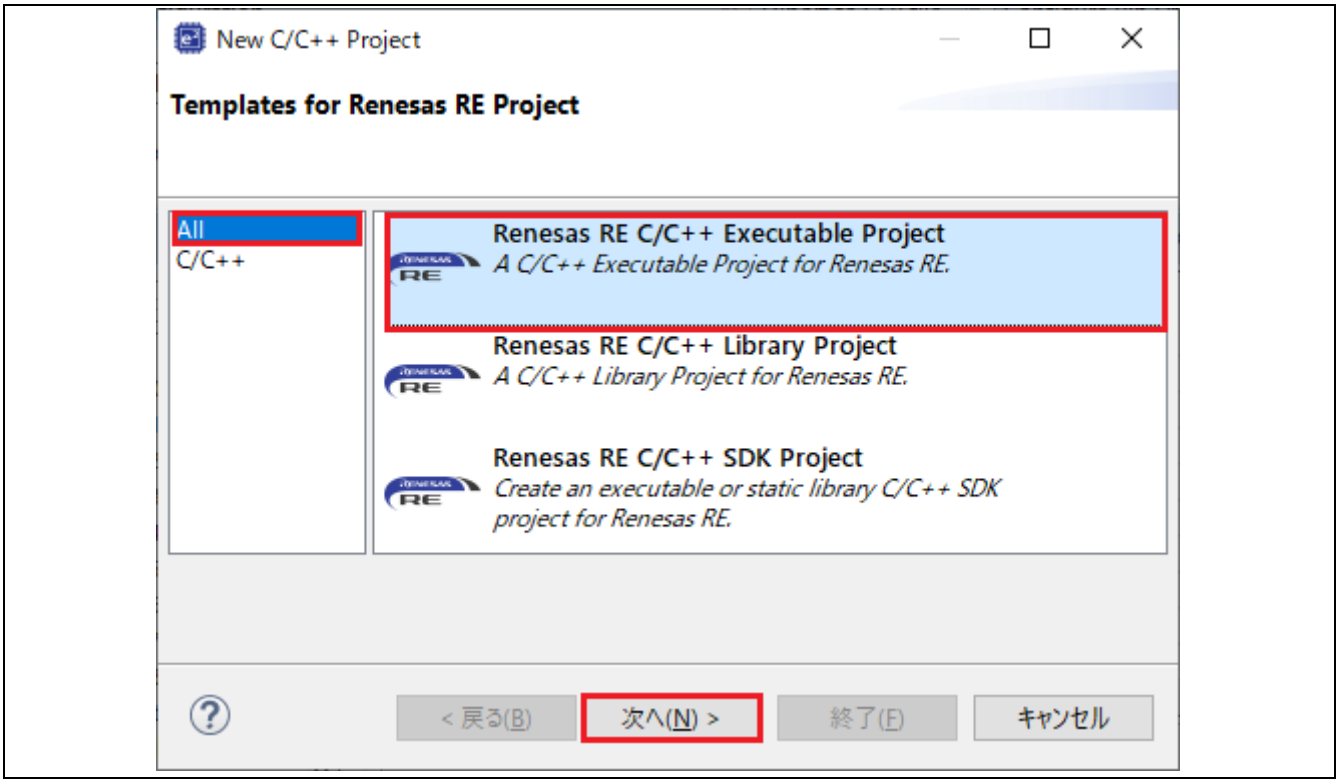


図3-9 新規プロジェクト作成用ウィザード(1/3)

3. プロジェクト名（例：Tutorial）を入力します。[次へ] で次に進みます。

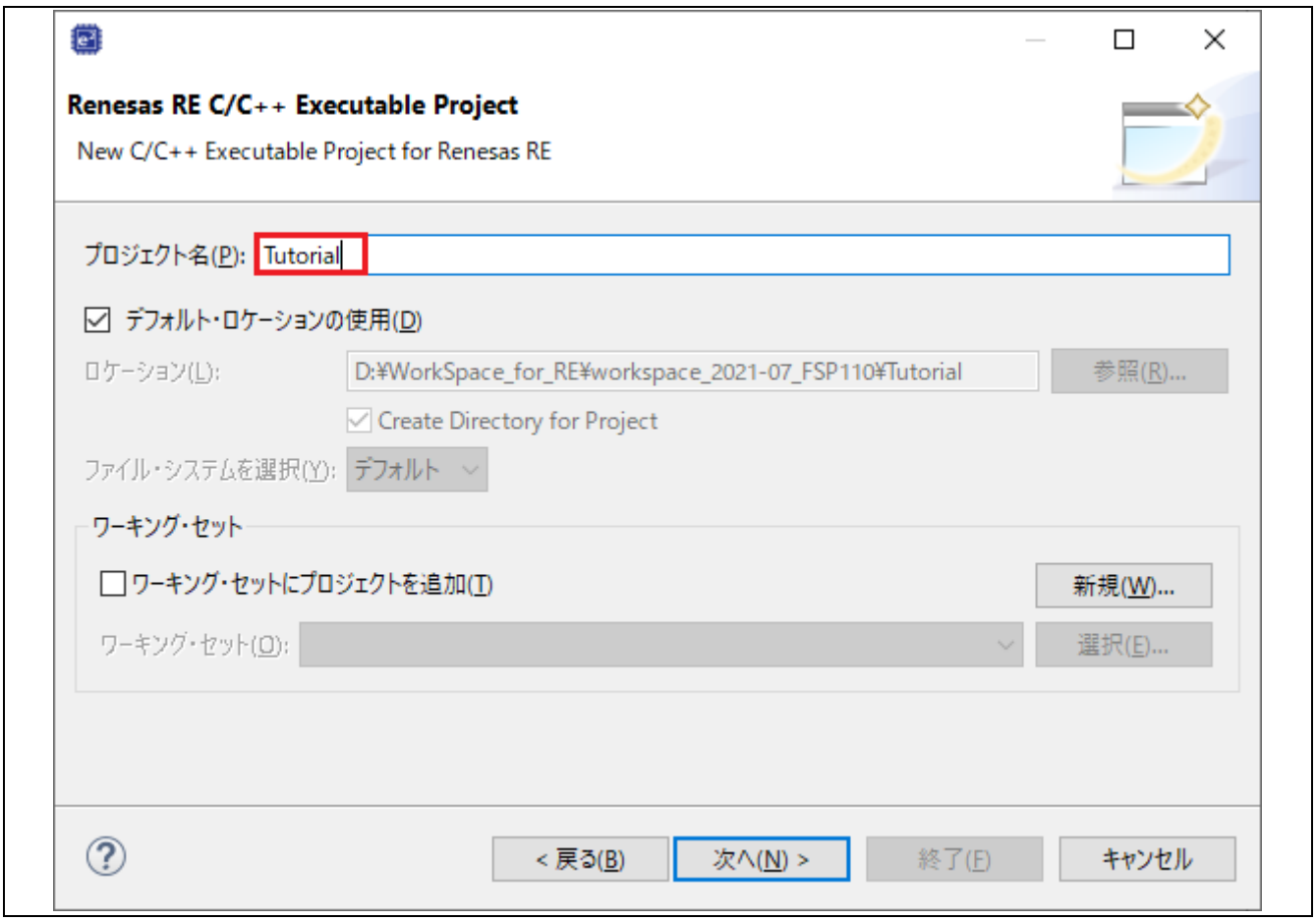


図3-10　新規プロジェクト作成用ウィザード(2/3)

4. デバイスとツールの情報を入力します。

- Target Device：R7F0E015D2CFB（例）
- Toolchain Settings：ルネサス RE ファミリ用に認定された最新の GNU Arm Embedded Toolchain を選択します（例：GNU ARM Embedded 6.3.1.20170620）。
- デバッガ：J-Link (ARM)
- その他はすべてデフォルト設定のままにします。[終了] ボタンを押すとプロジェクトが作成されます。

注：E2やE2 Liteは、J-Linkと同様に [Create Hardware Debug Configuration] プルダウンメニューから選択できます。

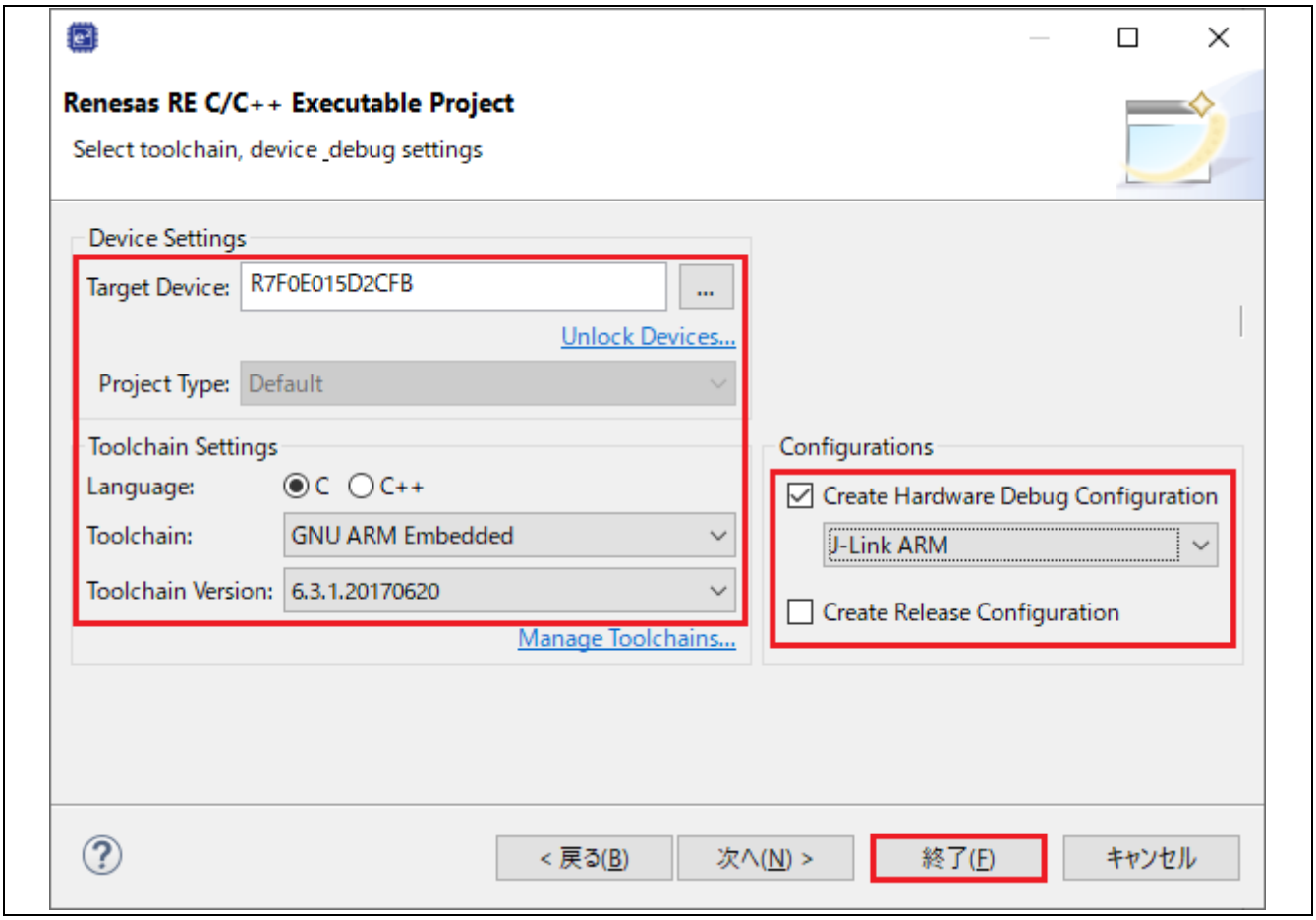


図3-11　新規プロジェクト作成用ウィザード(3/3)

5. “Tutorial”という名前の新規のCプロジェクトが作成されます。

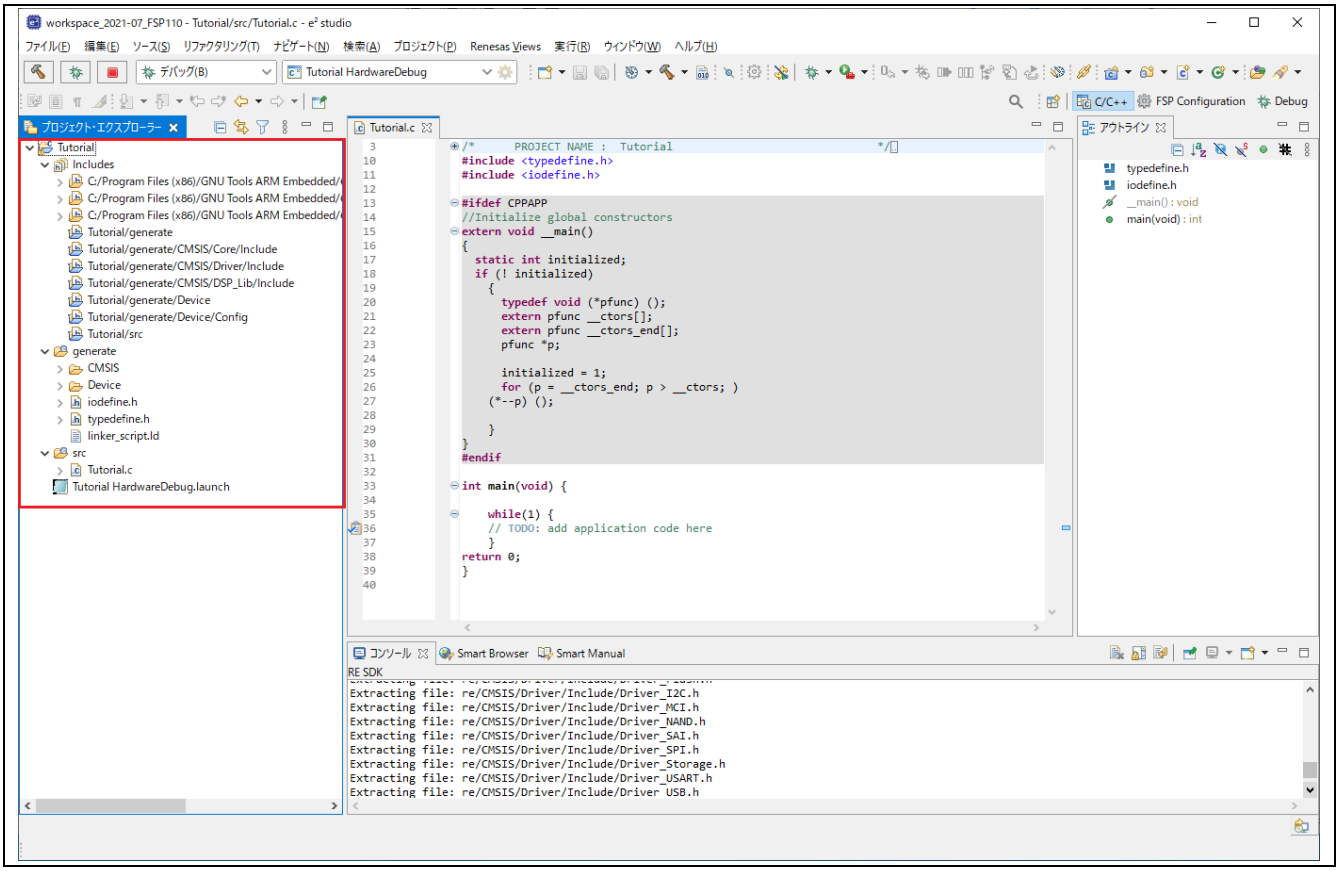


図3-12 作成された新規のCプロジェクト

3.3 REスタティックライブラリの作成と使用

この章ではREスタティックライブラリのプロジェクトの作成と、作成したライブラリプロジェクトを参照する実行プロジェクトの作成について説明します。

3.3.1 スタティックライブラリプロジェクトの作成

REスタティックライブラリのプロジェクトを作成する手順例を以下に示します。

1. 「3.1 スマート・コンフィグレータを使用した新規プロジェクトの作成」の説明にしたがい、以下のオプションを使用して、新規のSDKプロジェクトを作成します。

   - ステップ3でプロジェクトに名前を付けます（例：RE_Lib）。
   - ステップ5で [Static Library] をビルドアーティファクトとして選択します。

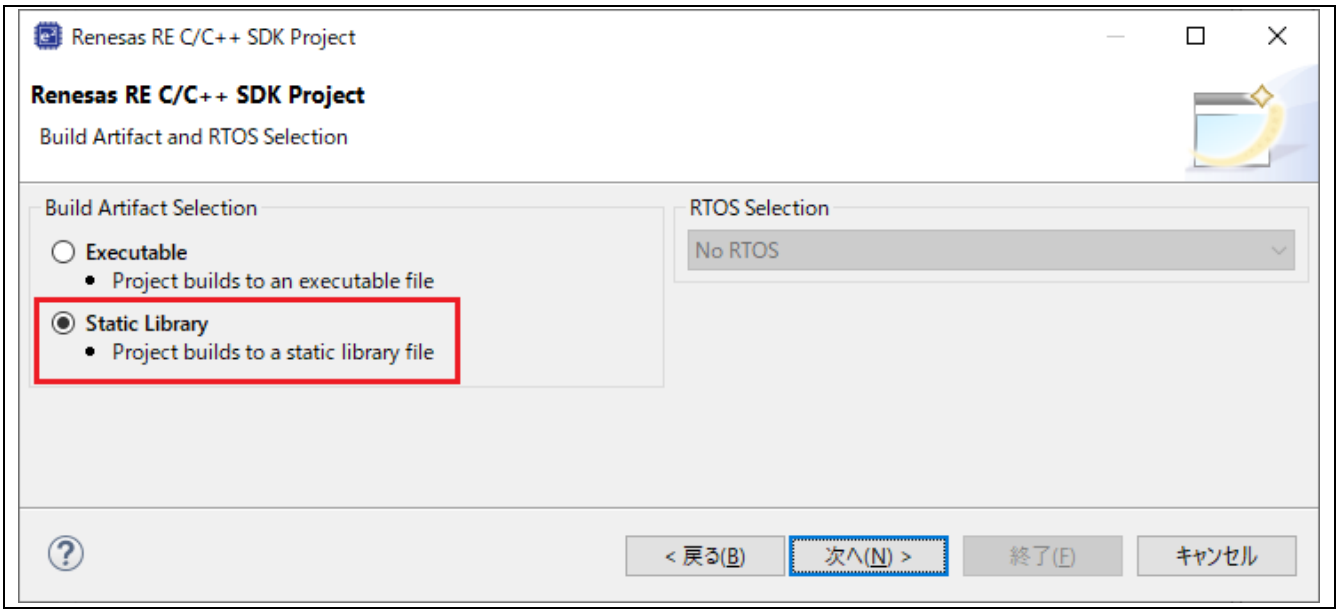


図3-13 ビルドアーティファクトに [Static Library] を選択

   - ステップ6で [Bare Metal – Blinky] をプロジェクトテンプレートとして選択します。

2. プロジェクトを作成したら、[コンフィグレーションエディタ] ビューで [Generate Project Content] をクリックしてください。

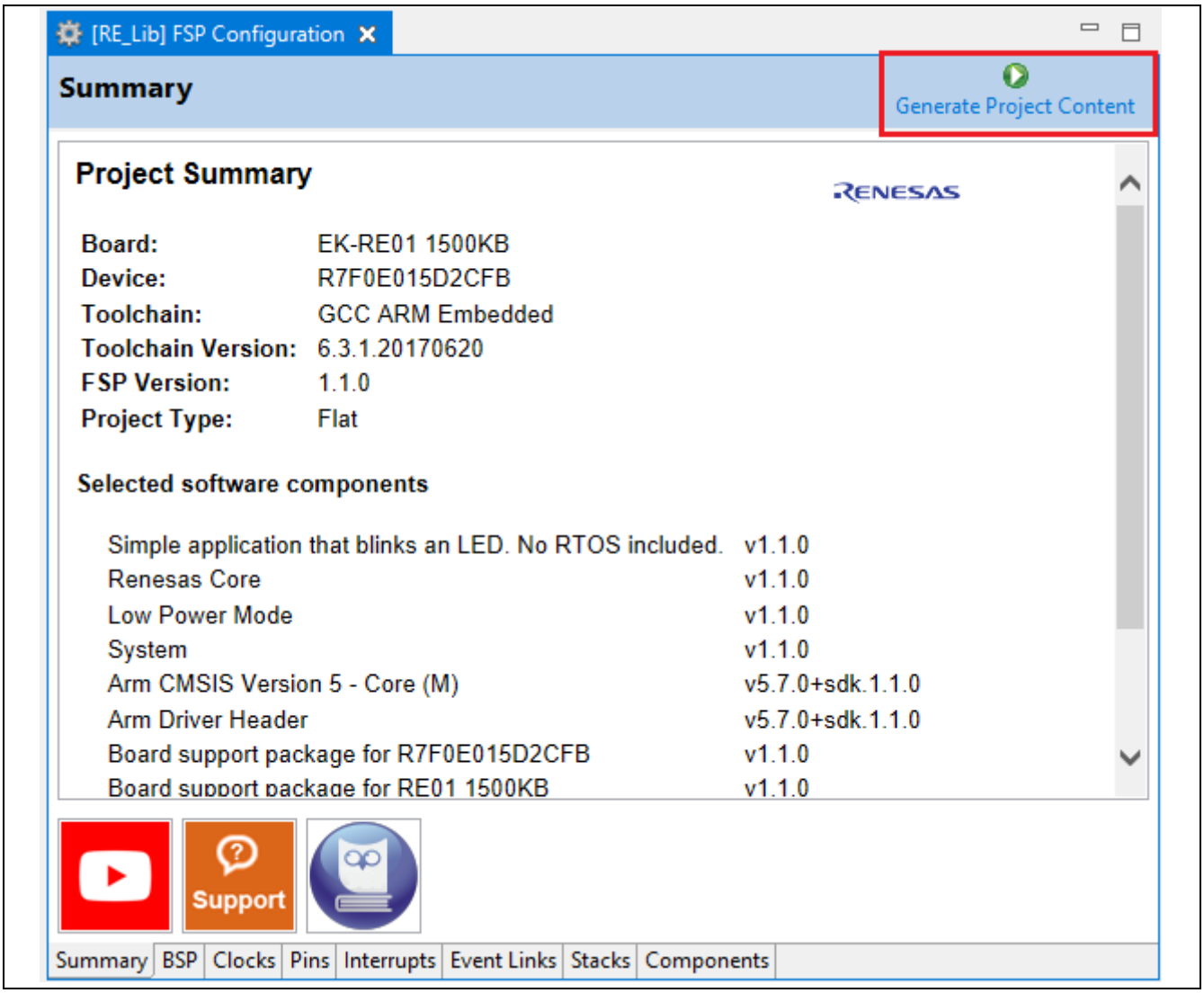


図3-14 ライブラリプロジェクトの生成

3. [プロジェクトエクスプローラー] ビューで、RE_Lib\src\ 下の “hal_entry.c” を開いてください。

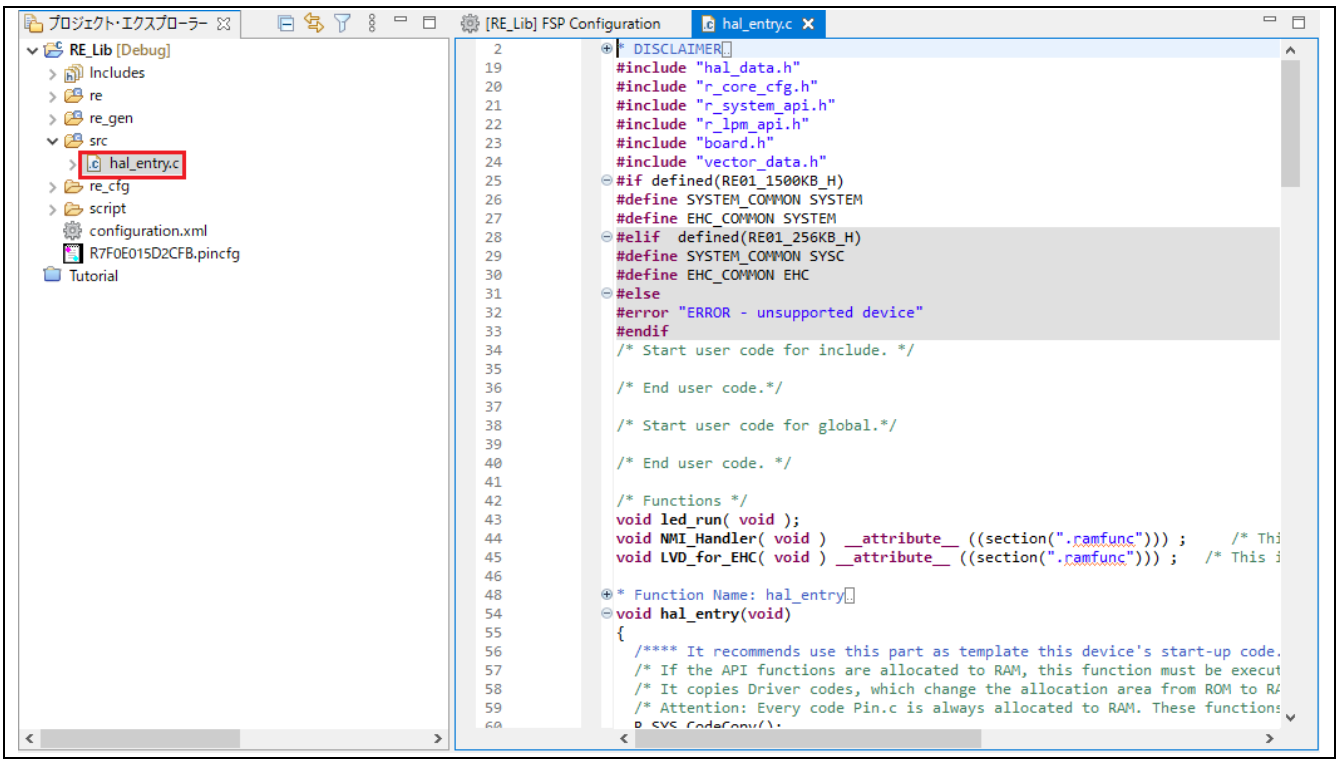


図3-15 元の “hal_entry.c”

4. hal_entry()関数の名称をhal_entry_lib()に変更し、hal_entry_lib()の宣言を追加します。

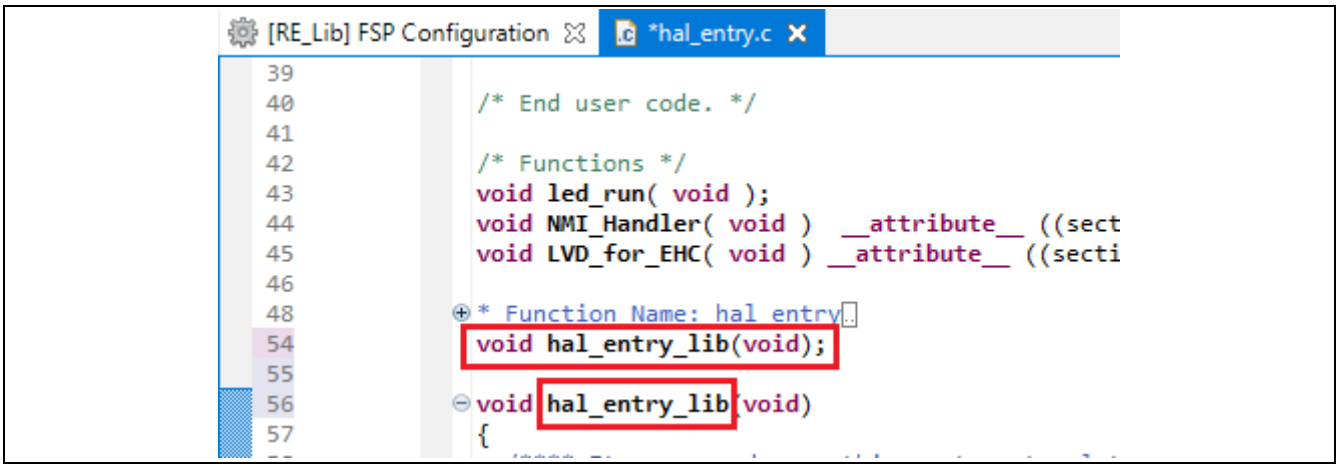


図3-16 変更された “hal_entry.c”

5. 作成したライブラリプロジェクトをビルドすると、スタティックライブラリファイル “RE_Lib\Debug\“libRE_Lib.a” が出力されます。

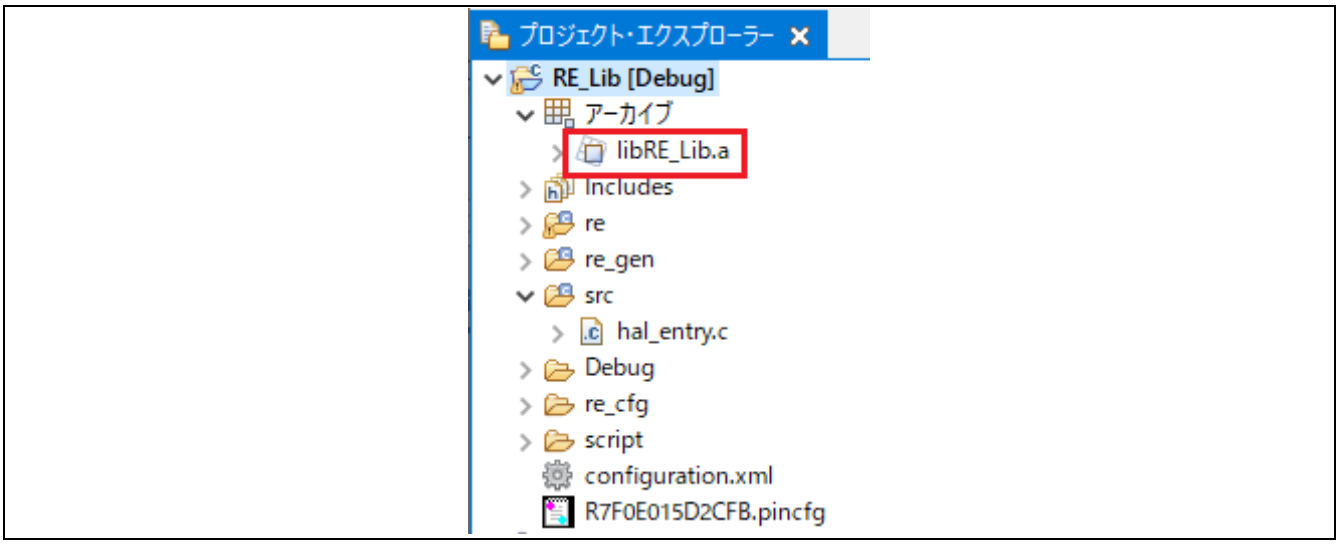


図3-17 ビルドされたスタティックライブラリ

3.3.2 既存のスタティックライブラリを使用した実行プロジェクトの作成

前節（3.3.1 スタティックライブラリプロジェクトの作成）で作成したスタティックライブラリを RE 実行プロジェクトで使用する方法を、以下の手順で説明します。

- RE 実行プロジェクトを作成する
- スタティックライブラリプロジェクトで宣言した hal_entry_lib()関数を呼び出すようにソースコードを変更する
- スタティックライブラリを追加するようにビルド設定を変更する
- RE 実行プロジェクトをビルドする

手順は、以下のようになります。

1. 「3.2 スマート・コンフィグレータを使用しない新規実行プロジェクトの作成」で説明した方法で実行プロジェクトを作成し、プロジェクトに名前を付けます（例：RE_App）。

2. [プロジェクトエクスプローラー] ビューで、RE_App\src\ 下の “RE_App.c” を開いてください。

3. ライブラリ関数 “hal_entry_lib()” の宣言を追加し、この関数をmain() 関数から呼び出してください。

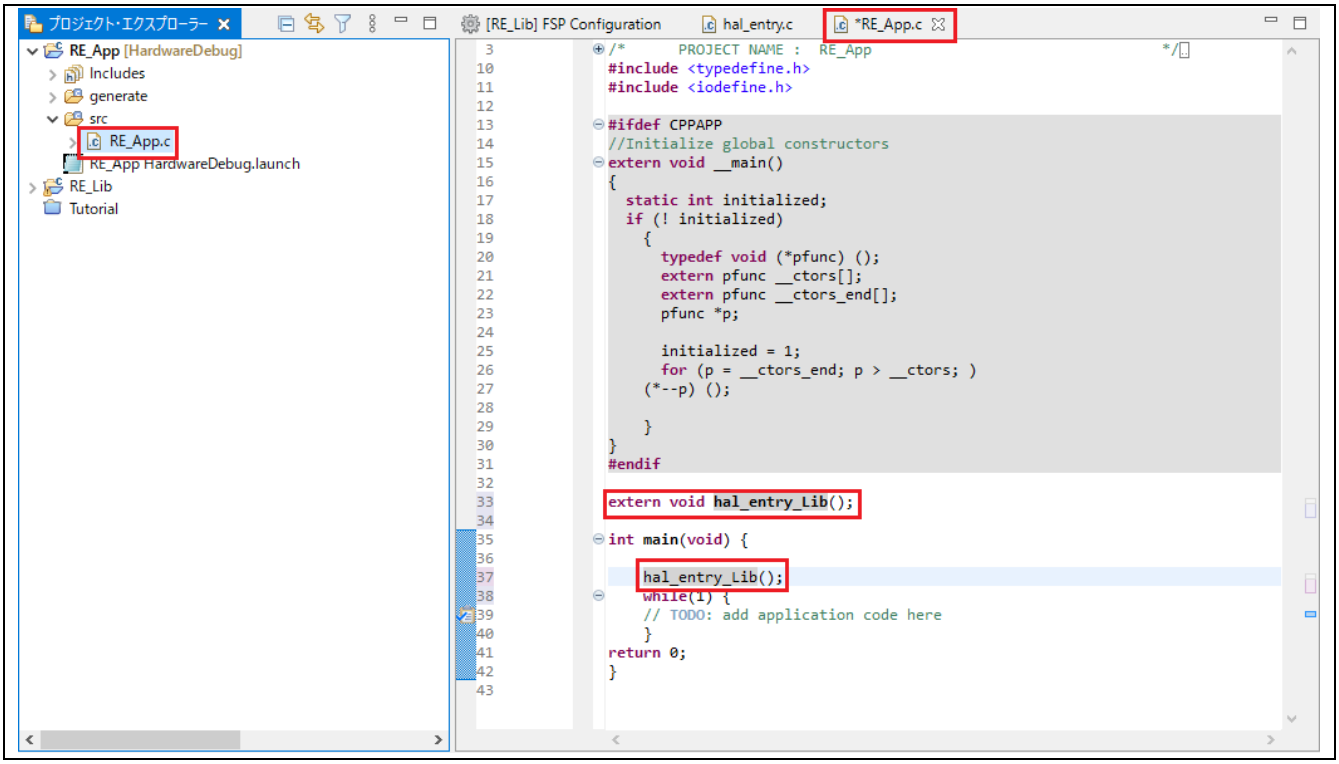


図3-18 “RE_App.c” の更新

4. プロジェクト “RE_App” を右クリックし、[C/C++ Project Settings] を選択してください。

5. プロジェクトのプロパティ画面で、[C/C++ビルド] → [設定] を選択し、[ツール設定] タブを選択してください。[Cross ARM C Linker] → [Libraries] を選択し、“User defined archive (library) files (-l)” で [追加…] ボタンを押して、“RE_Lib” をライブラリファイルとして追加してください。

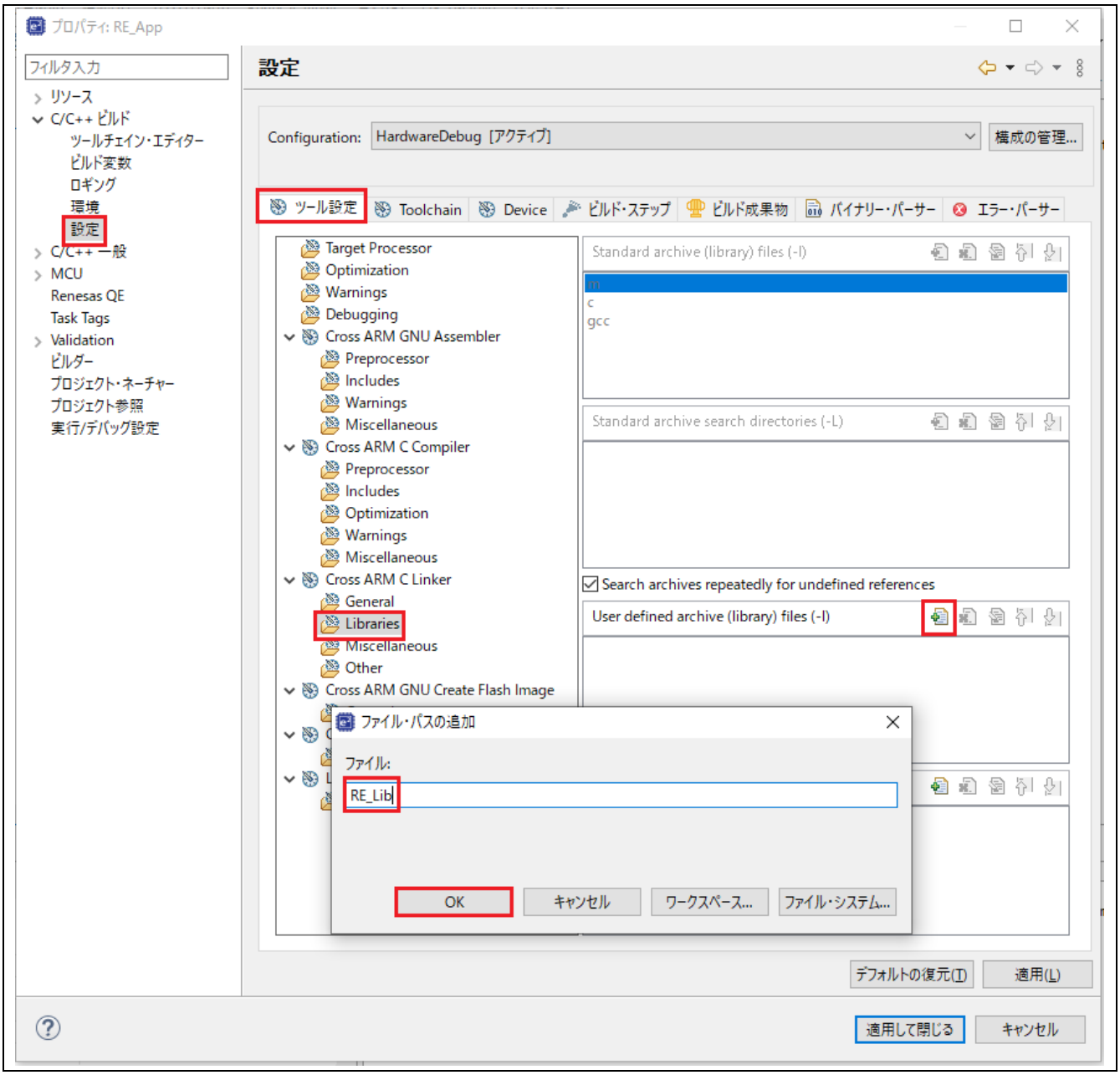


図3-19　ライブラリファイルへの参照を追加

6. “User defined archive search directories (-L)” で [追加…] ボタンを押して、
“${workspace_loc:/RE_Lib/Debug}” を追加してください。

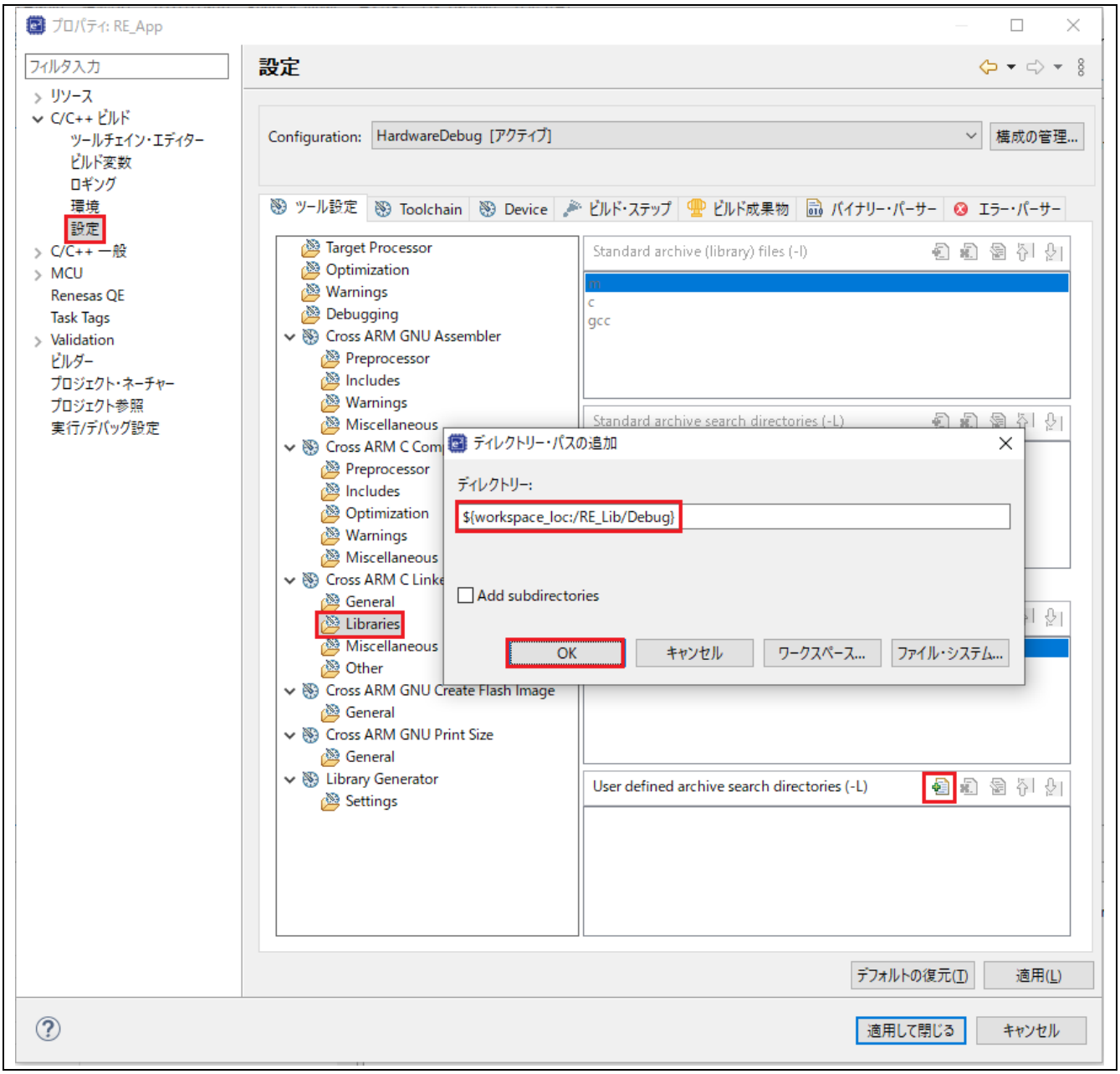


図3-20　ライブラリの場所への参照を追加

7. [Cross ARM C Linker] → [Libraries] の順に選択して、新しいパスが更新されたことを確認してください。

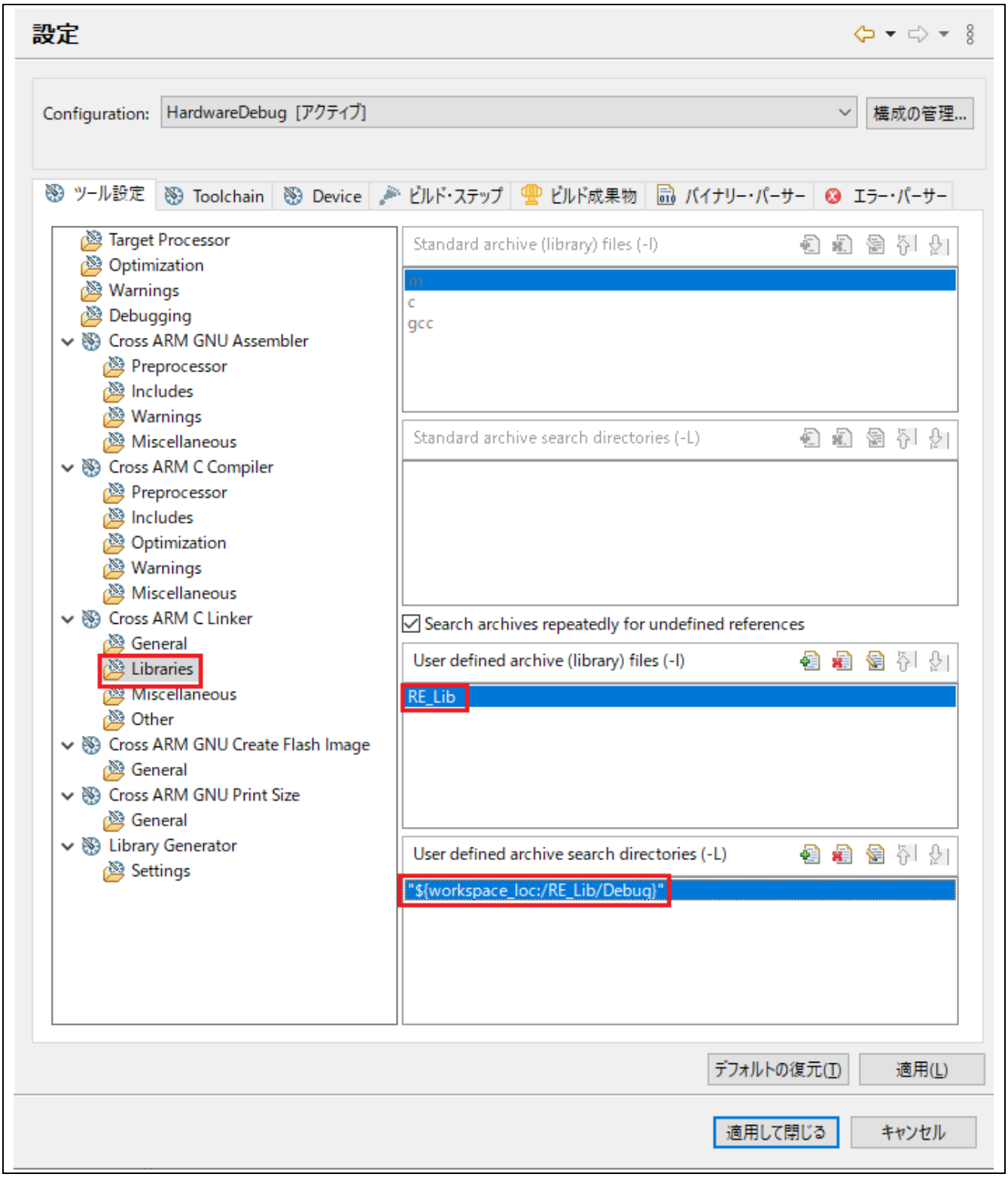


図3-21 [Cross ARM C Linker] 設定が更新済

8. プロパティ画面の左ペインで [プロジェクト参照] を選択し、“RE_Lib” のチェックボックスにチェックを入れてください。実行プロジェクトがスタティックライブラリプロジェクトに依存することを示すためです。その後、[適用して閉じる] ボタンを押してください。

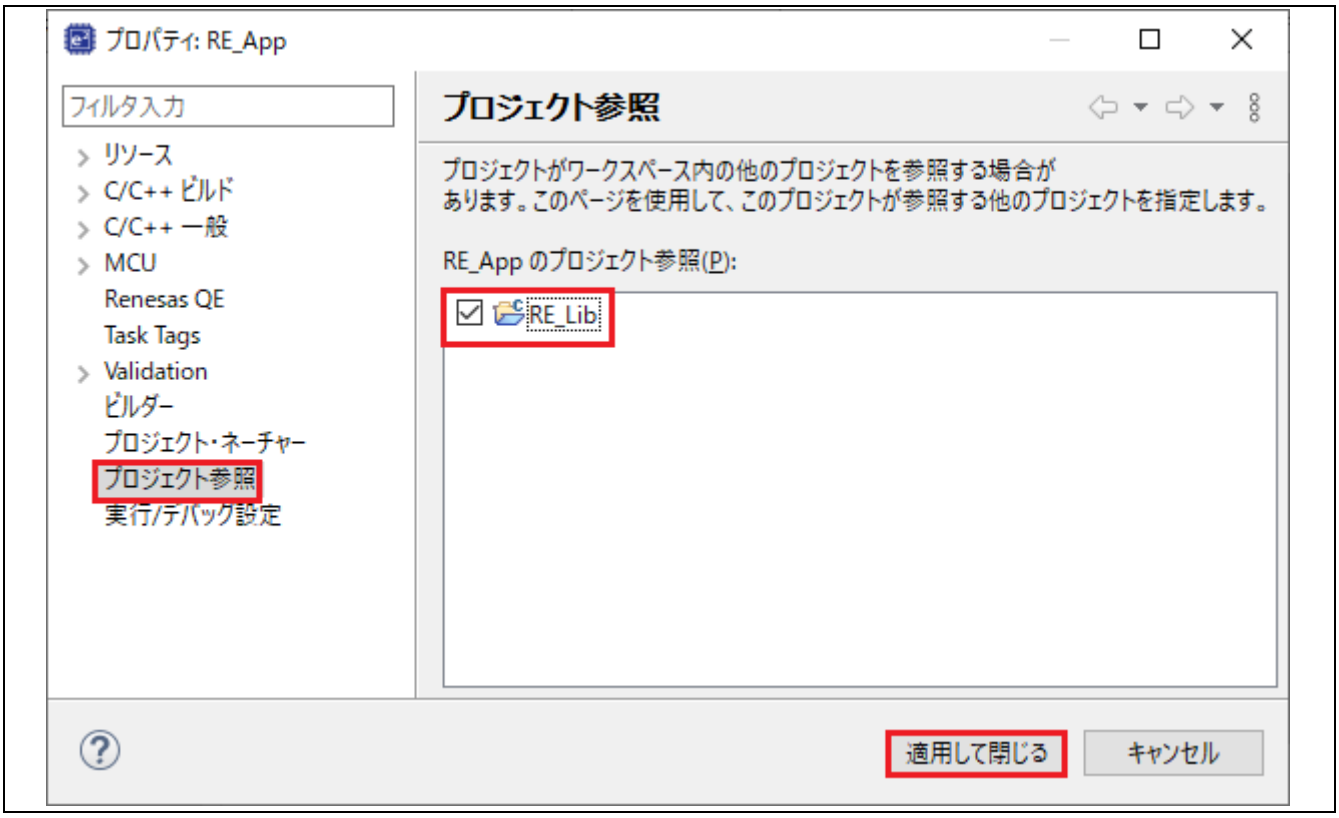


図3-22 実行プロジェクトがスタティックライブラリプロジェクトに依存することを示す

9. アプリケーションプロジェクトをビルドします。

10. ライブラリ関数 “hal_entry_lib()” の呼び出し位置にブレークポイントを設定してRE_Appプロジェクトを実行します。

11. ブレークポイントでプログラムの停止後、実行を再開してください。LEDを点滅するライブラリ関数 “hal_entry_lib()” が実行されることを確認します。

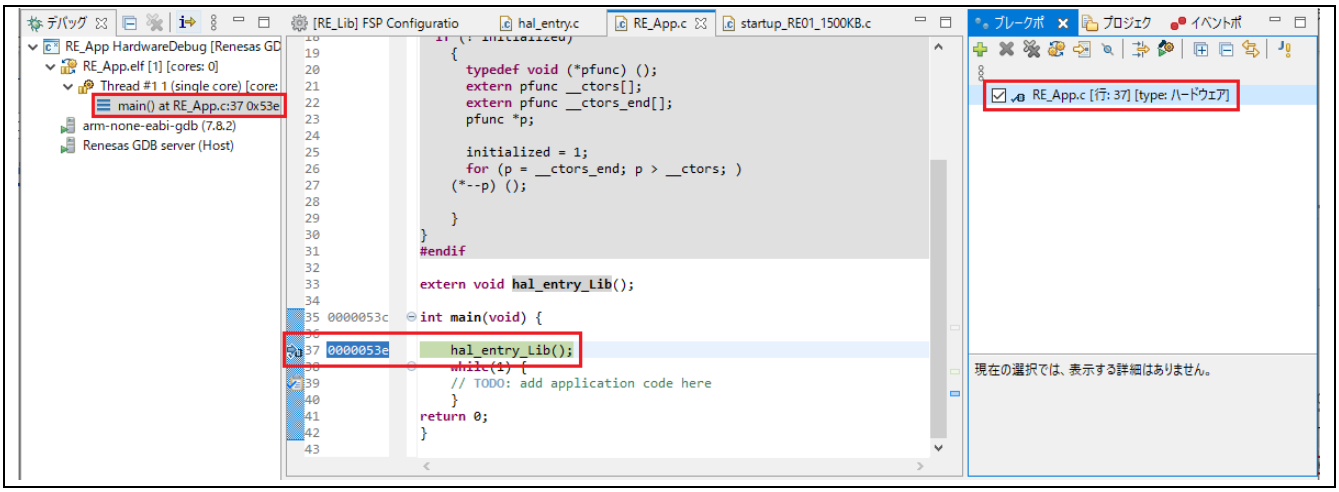


図3-23　アプリケーションプロジェクトでのライブラリ関数の起動

3.4 既存プロジェクトのワークスペースへのインポート

既存のe² studioプロジェクトを現在のワークスペースにインポートするには、以下の手順を実行してください。この手順により、ルネサスのWebサイトからサンプルプロジェクトをインポートできます。サンプルプロジェクトは「5.4 基本的なデバッグ機能」のデバッグ機能のデモに使われます。

1. RE01用のLED点滅サンプルコードをルネサスのWebサイトからダウンロードしてください。
   https://www.renesas.com/jp/ja/document/scd/led-blinker-sample-code-re01-1500kb-and-256kb-group-application-note-sample-code

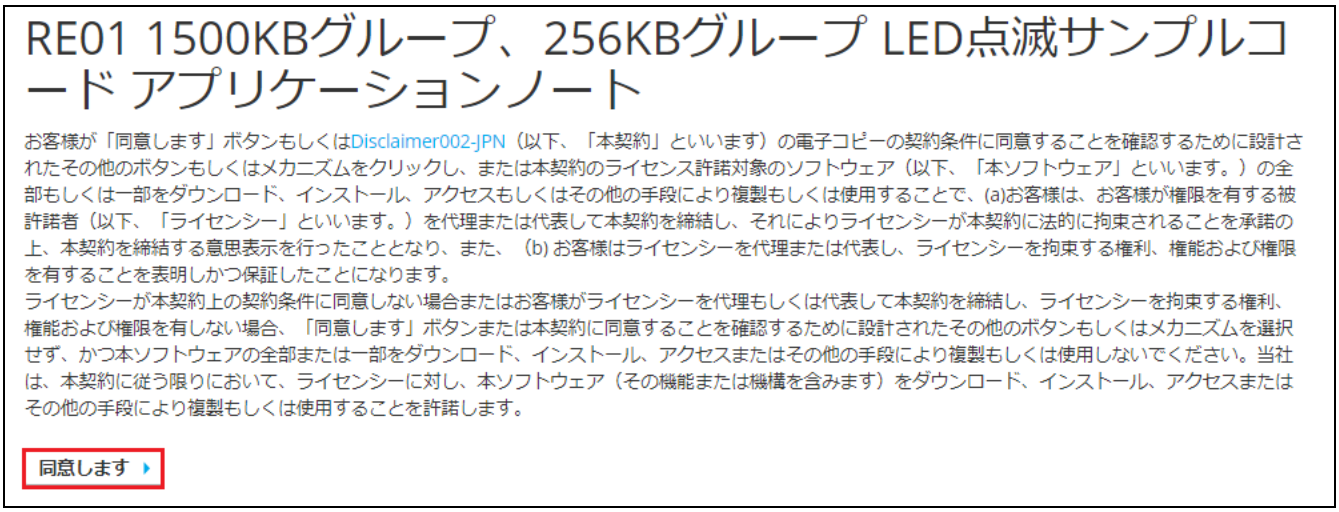


図3-24 サンプルコードのダウンロード

2. パッケージをダウンロードしてから、zipファイルを展開してください。

図3-25 サンプルコードパッケージ

3. e² studioで、[ファイル] → [インポート] の順に選択します。

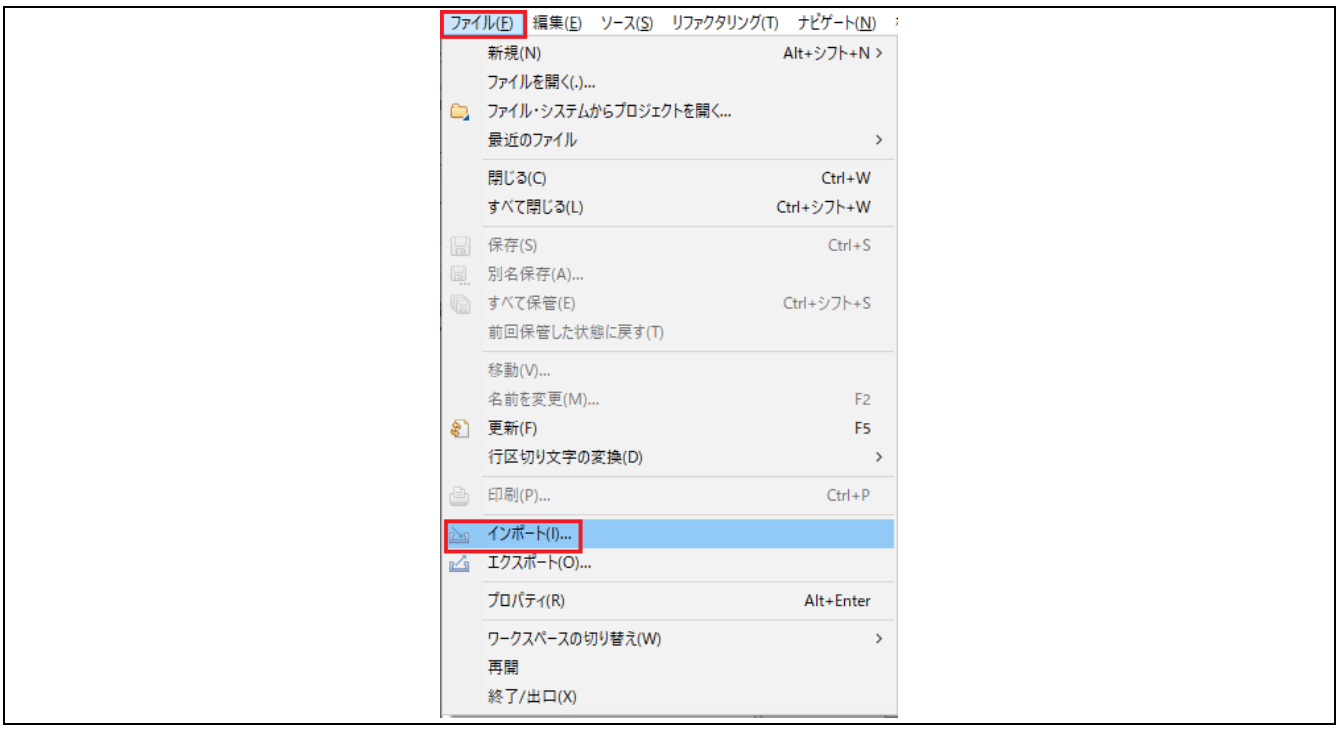


図3-26 サンプルプロジェクトのインポート

4. [インポート] 画面の [一般] → [既存プロジェクトをワークスペースへ] を選択します。[次へ] で次に進みます。

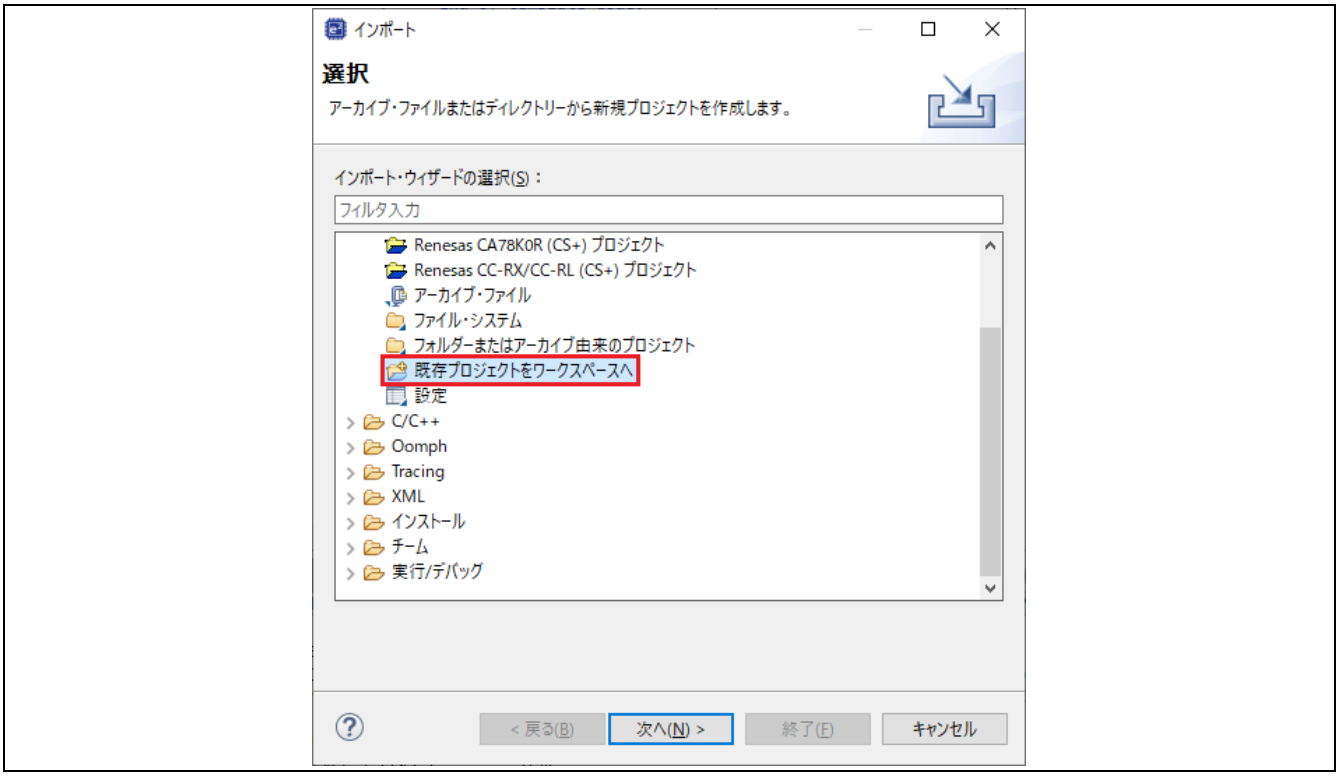


図3-27 インポートウィザードの選択

5. [プロジェクトをインポート] 画面で “アーカイブ・ファイルの選択” を選びます。右の [参照...] ボタンをクリックし、サンプルコードパッケージ中の “an4950_gpio_re_1500kb.zip” という名前のzipファイルを選択します。“プロジェクト” に表示されているプロジェクトを選択して、[終了] ボタンを押してください。

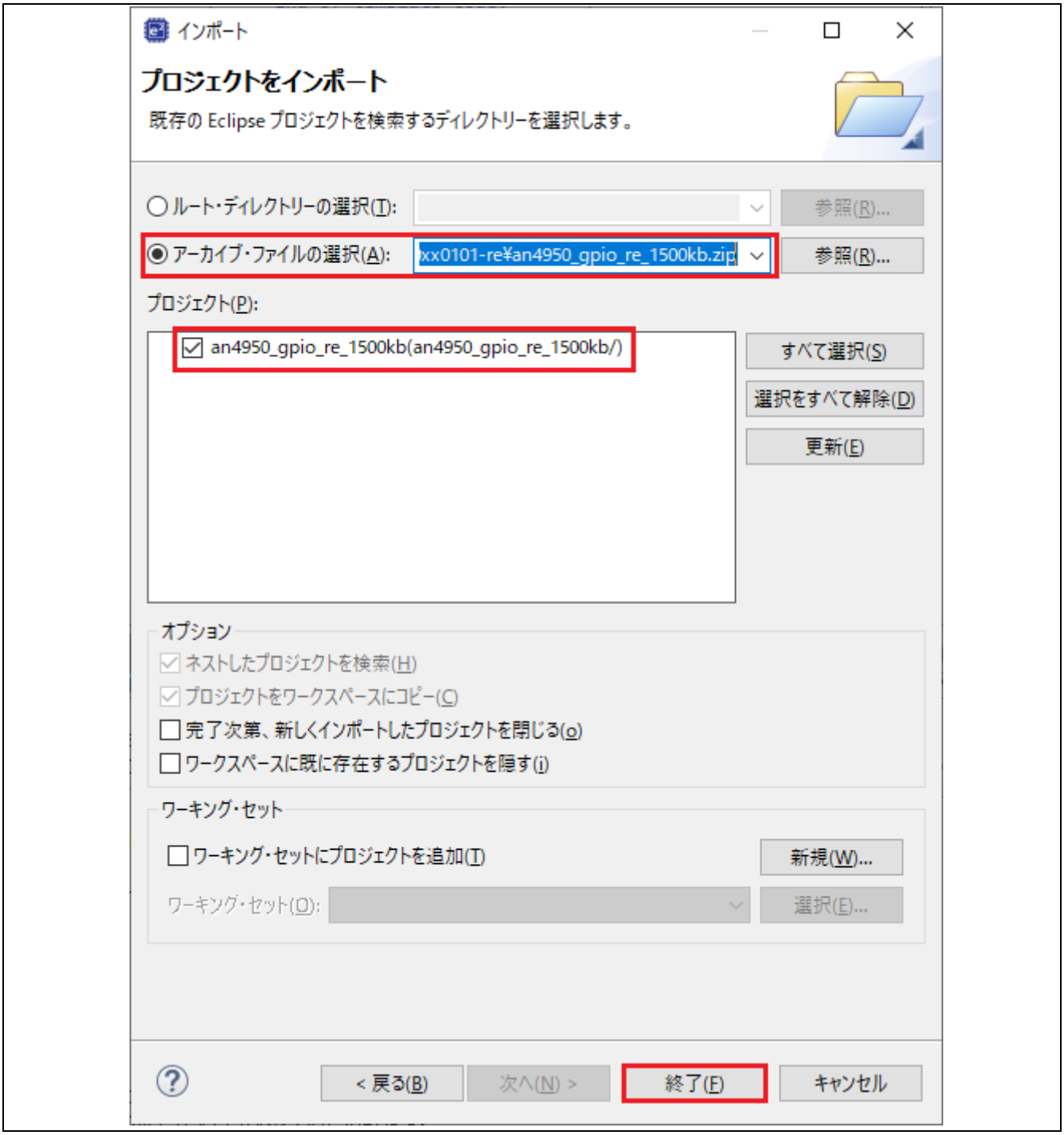


図3-28　インポートするプロジェクトの選択

6. プロジェクトのプロパティ画面を開き、左のペインで [C/C++ ビルド] → [設定] を選択します。[Toolchain] タブを選び、そのプロジェクトの正しいツールチェーンを選択します。その後、[適用して閉じる] ボタンを押してください。

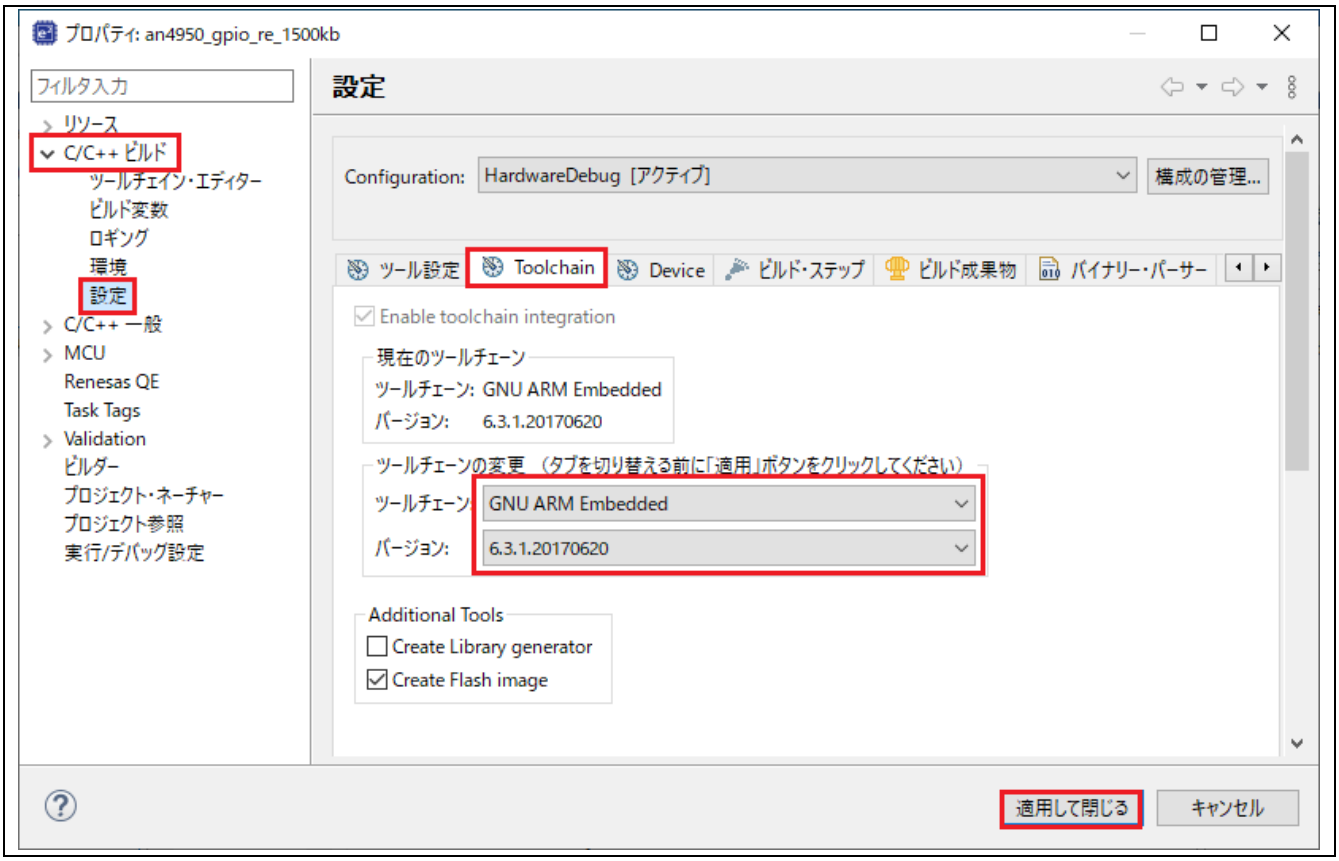


図3-29 プロジェクトツールチェーンの更新（スマート・コンフィグレータを使用しない場合）

インポートされたプロジェクトがスマート・コンフィグレータを使用する場合、GUIが異なる可能性があります。

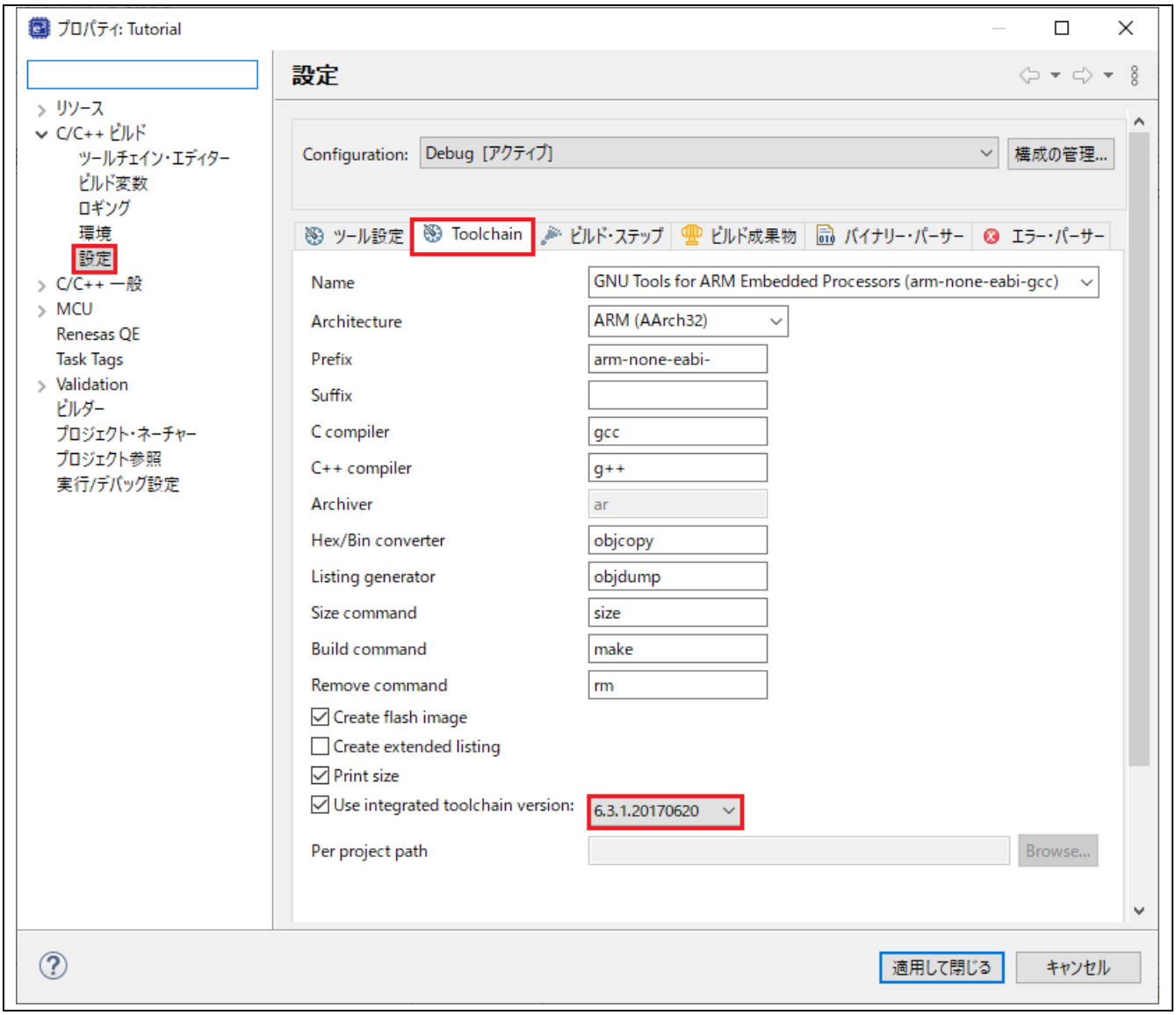


図3-30　プロジェクトツールチェーンの更新（スマート・コンフィグレータを使用する場合）

7.　プロジェクトをビルドして、正しくビルドできたかを確認してください。

3.5 コンフィグレーションエディタ

[コンフィグレーションエディタ] ビューは、現在のプロジェクト構成（configuration）を表示します。構成は “configuration.xml” ファイルに保存されています。プロジェクト構成の設定はいくつかのページに分類してあり、項目別にプロジェクトの設定を行なうことができます（端子やクロックの設定、使用するドライバなど）。

プロジェクト構成を編集するには、まず、以下を確認してください。

- [RE Configuration] パースペクティブが開いていることを確認してください。[ウインドウ] メニュー → [パースペクティブ] → [パースペクティブを開く] → [その他] → [FSP Configuration] の順に選択すると、開くことができます。
- “configuration.xml” ファイルが開いていることを確認してください。

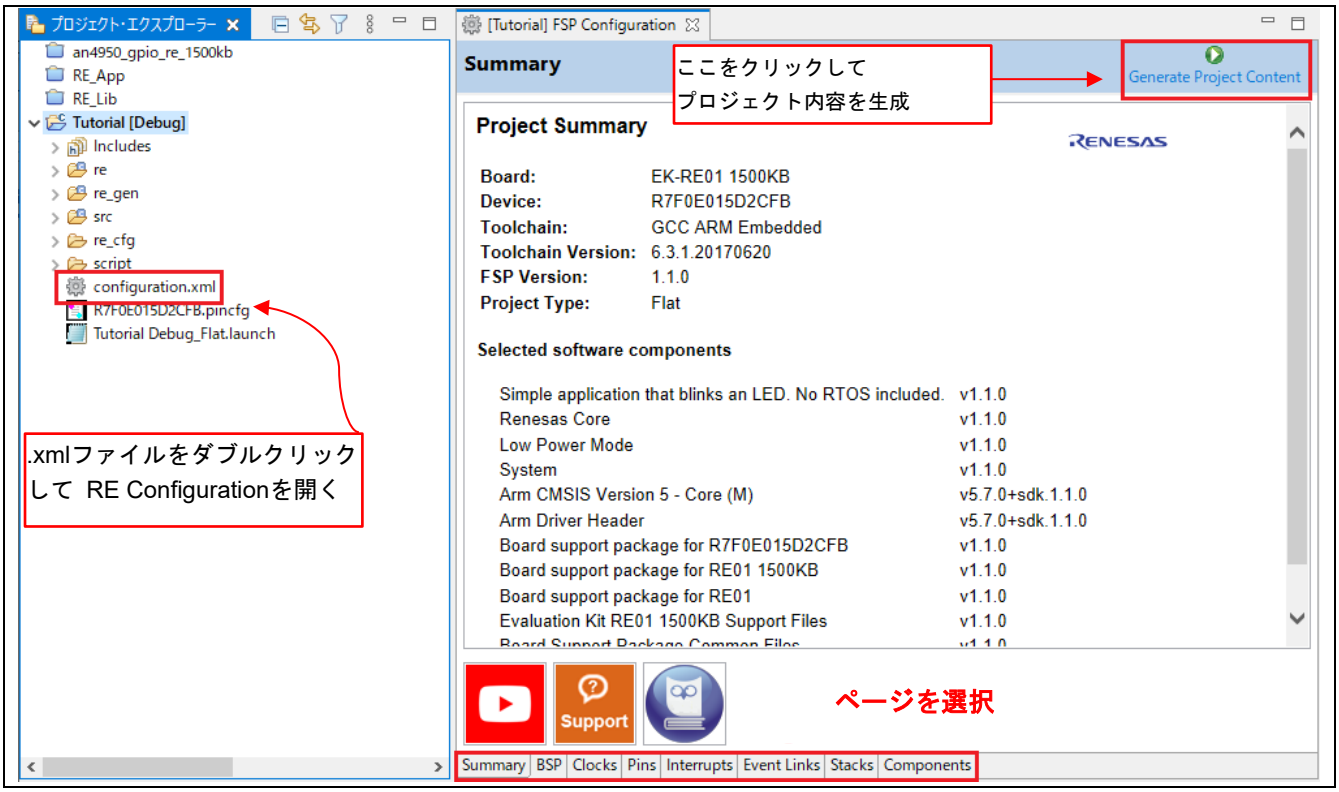


図3-31 [コンフィグレーションエディタ] ビュー

[コンフィグレーションエディタ] ビューには 8 つのページ（タブ）があります。

概要（Summary）ページは、プロジェクトの概要を表示します。

BSP ページでは、FSP のバージョン、RE 用ボードの種類、デバイスを選択できます。

クロック（Clocks）、端子（Pins）、割り込み（Interrupts）、イベントリンク（Event Links）、スタック（Stacks）、コンポーネント（Components）ページの設定手順やオプションについても後述します。

3.5.1 概要（Summary）ページ

[Summary] ページは、現在選択しているデバイス、ボード、RE 用ソフトウェアコンポーネントなど、プロジェクトの概要を表示します。また、YouTube の ‘Renesas Presents’ チャンネルやソフトウェアパッケージのユーザーズマニュアルなど便利なサイトへのリンクもあります。

スレッドにモジュール／オブジェクトを追加すると、その情報は [Summary] ページにも追加されます。

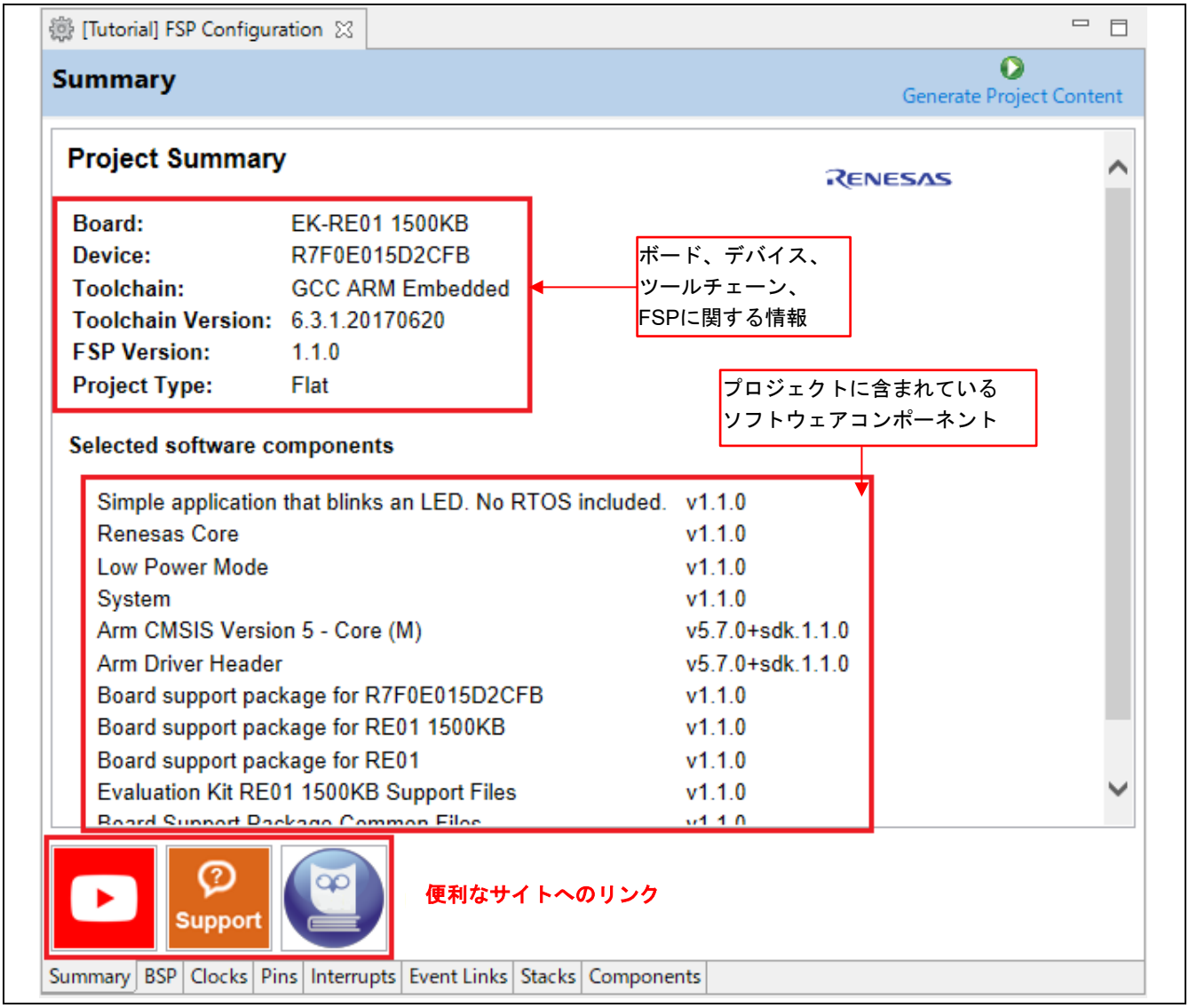


図3-32 [Summary] ページ

3.5.2 BSP ページ

[BSP] ページでは、FSPのバージョン、ボード、デバイスを選択できます。CMSISパックもこのページからインポートできます。

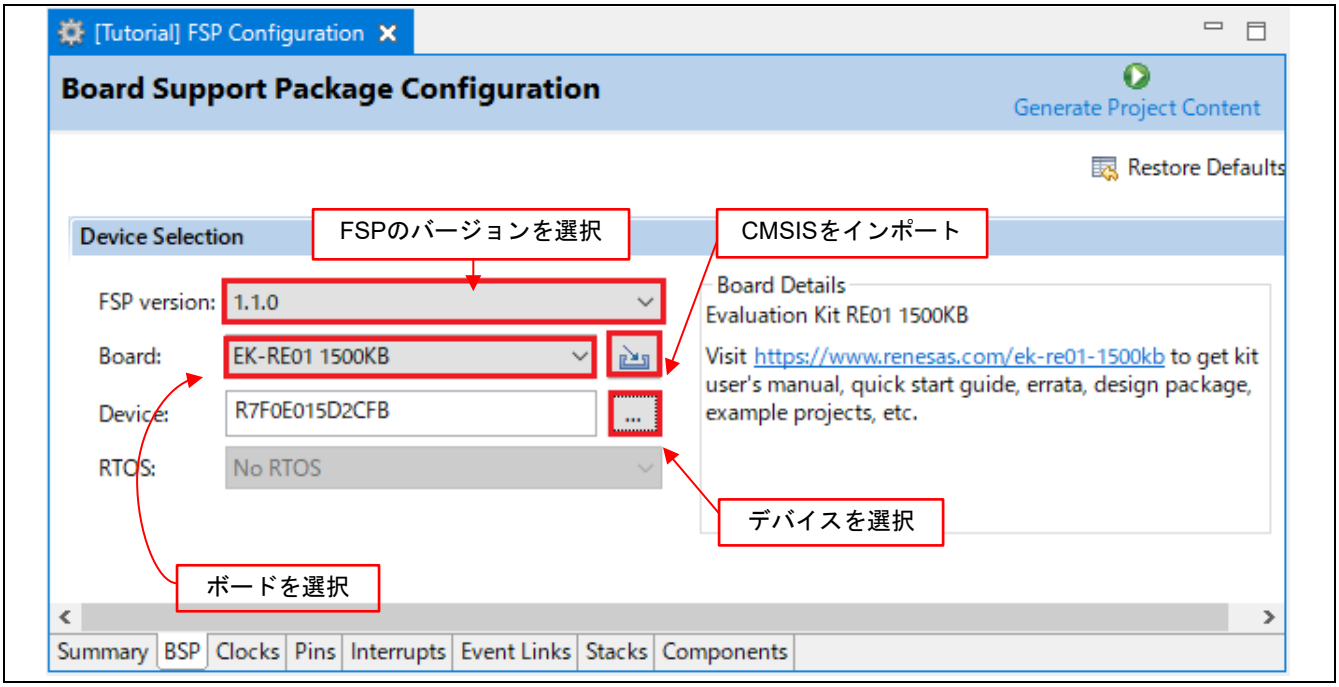


図3-33 [BSP] ページ

3.5.3 クロック（Clocks）ページ

[Clocks] ページでは、アプリケーションの初期クロックを設定します。出力クロックごとに、クロックソース、PLL設定、クロック分周設定を選択できます。

クロック生成回路（CGC）の詳細は、REファミリのユーザーズマニュアル：ハードウェア編を参照してください。以下の手順でプロジェクトを更新します。

1. 画面上のクロック設定のドロップダウンリストから、設定したい値を選択します。

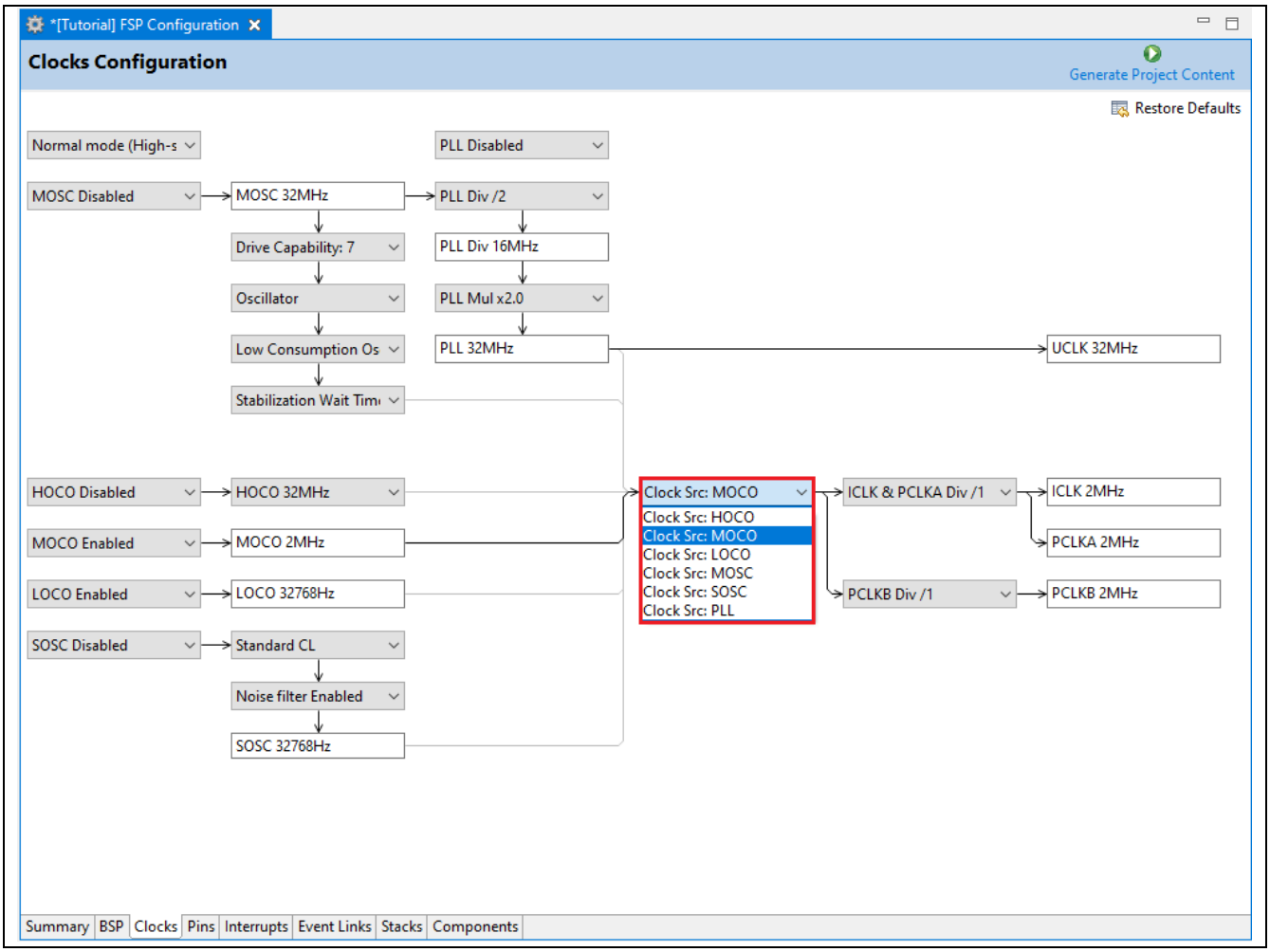
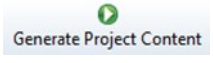


図3-34 [Clocks] ページ

2. [Ctrl+S] キーを押すことなどで、プロジェクト構成の設定を保存します。

3. [Generate Project Content] 

ボタンをクリックします。

4. 選択したクロック設定で、“bsp_clock_cfg.h” ファイルが更新されます。

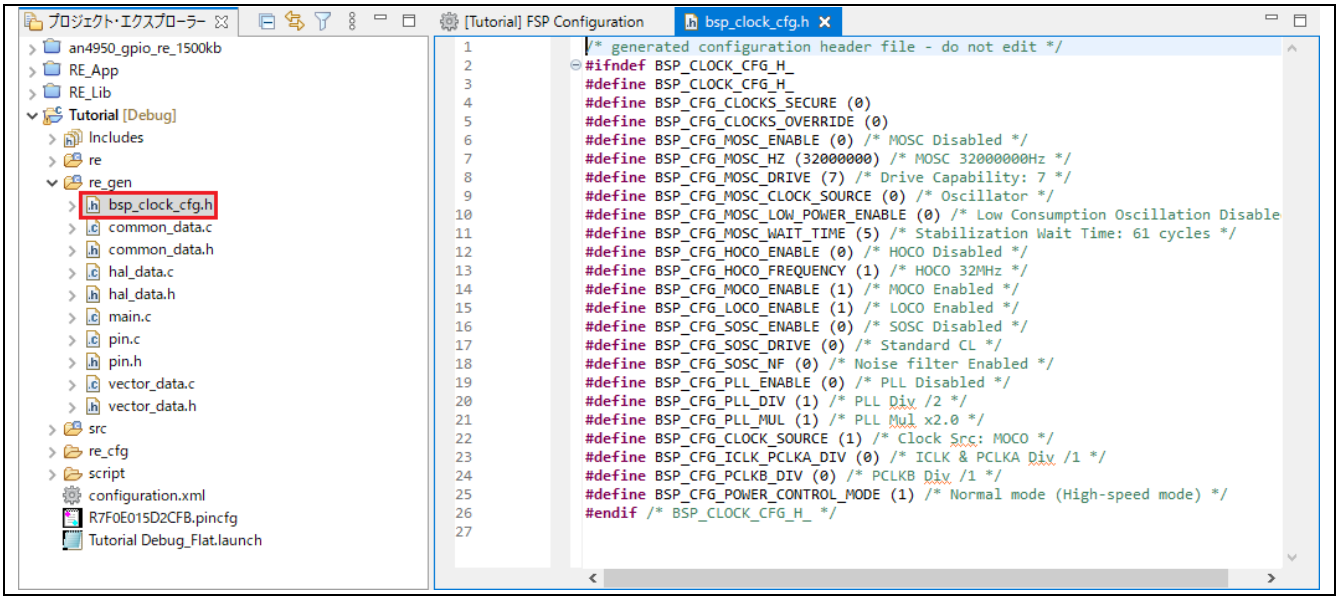


図3-35　bsp_clock_cfg.hの更新

3.5.4 端子（Pins）ページ

[Pins] ページでは、プロジェクトの端子構成を生成します。

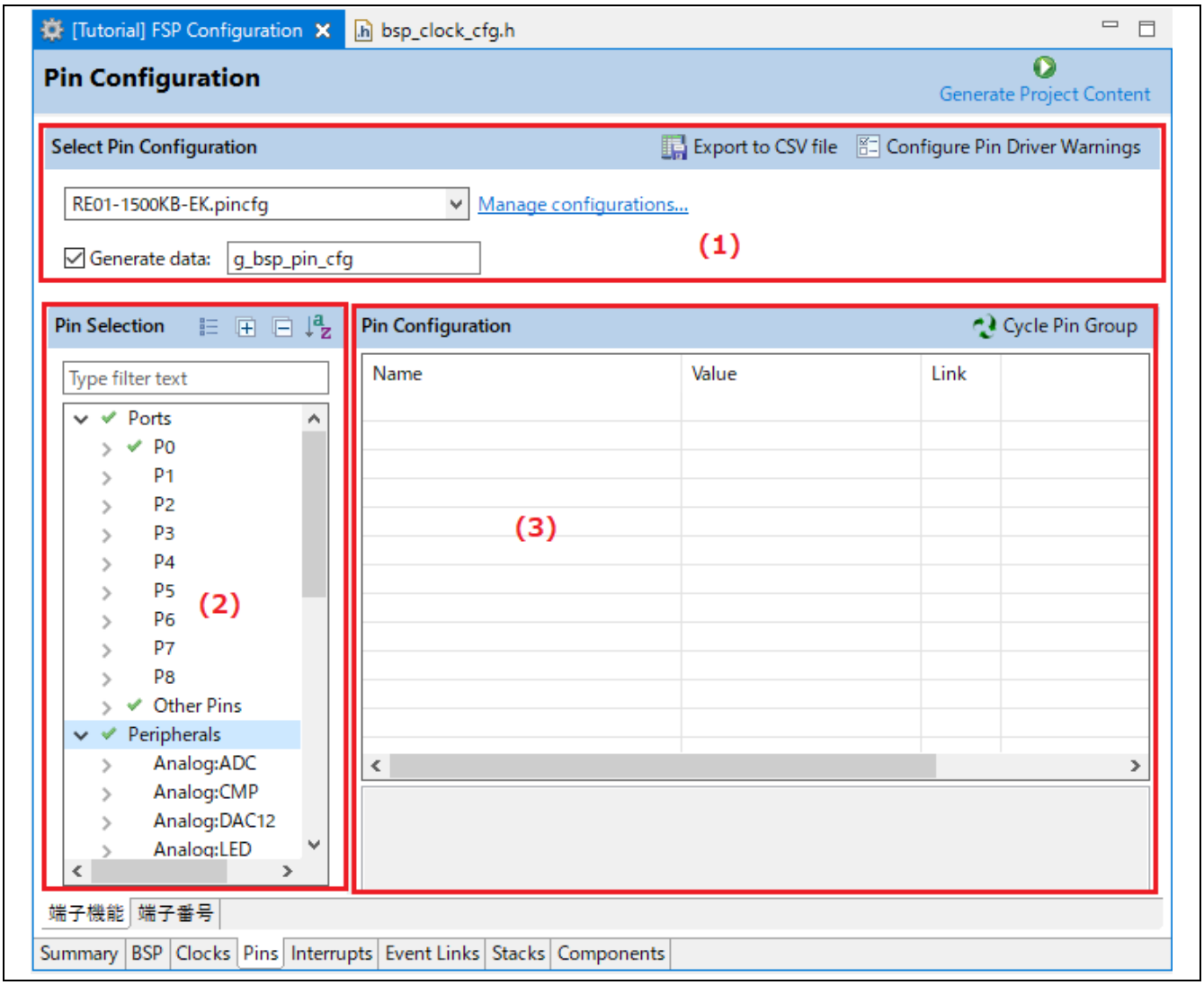


図3-36 [Pins] ページ

[Pins] ページは3つのペインで構成されています。

1. Select Pin Configuration：端子構成ファイルを選択し、関連するデータ構造に名前を付けます。以下の手順で、複数の端子構成を設定できます。

   - 既存の .pincfgファイルをコピーして、新規の .pincfgファイル（例：NewName.pincfg）をプロジェクトエクスプローラーに作成します。
   - 作成した新規 .pincfgファイル（例：NewName.pincfg）を、“Select Pin Configuration” のドロップダウンリストで選択します。
   - [Generate data] チェックボックスを選択し、横にあるテキストボックスで、新規端子構成のデータ構造に固有の名称を付けます。
   - 複数の異なるデータ構造を用いて、複数の端子構成を作成できます。

2. Pin Selection：設定する端子や周辺機能を選択します。

3. Pin Configuration：選択した端子や周辺機能のプロパティや機能を設定します。

プロジェクトで使用する周辺機能ごとに端子構成の設定を以下の手順で行ないます。

1. [Pin Selection] ペインで周辺機能を選択します（例：[Connectivity:SCI] → [SCI4]）。選択した周辺機能の構成が [Pin Configuration] ペインに表示されます。

2. 選択した周辺機能の動作モードを “Operation Mode” で選択します（例：“Simple SPI”）。

3. 選択したモードで周辺機能の入出力に使用したい端子を “Input/Output” で選択します。

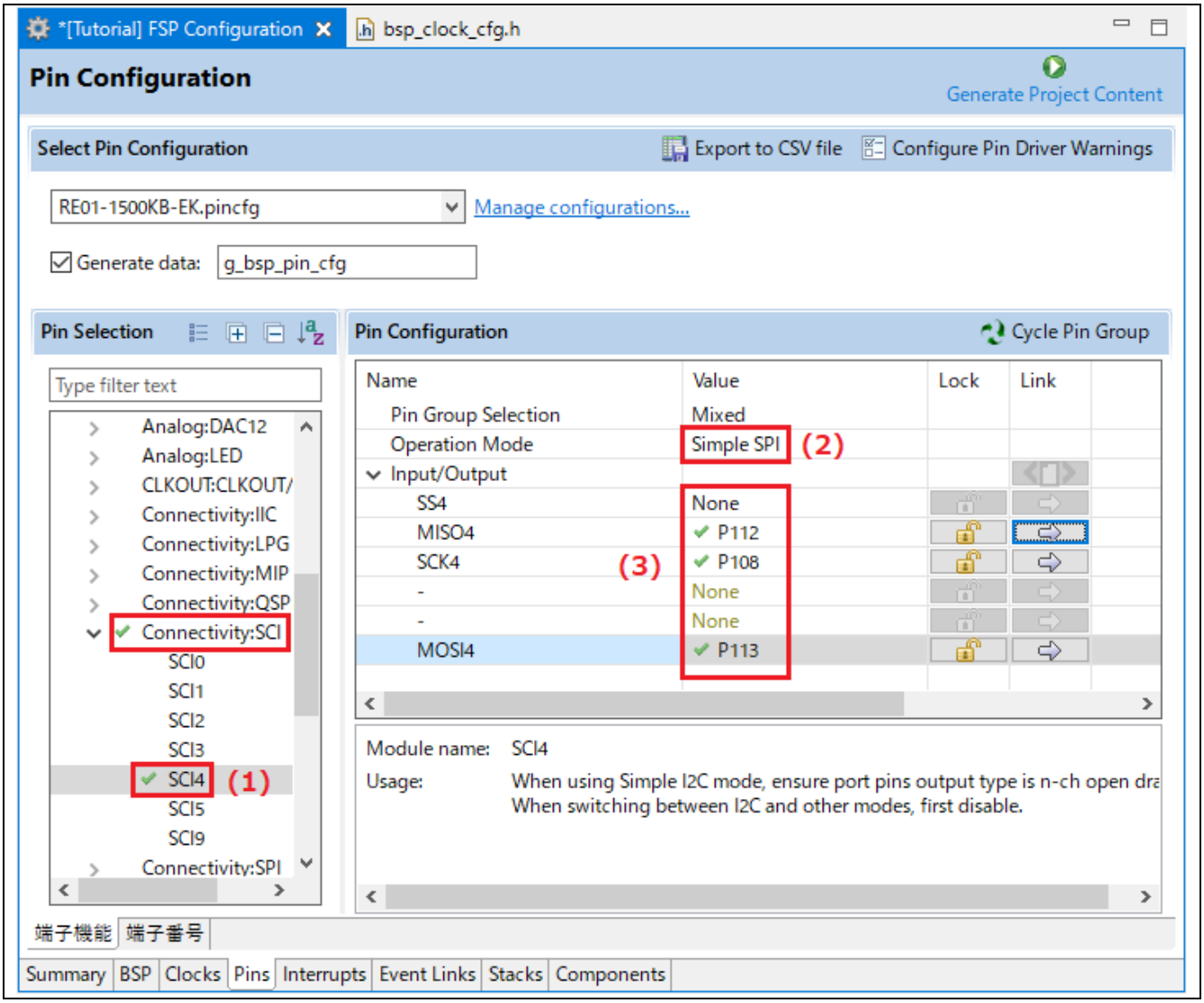


図3-37　端子構成の設定（周辺機能ごとに設定）

端子ごとに設定することもできます。以下の手順で行ないます。

1. [Pin Selection] ペインで 1 つの端子を選択します（例：[Ports] → [P0] → [P003]）。選択した端子の設定が [Pin Configuration] ペインに表示されます。
2. 下図の例のように、端子のプロパティを入力します。

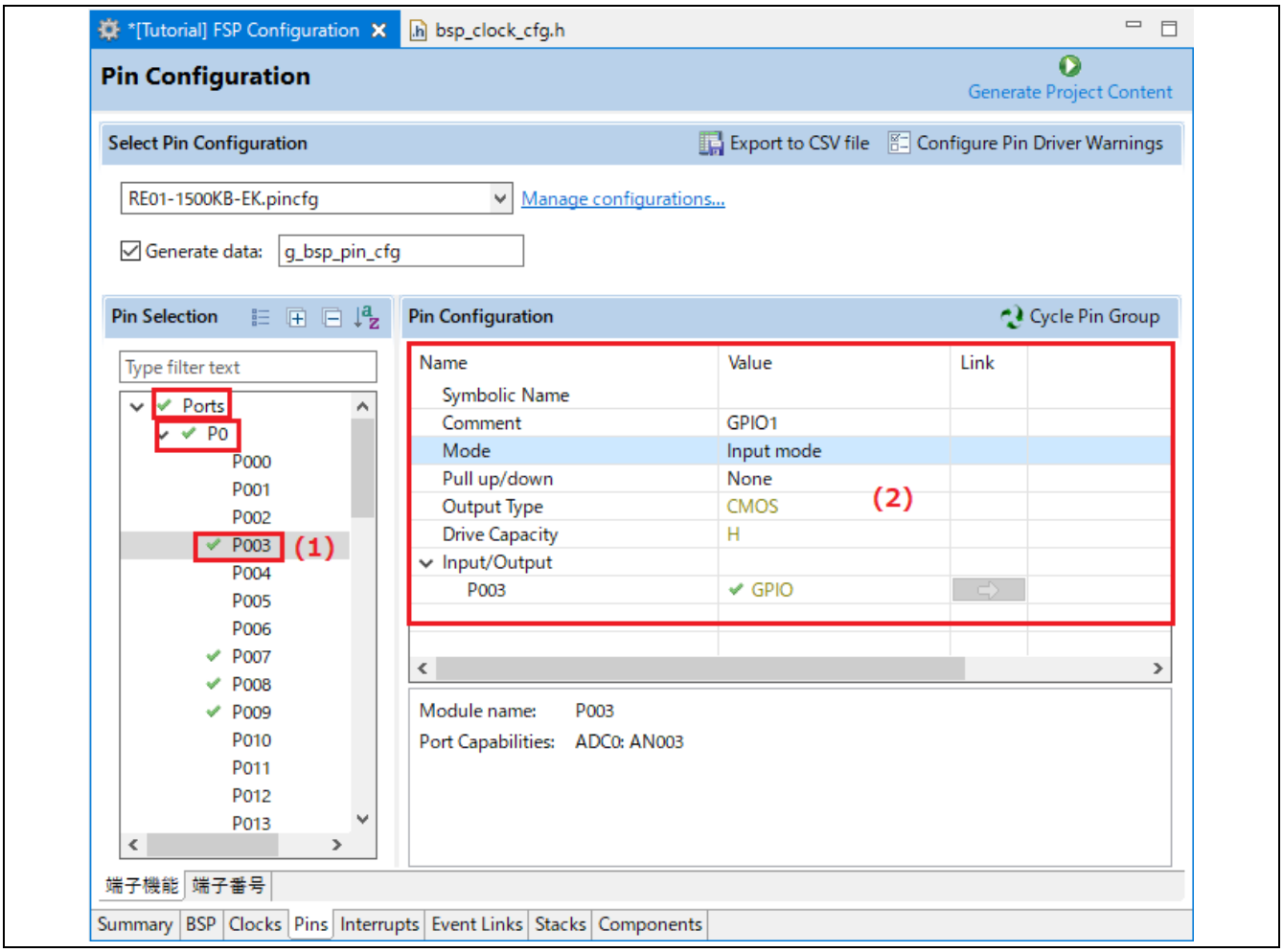


図3-38　端子構成の設定（端子ごとに設定）

3. 変更した端子設定は [MCU パッケージ] ビューに表示されます。

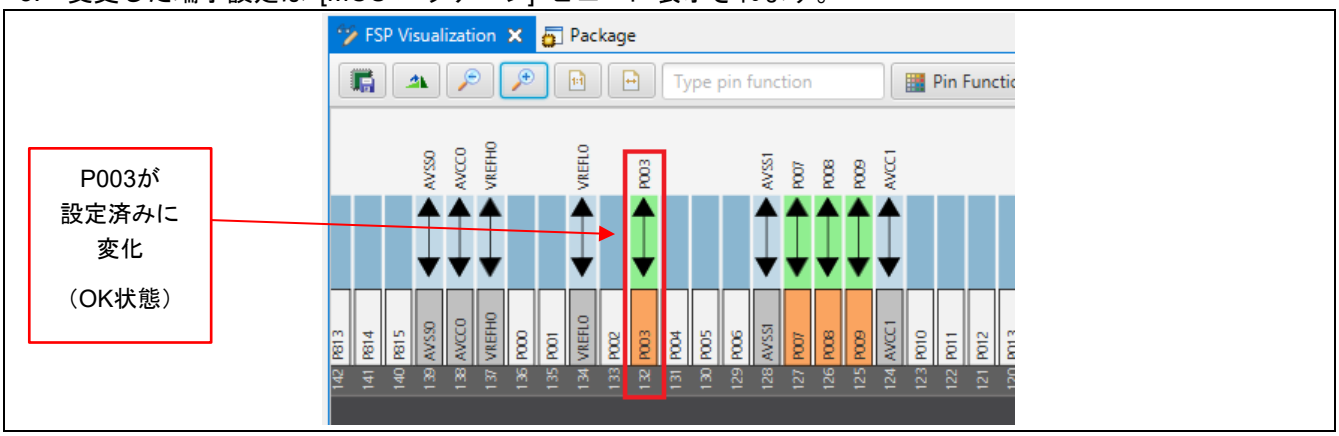


図3-39　[MCUパッケージ] ビュー（接続状態）

3.5.5 スタック（Stacks）ページ

[Stacks] ページでは次の操作が可能です。

- RE プロジェクト内でスレッドを設定する
- スレッドに RE のソフトウェアモジュールやオブジェクトを追加する
- [プロパティ] ビューでモジュールやオブジェクトのプロパティを変更する

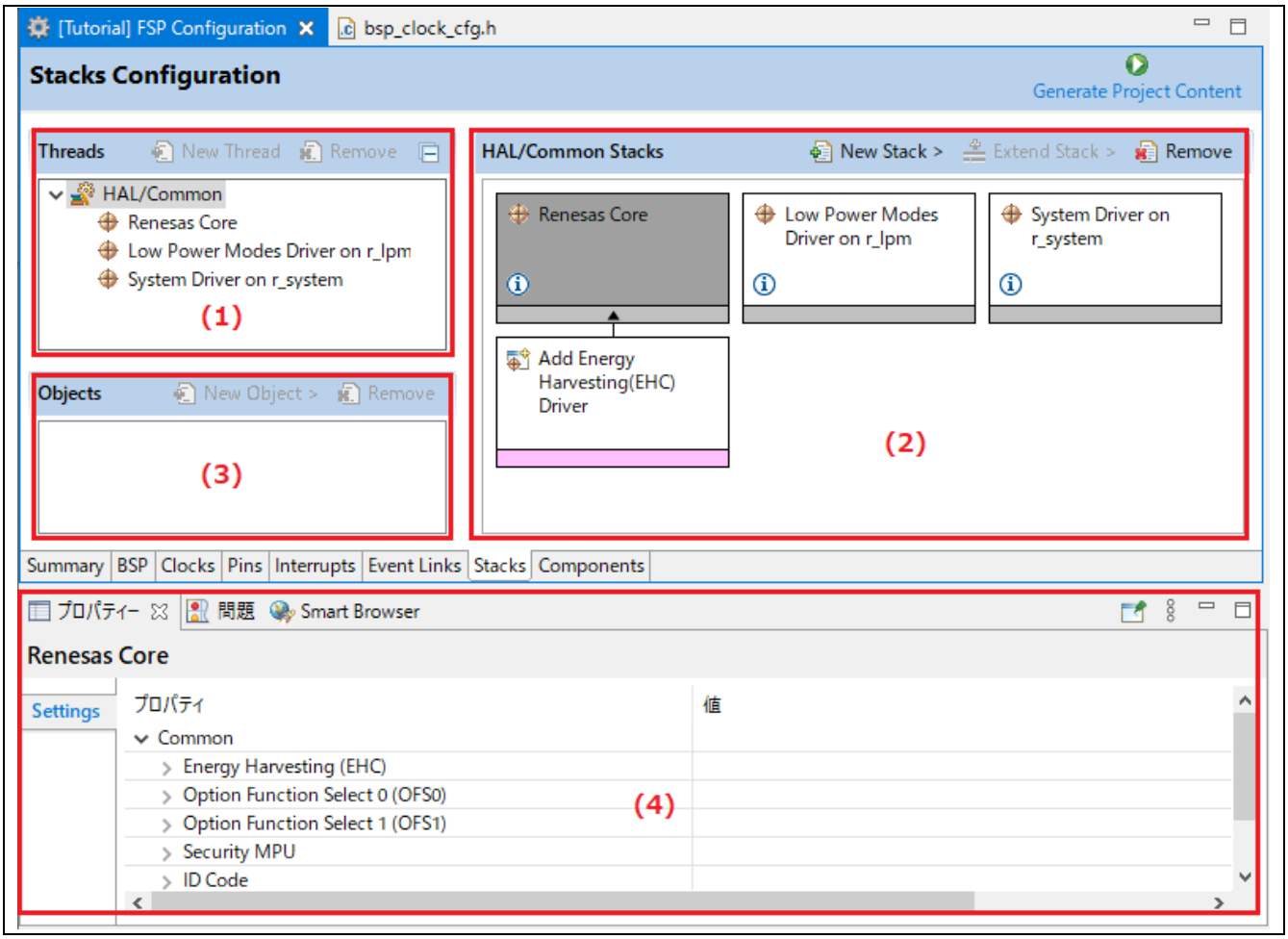


図3-40 [Stacks] ページ

[Stacks] ページは3つのペインで構成されています。

1. [Threads] ペイン：スレッドを追加または削除します。
2. [Stacks] ペイン：ソフトウェアのモジュールインスタンス（I/Oポート、SCI、UARTなど）を追加または削除します。
3. [Objects] ペイン：カーネルオブジェクトを追加または削除します。

また、[プロパティ] ビューではスレッドの設定機能をサポートしており、モジュールやオブジェクトのプロパティを変更できます。

以下の手順でモジュールを既存のプロジェクトに追加できます。

1. スレッド（HAL/Common）を選択します。選択したスレッド内のモジュールとオブジェクトが表示されます。
2. [Stacks] ペインの [New Stack] ボタンを押してスレッドにモジュールを追加します（例：[New Stack] → [Driver] → [Connectivity] → [CMSIS Driver for USART on r_usart ch(SCI)] の順に選択）。
3. [Generate Project Content] Generate Project Content ボタンをクリックしてソースコードを生成します。
4. 必要に応じて、[プロパティ] ビューで選択したモジュールのプロパティを変更できます。

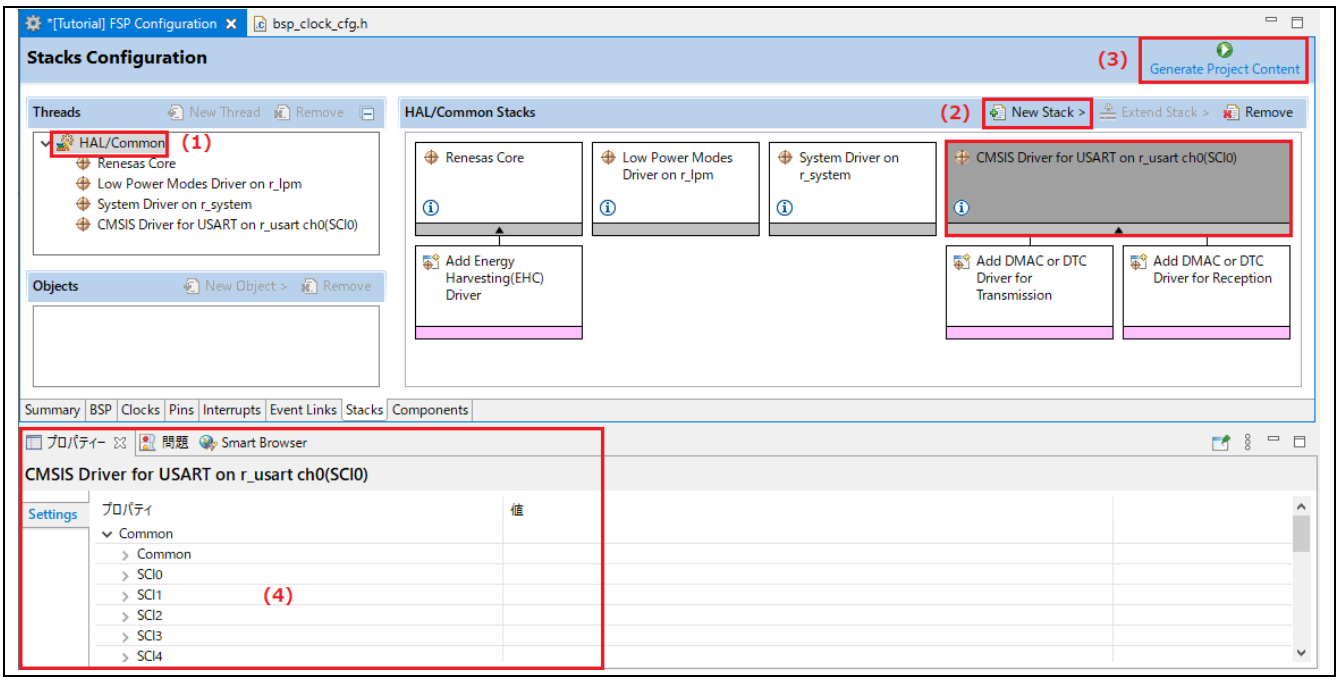


図3-41　スレッドにモジュールを追加

モジュールを追加すると、依存関係がある別のモジュールの追加や、構成の設定が必要になる場合があります。依存関係があるモジュールのうち、必要なものは自動で追加されます。任意で追加可能なものは別に表示されますので、追加したいモジュールをクリックして選択し、プロパティを設定してください。

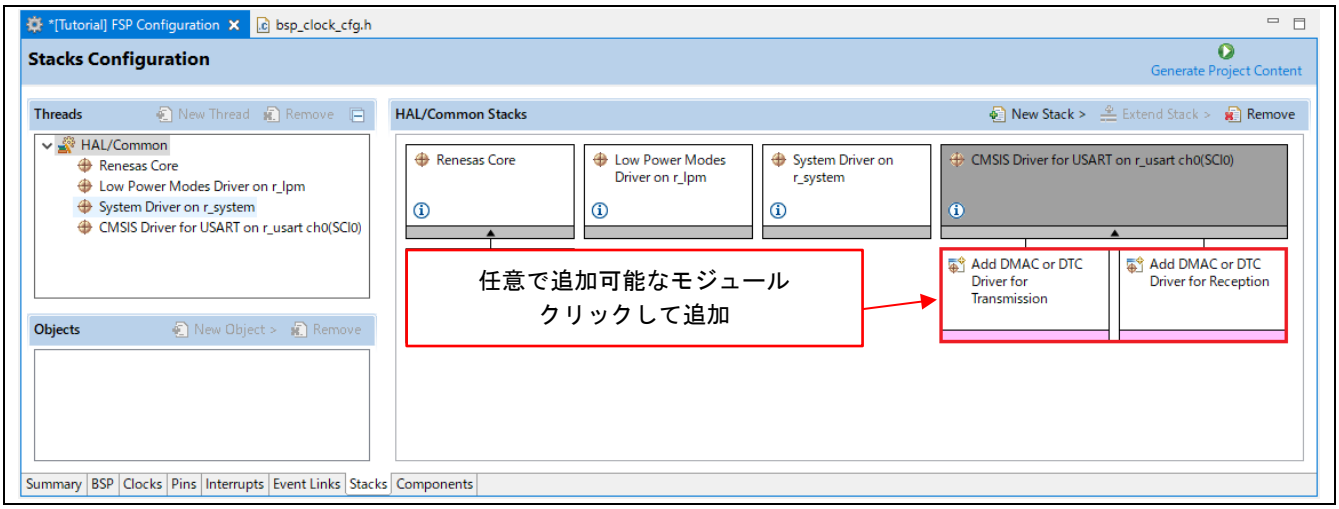


図3-42　依存関係のあるモジュール

[Stacks] ページでは、コピー&ペーストでモジュールやモジュールスタックを追加することもできます。モジュール上で右クリックし、[コピー] を選択してコピーしてください。次に、同じプロジェクト内の同じスレッドまたは異なるスレッドの [Stacks] ペインで右クリックし、[貼り付け] を選択してください。

カット&ペーストも可能です。

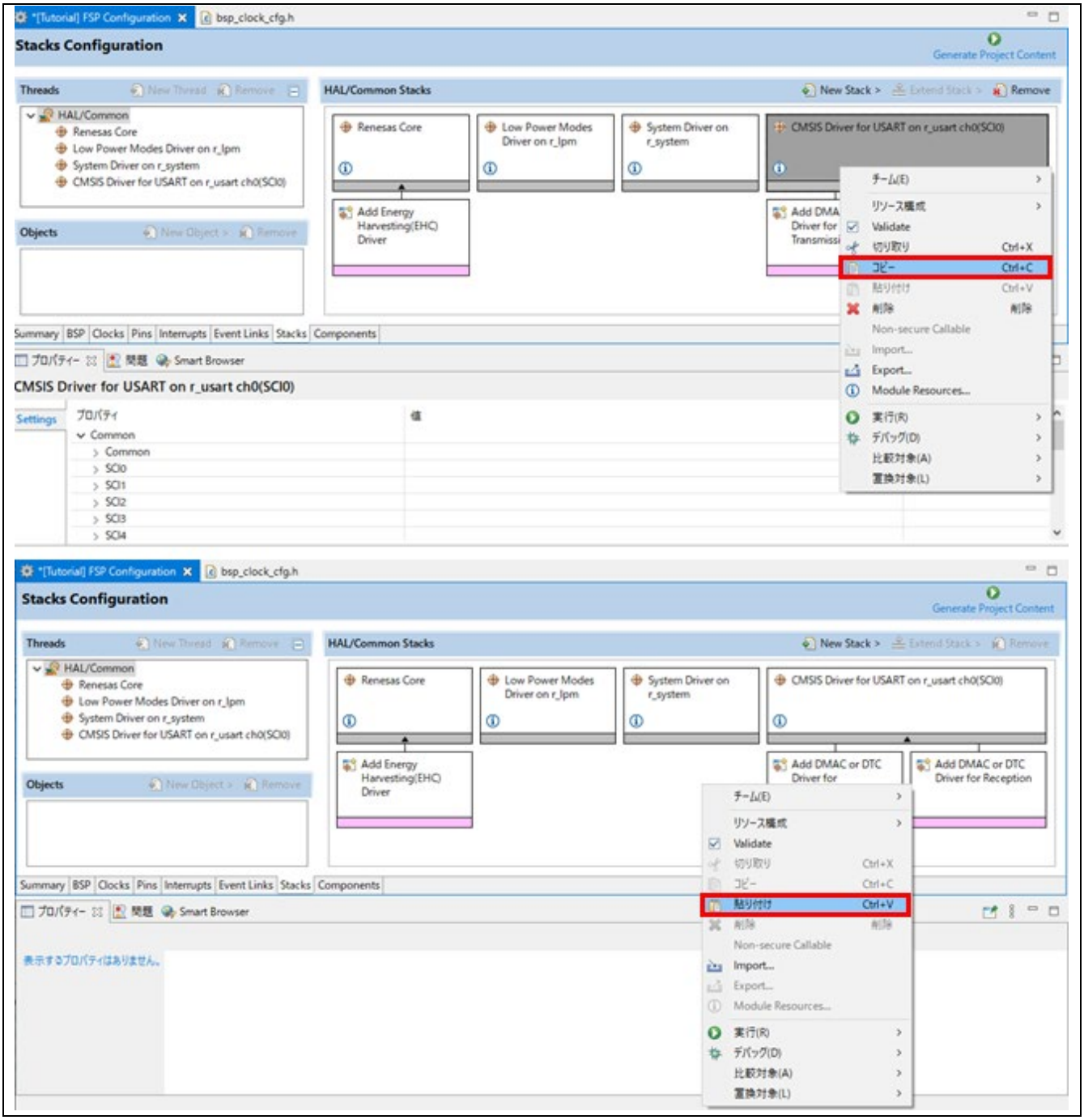


図3-43 コピー&ペーストの操作

モジュールをコピーすると、既存のモジュールとの間でモジュール名が競合します。一方のモジュールインスタンス名を変更して競合を解消してください。ハードウェアの競合がある場合、一方のモジュールを別のハードウェアチャネルに変更してください。

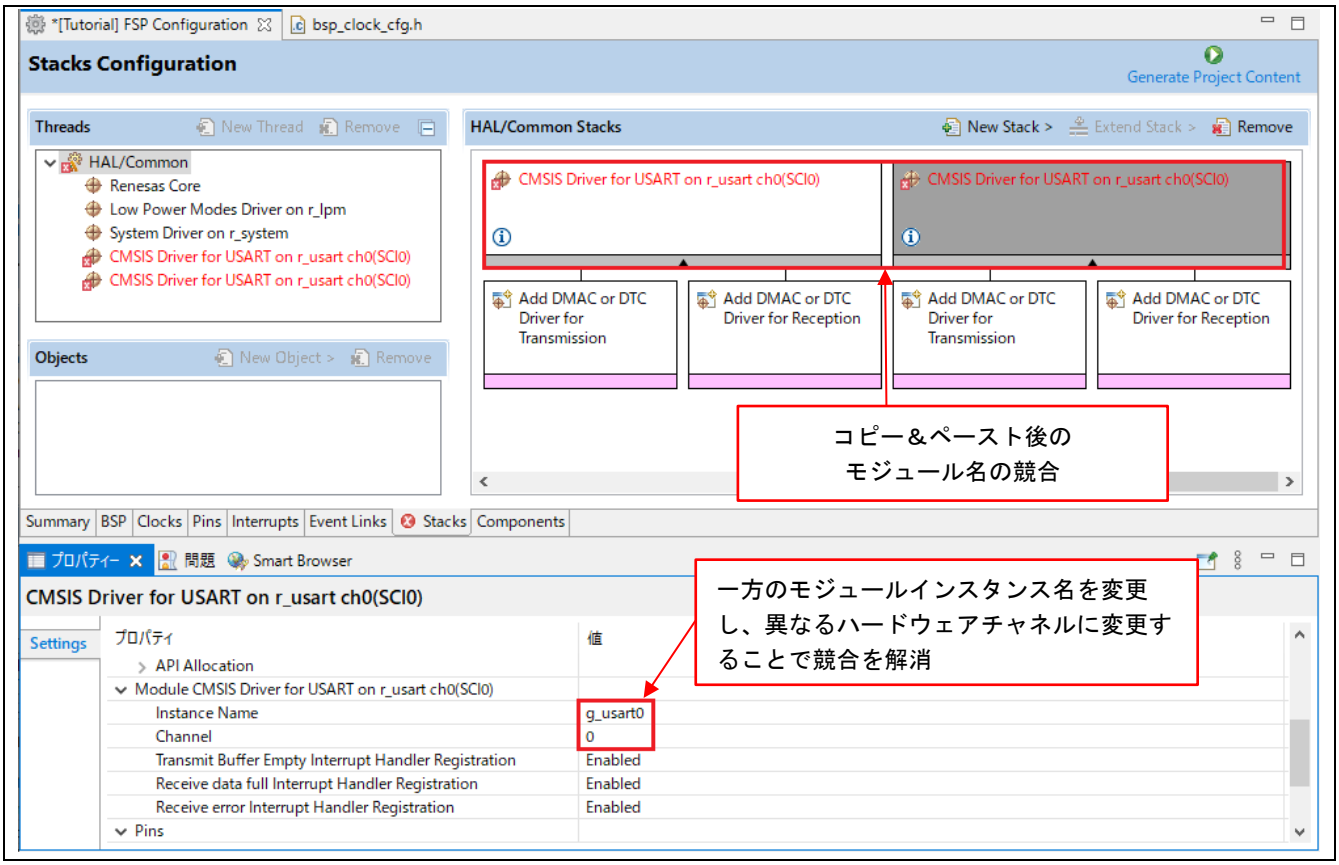


図3-44　モジュールインスタンス名の競合

[Stacks] ページでは、エクスポート&インポート操作でモジュールやモジュールスタックを追加することもできます。モジュール上で右クリックし、[Export…] を選択してモジュールの構成をXMLファイルにエクスポートしてください。次に、同じプロジェクト内の同じスレッドまたは異なるスレッドの [Stacks] ペインで右クリックし、[Import...] を選択して、エクスポート済みのXMLファイルから構成をインポートしてください。

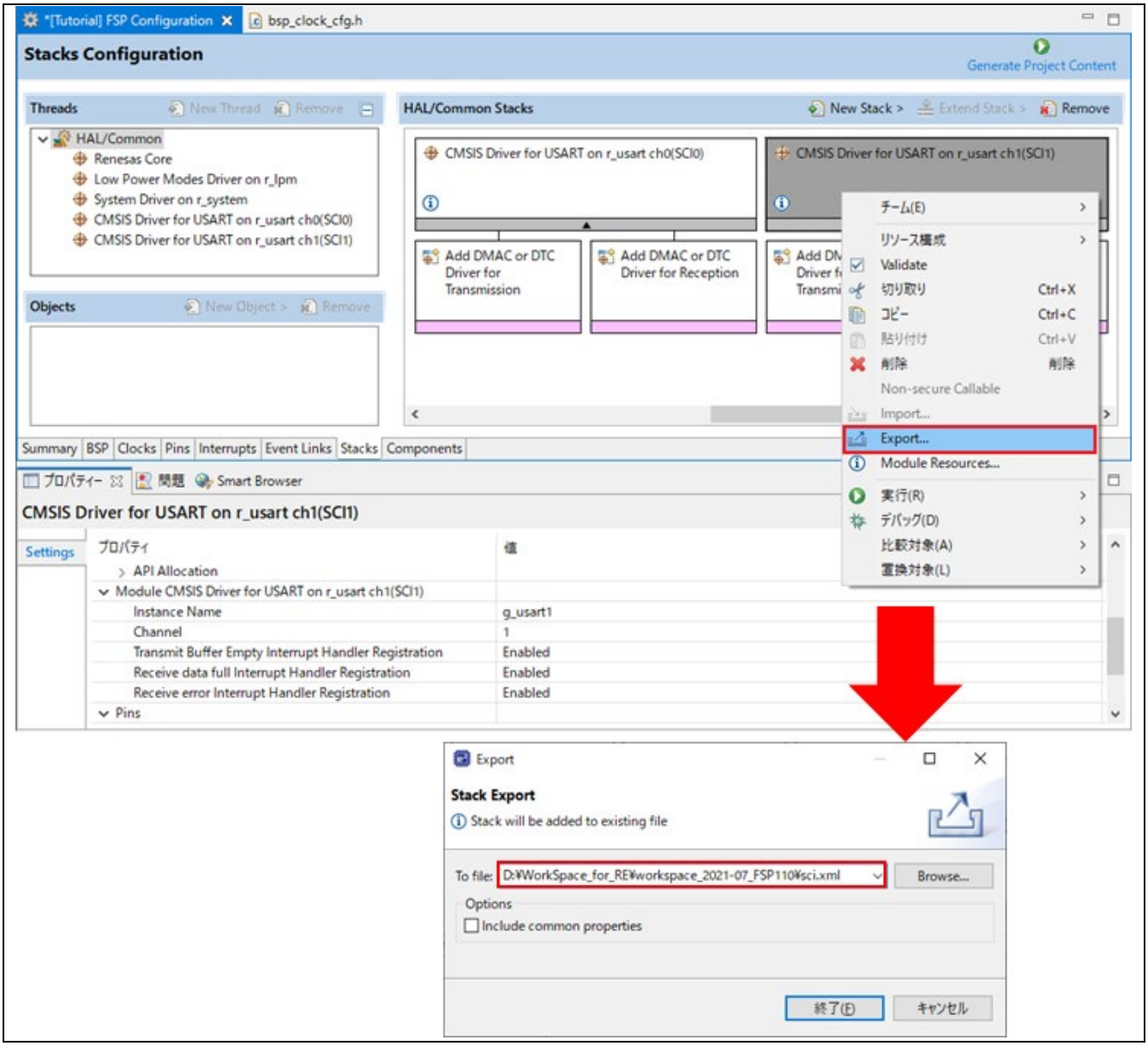


図3-45　REスタックのエクスポート

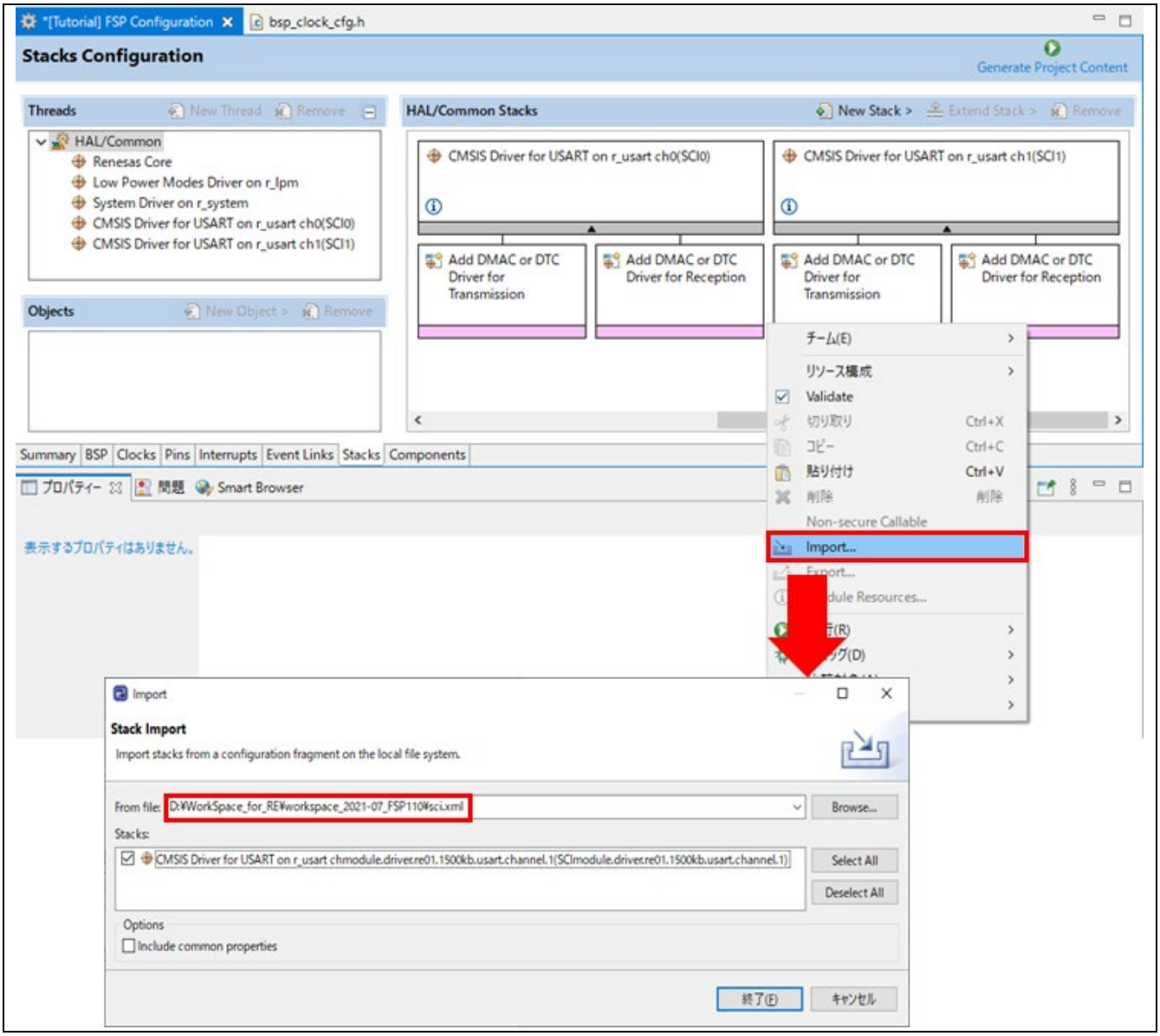


図3-46　REスタックのインポート

3.5.6 割り込み（Interrupts）ページ

[Interrupts] ページでは、RE割り込みフレームワークで使用するイベント（割り込み）やISR（割り込みサービスルーチン）の管理ができます。

[Interrupts] ページは 2 つのペインで構成されています。

1. [User Events] ペイン：ユーザが独自に作成したイベントの一覧を表示します。
2. [Allocations] ペイン：[Stacks] ページで追加した RE モジュールのイベント一覧を表示します。

   両ペインで、“Event” 欄はイベント名を表示します。“ISR” 欄は、対応する “Event” 欄のイベントの割り込みハンドラ名を表示します。

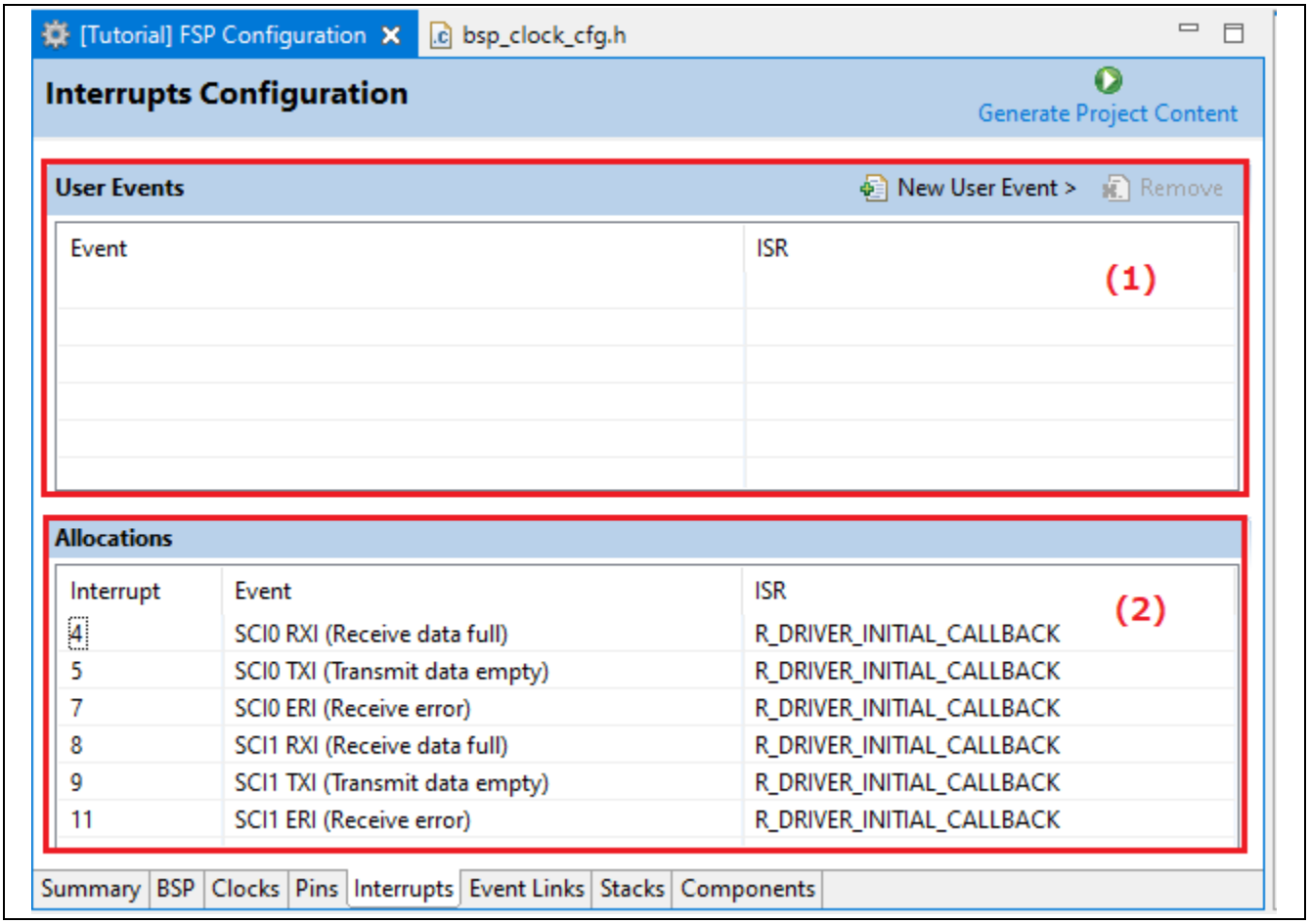


図3-47 [Interrupts] ページ

ユーザイベントと、それに対応するISRを作成するには、[New User Event] ボタンを押し、作成するイベントを選択してください。

注：割り込みをDTCやDMACのトリガとして使用する場合にもISRの登録は必要です。ISRのダミー関数を作成してください。

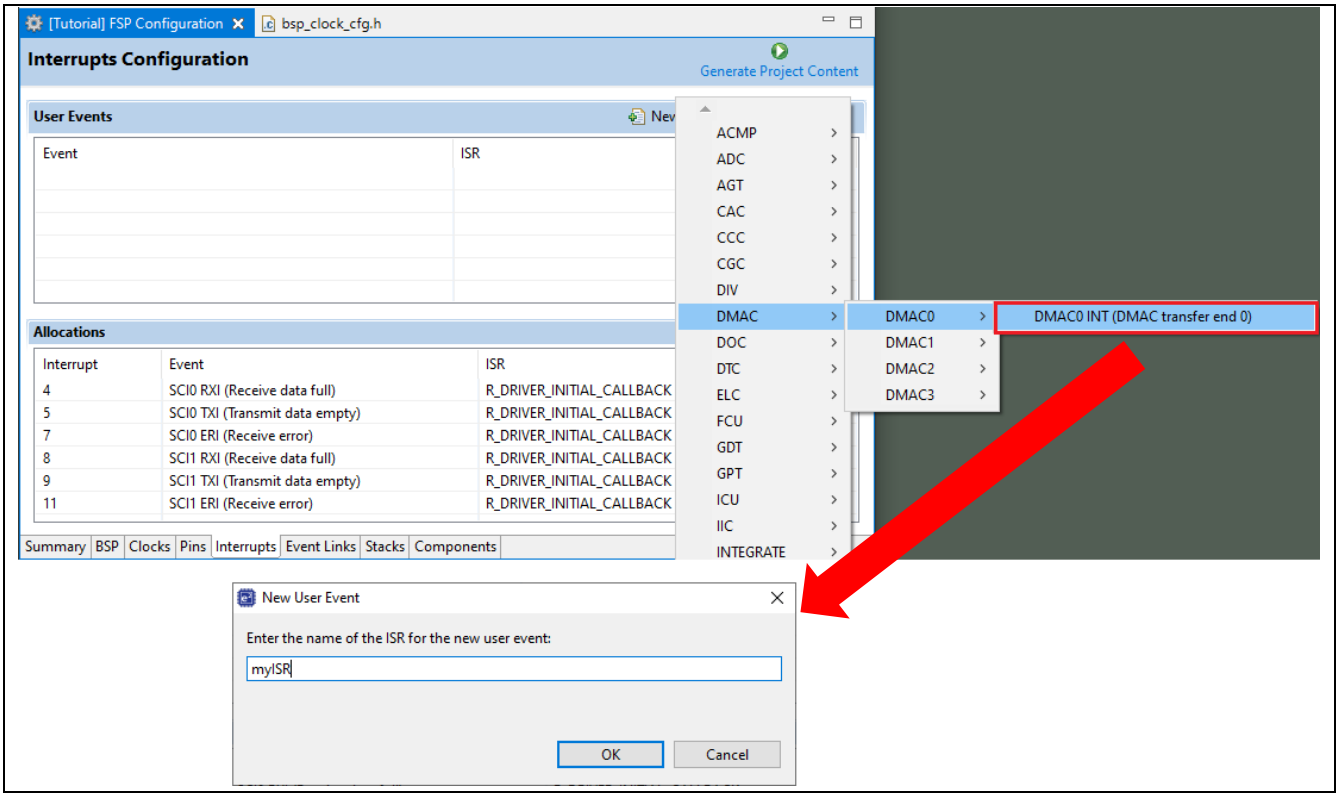


図3-48　新規ユーザイベントの追加

作成された新規イベントは “User Events” ペインに表示されます。

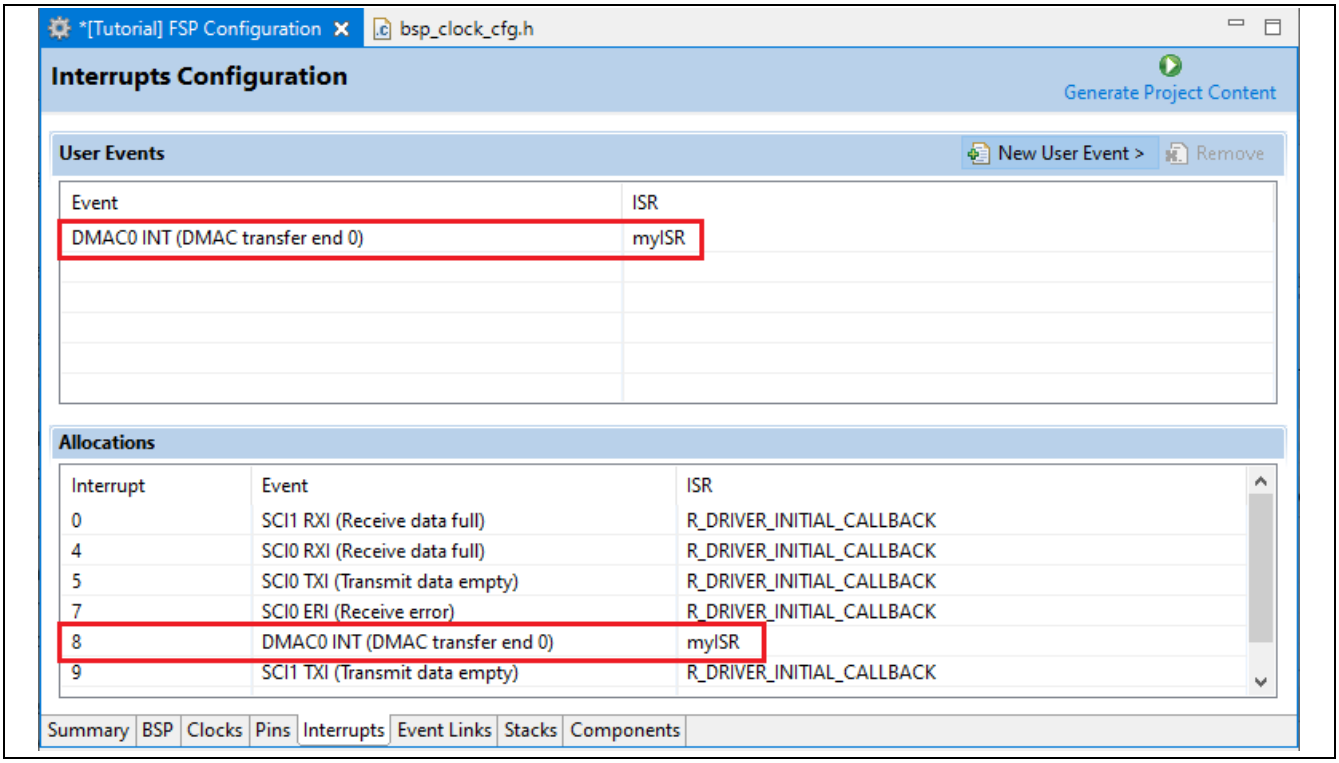


図3-49　作成されたユーザイベント

ユーザイベントを削除するには、“User Events” ペインでイベントを選択し、 ボタンを押してください（REのモジュール追加によって “Allocations” ペインに表示されたイベントは削除できません）。

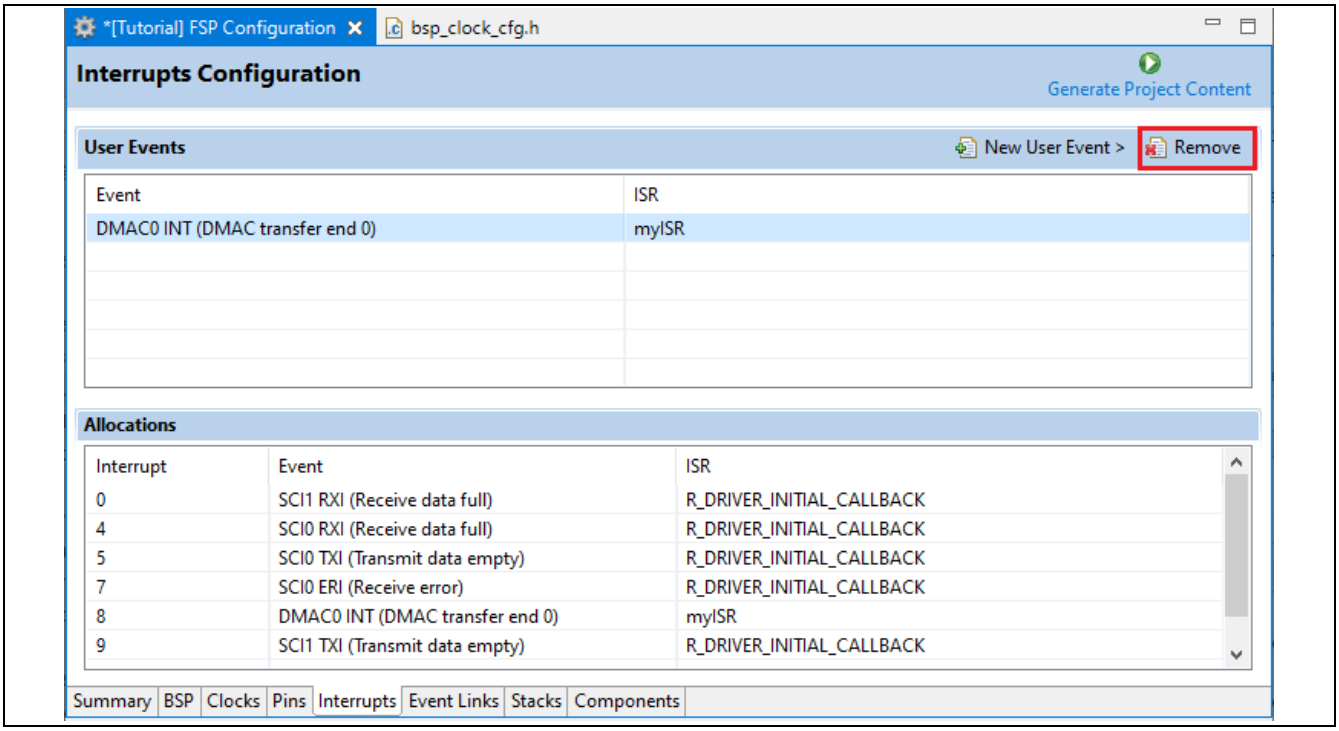


図3-50　ユーザイベントの削除

ユーザイベントの削除は、[プロパティ] ウィンドウで “Disabled” を指定することでも行えます。

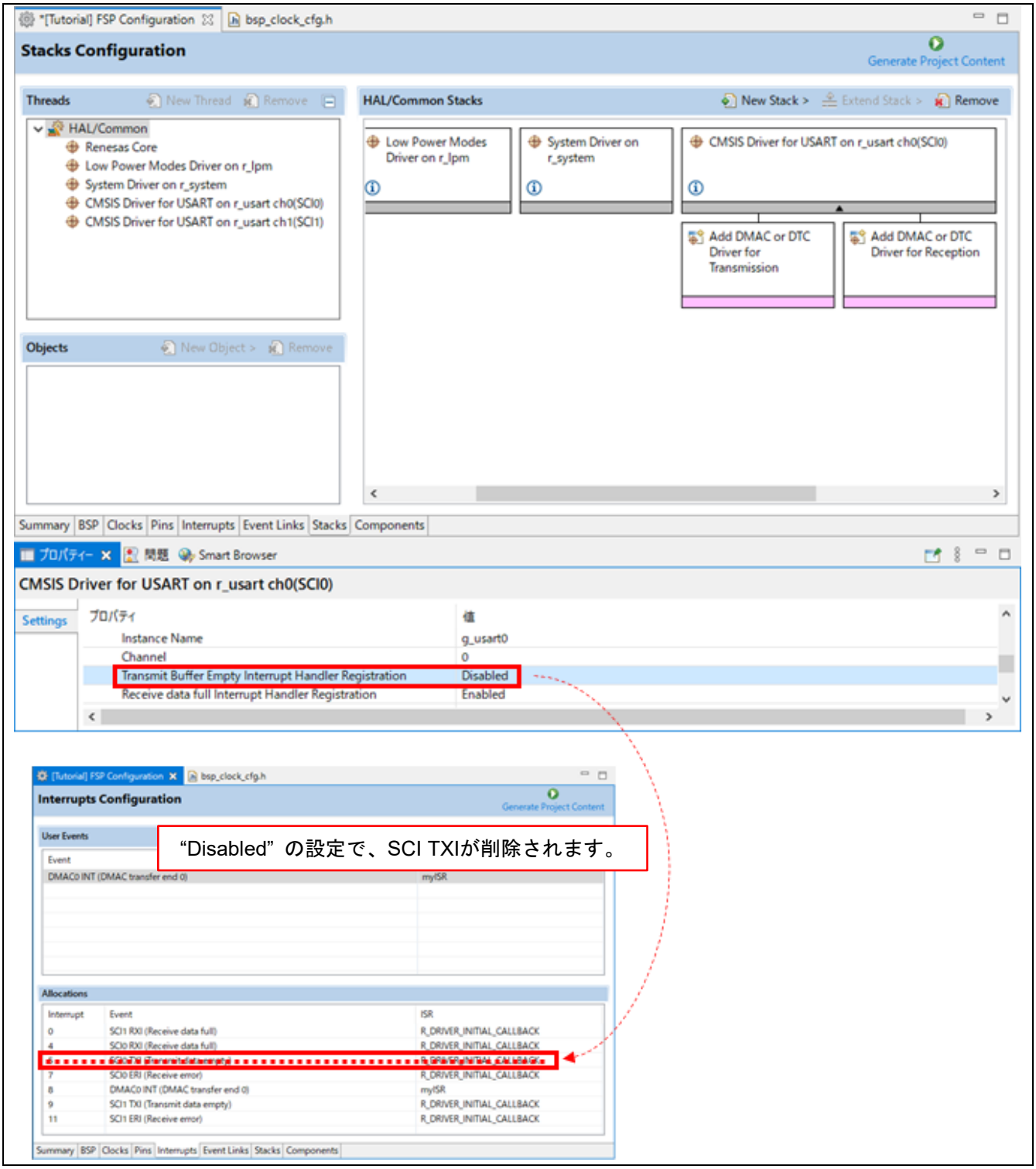


図3-51 [プロパティ] ウィンドウでのユーザイベントの削除

3.5.7 コンポーネント（Components）ページ

[Components] ページでは、アプリケーションで必要な個々のモジュールを取り込んだり、削除したりできます。

すべてのREプロジェクトに共通して必要なモジュールはあらかじめ選択されています（例：HAL Drivers → re01_1500kb → r_adc）。

[Stacks] ページで選択したドライバに必要なモジュールは、すべて自動で取り込まれます。その他のモジュールについては、必要なコンポーネントのチェックボックスをクリックすることで、取り込んだり削除したりできます。

注：アプリケーションにモジュールを追加するには、通常は [Stacks] ページを用います。[Components] ページは、主に、インストール済みソフトウェアパッケージで使用可能なコンポーネント一覧を参照するために用います。

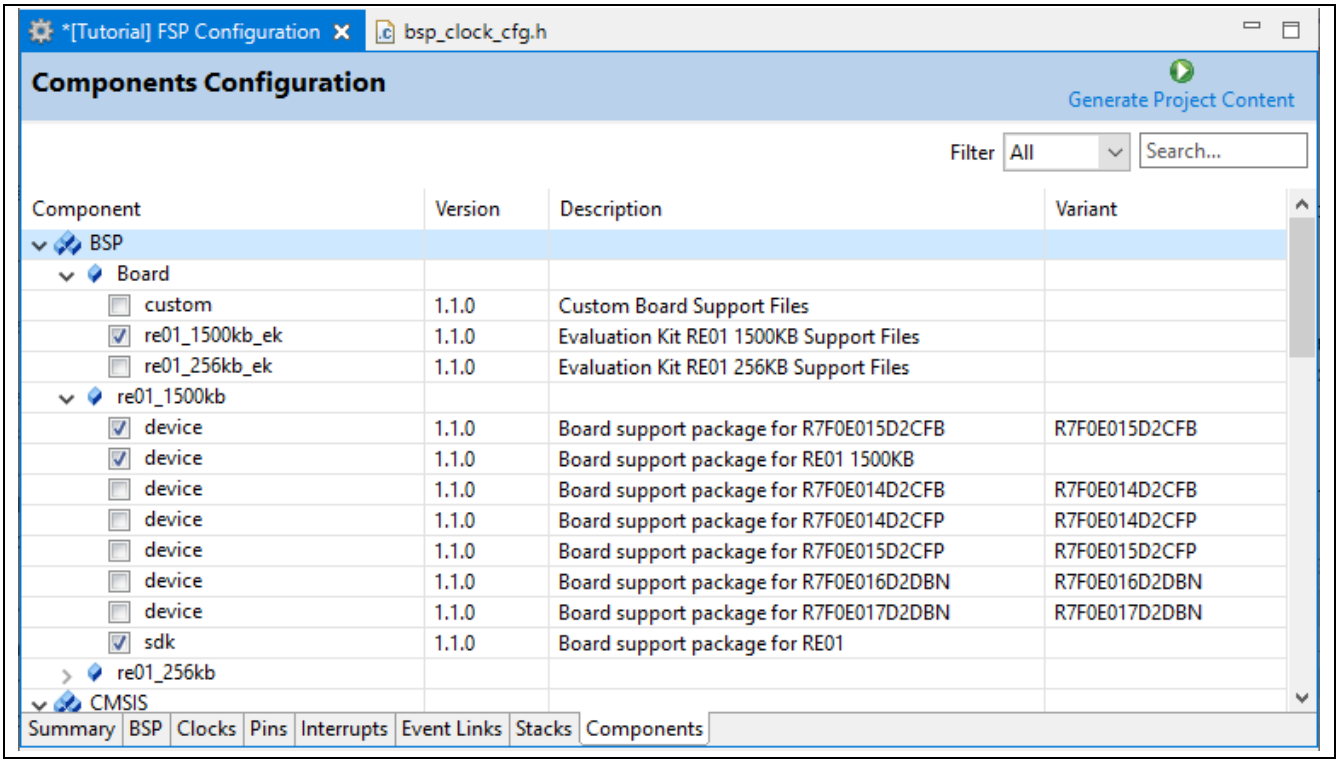


図3-52 [Components] ページ

注：ユーザアプリケーションのコーディング

ユーザアプリケーションのコードは絶対に main.c に追加しないでください。各ページで [Generate Project Content] ボタンをクリックするたびに main.c が上書きされることに注意してください。これはスマート・コンフィグレータの特徴です。hal_entry.c やその他の*.c ファイルにユーザアプリケーションのコードを適切に実装してください。

4. ビルド

この章では e² studio のビルド構成（configuration）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設定でビルドされたプロジェクトは正しく動作します。しかし、ビルドオプション（ツールチェーンのバージョンや、最適化オプションなど）を変更したい場合は、プロジェクトをビルドする前に以下の手順を実行してください。

1. サンプルプロジェクト（例：an4950_gpio_re_1500kb）を右クリックし、[プロパティ] を選択するか、ショートカットキー[Alt] +[Enter] で [プロパティ] ウィンドウを開きます。

   [プロパティ] ウィンドウはワークスペース単位、プロジェクト単位、およびソースファイル単位で設定できます。プロジェクト単位の [プロパティ] ウィンドウではソースファイル単位より多くのプロパティを設定でき、その設定はプロジェクトのワークスペース内の全てのファイルに共通に適用されます。

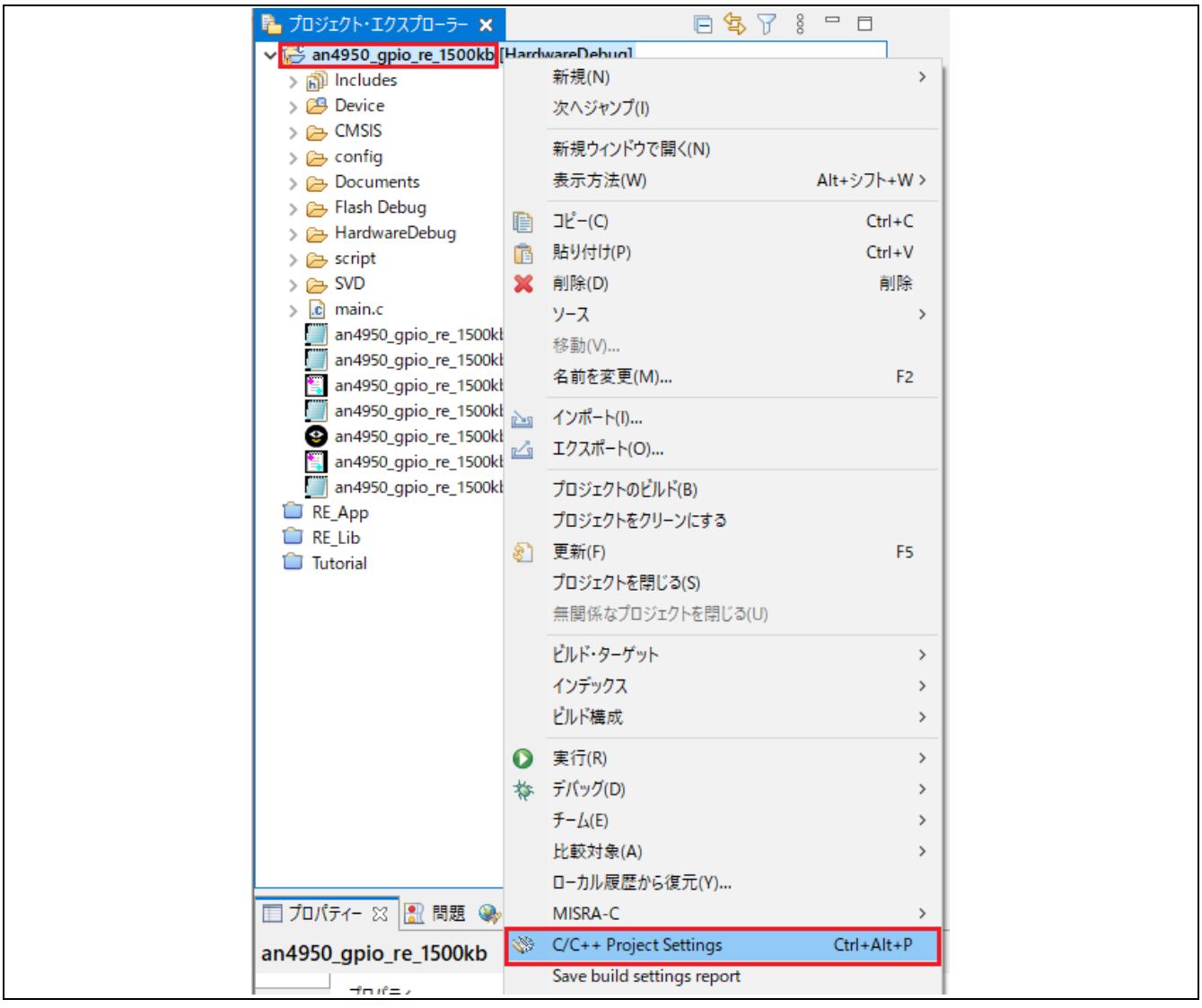


図4-1 [プロパティ] ウィンドウを開く

2. [C/C++ビルド] → [設定] → [Toolchain] タブの順に選択して、ツールチェーンのバージョンの表示や編集を行います。
図4-2を参照してください。現在のバージョンは6.3.1.20170620ですが、必要に応じて [バージョン] オプションを選択して、ツールチェーンのバージョンを変更してください。

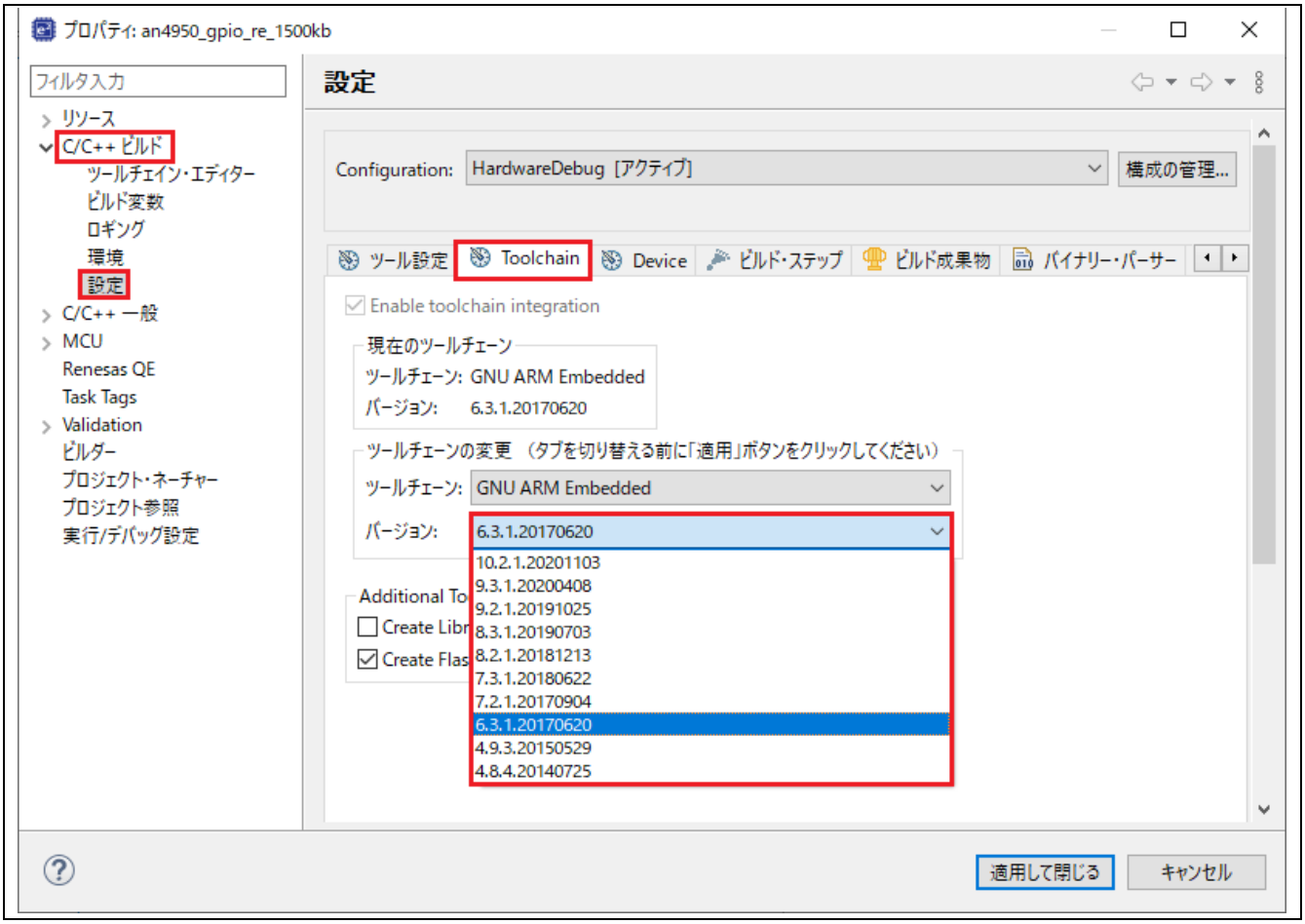


図4-2 ツールチェーンのバージョンの変更

3. [C/C++ビルド] → [環境] の順に選択して、ビルドオプションの設定および環境変数の追加や編集を行います。

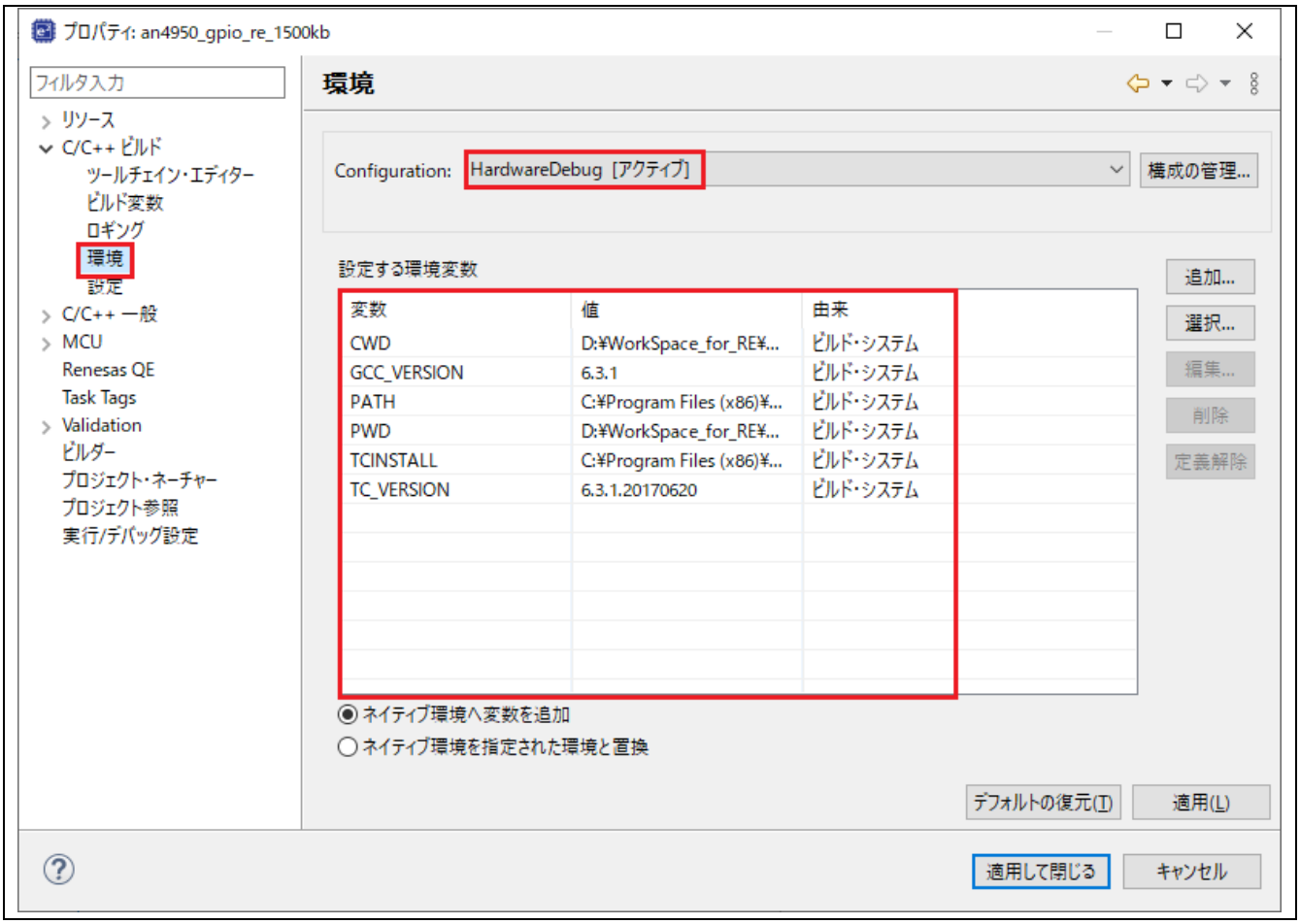


図4-3 ビルド環境の設定

4. ビルドオプションを設定します。

   [C/C++ビルド] → [設定] → [ツール設定] タブの順に選択して、コンパイラやリンカ等で使用するビルドオプションを設定できます。

   すべての設定は [すべてのオプション] に要約されており、ビルド処理で使われるコマンドが表示されます。

   “ビルド構成“ は画面の上部にある [Configuration] ドロップダウンリストで切り替えができます。各ビルド構成はビルドオプション一式を管理します。

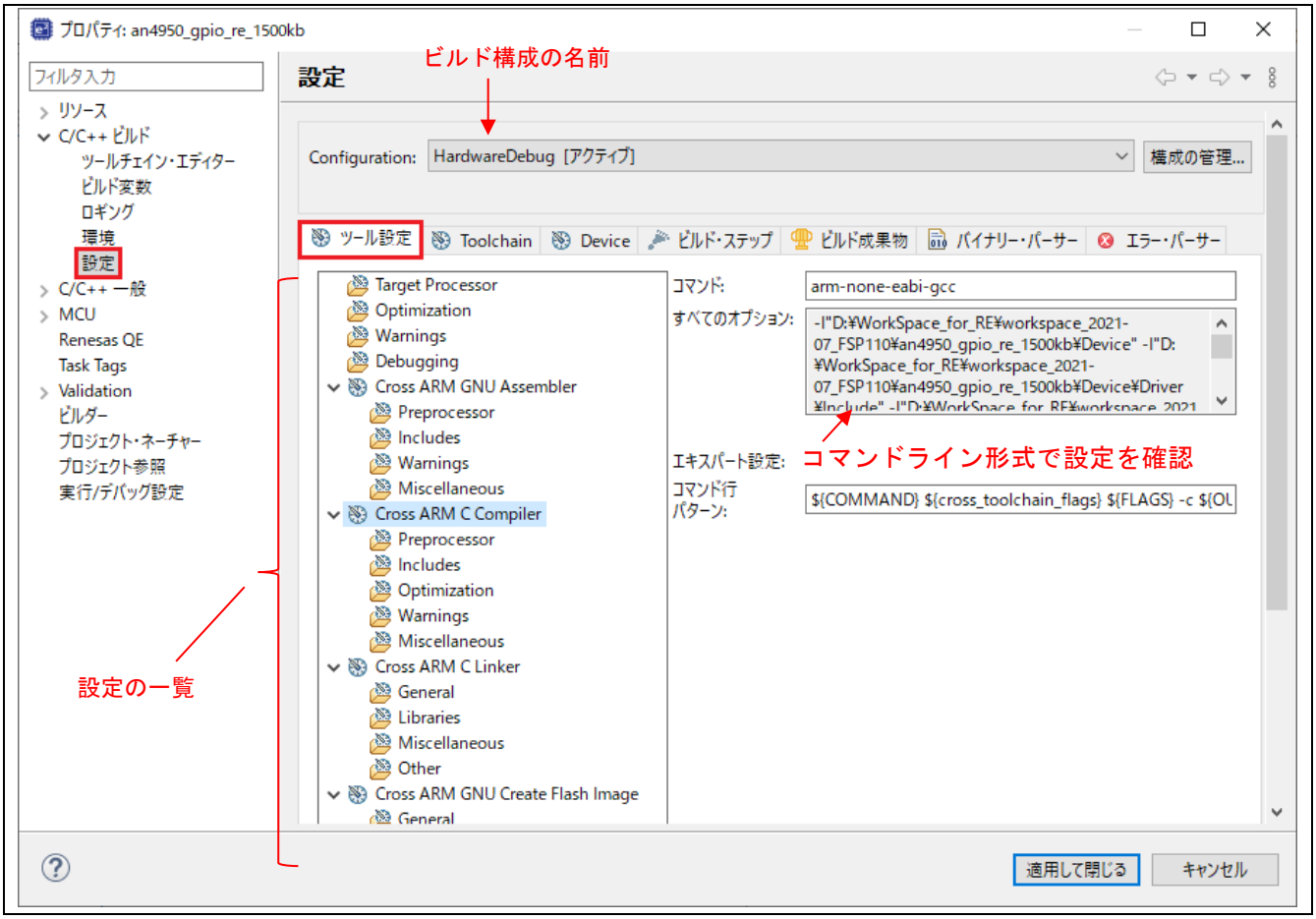


図4-4 ビルドオプションの設定

ビルドオプションの詳細は、“{コンパイラのインストールディレクトリ}\share\doc” に保管されているコンパイラのユーザーズマニュアルで説明されています。例えば、“C:\Program Files (x86)\GNU Tools ARM Embedded\6 2017-q2-update\share\doc" に用意されています。

4.1.1 推奨されるビルド設定

- 下図で示すように、[C/C++ビルド] の下に [ツールチェイン・エディター]があります。**この設定を変更しないでください**。ツールチェイン・エディターはルネサス製ビルド・プラグインでサポートされていないツールチェーンに使用します。
- [ツール設定] → [Warnings] でチェックされているオプションを一旦すべて解除してください。

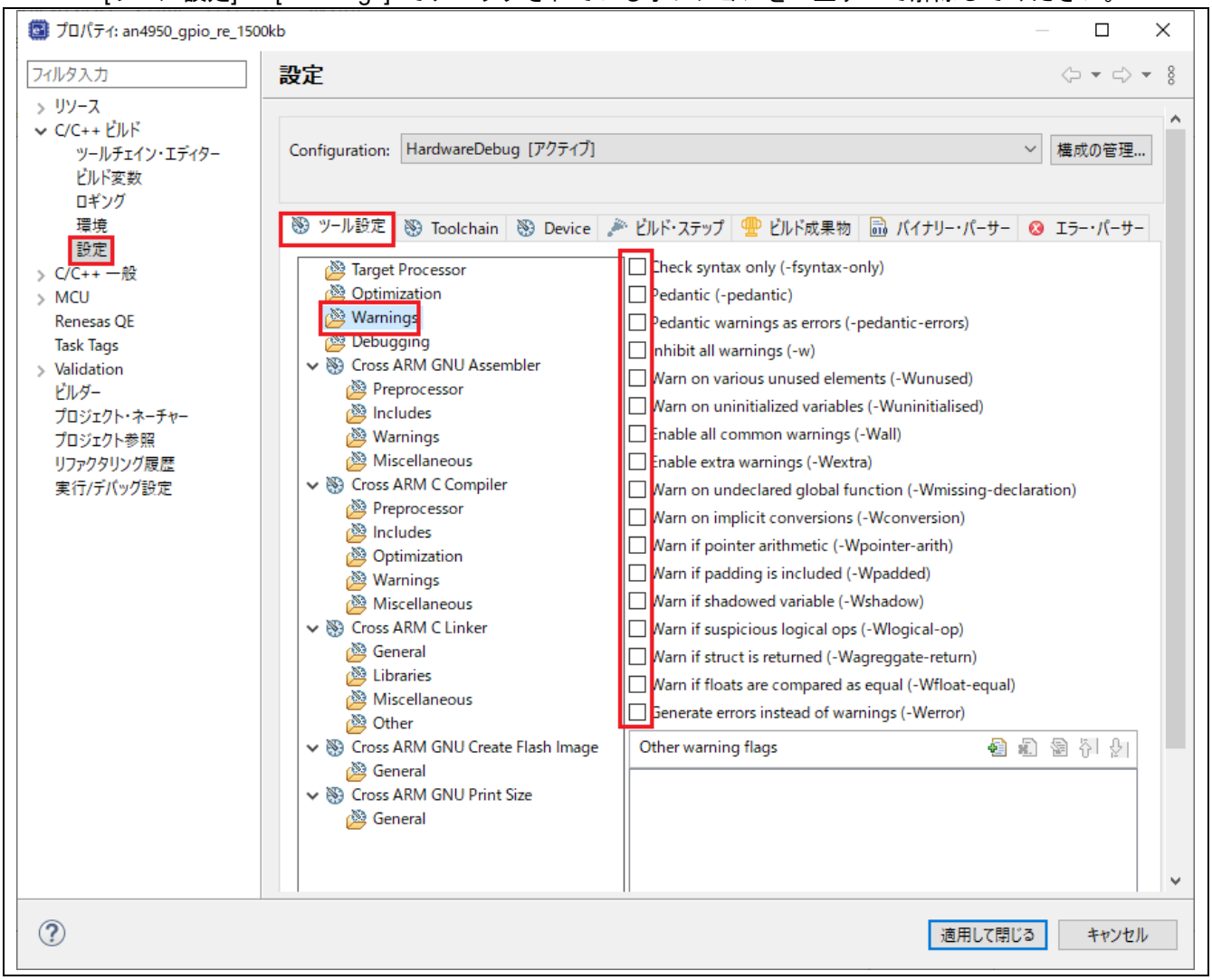


図4-5 すべての “Warning” オプションを解除

- 本ドライバは “newlib-nano” オプションなしで評価されました。標準ライブラリを使う上で問題がある場合、このオプションを無効にしてください（プロジェクト作成時にデフォルトで有効になっています）。

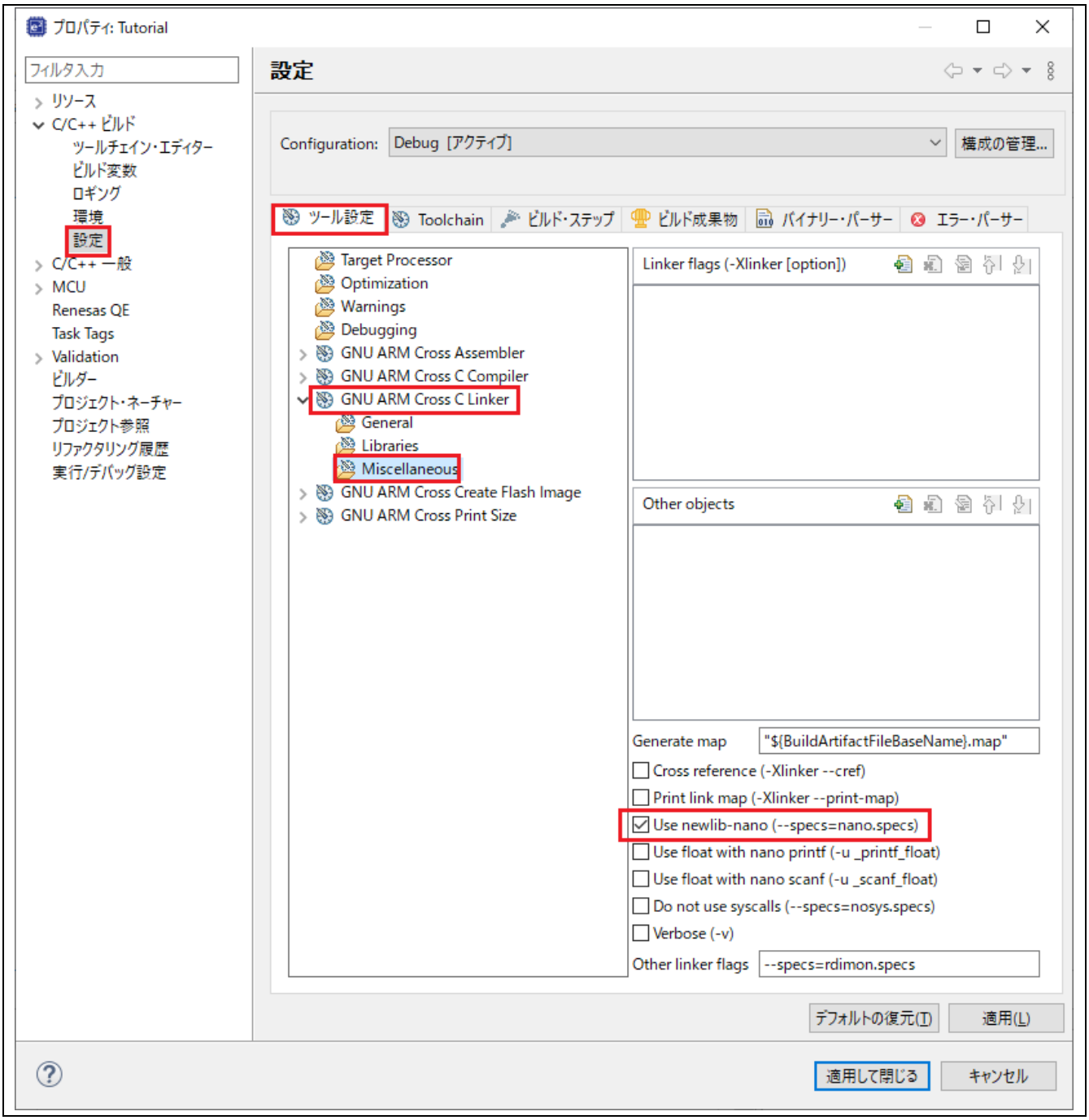


図4-6 “newlib-nano” オプション

4.2 サンプルプロジェクトのビルド

以下の手順でプロジェクトをビルドします。

1. プロジェクト名を右クリックし、[プロジェクトのビルド] を選択します。

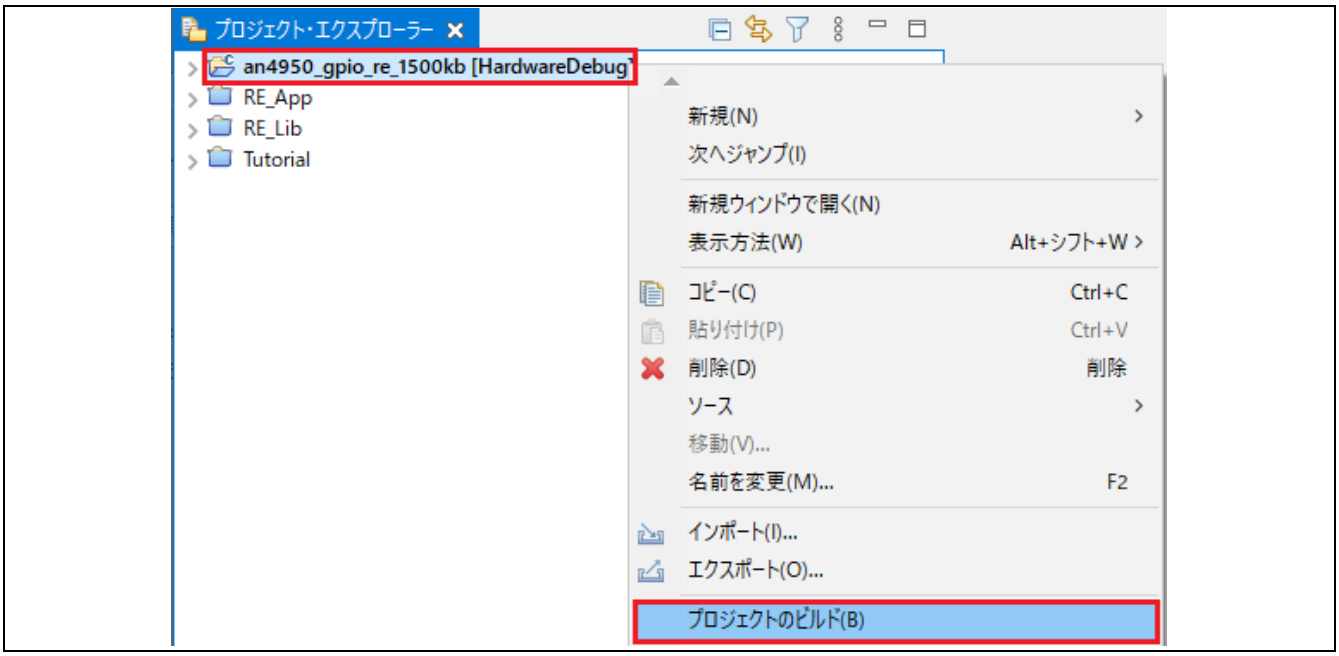


図4-7 サンプルプロジェクトのビルド

2. [コンソール] ウィンドウに、ビルドが成功したことを示すビルド完了メッセージが表示されます。

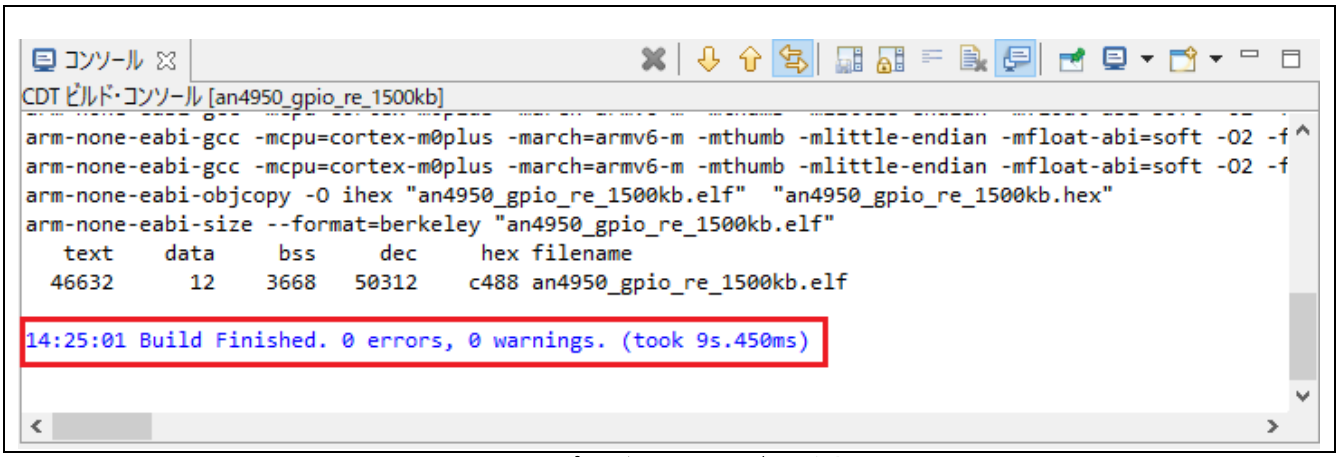


図4-8 プロジェクトのビルド成功

3. ビルドに失敗した場合、エラーの詳細（エラーの発生位置（ファイル、カラム、行）、要因など）を含むエラーメッセージを [コンソール] ウィンドウに表示します。エラーを確認し、ソースコードまたは設定を修正してプロジェクトを再度ビルドしてください。

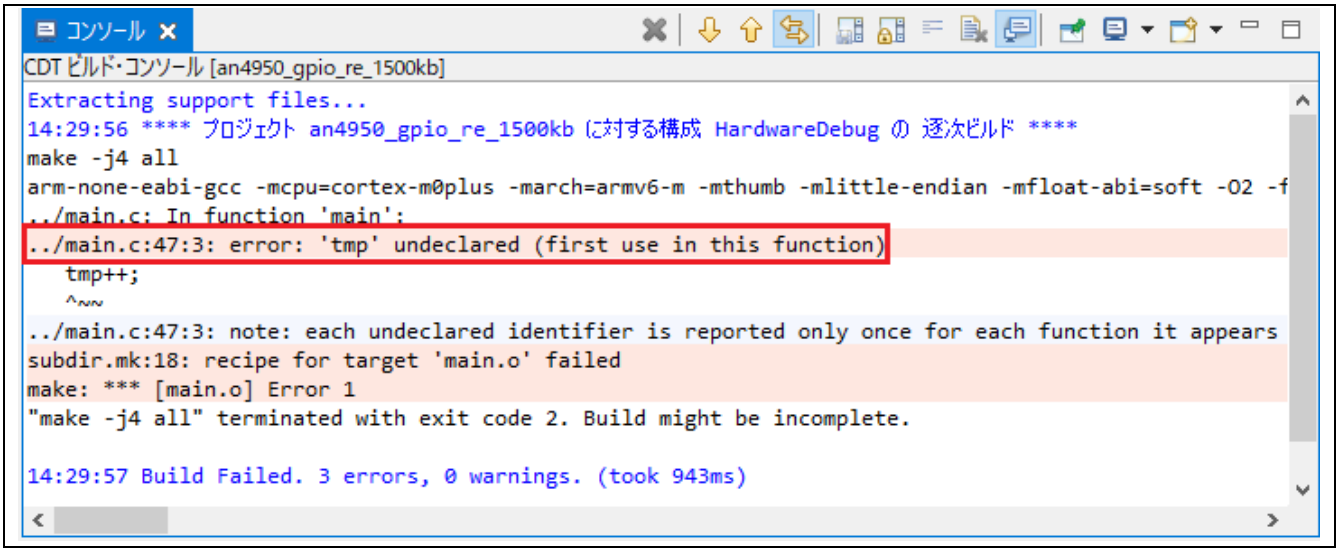


図4-9　ビルドの失敗時に表示されるエラーメッセージ

4.3 ビルド構成の設定のエクスポート

プロジェクトレポート機能により、e² studioのプロジェクトのビルド設定をファイルに保存できます。

1. [プロジェクトエクスプローラー] ビューで右クリックし、ポップアップメニューを表示します。
2. [Save build settings report] を選択してビルド設定のレポートを保存します。

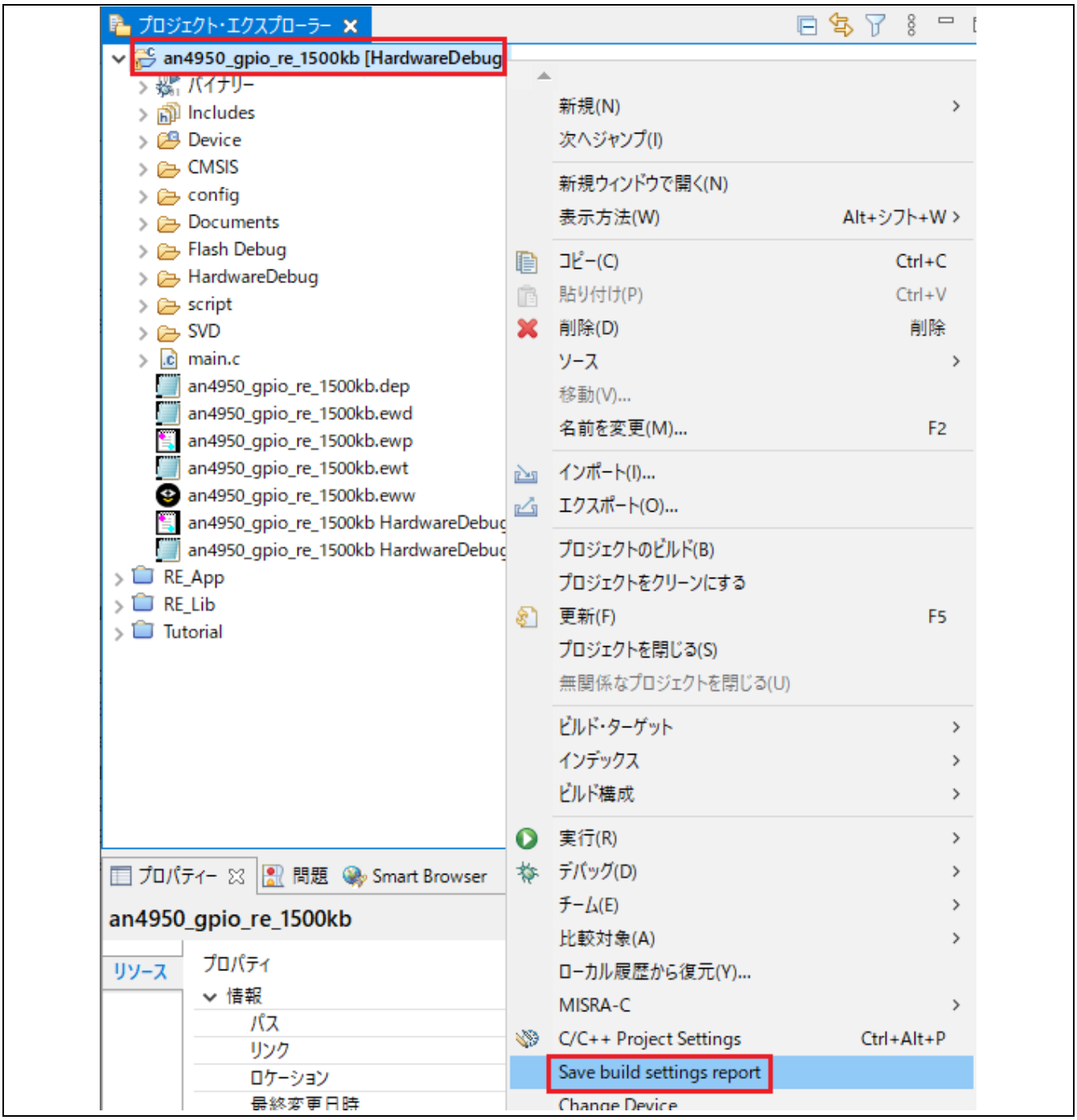


図4-10 プロジェクトレポート機能

5. デバッグ

この章では、e² studioのデバッグ構成と主なデバッグ機能の使い方について説明します。3.4節でダウンロードおよびインポートされたサンプルプロジェクト（例：r01an4950ej0101、以降「サンプルプロジェクト」と呼ぶ）が、J-Link ARMとEK-RE01 1500KBボードを動作環境とする例を説明します。

1. e² studioでサンプルプロジェクトのワークスペースを開き、[デバッグ] パースペクティブを選択します。

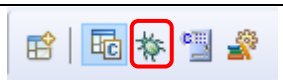


図5-1 [デバッグ] パースペクティブへの切り替え

「パースペクティブ」とは、あらかじめ（開発ツールと関連した）ビューを組み合わせてワークベンチウィンドウのレイアウトを定義したものを指します。それぞれのパースペクティブは、特定の用途向けのビュー、メニュー、ツールバーの組み合わせで構成されます。たとえば、以下のようなパースペクティブがあります。

- [デバッグ] パースペクティブ：プログラムのデバッグに必要なビューを表示します。
- [RE Configuration] パースペクティブ：エディタ画面の “configuration.xml” とあわせて RE configuration を開き、プロジェクト構成を設定する [パッケージ] ビューや [プロパティ] ビューを表示します。
- [C/C++] パースペクティブ：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表示します。

[デバッグ] パースペクティブ以外でユーザがデバッガに接続しようとすると、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に促します。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。ユーザはパースペクティグをカスタマイズしたり、新しいパースペクティブを追加したりすることができます。

5.1 既存デバッグ構成の変更

初回のデバッグ時にのみ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] ペインでサンプル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。

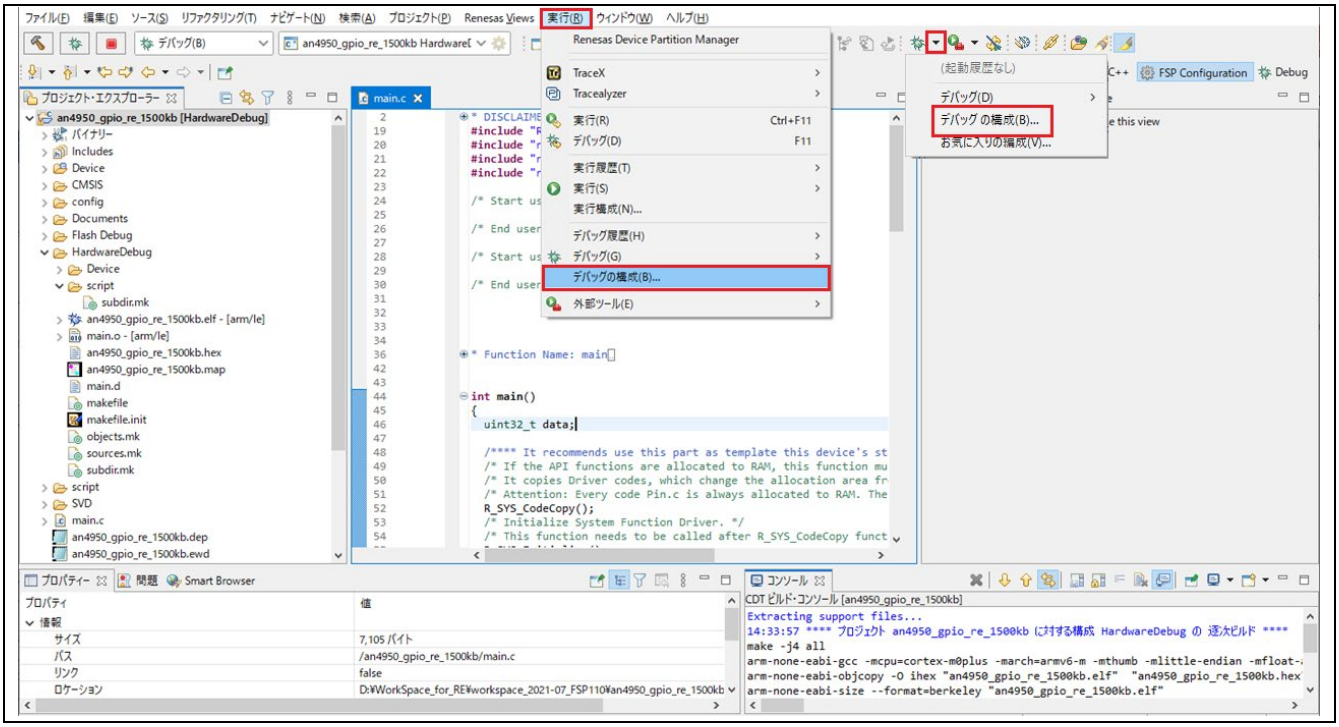


図5-2 [デバッグ構成] ウィンドウを開く

2. [デバッグ構成] ウィンドウで、[Renesas GDB Hardware Debugging] → [an4950_gpio_re_1500kb HardwareDebug] の順に選択します。[メイン] タブをクリックして、ロードモジュールが “an4950_gpio_re_1500kb.elf” であることを確認してください。

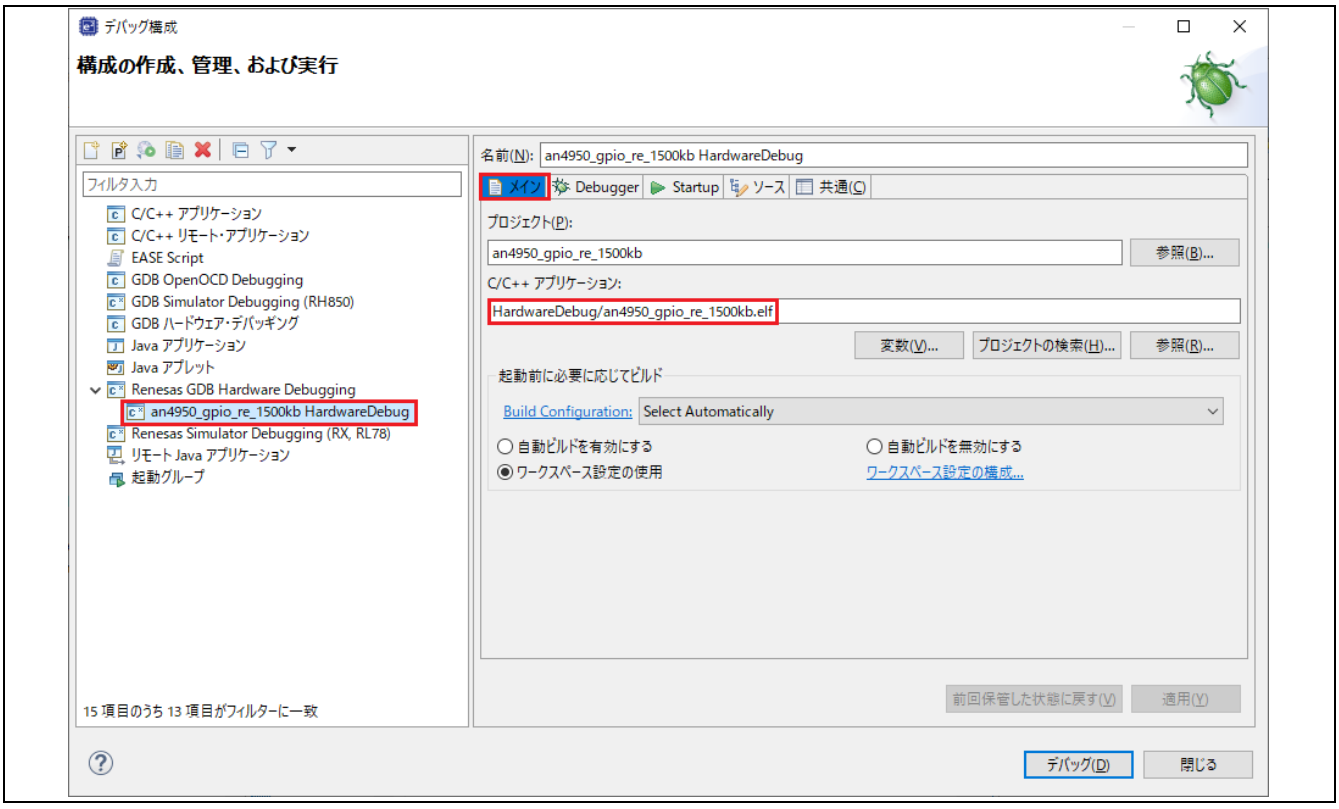


図5-3 ロードモジュールの選択

3. [Debugger] タブに切り替え、[Debug hardware] に “J-Link ARM”、[Target Device] に “R7F0E015D2CFB” を設定してください。

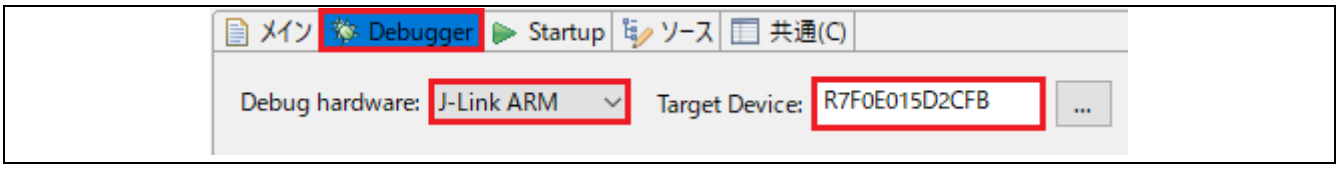


図5-4 ターゲットデバイスの選択

4. [Debugger] タブの下にあり、エミュレータ接続に関する [Connection Settings] サブタブを参照してください。以下の例は、J-Link ARMエミュレータとRSK RE01ボードを動作環境としています。

   - J-Link
     - Type：**USB**
     - Low Power Handling：**Yes**
   - Interface
     - Type：**SWD**
     - Speed：**320**
   - 接続
     - 実行前にリセット：**はい**
     - ダウンロード前にリセット：**いいえ**

   注：図5-5のデバッグ構成は例として示しています。間違った設定は誤動作やハードウェアの故障を引き起こす可能性があります。したがって、接続前にボードとエミュレータの設定を注意して確認してください。

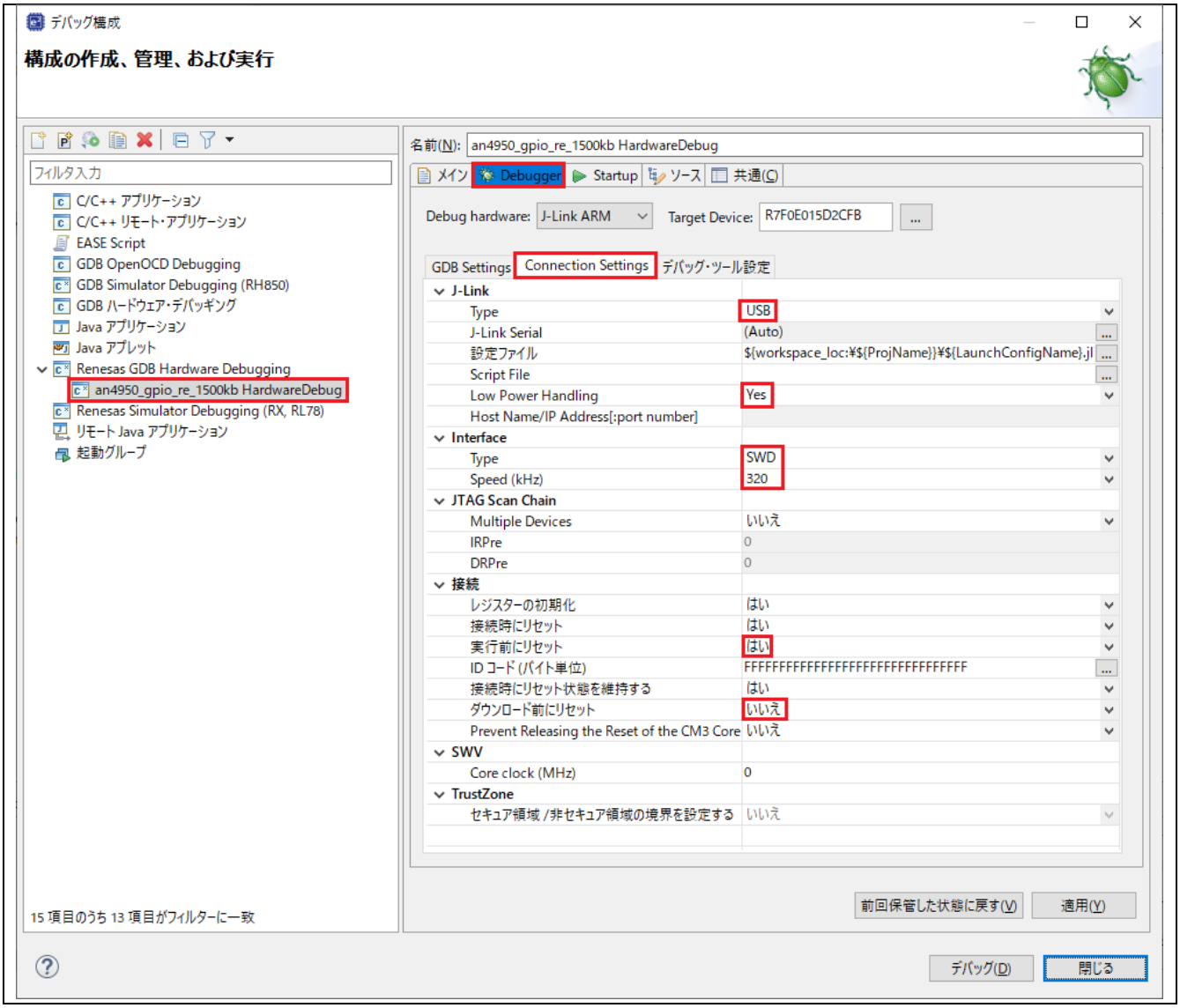

デバッグ構成
構成の作成、管理、および実行
フィルタ入力
C/C++ アプリケーション
C/C++ リモート・アプリケーション
EASE Script
GDB OpenOCD Debugging
GDB Simulator Debugging (RH850)
GDB ハードウェア・デバッギング
Java アプリケーション
Java アプレット
Renesas GDB Hardware Debugging
an4950_gpio_re_1500kb HardwareDebug
Renesas Simulator Debugging (RX, RL78)
リモート Java アプリケーション
起動グループ
15 項目のうち 13 項目がフィルターに一致
名前(N): an4950_gpio_re_1500kb HardwareDebug
メイン
Debugger
Startup
ソース
共通(C)
Debug hardware: J-Link ARM
Target Device: R7F0E015D2CFB
GDB Settings
Connection Settings
デバッグ・ツール設定
J-Link
Type USB
J-Link Serial (Auto)
設定ファイル ${workspace_loc:¥${ProjName}}¥${LaunchConfigName}.jl
Script File
Low Power Handling Yes
Host Name/IP Address[:port number]
Interface
Type SWD
Speed (kHz) 320
JTAG Scan Chain
Multiple Devices いいえ
IRPre 0
DRPre 0
接続
レジスターの初期化 はい
接続時にリセット はい
実行前にリセット はい
ID コード (バイト単位) FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
接続時にリセット状態を維持する はい
ダウンロード前にリセット いいえ
Prevent Releasing the Reset of the CM3 Core いいえ
SWV
Core clock (MHz) 0
TrustZone
セキュア領域 /非セキュア領域の境界を設定する いいえ
前回保管した状態に戻す(V)
適用(Y)
デバッグ(D)
閉じる

図5-5 接続設定の変更

5. デバッガの動作に関する [デバッグ・ツール設定] サブタブに切り替えます。詳細については、e² studio のヘルプ目次から[e2 studio ユーザガイド] → [デバッグに関する機能] を参照してください。

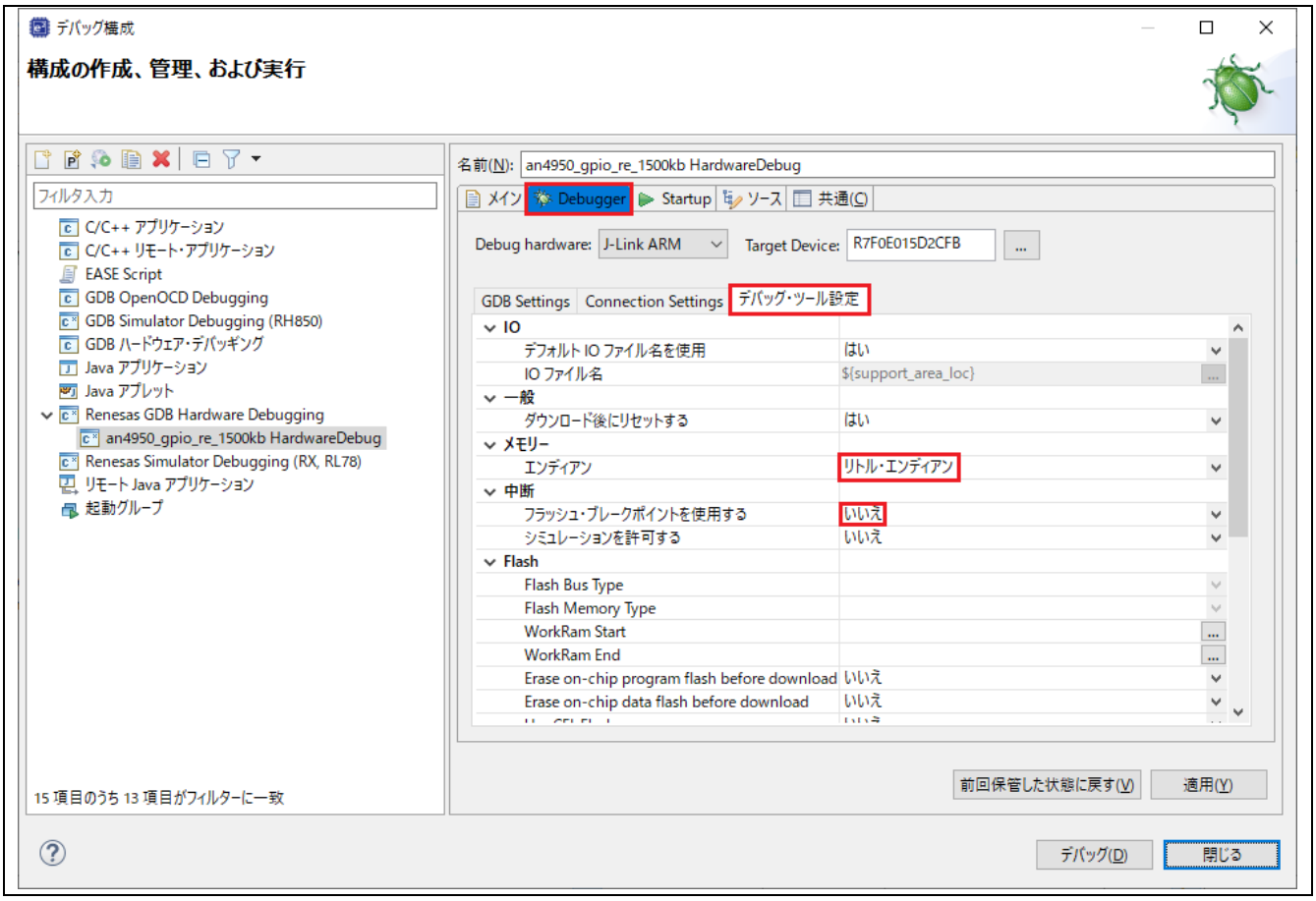


図5-6 デバッグ・ツール設定の変更

- メモリー

  — エンディアン：リトル・エンディアン

  デバッガメモリ参照時のエンディアン設定です。この設定はターゲットプログラムの動作には影響しません。

- 中断

  — フラッシュ・ブレークポイントを使用する：いいえ

  ソフトウェア・ブレークポイントをフラッシュメモリに設定することは制限されているため、ハードウェア・ブレークポイントを使う必要があります。

6. [適用] をクリックして、設定を確認してください。確認後、[デバッグ] をクリックして、デバッガを起動してください。

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（例：“startup_RE01_1500KB.c” の “Reset_Handler()”）でプログラムの実行が一旦中断されます。

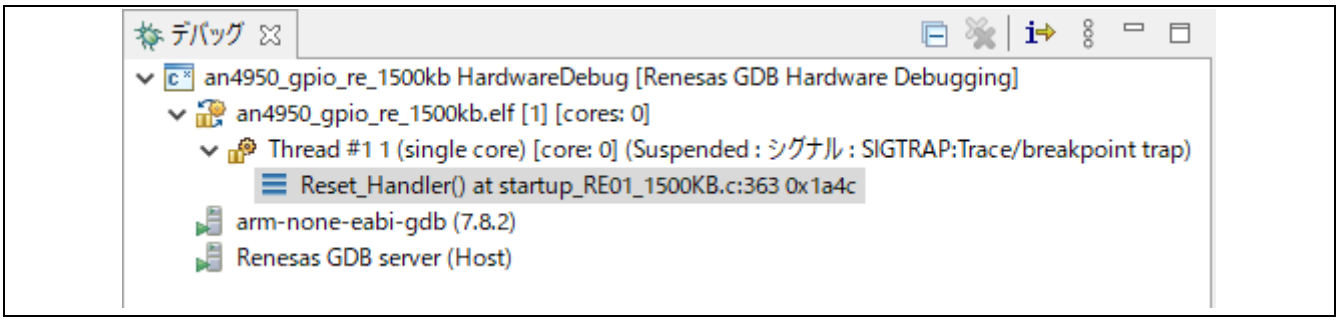


図5-7　ターゲット接続後の [デバッグ] ビュー表示

5.2 新規デバッグ構成の作成

別の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 既存デバッグ構成の変更」のステップ1を再度行い、[デバッグ構成] ウィンドウを開きます。

2. デバッグ構成（例：“an4950_gpio_re_1500kb 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起動構成（例：“an4950_gpio_re_1500kb HardwareDebug(1)”）が作成されます。設定を識別するために構成の名前を書き換えることも可能です。[名前] テキストボックスに新しい名前を入力し、[適用] ボタンをクリックしてください。

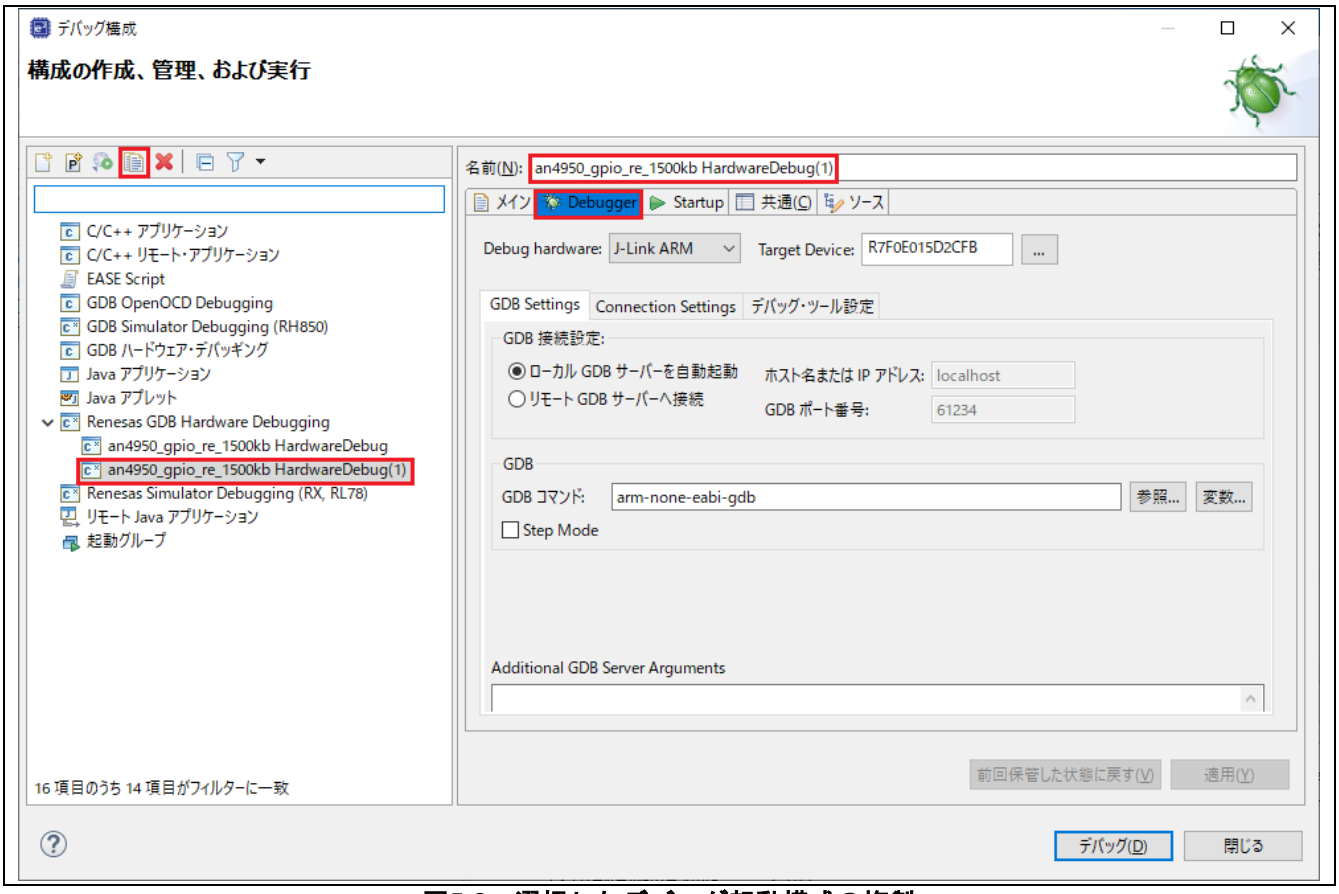


図5-8 選択したデバッグ起動構成の複製

3. デバッグ起動構成は「5.1 既存デバッグ構成の変更」で説明した方法で設定できます。例えば、[Debug hardware] を “E2 Lite (ARM)” に変更してください。

4. [デバッグ構成] ウィンドウの左のペインのツリーで、新規に追加した起動構成に[Local] と * マーク（赤い星印）が表示されている場合、どのプロジェクトにもまだ紐づいていません。その場合は [共通] タブでプロジェクト名を指定してください。

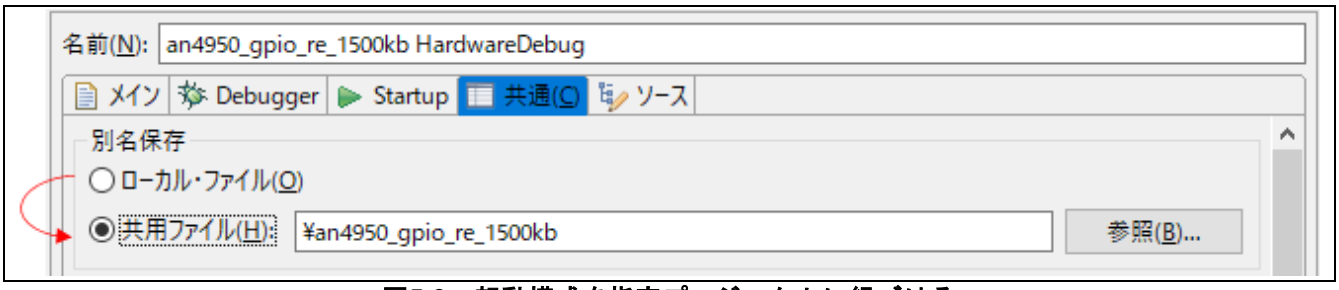


図5-9 起動構成を指定プロジェクトに紐づける

5.3 起動バー

この節では、V6.0.0以降のバージョンでサポートされている「起動バー」の使用について説明します。
起動バーはe² studioメインウィンドウのツールバー領域に位置しています。
インタフェースはとてもシンプルで、以下に示すように、選択した起動対象のビルドとデバッグに使用します。

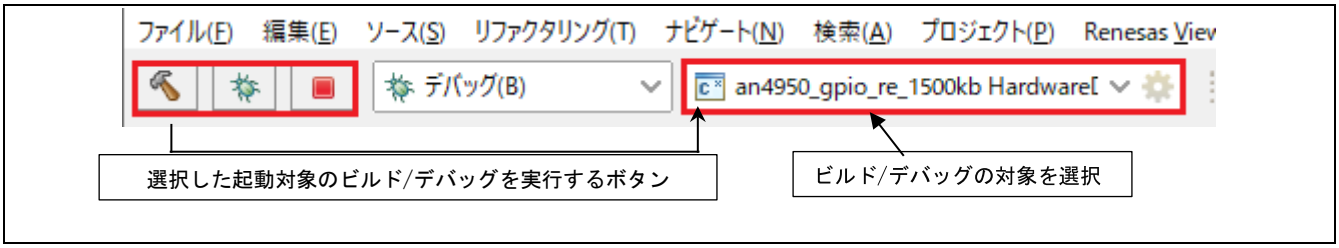


図5-10 起動バーのインタフェース

起動バーのボタンは以下の機能を持っています。（ビルドやデバッグの対象を事前に選択してください。）

-  ボタンは選択した起動構成のロードモジュールをビルドします。

  注：ファイルのツールバーには別のビルドボタン  があり、プロジェクトエクスプローラーのアクティブなビルド構成をビルドします。それに対して、起動バーのボタンはプロジェクトエクスプローラーのアクティブな状態を反映しません。

-  ボタンはそれぞれデバッガの起動と選択した起動構成の終了のトリガです。

起動バーとビルドボタンは以下のダイアログで非表示にすることができます。

- [ウィンドウ] メニュー → [設定] → [実行/デバッグ] → [起動中] → [Launch Bar] の順に選択します。

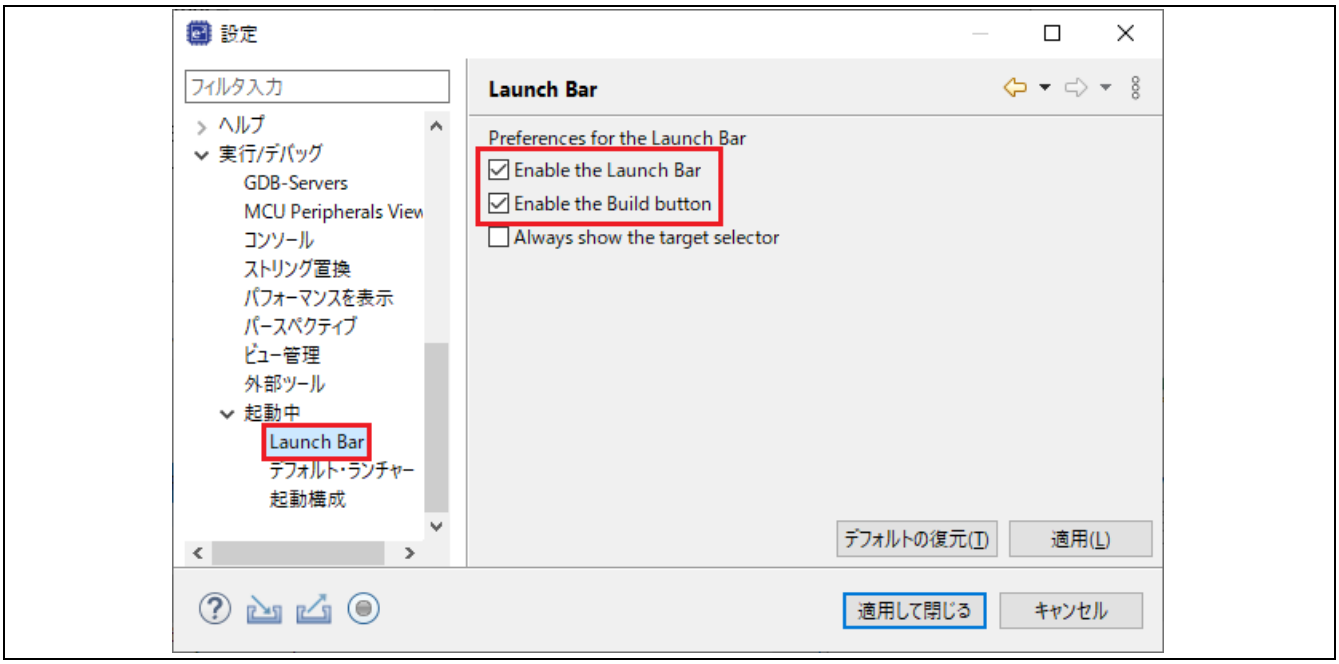


図5-11 起動バーの表示／非表示

5.4 基本的なデバッグ機能

この章では、e² studioがサポートする以下の代表的なデバッグビューを説明します。

- 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ボタンがあります。

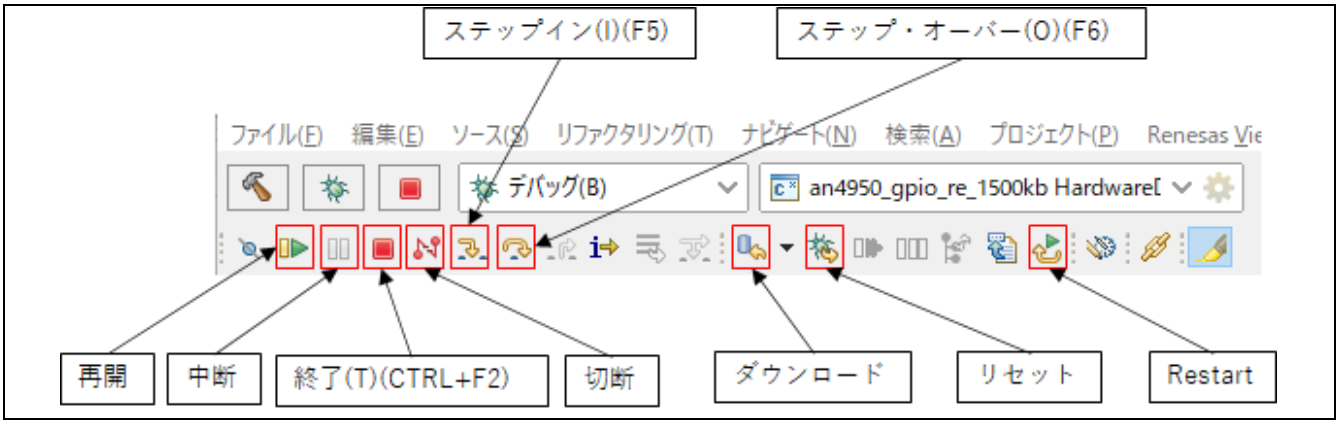


図5-12 [デバッグ] ビューの便利なツールバーのボタン

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、プログラムを再開します。
- 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、新しいデバッグセッションを開始します。
- 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。

注：以下の節の機能を実演するには、「3.4 既存プロジェクトのワークスペースへのインポート」で説明したように、ルネサスのWebサイトからインポートしたRE01用サンプルコードを使用してください。

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。

注：ソフトウェアブレークポイントをフラッシュメモリ領域に設定することは禁止されています。ハードウェアブレークポイントはRAM領域でサポートされていません。

フラッシュメモリ領域にブレークポイントを設定するには、ハードウェアブレークポイントを使用してください。

RAM領域にブレークポイントを設定するには、ソフトウェアブレークポイントを使用してください。

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
   “main.c” を開いて、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）を開くとソフトウェア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、コンテキストメニューの「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 ）。
  再度コンテキストメニューの「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。

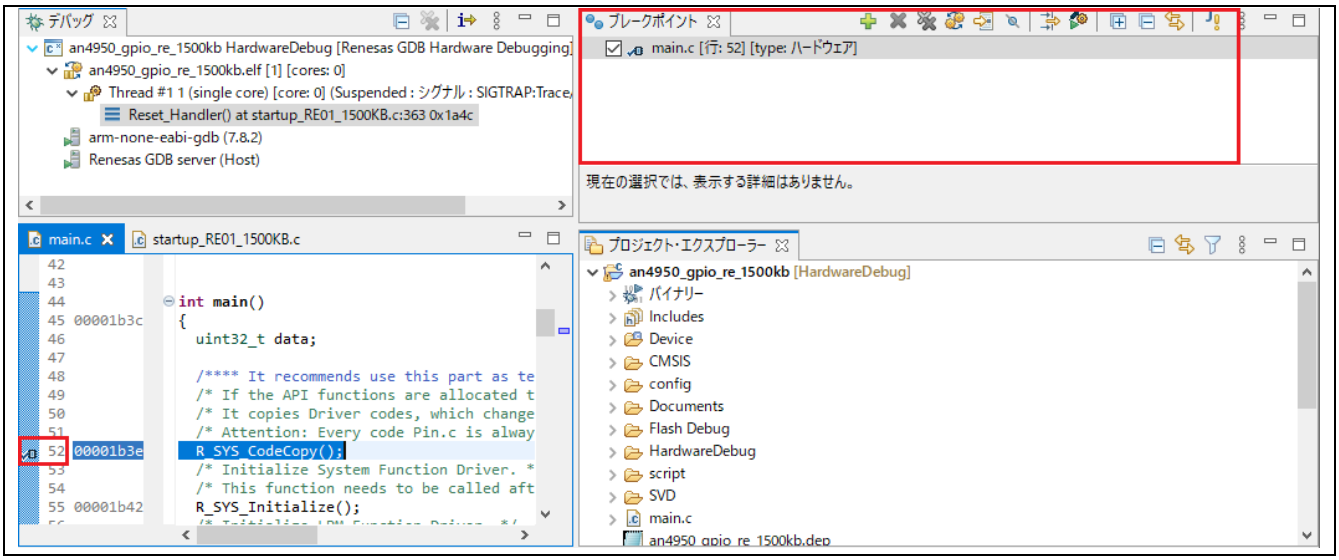


図5-13 [ブレークポイント] ビュー

5.4.2 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モニタできます。これらの変数（スコープ内のローカル変数を含む）に対して、リアルタイム・リフレッシュを行うよう設定できます。

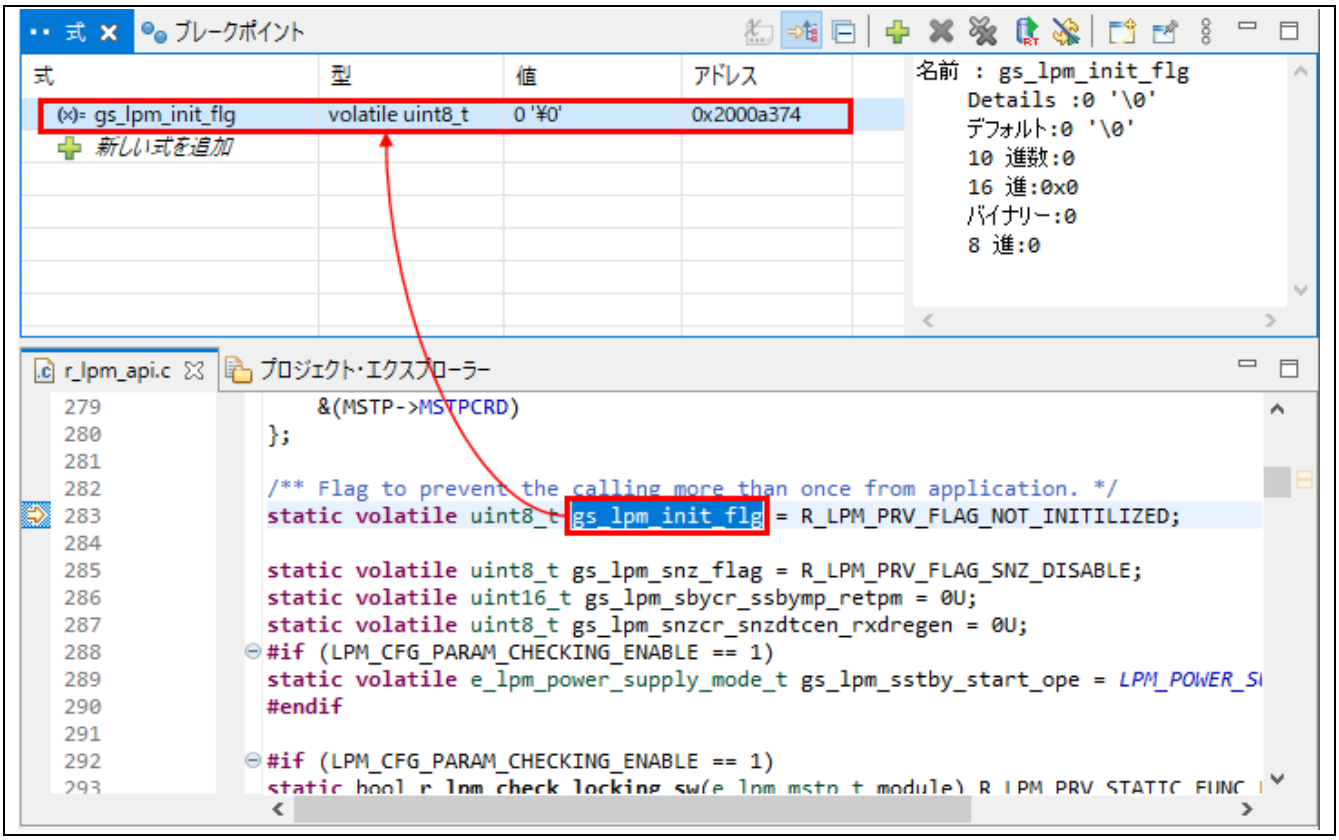


図5-14 [式] ビュー

グローバル変数を見るには以下の手順を行なってください。

1. [ウィンドウ] メニュー →[ビューの表示] →[式] の順に選択、あるいは アイコンをクリックし、[式] ビューを開きます。
2. エディタ上でグローバル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)...] メニュー項目を選択し、[式] ビューに追加します。）
3. [式] ビュー内で右クリックして、“Real-time Refresh” メニュー項目を選択します。これにより、プログラム実行中にリアルタイムで式の値をリフレッシュすることが可能です。文字 “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示します。
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ください。

ローカル変数も同様に追加できますが、プログラムが変数のスコープ外で実行されている場合の監視は不可能です。

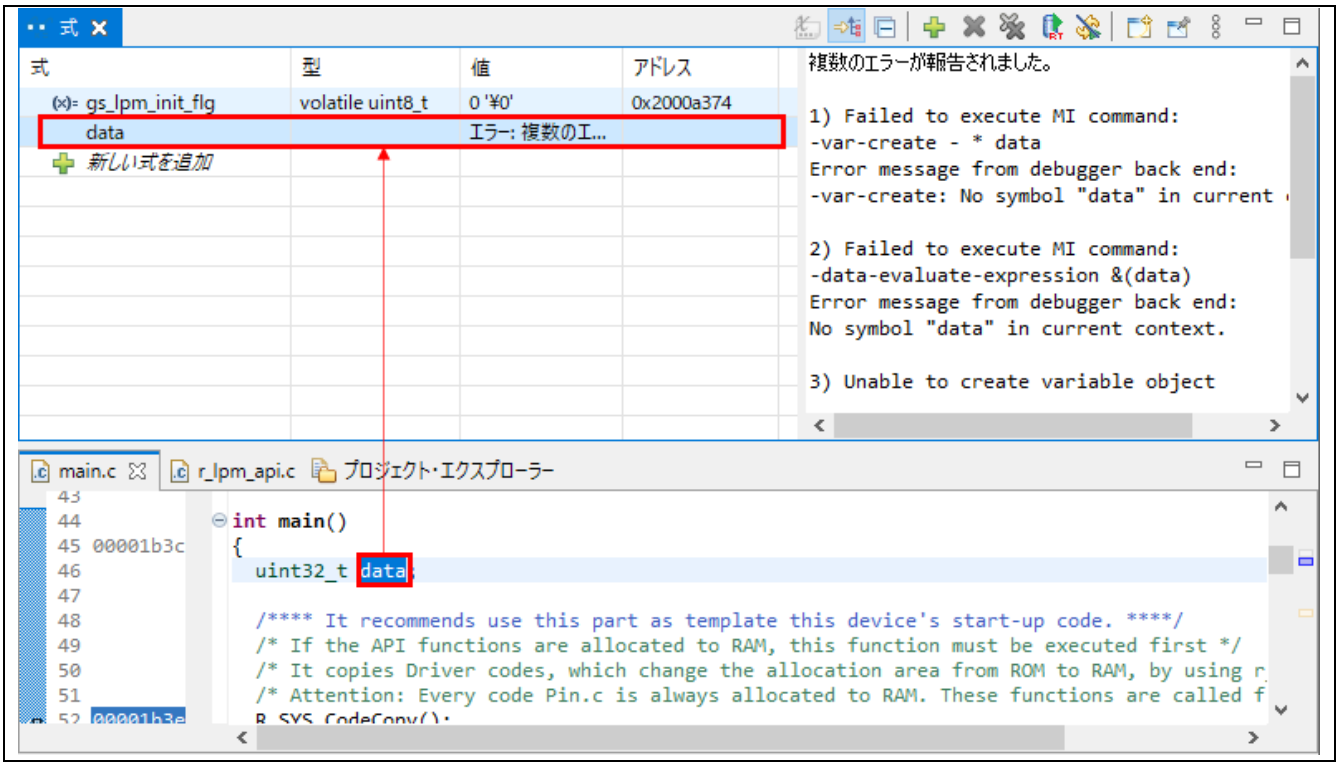


図5-15 ローカル変数を[式] ビューに追加

5.4.3 レジスタービュー

[レジスター] ビューは、ターゲットデバイスの汎用レジスタについての情報を表示します。

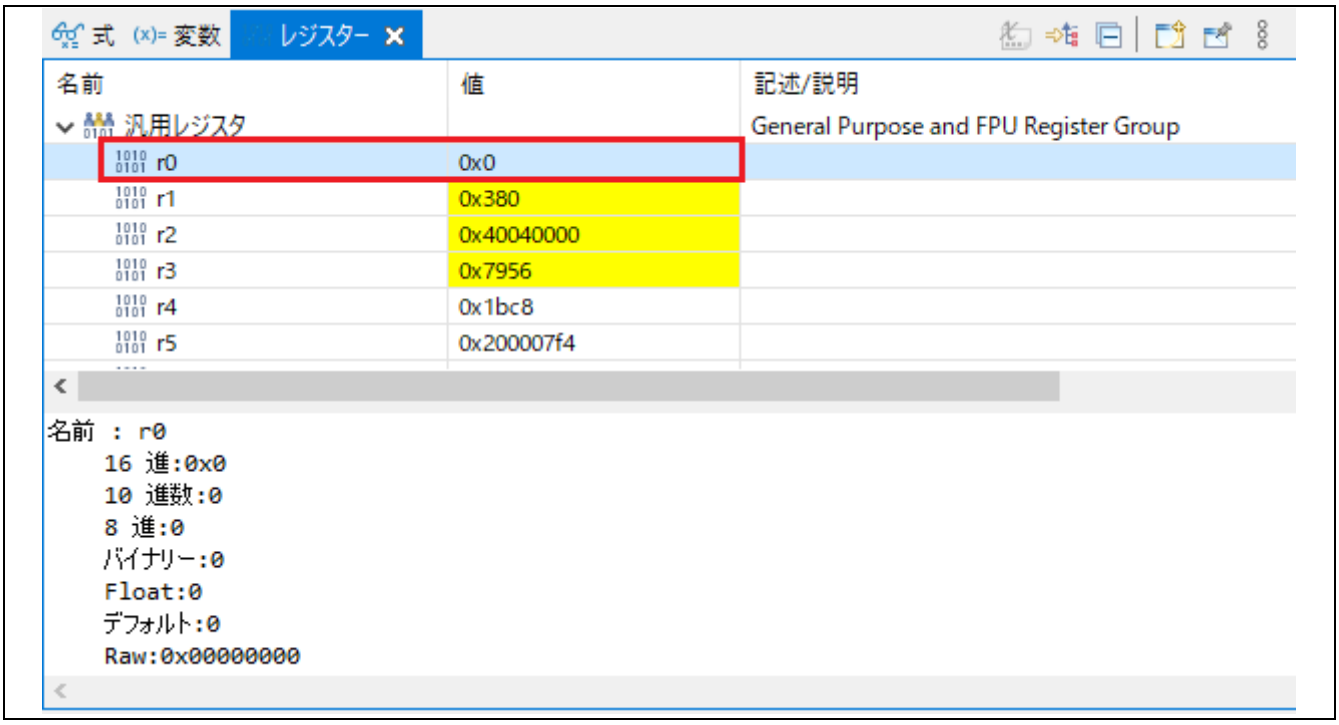


図5-16 [レジスター] ビュー

汎用レジスタ “r0” を見るには以下の手順を行なってください。

1. 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. “r0” 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：“gs_lpm_init_flg”）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] メニュー →[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 アイコンをクリックして [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 “gs_lpm_init_flg” を指定しています。

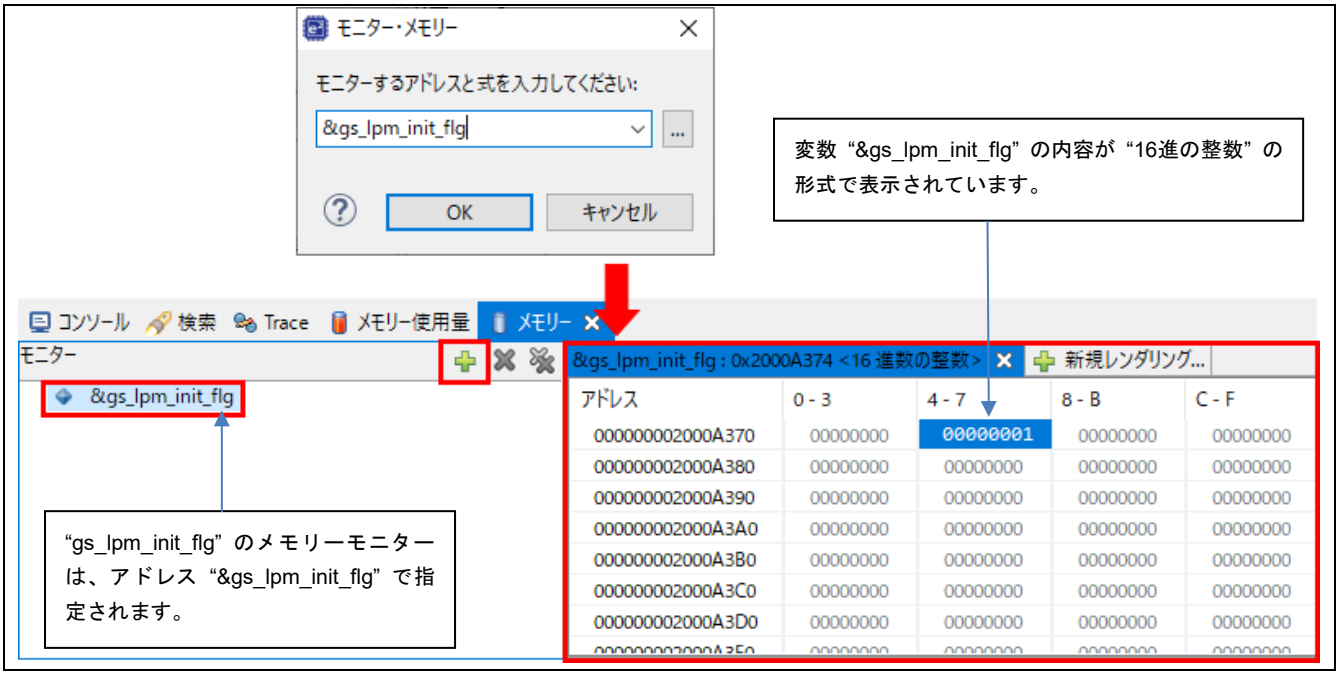


図5-17 [メモリー] ビュー(1/2)

変数“gs_lpm_init_flg”に対する新しいレンダリング（表示形式）（例：Raw Hex）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、[浮動小数点] の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“&gs_lpm_init_flg”を「16進数の整数」と「浮動小数点」の二種類で表示を切り替えられるようにしています。

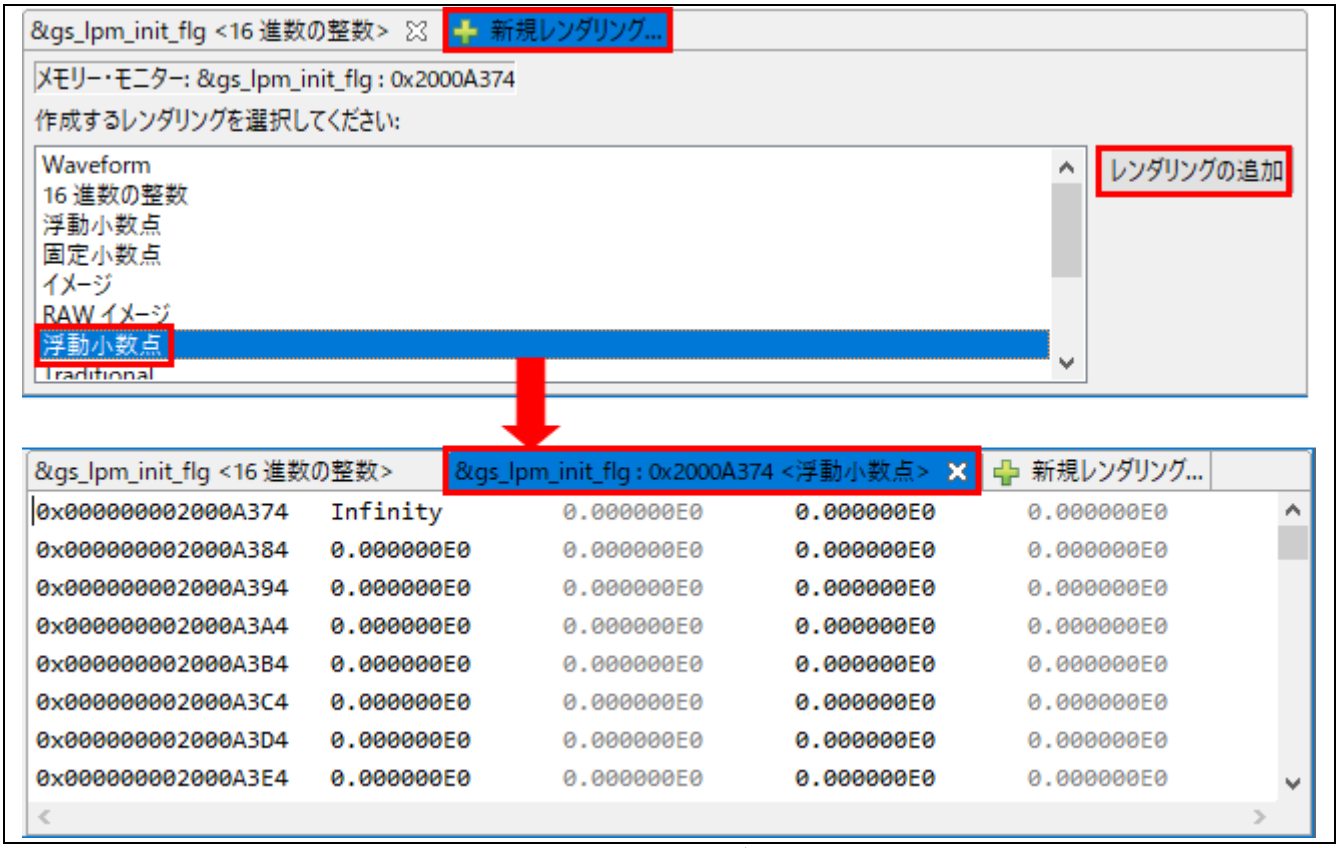


図5-18 [メモリー] ビュー(2/2)

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。

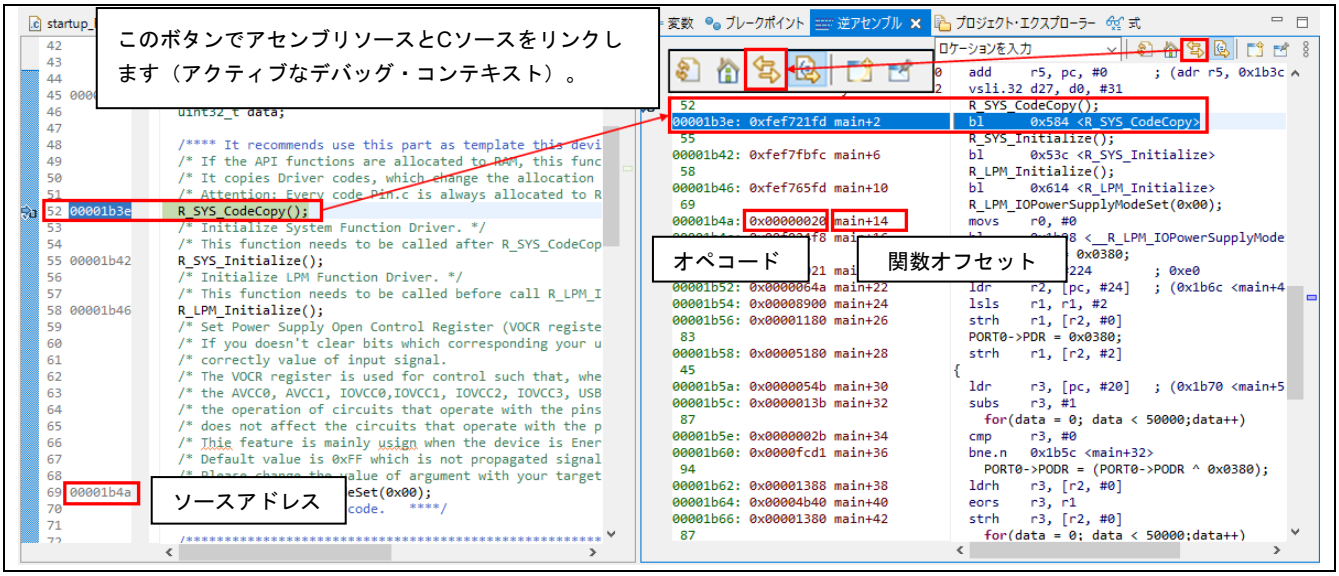


図5-19 [逆アセンブル] ビュー

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] メニュー →[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・コンテキストにリンク）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。

4. コンテキストメニューを用いると、エディタ内でソースアドレスを有効にできます。

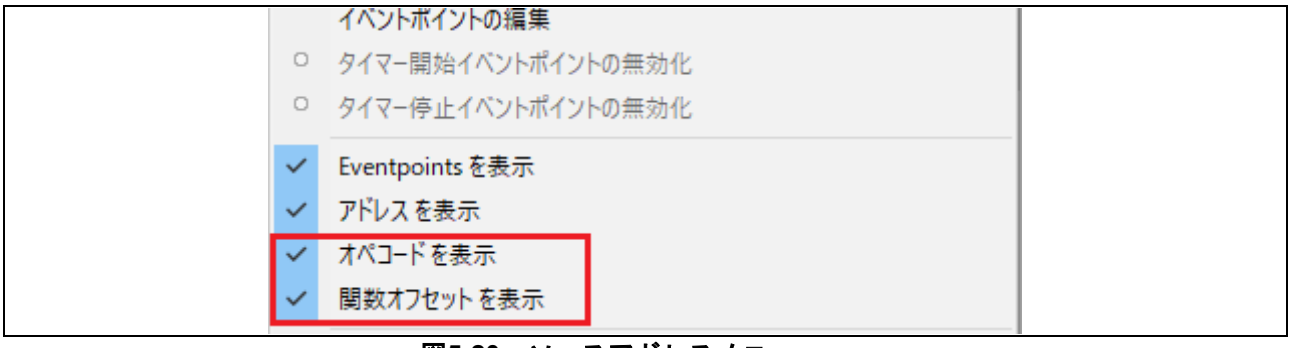


図5-20 ソースアドレスメニュー

5.4.6 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。

現在実行中のスコープ外のグローバル変数や外部変数を見るには、[式] ビューを参照してください。

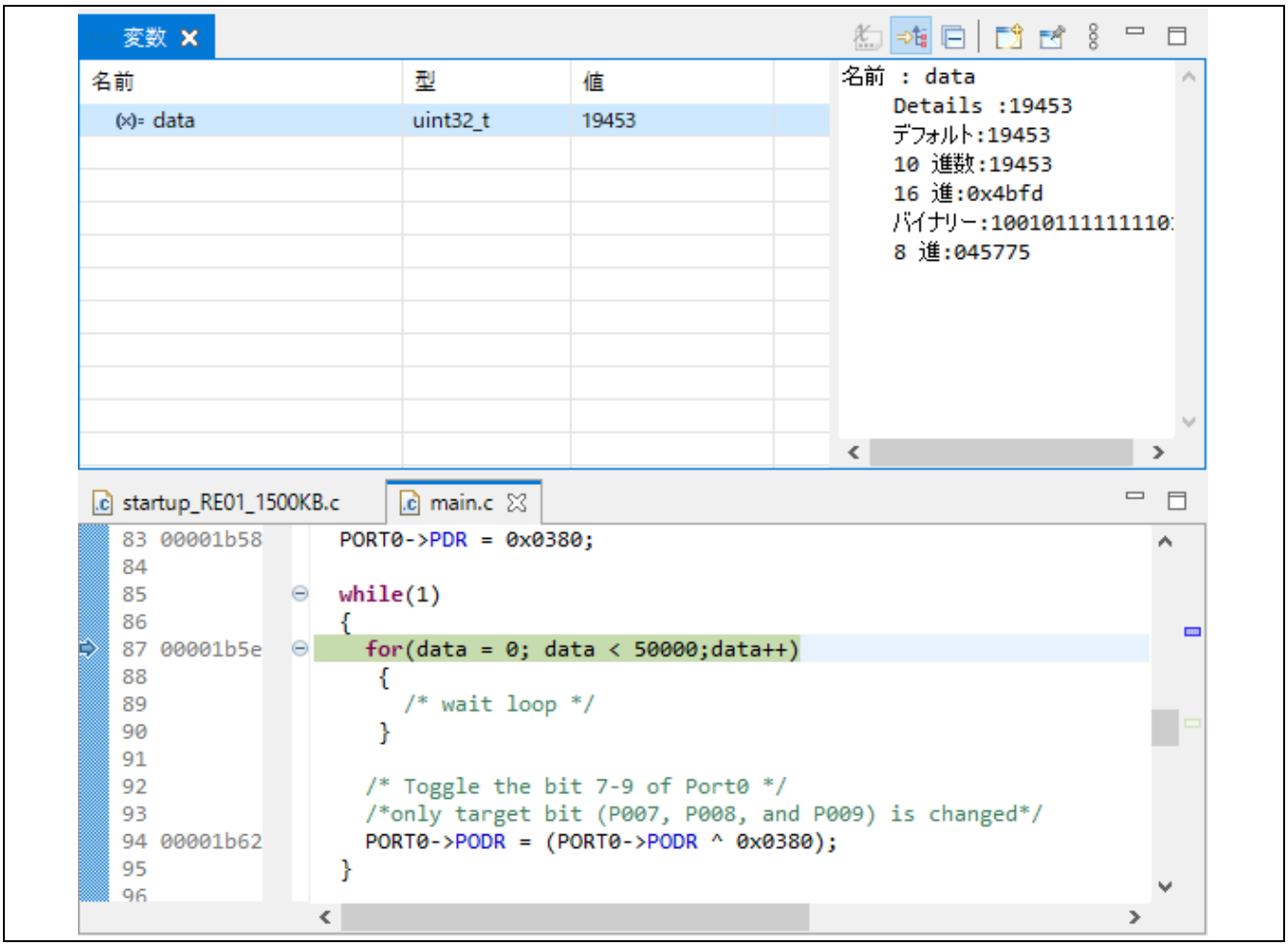


図5-21 [変数] ビュー

ローカル変数を見るには、

1. [ウィンドウ] メニュー →[ビューの表示] →[変数] の順に選択、あるいは (x)= 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内に入ると、変数ビューにローカル変数とその値が表示されます。（上記の例では main ()内の “data”）

注:
最適化された変数や、アキュムレータレジスタに一時的に割り当てられた変数は [変数] ビューに表示されないことがあります。必要に応じて、[逆アセンブル] ビューを参照してください。

多くの場合、最適化を無効にすることで変数は表示されますが、メモリ効率、コードサイズの縮小、性能の向上といった最適化の利点を失うことになります。

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、トレースの記録、イベント・ブレーク、PC前ブレーク、パフォーマンス（タイマ）開始、パフォーマンス（タイマ）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の数や設定条件はMCUによって異なります。2種類のイベントを以下に示します。

- 実行アドレス：エミュレータはCPUによる指定アドレスの命令の実行を検出します。“PC前”ブレーク（イベント条件は指定アドレスの命令の実行の直前に成立する）あるいは他のイベント（イベント条件は指定アドレスの命令の実行の直後に成立する）になります。
- 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（OR、AND（およびその組み合わせ）、シーケンシャル）は2つ以上のイベントに使用できます。

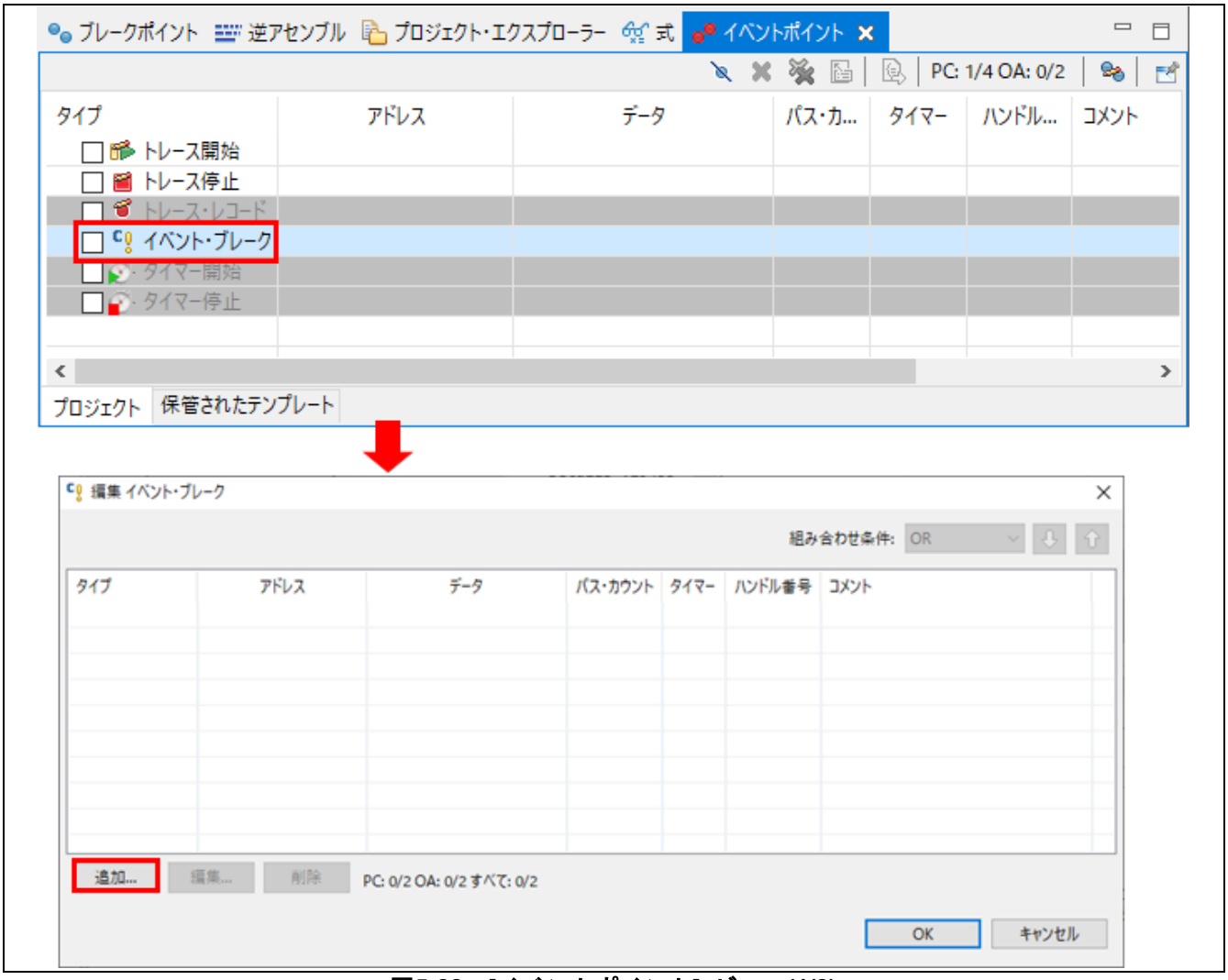


図5-22 [イベントポイント] ビュー(1/2)

アドレスまたはデータが一致する条件（例：gs_lpm_init_flg への書き込み時）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. [ウィンドウ] メニュー →[ビューの表示] →[イベントポイント] の順に選択、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. 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次に進みます。

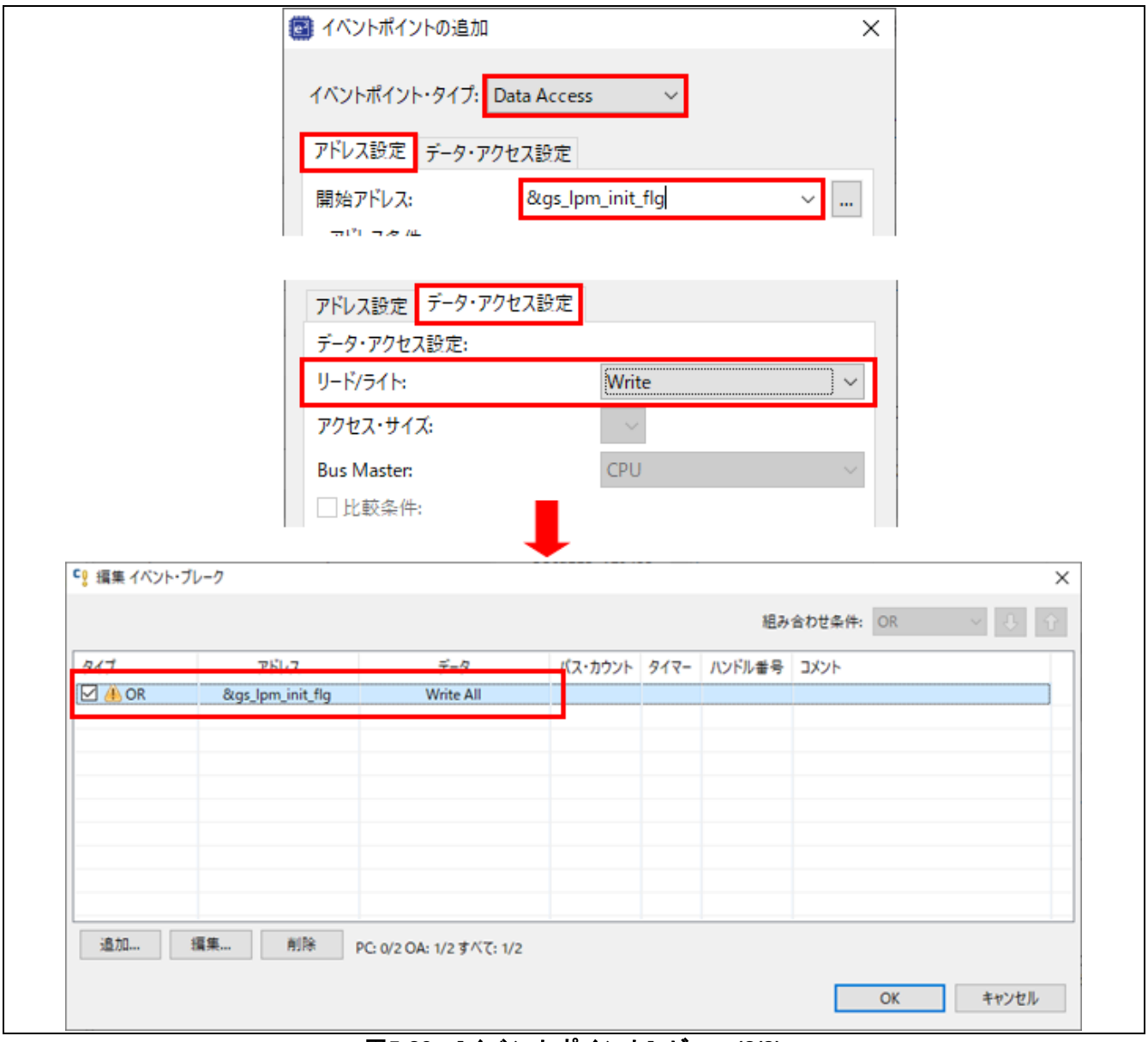


図5-23 [イベントポイント] ビュー(2/2)

4. [イベントポイント・タイプ] 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 “gs_lpm_init_flg” を検索します。（グローバル変数のアドレスは “&gs_lpm_init_flg” 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、[リード/ライト] に “Write” を選択します。[OK] ボタンで次に進みます。

7. “gs_lpm_init_flg Write All” のイベント・ブレークが設定され、[イベントポイント] ビューで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にはリセットを行います。

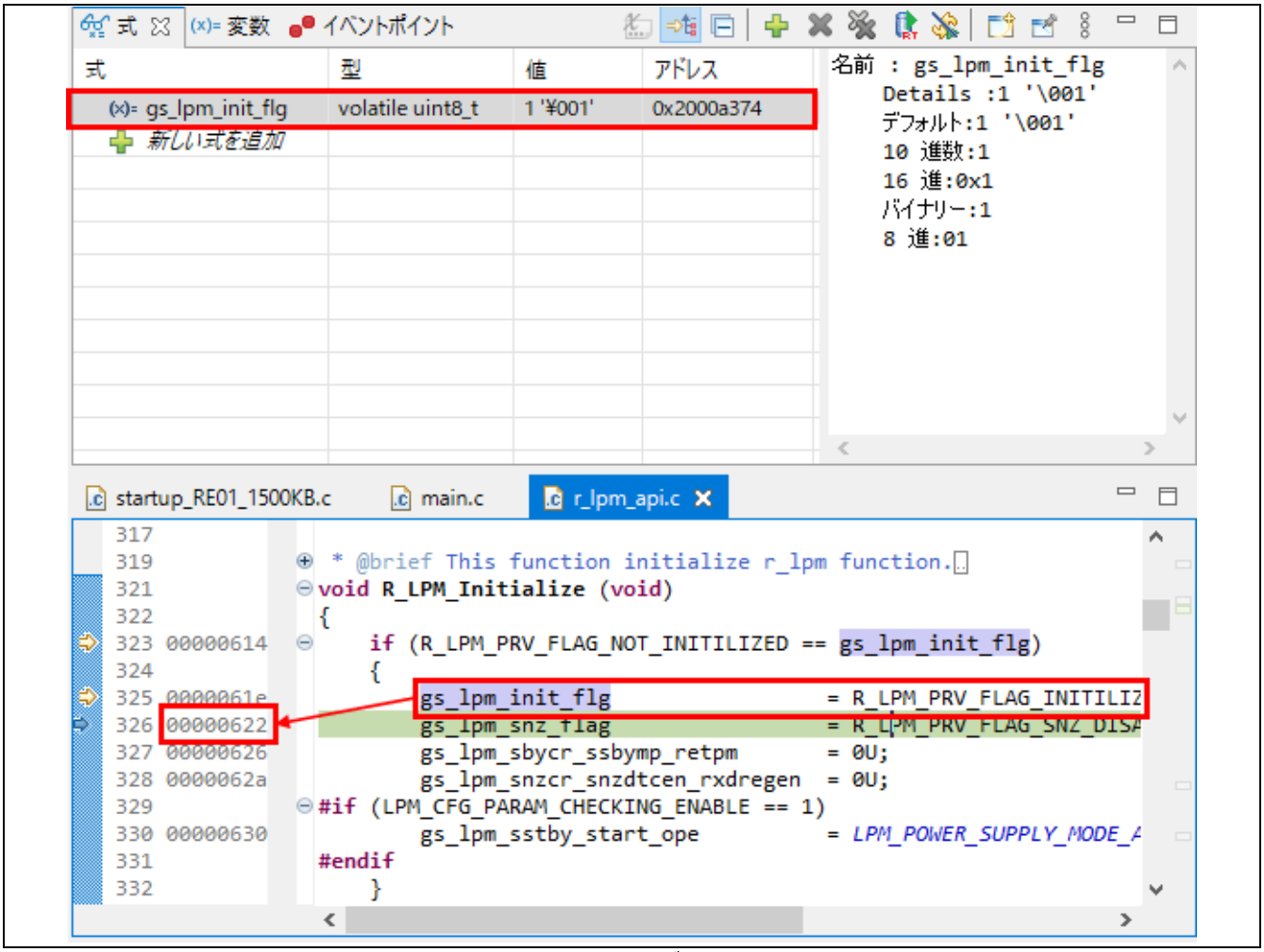


図5-24 イベント・ブレークの実行

図5-24は、gs_lpm_init_flg に R_LPM_PRV_FLAG_INITILIZED の値が割り当てられている場合、プログラムがコードの326行目（gs_lpm_init_flgへの書き込みの行の直後）で停止することを示します。

5.4.8 IOレジスタ（IO Registers）ビュー

[IO Registers] 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を、それぞれのアドレス、16進数と2進数の値を含めて表示します。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。

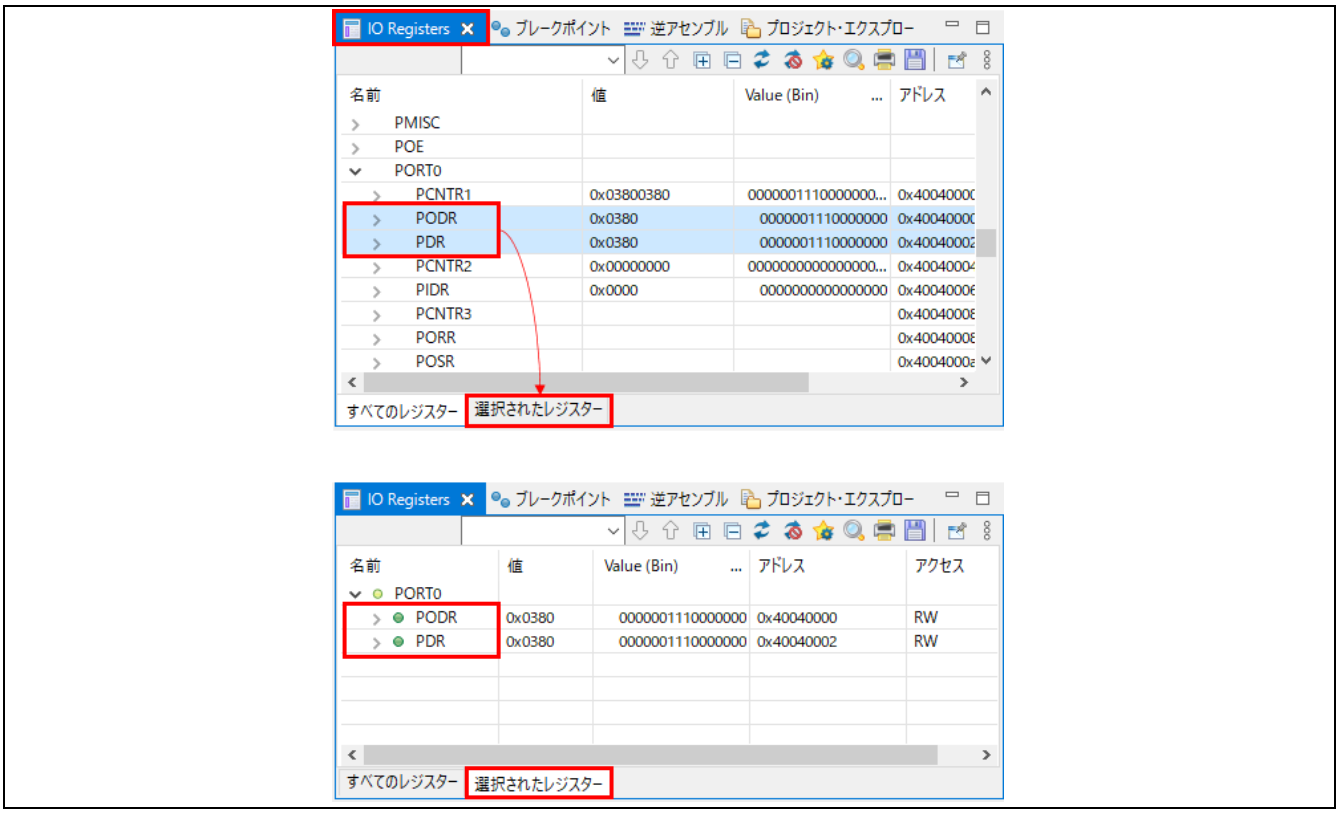


図5-25 [IO Registers] ビュー

選択したIOレジスタ（例：PORT0のPODR、PDR）の値を見るには、

1. [ウィンドウ] メニュー →[ビューの表示] →[その他...] の順に選択します。[ビューの表示] ダイアログで[デバッグ] →[IO Registers]をクリックするか、あるいは アイコンをクリックし、[IO Registers] ビューを開きます。

2. [すべてのレジスター] タブで、[PORT0]を探し、PORT0のIOレジスター覧を展開してください。[IO Registers] ビューのツールバーの ボタンで名前による迅速な検索も可能です。

3. PODRとPDR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のIOレジスタであるPODRとPDR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] タブでの表示には時間がかかるため、複数のレジスタを見るには [選択されたレジスター] を使うことをお勧めします。

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、問題が発生したプログラムの箇所を探すことができます。

トレースバッファは有限なため、バッファがいっぱいになると、一番古いトレースデータを新しいデータで上書きします。

プログラム実行が停止するまでのトレース取得を設定するには、

1. [Renesas Views] → [デバッグ] → [Trace] の順に選択、あるいは アイコンをクリックし、[Trace] ビューを開きます。

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。

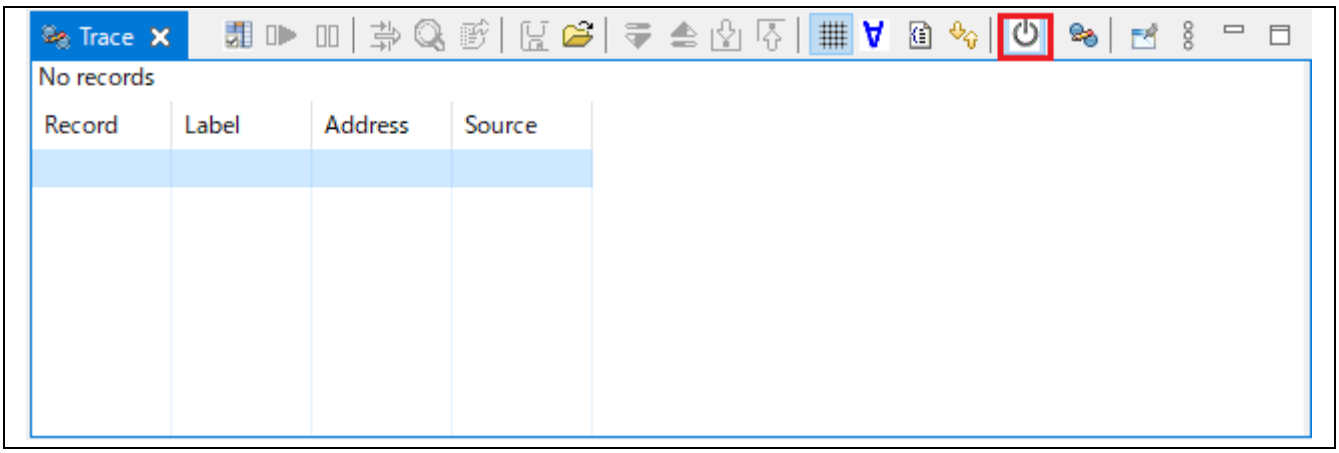


図5-26 トレース有効

3. プログラムを実行し、ブレークポイントで停止させるか、あるいは [デバッグ] ツールバーの [中断] ボタンで停止させます。それまでにトレースメモリに格納された情報がトレース結果として表示されます。

4. [トレース] ビュー上のボタンで表示モードを選択します。

下図は、main() 関数実行前までのトレース結果を示しています。

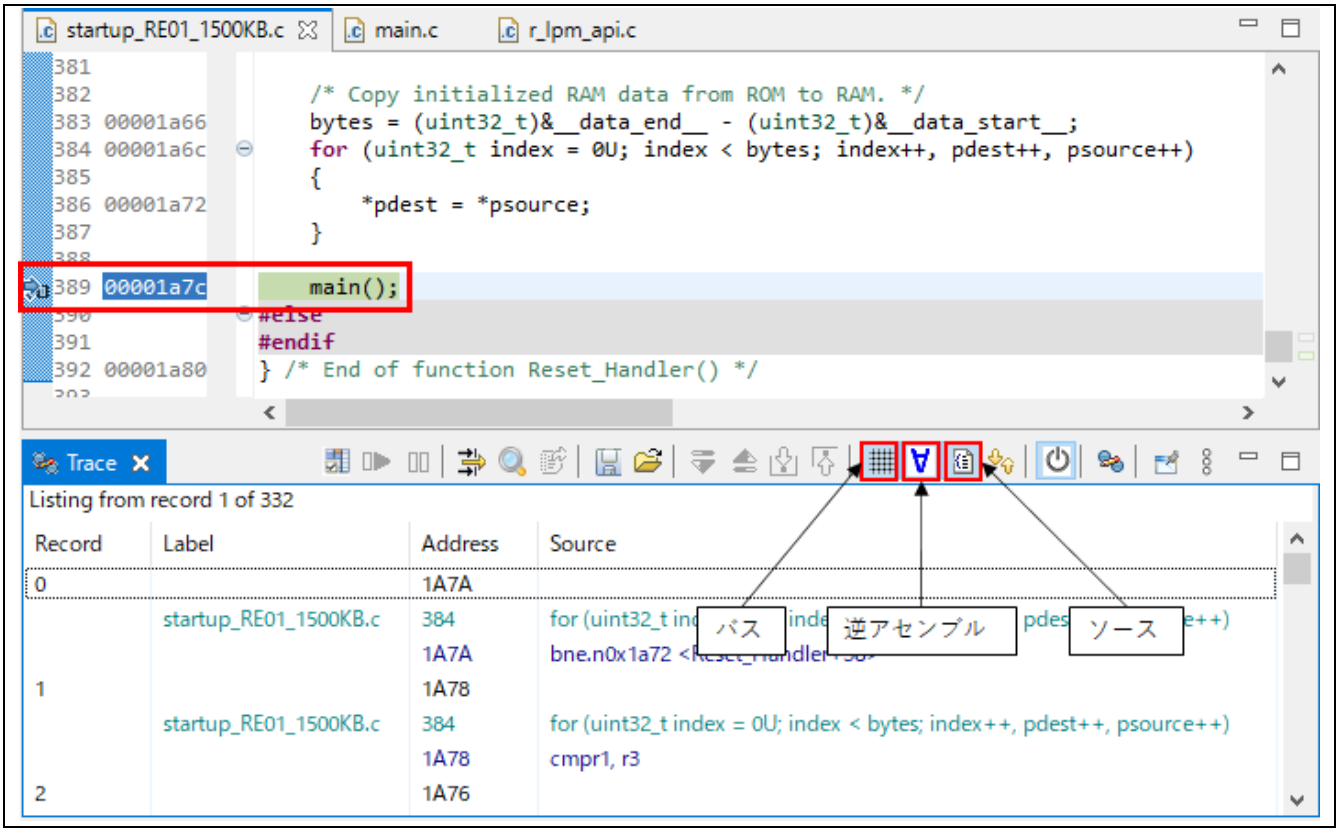


図5-27 [トレース] ビューの表示モードの選択

5. トレース結果は、デフォルト設定では古い順から表示されます。表示順は ボタンで変更できます。

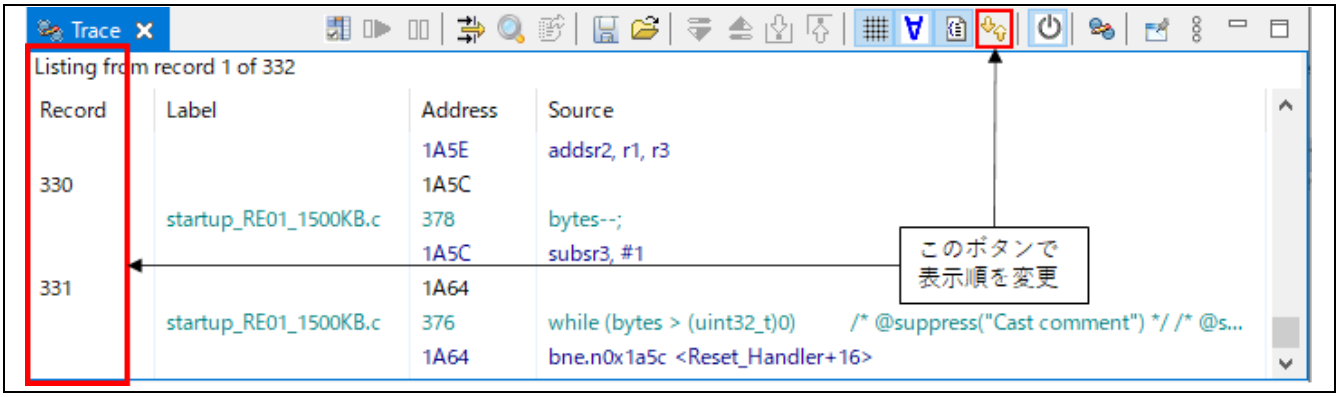


図5-28 表示順の変更

6. トレース結果は ボタンでフィルタすることができます。フィルタ条件として、[Record] と [Address] とが選択できます。

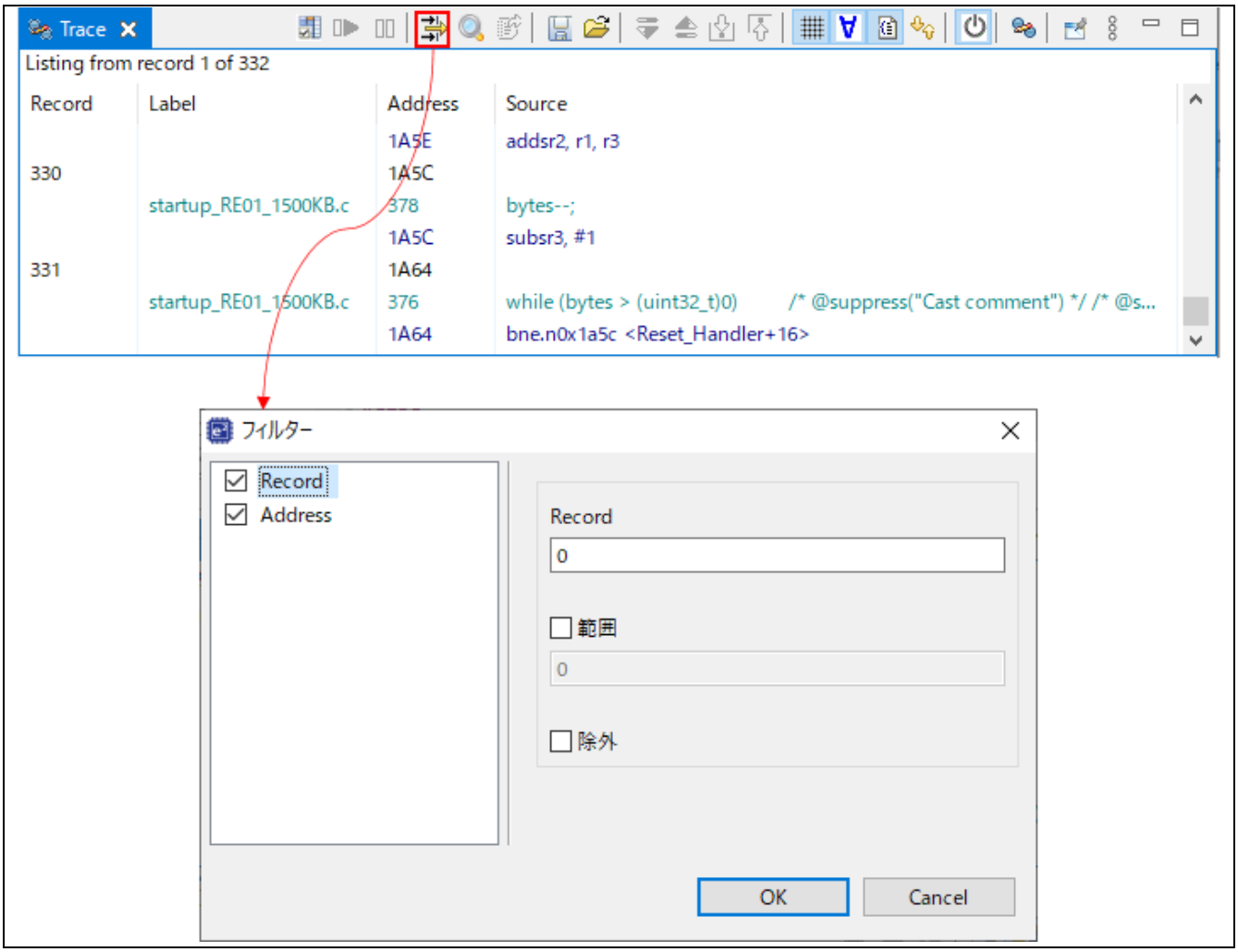


図5-29 トレース結果のフィルタ

7. トレース結果は .csvファイルに保管することができます（バス、アセンブリ、ソース情報を含む）。また、トレース結果を .csvファイルから [トレース] ビューにロードすることも可能です。

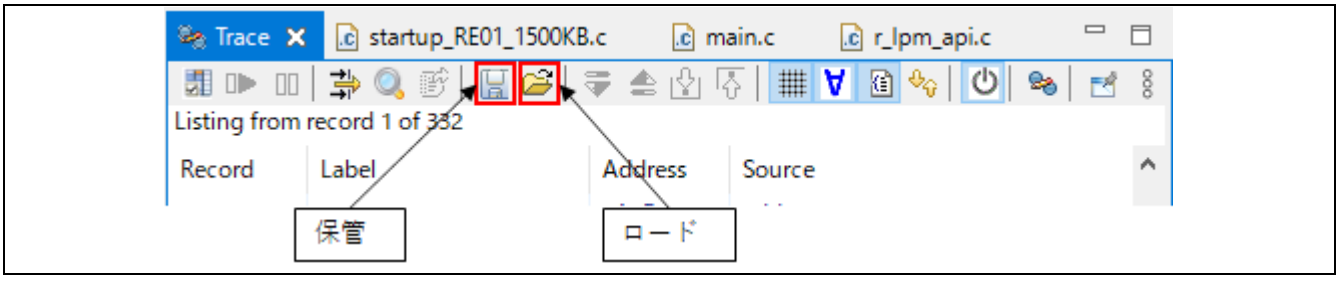


図5-30 トレース結果の保管とロード

5.4.10 メモリー使用量ビュー

[メモリー使用量] ビューを使用して、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を見ることができます。

プロジェクトのメモリー使用量を見るには、

1. [ウィンドウ] メニュー →[ビューの表示] →[その他...] →[デバッグ] →[メモリー使用量] の順に選択して、[メモリー使用量] ビューを開きます。

2. 実行プロジェクト（GNU ARM Embedded Toolchainを使用）の [メモリー使用量] ビューのデフォルト表示は3つのエリアで構成されています：
(1) サイズ、(2) メモリ領域使用量／アドレス空間使用量、(3) 詳細

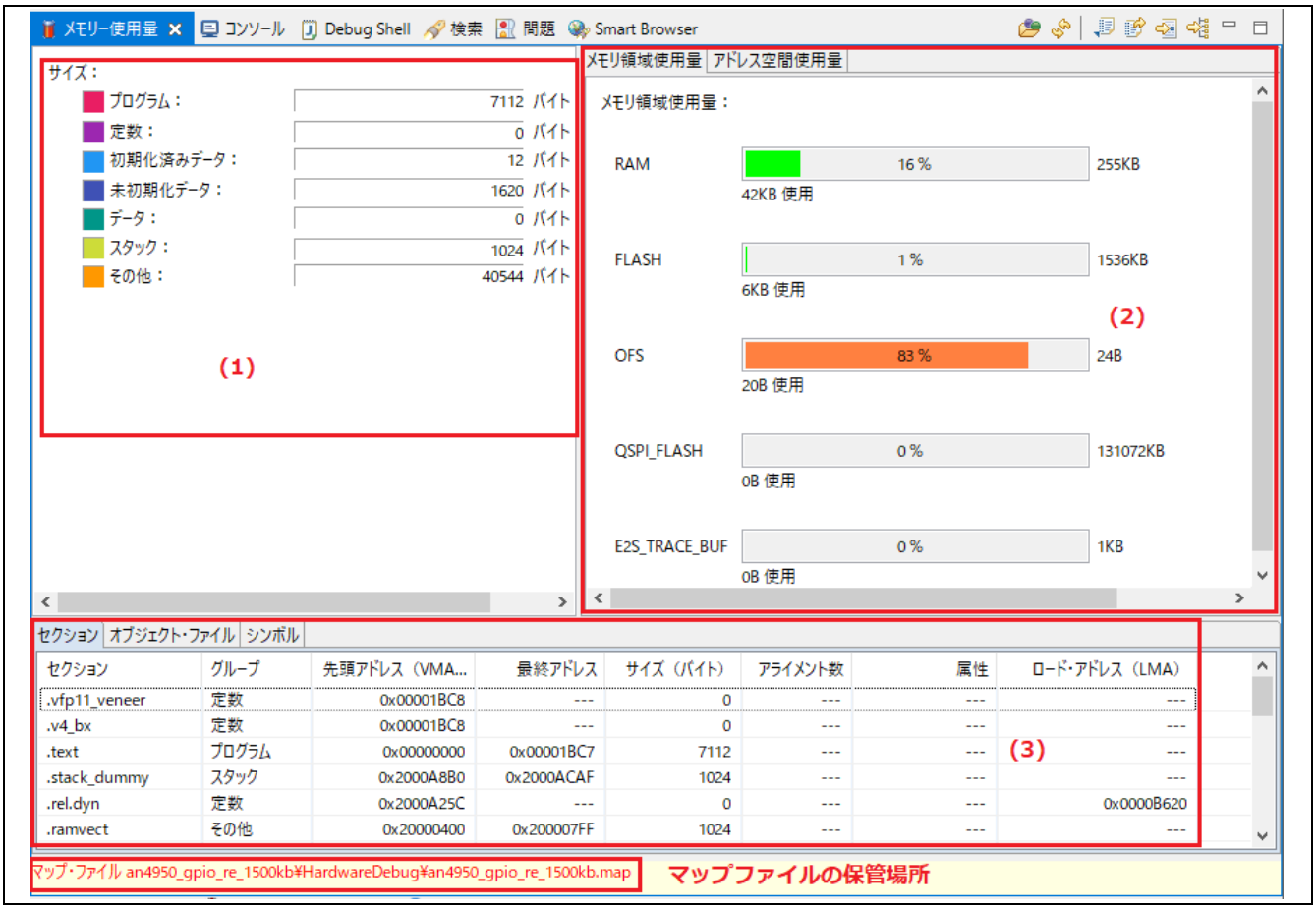


図5-31 [メモリー使用量] ビュー

(1) [サイズ] エリア（実行プロジェクト用）

表示されるサイズは、選択したマップファイルのプログラムの合計、定数、初期化済みデータ、未初期化データ、データ、スタック、その他です。

注：このビューは、サポートするツールチェーンの実行プロジェクトについてのみ表示します。

(2) [メモリ領域使用量]／[アドレス空間使用量] エリア

[メモリ領域使用量] エリアは、RAM、ROM、フラッシュメモリの使用量の割合を数値とバーのステータスで表示します。バーの色はパーセント値に基づいています。

- 75%未満：緑
- 75%以上90%未満：オレンジ
- 90%以上：赤

[アドレス空間使用量] エリアは選択したプロジェクトのデバイスのメモリを表示します。表示される情報は、各メモリ領域の名前、開始アドレス、終了アドレス、使用量、容量です。

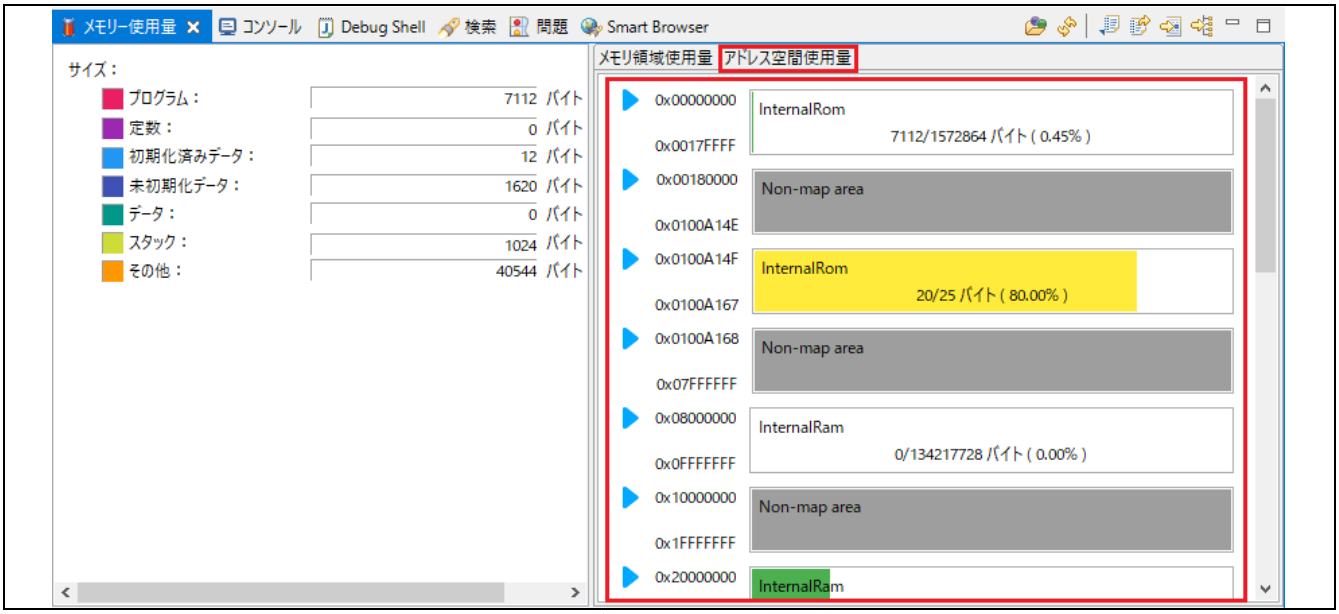


図5-32 [アドレス空間使用量] エリア

メモリ領域のセクションを見るには、メモリ領域を拡張してください。セクションの色は [サイズ] エリアの色と対応しています。

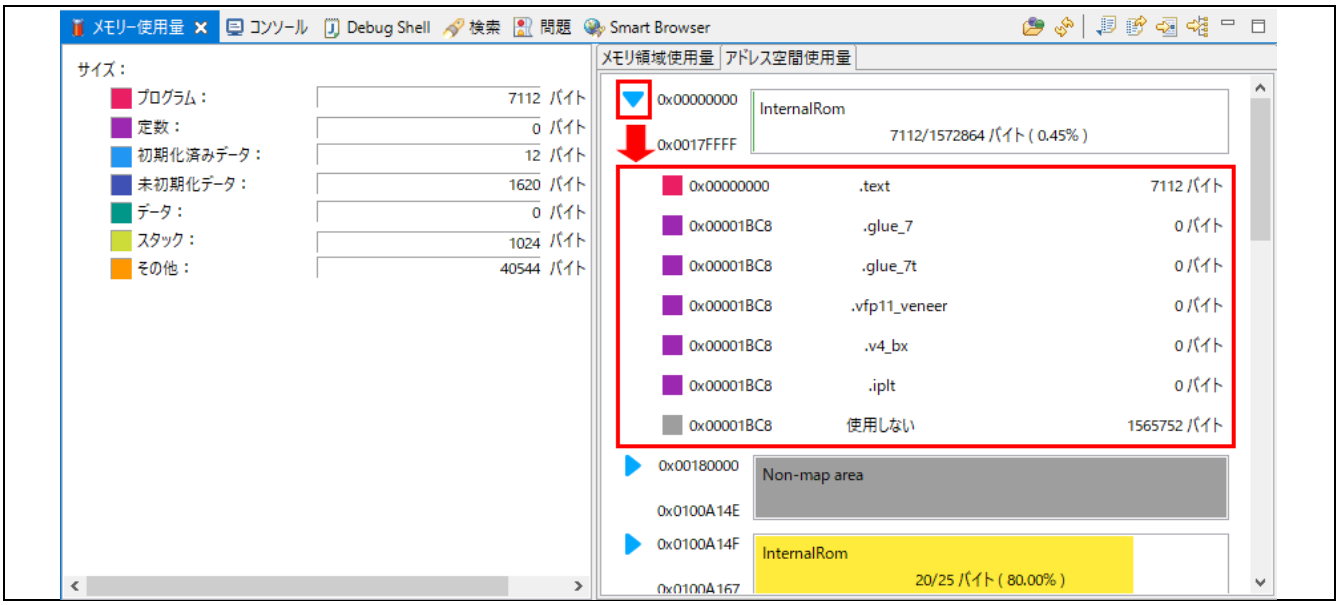


図5-33 メモリ領域の拡張

(3) 詳細エリア

アクティブなプロジェクトや開いているマップファイルのマップファイル情報を表示します。

- [セクション] タブ : “マップファイル” 表を含みます。“マップファイル” 表は、マップファイルとその詳細な情報から分析されたセクション一覧を表示します。
- [オブジェクト] タブ : “オブジェクト” 表を含みます。“オブジェクト” 表は、マップファイルとその詳細な情報から分析されたオブジェクト一覧を表示します。
- [シンボル] タブ : “シンボル” 表を含みます。“シンボル” 表は、マップファイルとその詳細な情報から分析されたシンボル一覧を表示します。
- [クロスリファレンス] タブ : マップファイルから取得したクロスリファレンス情報を表示します。このタブは実行プロジェクトの場合のみ存在します。

マップファイルの場所

[メモリー使用量] ビューに、プロジェクトのマップファイル（*.map）やライブラリリストファイル（*.lbp）の情報が表示されます。選択したマップファイルやライブラリリストファイルの相対パスを [メモリー使用量] ビューの一番下で確認することができます。

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。

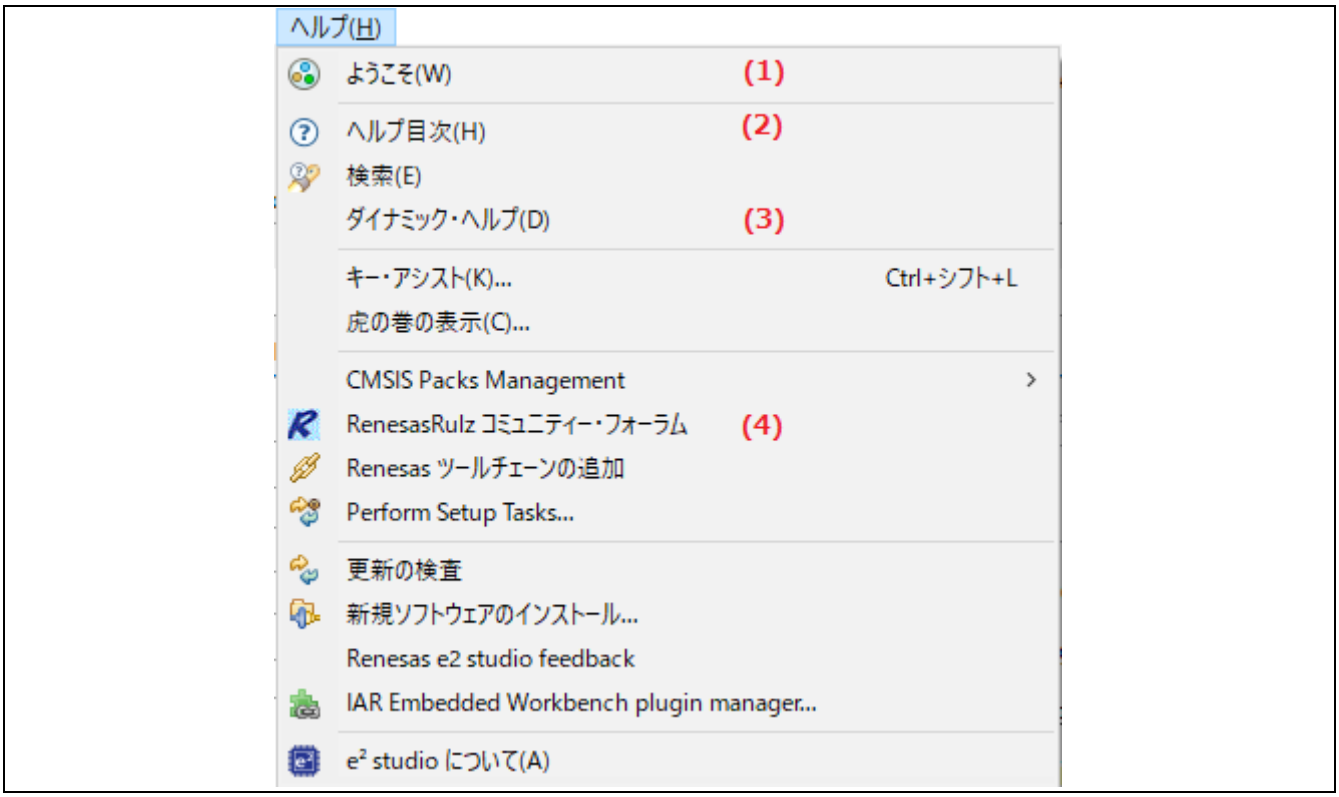


図6-1 ヘルプメニュー

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、IDEのチュートリアルやサンプルコードにアクセスするためのリンク、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、ワークベンチ内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	Renesas e^2 studio 2021-07以上 ユーザーズマニュアル：クイックスタートガイド

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2021.09.08	ー	初版発行

Renesas
e2 studio 2021-07以上 ユーザーズマニュアル
クイックスタートガイド

発行年月日　　2021年09月08日　　Rev.1.00

発行　　ルネサス エレクトロニクス株式会社

Renesas e² studio 2021-07以上
ユーザーズマニュアル：クイックスタートガイド

Renesas Electronics Corporation

R20UT5034JJ0100