

e² studio v7.3

統合開発環境（IDE）

ユーザーズマニュアル 入門ガイド
(参考資料)

Renesas Synergy™ プラットフォーム

本資料は英語版を翻訳した参考資料です。内容に相違がある場合には英語版を優先します。資料によっては英語版のバージョンが更新され、内容が変わっている場合があります。日本語版は参考用としてご使用のうえ、最新および正式な内容については英語版のドキュメントをご参照ください。

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）、交通管制（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。

Renesas Synergy™ プラットフォーム

e² studio 入門ガイド

目次

1. 概要 (Overview)	3
1.1 システム構成 (System Configuration)	4
1.2 動作環境 (Operating Environment)	5
1.2.1 システム要件 (System Requirements)	5
1.2.2 サポート対象ツールチェーン (Supported Toolchains)	5
1.2.3 サポート対象エミュレータデバイス (Supported Emulator Device)	5
1.3 Synergy プロジェクト開発の概要 (Outline of a Synergy Project Development)	5
2. インストール (Installation)	5
2.1 プラットフォームインストーラでのインストール (Installing the Platform Installer)	5
2.2 スタンドアロンインストーラでのインストール (Installing the Standalone Installer)	10
2.2.1 e ² studio のインストール (Installing e ² studio)	10
2.2.2 GNUARM コンパイラのセットアップ (Setting Up the GNUARM Compiler)	15
2.2.3 Renesas Synergy™ ソフトウェアパッケージ (SSP) のインストール (Installing the Renesas Synergy™ Software Package (SSP))	15
2.3 e ² studio のアンインストール (Uninstalling e ² studio)	18
2.4 e ² studio の更新 (Updating e ² studio)	18
2.5 Synergy ライセンスの登録 (Register Synergy license)	19
3. プロジェクトの作成 (Project Generation)	20
3.1 新しい Synergy プロジェクトの作成 (Generating a New Synergy Project)	21
3.2 既存のプロジェクトのインポート (Importing an Existing Project)	26
3.3 Synergy 静的ライブラリの生成と使用 (Generating and Using a Synergy Static Library)	29
3.3.1 静的ライブラリプロジェクトの作成 (Creating the Static Library Project)	29
3.3.2 実行可能プロジェクト内での静的ライブラリプロジェクトの使用 (Using the Static Library in the Executable Project)	33
3.4 Synergy Project Configuration Editor	39
3.4.1 Summary (概要) ページ	40
3.4.2 [BSP] ページ	41
3.4.3 [Clocks Configuration] (クロック設定) ページ	41
3.4.4 [Pin Configuration] (端子設定) ページ	43
3.4.5 [Threads Configuration] (スレッド設定) ページ	47
3.4.6 [Messaging] (メッセージング) ページ	54
3.4.7 [Components Configuration] (コンポーネント設定) ページ	55
3.5 エディタのホバー (Editor hover)	56
3.6 [Developer Assistance] (開発者支援機能)	57

4. ビルド (Building)	60
4.1 ビルド構成 (Build Configurations)	60
4.2 サンプルプロジェクトのビルド (Building a Sample Project)	61
4.3 ビルド設定レポートの保存 (Saving the Build Settings Report)	61
5. デバッグ (Debugging)	63
5.1 既存のデバッグ構成の変更 (Changing an Existing Debug Configuration)	64
5.2 新しいデバッグ構成の作成 (Creating a New Debug Configuration)	67
5.3 基本的なデバッグ機能 (Basic Debugging Features)	68
5.3.1 [Breakpoints] (ブレークポイント) ビュー	69
5.3.2 [Expressions] (式) ビュー	71
5.3.3 [Registers] (レジスタ) ビュー	72
5.3.4 [Memory] (メモリ) ビュー	73
5.3.5 Memory Usage (メモリ使用状況) ビュー	75
5.3.6 [Disassembly] (逆アセンブル) ビュー	77
5.3.7 [Variables] (変数) ビュー	78
5.3.8 [IO Registers] (IO レジスタ) ビュー	79
5.3.9 [Eventpoints] (イベントポイント) ビュー	80
5.3.10 [Trace] (トレース) ビュー	83
5.3.11 [Fault Status] (障害ステータス) ビュー	86
5.3.12 実行ブレークタイマ (Run Break Timer)	87
6. ThreadX アプリケーションの設定 (Setting up a ThreadX Application)	88
6.1 ThreadX 内の General Purpose Timer の例 (General Purpose Timer Example in ThreadX)	88
6.2 サンプルプロジェクトの作成 (Creating the Sample Project)	89
7. [Help] (ヘルプ)	95
改訂記録	97

1. 概要 (Overview)

Renesas e² studio は、Renesas Synergy™ マイクロコントローラ向けの統合開発環境 (Integrated Development Environment: IDE) です。e² studio は、ビルドフェーズ (エディタ、コンパイラ、リンカコントロールなど) および拡張 GDB (GNU Debug) インタフェースをサポートするデバッグフェーズに対応した業界標準のオープンソース Eclipse IDE フレームワークと C/C++ 開発ツール (CDT) プロジェクトに基づいて開発されています。

e² studio IDE は、Renesas Synergy™ プロジェクトに対応するフレームワーク、ハードウェア抽象化レイヤ (HAL) ドライバ、ボードサポートパッケージ (BSP) ドライバを含む、Renesas Synergy™ ソフトウェアパッケージ (SSP) に対応しています。SSP は、e² studio で Renesas Synergy™ アプリケーションを開発するための総合的なドライバライブラリを提供しています。

e² studio IDE は、複数のグラフィカルユーザインタフェース (GUI) ウィザードを採用しており、コードの自動生成、既存のドライバのインクルードと構成、ビルドオプションとデバッグオプションの構成、開発したアプリケーションの実行に対応できます。ドライバのドキュメントは、ツールヒントの形で統合されており、コードエディタビューで利用できます。

Renesas Synergy™ のサポートは、Renesas e² studio の release 4.1 およびそれ以降のバージョンに含まれています。特に Renesas Synergy ARM® Cortex®-M ベースのマイクロコントローラ (MCU) とオープンソース GNU ARM ツールチェーンをサポートするために、複数のビューとエディタを利用できます。

Renesas Synergy™ 固有のアドオンには、操作しやすいウィザードがあり、ハードウェアの構成や、Renesas Synergy™ の大規模なソフトウェアライブラリの管理に活用できます。

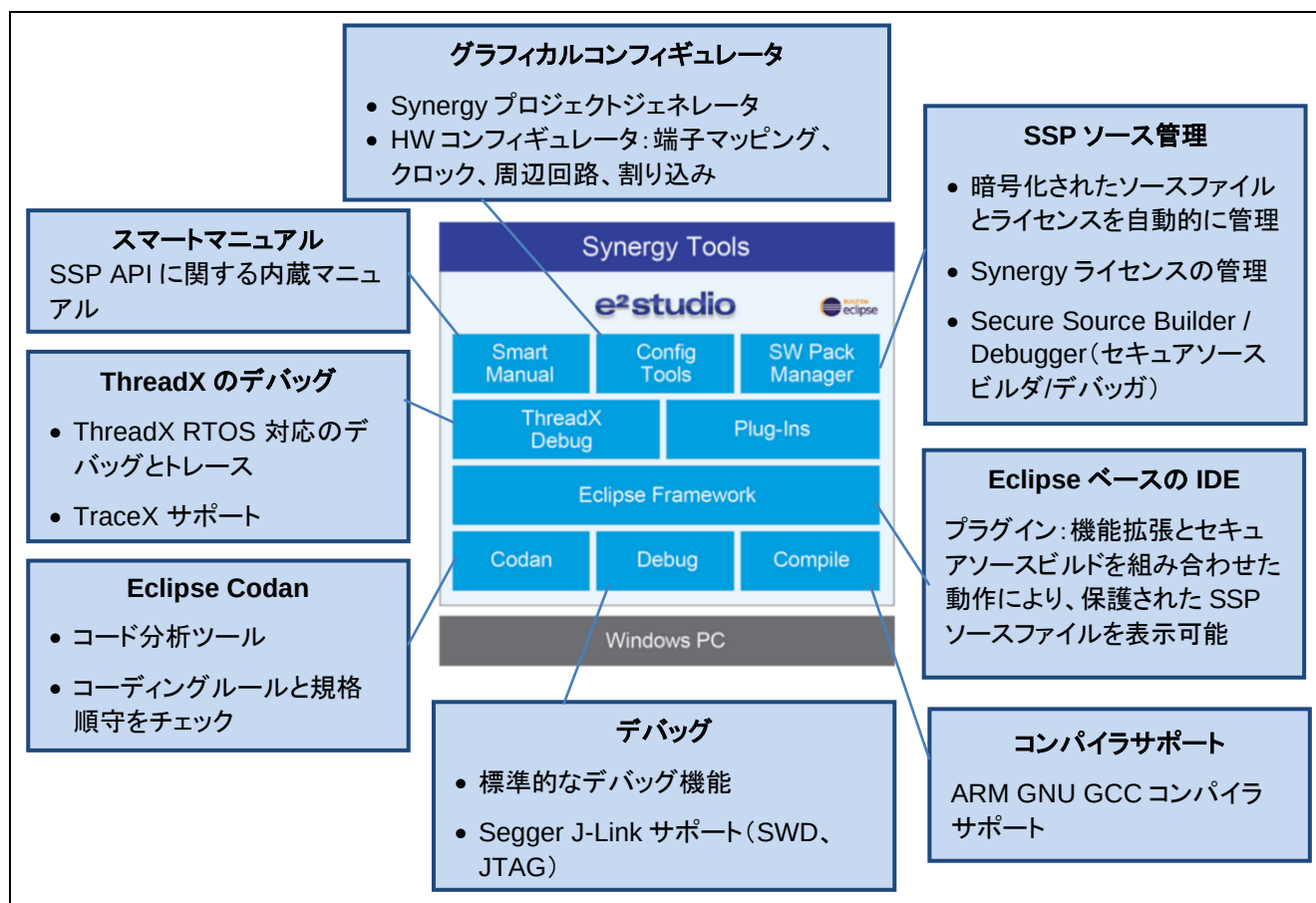


図 1. Renesas Synergy™ を e² studio 環境で開発

e² studio でサポートされるほとんどの機能は、Renesas 製品ファミリに共通しています。Synergy 製品固有の機能は、GCC Arm Embedded Toolchain のサポートと、Renesas Synergy™ Project Generator、および Renesas Synergy™ Configuration Editor です。

表 1. e² studio の機能比較

機能	Renesas Synergy™	RX / RL78 / RZ / RH850
IDE フレームワークと C/C++ のサポート	Eclipse + CDT	Eclipse + CDT
コード生成ツール	Synergy プロジェクトジェネレータ Synergy Configuration Editor	コード生成ツール/スマートコンフィギュレータ
Toolchain	GCC Arm Embedded	RX ファミリ (GNURX-ELF、Renesas CC-RX ビルドプラグイン) RL ファミリ (GNURL78-ELF、Renesas CCRL ビルドプラグイン) RZ ファミリ (GNUARM-NONE-EABI) RH850 ファミリ GHS (*デバッグのみサポート)
HEW / CS+ プロジェクトインポート	サポート対象外	サポート対象は、HEW/CS+ IDE をサポートする MCU
対象デバッガ	SEGGER J-Link®	E1、E2、E2 lite、E20、IECUBE、E10A-USB、Segger J-Link
スマートマニュアルツールヒント	サポート対象 (SSP API)	サポート対象
コード分析 (CODAN)	サポート対象	サポート対象
シミュレータ	サポート対象外	選択された RX および RL ファミリのデバイスをサポート
デバッガ	トレース機能とリアルタイムメモリアクセス機能を搭載した GDB	トレース機能とリアルタイムメモリアクセス機能を搭載した GDB
RTOS	Express Logic ThreadX®	各種オペレーティングシステム
ThreadX の構成	サポート対象	サポート対象外
ThreadX のデバッグ	サポート対象	サポート対象外
メモリ使用状況ビュー	サポート対象	サポート対象
Visual Expressions ビュー	サポート対象	サポート対象

1.1 システム構成 (System Configuration)

以下の通り、標準的なシステム構成は、ホストコンピュータとターゲットボードで形成されています。

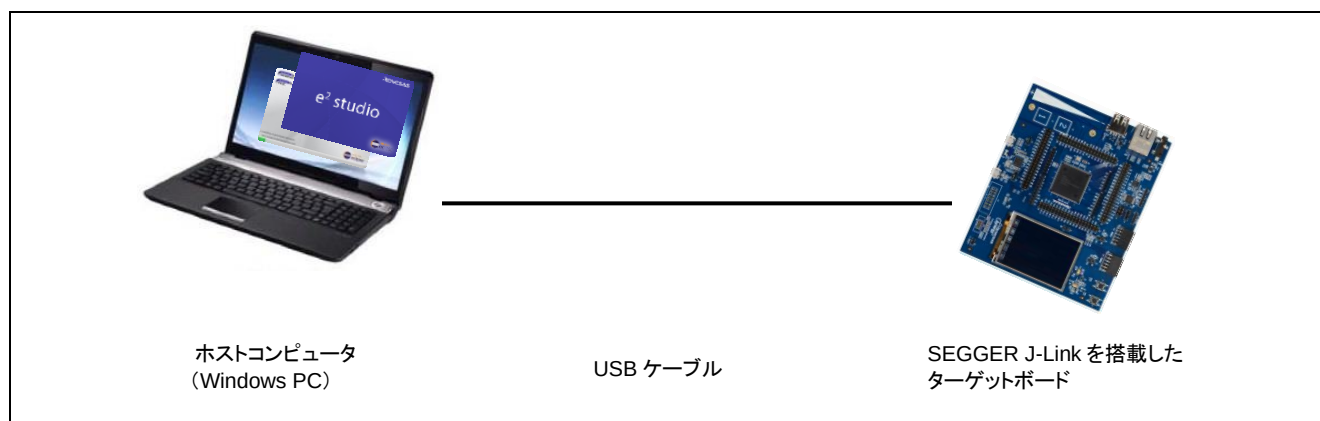


図 2. システム構成

1.2 動作環境 (Operating Environment)

以下に、本製品のシステム要件を示します。

1.2.1 システム要件 (System Requirements)

ホストコンピュータ	- プロセッサ：2GHz 以上 (Intel® Core™ ファミリプロセッサを搭載) - メモリ容量：1GB 以上 (8GB 以上を推奨) - ハードディスク容量：250GB 以上 - ディスプレイ：解像度 1,024 x 768 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。
オペレーティングシステム	ホストコンピュータでは以下のオペレーティングシステムがサポートされています。 - Windows 7 (32/64 ビット OS) - Windows 8.1 (32/64 ビット OS) - Windows 10 (32/64 ビット OS)

1.2.2 サポート対象ツールチェーン (Supported Toolchains)

GNU Arm®コンパイラ (バージョン：GCC_4.9_2015q3、および GCC_7.2_2017q4)

1.2.3 サポート対象エミュレータデバイス (Supported Emulator Device)

Segger J-Link

1.3 Synergy プロジェクト開発の概要 (Outline of a Synergy Project Development)

本資料では、Renesas Synergy™プラットフォームを使用して開発を開始する詳細な方法を説明します。主な手順の概要を以下に示します。これらを理解すると、第 3 章と第 4 章で説明する手順がわかりやすくなります。

1. Synergy プロジェクトの作成。
2. クロック、ICU、ピンの機能などのハードウェア仕様に合わせて Synergy プロジェクトを構成。
3. ThreadX OS の構成。
4. BSP の構成 (HAL ドライバモデルの選択)。
5. ユーザコードの追加。
6. プロジェクトのビルド。
7. デバッガの構成とデバッガの起動。

2. インストール (Installation)

開発ツールは、Platform Installer または Standalone Installer を使用してインストールします。

最新のインストーラパッケージは、Synergy Platform Web サイト

(<https://www.renesas.com/products/synergy.html>) のソリューションギャラリーからダウンロードできます。

2.1 プラットフォームインストーラでのインストール (Installing the Platform Installer)

プラットフォームインストーラは、以下のコンポーネントで構成されています。

- Synergy ソフトウェアパッケージ (SSP)
- e² studio または IAR Embedded Workshop® for Renesas Synergy™ (IAR EW for Synergy) IDE
- GCC Arm embedded コンパイラ、および
- SEGGER J-Link ドライバ

プラットフォームインストーラをダウンロードしてインストールするには、以下の手順を使用します。

1. [Renesas Synergy™ プラットフォーム](#) Web サイトの[ソリューションギャラリー]ページにアクセスします。
[ソフトウェア] -> [Synergy Software Package] (Synergy ソフトウェアパッケージ) を選択するか、[ソフ

トウェアツール] -> [e² studio] を選択した後、[ダウンロード]を選択します (My Renesas アカウントへのログインが必須です)。



図 3. インストール - プラットフォームインストーラのダウンロード

2. 開発環境として、e² studio を選択します。

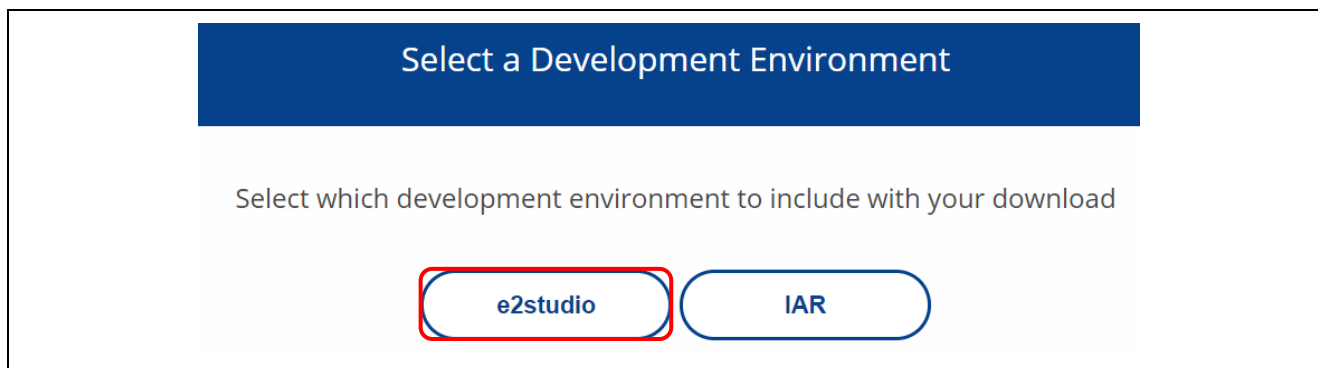


図 4. インストール - e² studio 開発環境の選択

3. [License Agreement] (ライセンス契約) の [I Accept] (同意する) をクリックします。インストールファイル (例えば、setup_ssp<version>_e2s_<version>.zip) がダウンロードされます。

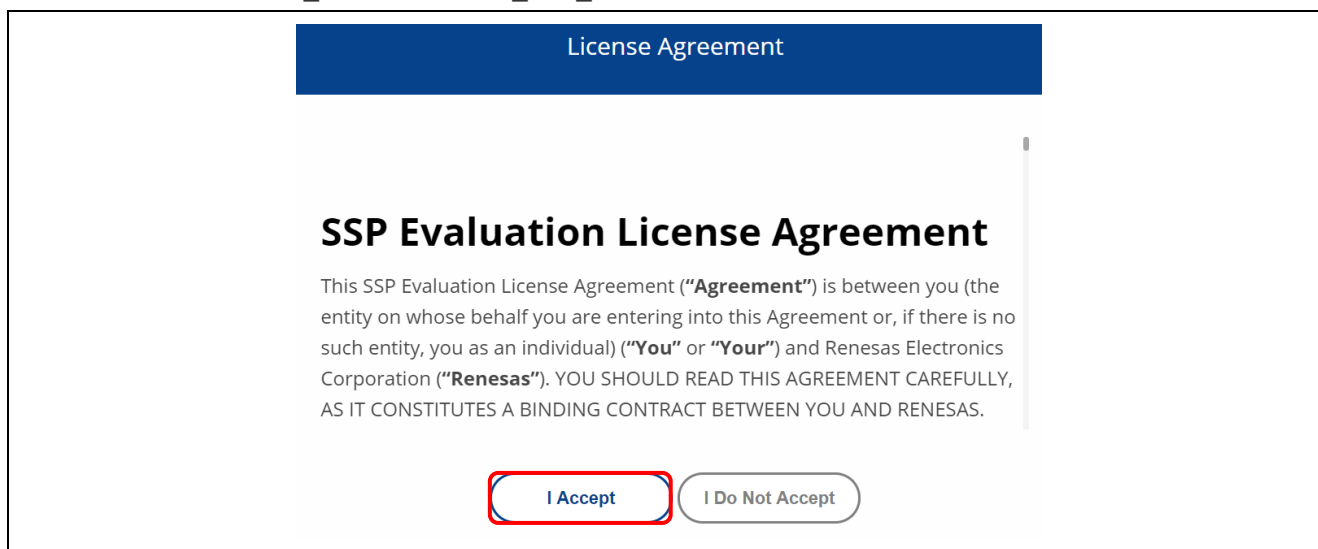


図 5. インストール - [License Agreement] (ライセンス契約) の同意

4. インストールファイルを解凍し、実行します。

5. **[Select Install Type]** (インストールの種類の選択) ページで、**[Quick Install]** (クイックインストール) を選択し、**[Next]** (次へ) をクリックします。インストールするコンポーネントをカスタマイズするには、**[Custom Install]** (カスタムインストール) を選択した後、**[Next]** (次へ) をクリックします。

新規ユーザは、最少の手順で設定するために、**[Quick Install]** (クイックインストール) オプションを推奨します。このオプションを選択すると、e² studio、SSP、および GCC Arm Embedded がデフォルトでインストールされます。ユーザが **[Quick Install]** (クイックインストール) を選択した場合、ステップ 7 は表示されません。

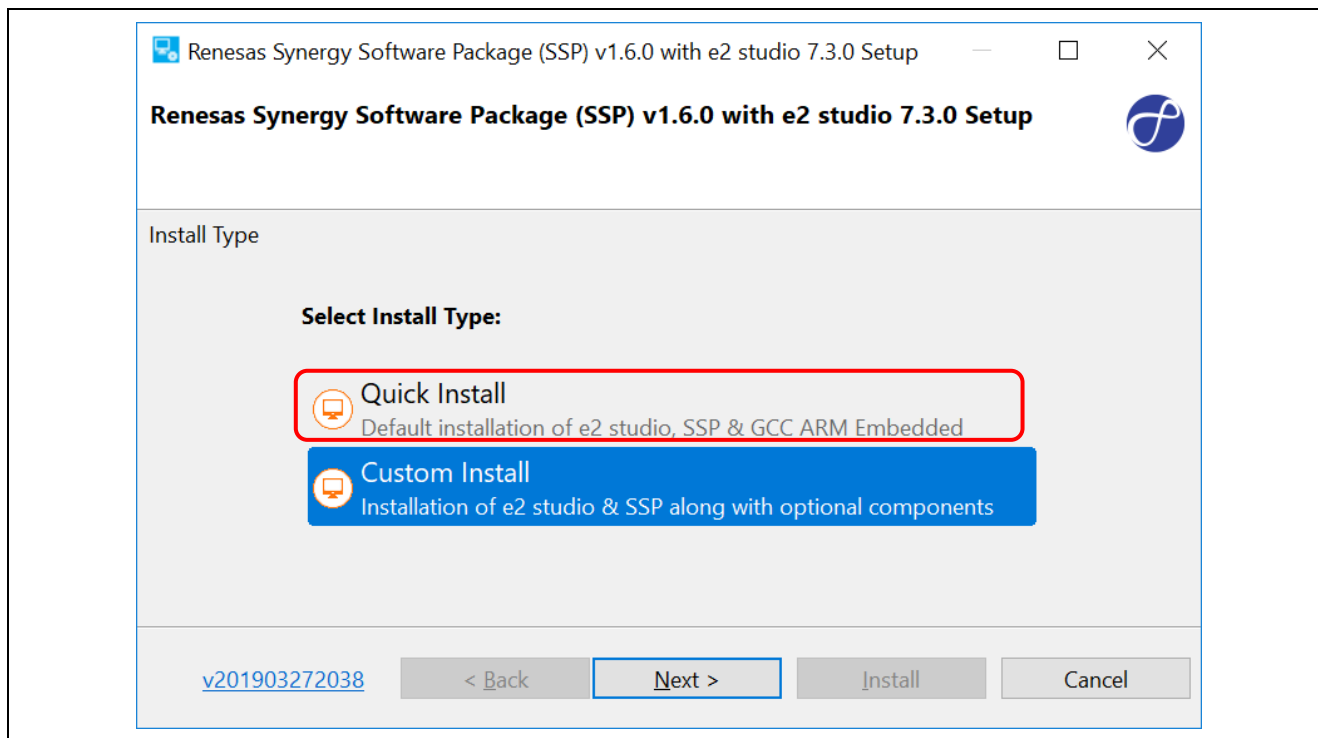


図 6. インストール - インストールの種類の選択

6. **[Welcome]** (ようこそ) ページで、デフォルトフォルダを使用します。**[Change...]** (変更...) をクリックしてフォルダを変更することもできます。**[Next]** (次へ) をクリックして続けます。

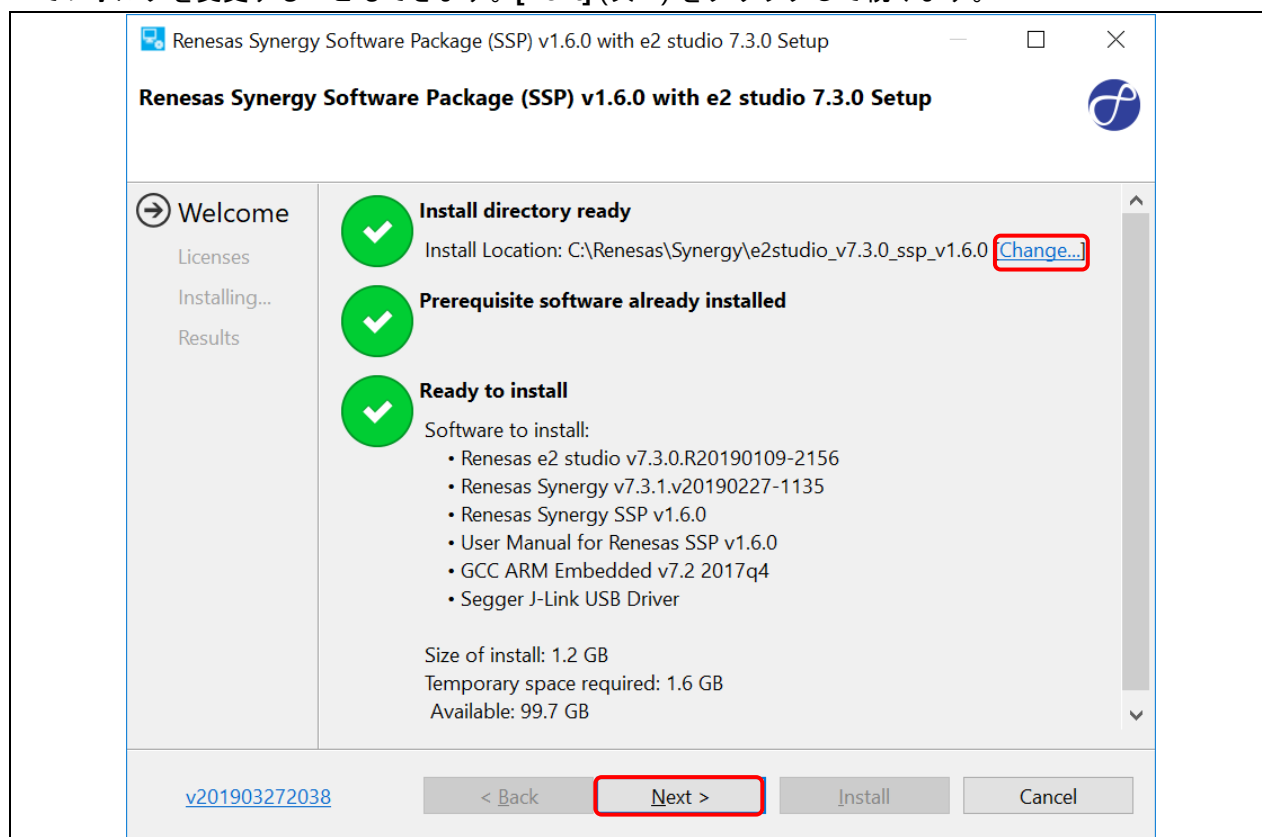


図 7. インストール - [Welcome] (ようこそ) ページ

7. **[Optional Components]** (オプションコンポーネント) ページで、インストールするコンポーネントを選択した後、**[Next]** (次へ) をクリックします。**[Quick Install]** (クイックインストール) を選択した場合、このページは表示されません。

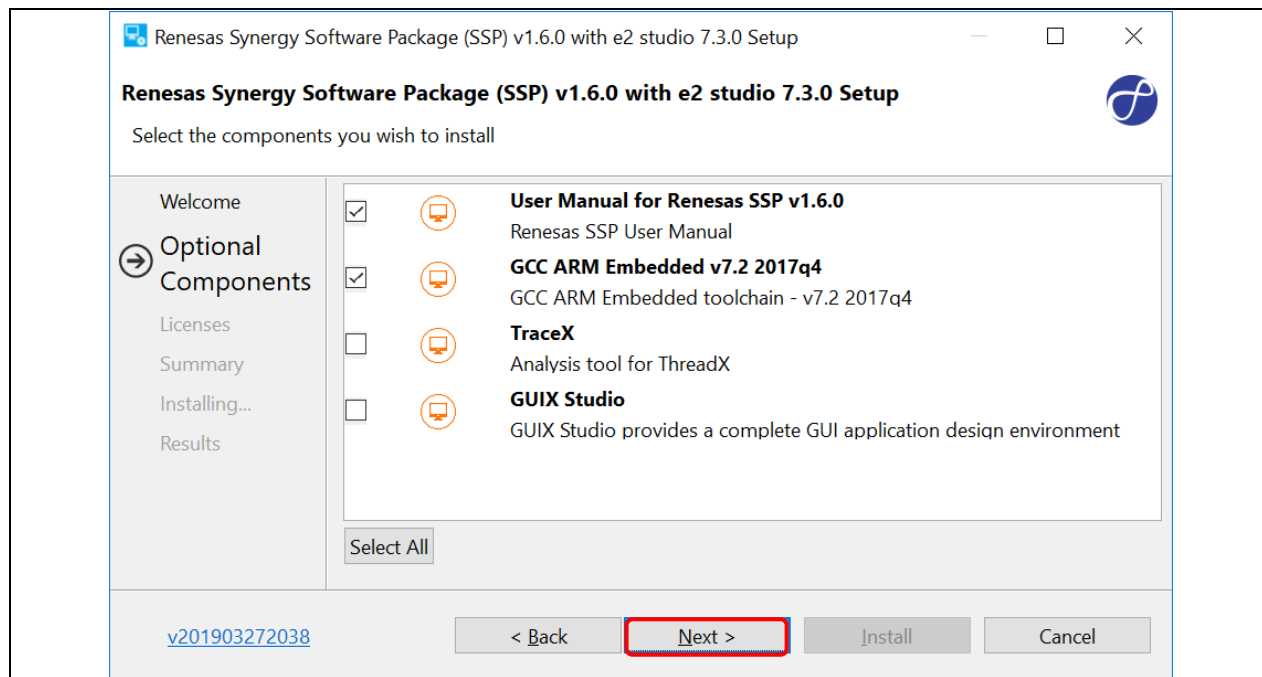


図 8. インストール - オプションコンポーネントの選択

8. チェックボックスをオンにしてライセンス契約に同意し、**[Install]** (インストール) をクリックして続けます。

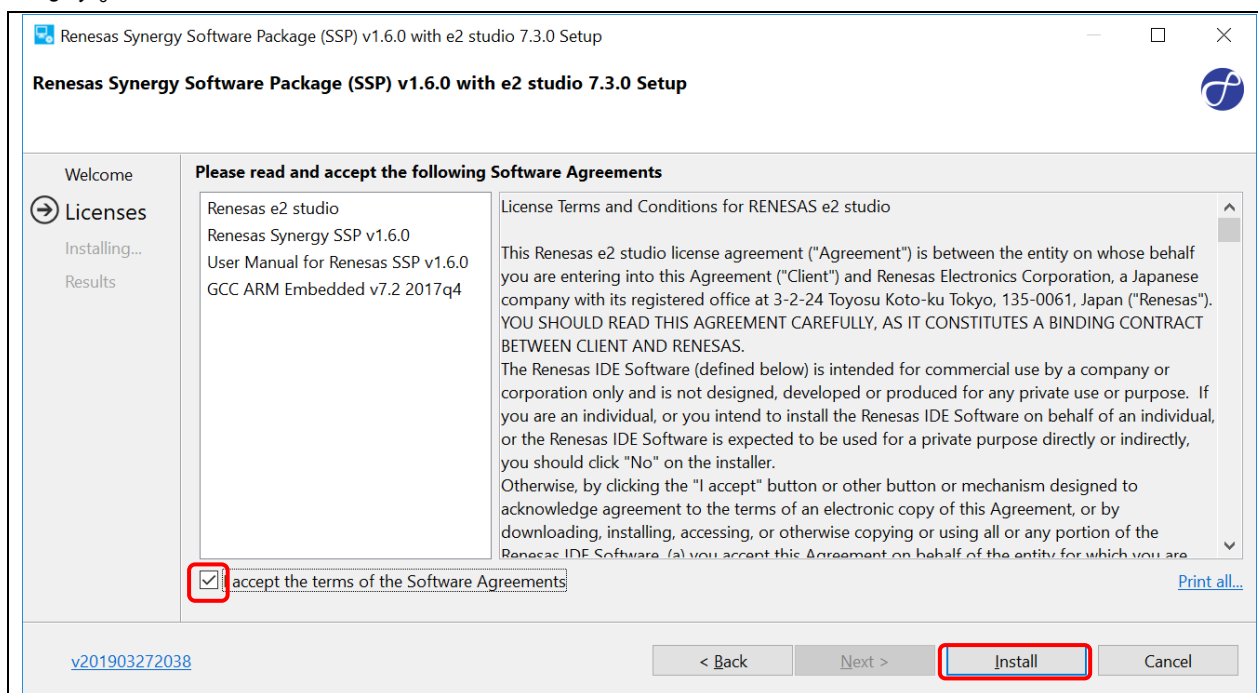


図 9. インストール - Software Agreements (ソフトウェア契約)

9. **[OK]** をクリックしてインストールを完了します。

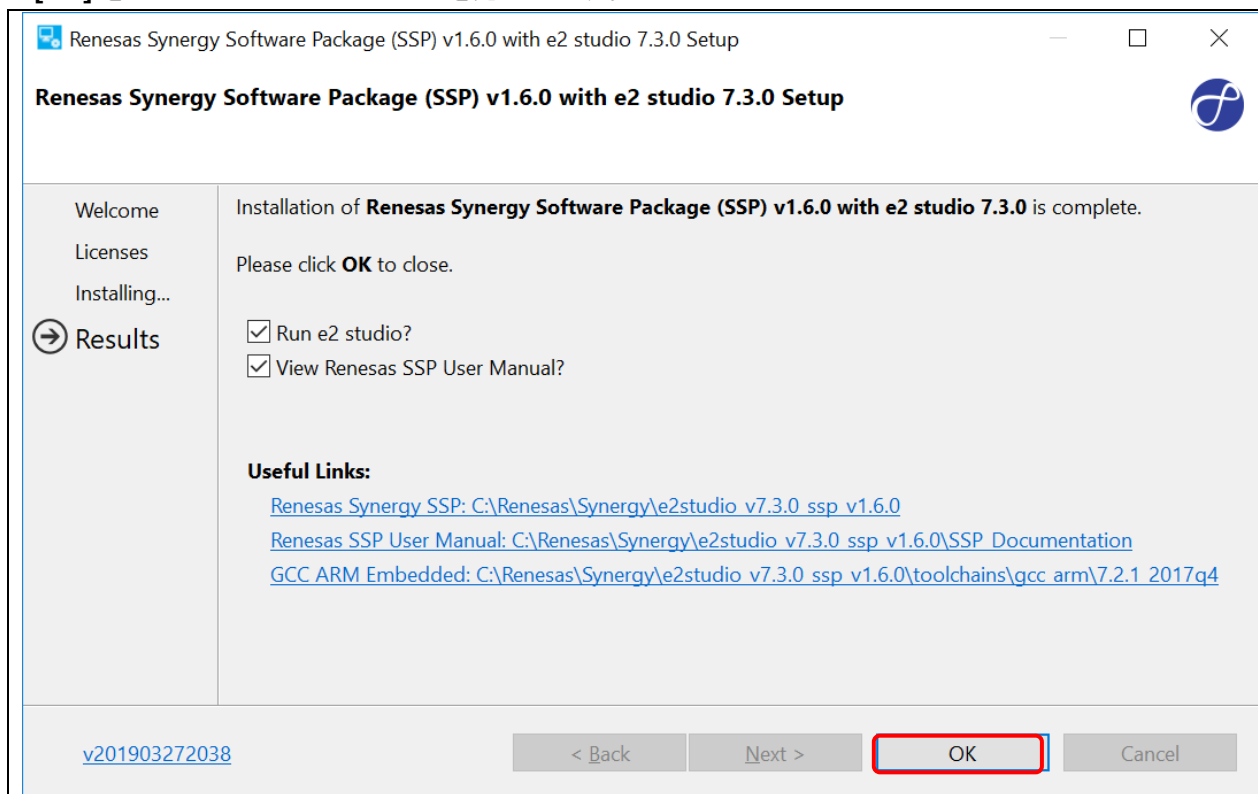


図 10. インストール - インストールの完了

2.2 スタンドアロンインストーラでのインストール (Installing the Standalone Installer)

この章では、スタンドアロンインストーラを使用して、以下の開発ツールを個別にインストールする方法を説明します。

- e² studio IDE
- GCC Arm Embedded コンパイラ
- Renesas Synergy™ Software Package (SSP)

最新のインストーラパッケージは、Synergy Platform Web サイト (<https://www.renesas.com/products/synergy.html>) の [Solutions Gallery] (ソリューション Gallery) からダウンロードできます。

2.2.1 e² studio のインストール (Installing e² studio)

Synergy 向けの e² studio をインストールするには、以下の手順に従ってください。

1. Synergy Platform Web サイト (<https://www.renesas.com/products/synergy.html>) の [ソリューション ギャラリー] をクリックします。
2. [ソリューションギャラリー] ページで、[ソフトウェアツール] の下の [e² studio] を選択します。
3. [e² studio] ページで、[ダウンロード] を選択します。

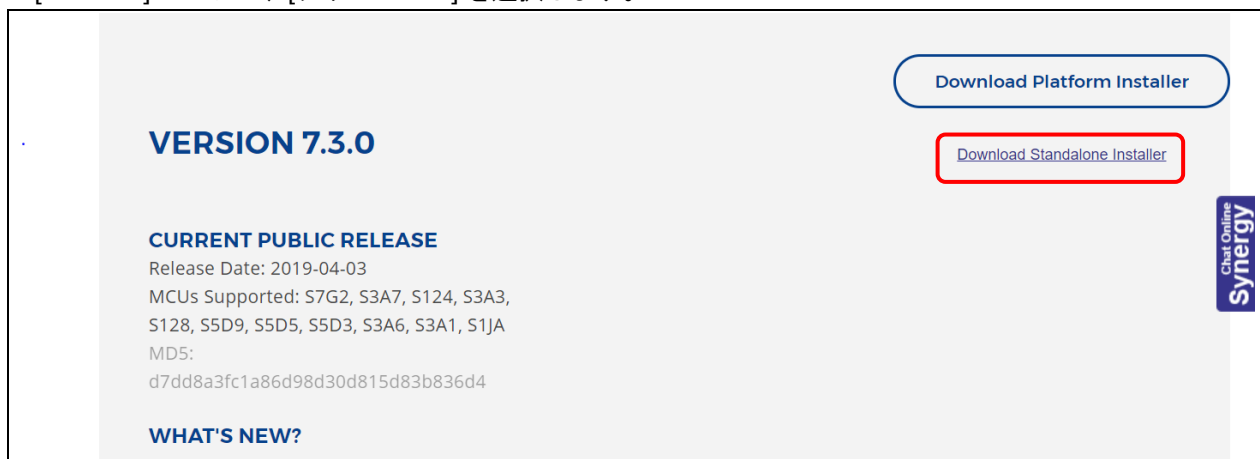


図 11. インストール - スタンドアロンインストーラのダウンロード

4. [License Agreement] (ライセンス契約) の [I Accept] (同意する) をクリックし、e² studio のスタンドアロンインストーラである `setup_e2studio_<version>.zip` をダウンロードします。 ([Download Standalone

Installer] (スタンドアロンインストーラのダウンロード) を有効にするには、MyRenesas アカウントにサインインする必要があります)。

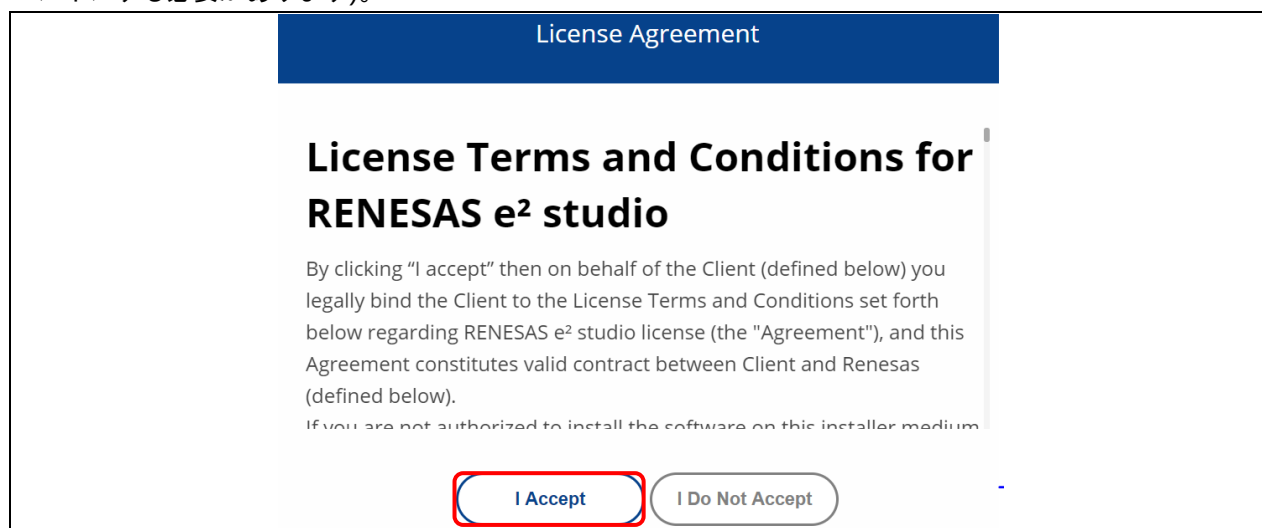


図 12. インストール - [License Agreement] (ライセンス契約) の同意

5. ダウンロードファイルを解凍し、e² studio インストーラを実行して、[e² studio インストールウィザード ページを開きます。[Next] (次へ) ボタンをクリックして続けます。

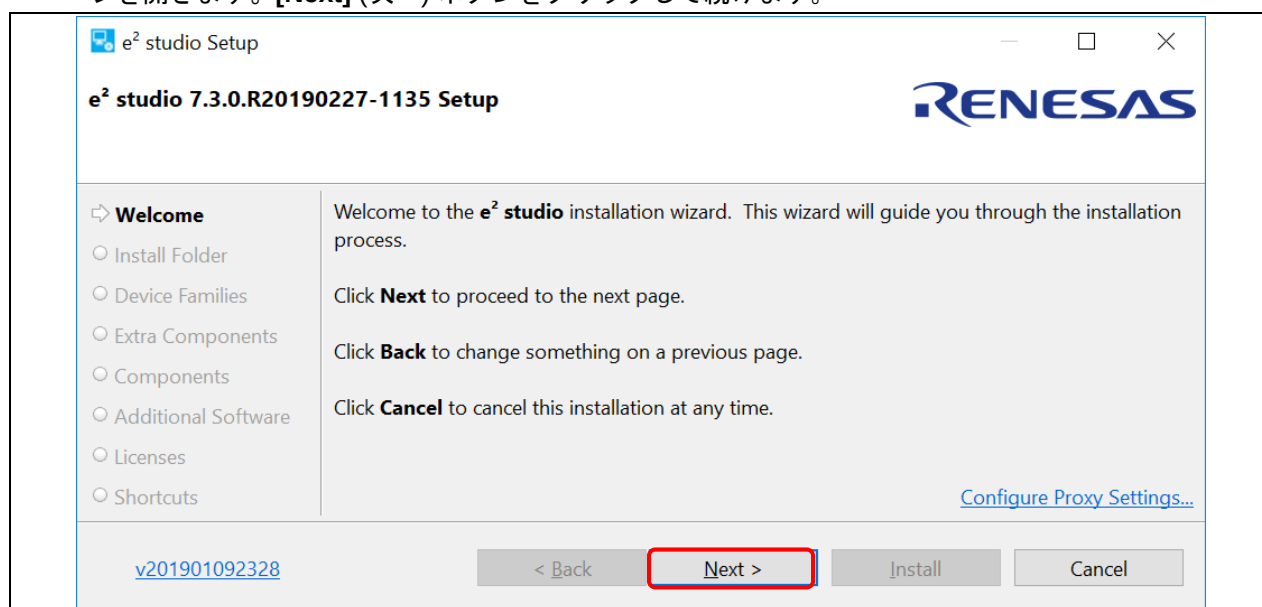


図 13. インストール - [Welcome] (ようこそ) ページ

6. Install Folder (インストール先フォルダ) :

デフォルトのインストール先は以下のように設定されています。C:\Renesas\e2_studio。インストール先フォルダをテキストボックスに直接入力するか **[Browse...]** (参照...) ボタンをクリックして指定します。

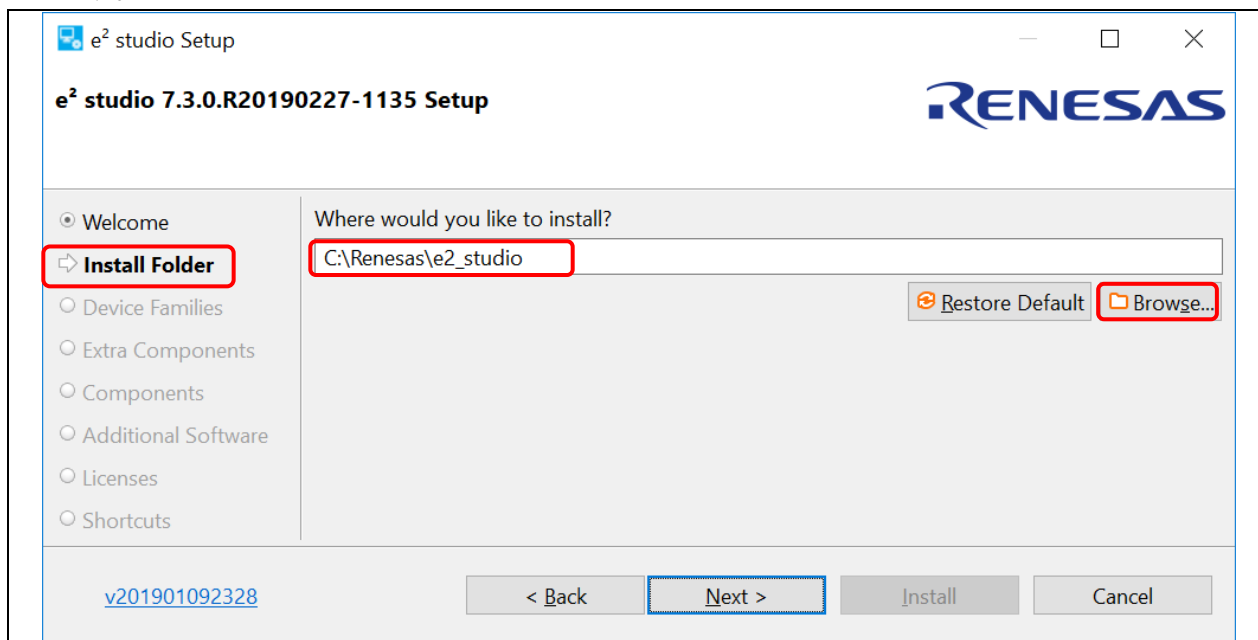


図 14. インストール - Install Folder (インストール先フォルダ)

7. Device Families (デバイスファミリ) :

[Renesas Synergy™] (ルネサス) チェックボックスをオンにします。他のデバイスに対応するチェックボックスはオプションです。

[Next] (次へ) ボタンをクリックして続けます。

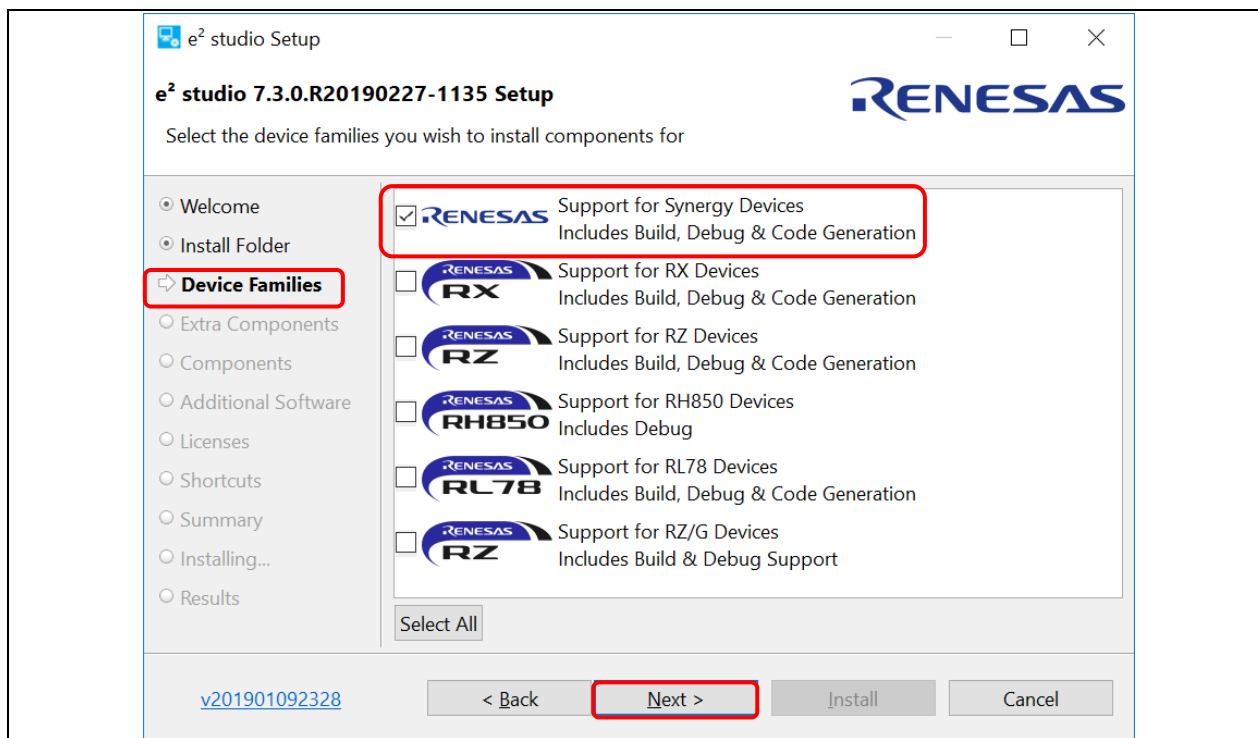


図 15. インストール - Device Families (デバイスファミリ)

8. Extra Components (追加コンポーネント)

インストールする追加コンポーネント（つまり、言語パック、SVN および Git サポート、RTOS サポート）を選択します。これらのコンポーネントはオプションです。[Next] (次へ) ボタンをクリックして続けます。

9. Components (コンポーネント)

[Renesas Synergy]、[Renesas Synergy Build Support Files] (Renesas Synergy ビルドサポートファイル)、[Renesas Synergy Debug Support Files] (Renesas Synergy デバッグサポートファイル) がオンになっていることを確認します。

[Next] (次へ) ボタンをクリックして続けます。

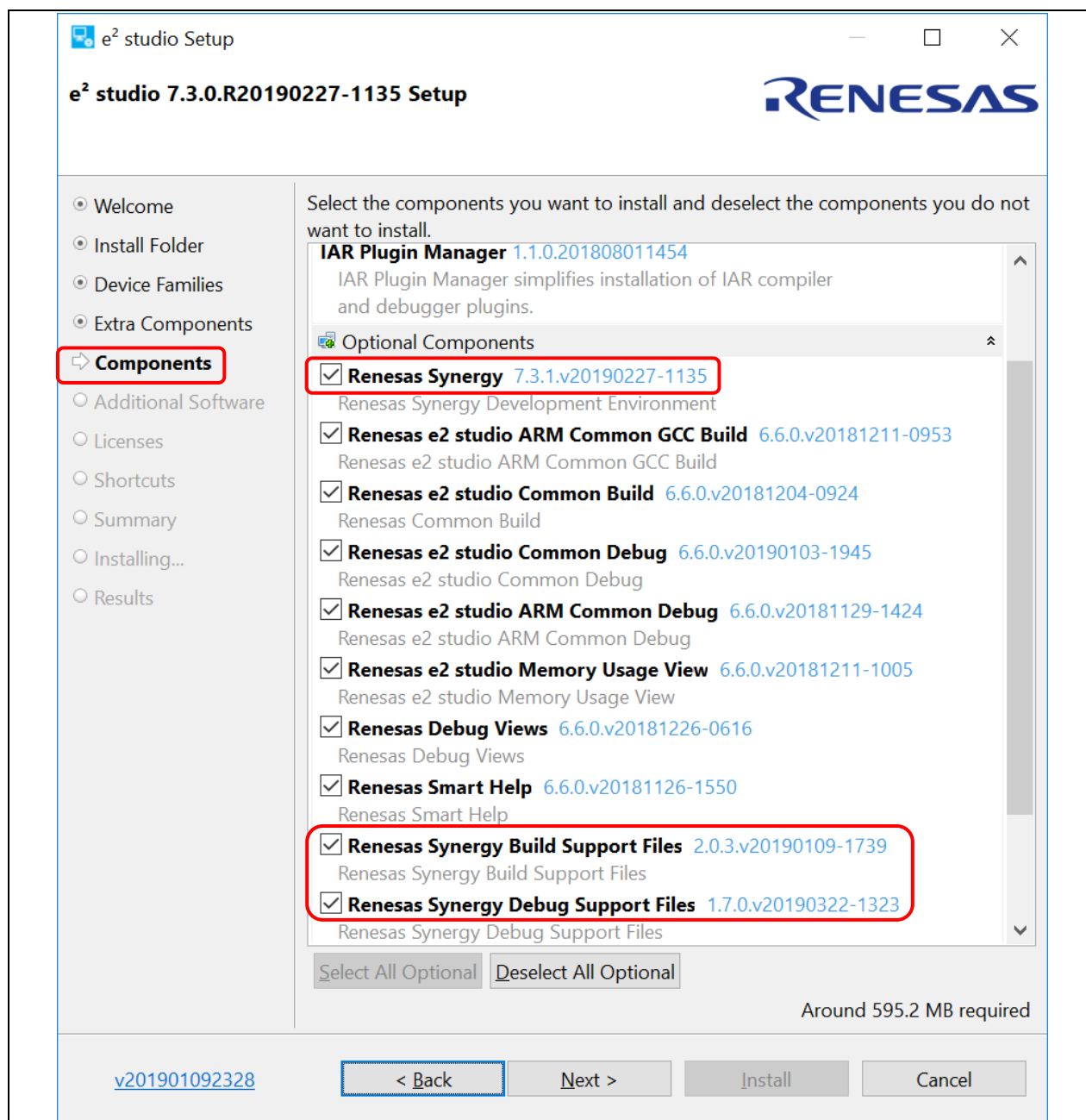


図 16. インストール - Components (コンポーネント)

10. Additional Software (追加ソフトウェア)

[Additional Software] (追加ソフトウェア) ダイアログで、[GCC Arm Embedded] の最新バージョンに対応するチェックボックスをオンにします (SSP v1.6 がサポートする GCC Arm Embedded の最新バージョンは、4.9 2015q3 と 7.2 2017q4 です)。それより古いバージョンはオプションです。

[Next] (次へ) ボタンをクリックして続けます。

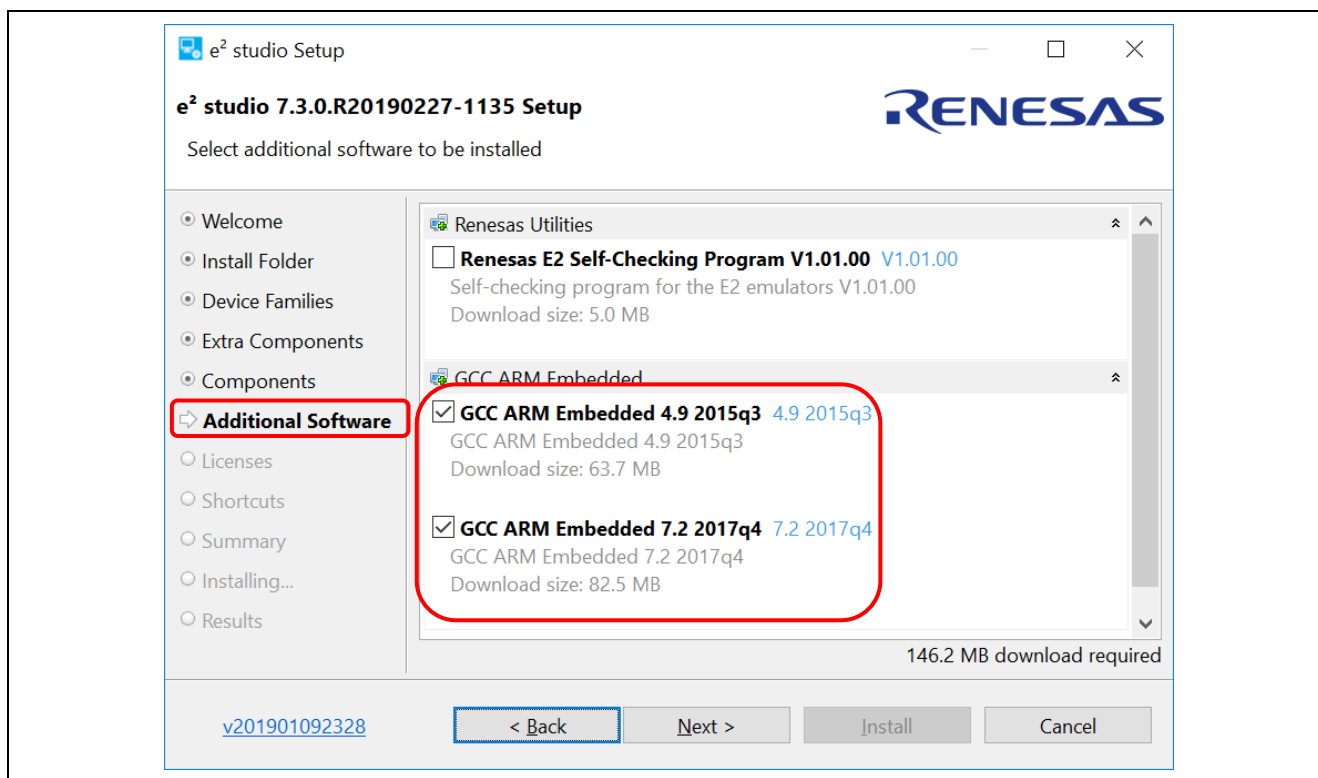


図 17. インストール - Additional Software (追加ソフトウェア)

11. Licenses (ライセンス)

ソフトウェアライセンス契約を読んで同意した後、[Next] (次へ) ボタンをクリックします。

ライセンス契約に同意する必要があることに注意してください。ライセンス契約に同意しない場合、インストールは続行できません。

12. Shortcuts (ショートカット)

スタートメニューに表示するショートカット名を選択し、[Next] (次へ) ボタンをクリックして続けます。

13. Summary (概要)

[Install] (インストール) ボタンをクリックし、Renesas e² studio をインストールします。

14. Installing... (インストール中...)

インストールが開始されます。[Additional Software] (追加ソフトウェア) ダイアログで選択した項目に応じて、それらのソフトウェアパッケージをインストールするためのダイアログが開きます。

[GCC ARM Embedded...] がインストールされます。インストール全体ですべてのデフォルト設定をそのままにします。また、最後のダイアログで、[Add path to environment variable] (パスを環境変数に追加) をオンにします。

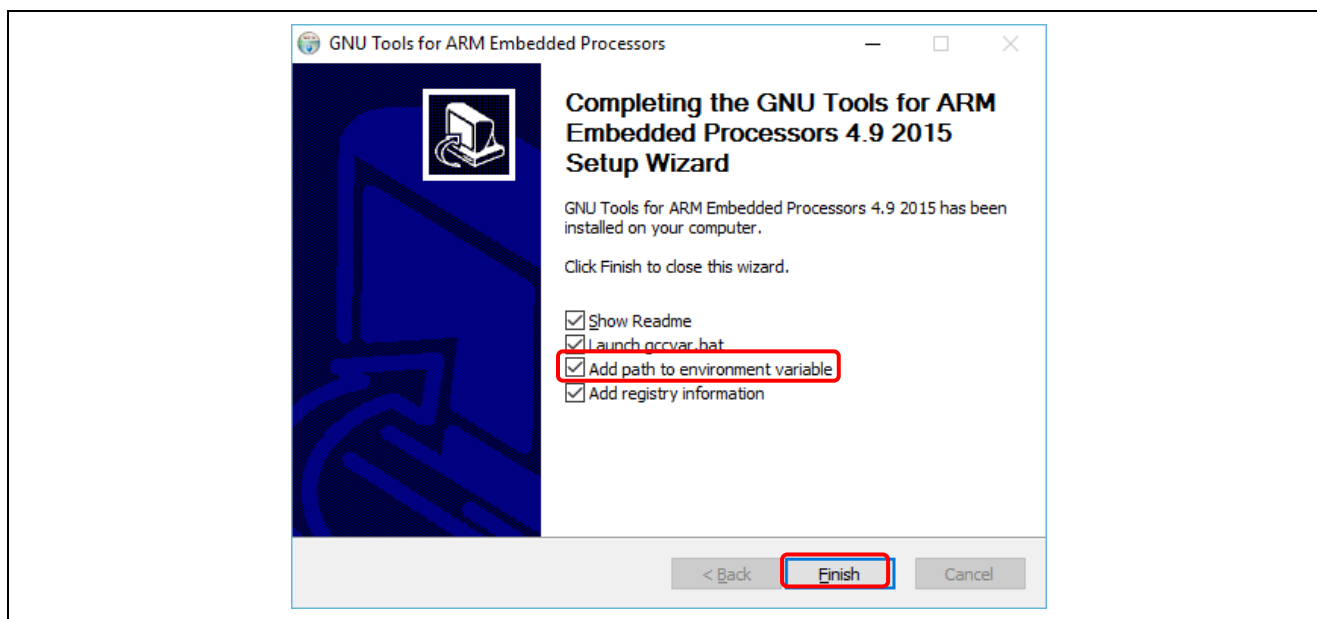


図 18. インストール - GCC Arm Embedded コンパイラのインストール

2.2.2 GNUARM コンパイラのセットアップ (Setting Up the GNUARM Compiler)

GNUARM toolchain は、e² studio インストールの際にインストールできます。GNUARM コンパイラを単独でインストールするには、以下の手順に従ってください。

1. Renesas Synergy™ GNU Arm コンパイラのサポート対象である (現時点で v7.2.1) を <https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-arm/downloads> からダウンロードします。
2. インストーラを実行し、GNU Arm コンパイラをホストコンピュータにインストールします。
3. インストールの言語を選択します。インストール確認ダイアログで **[Yes]** (はい) をクリックします。
4. インストールウィザードですべてのデフォルト設定をそのままにしてください。
5. **[Install wizard Complete]** (インストールウィザードの完了) ダイアログが表示されたときに、**[Add path to environment variable]** (パスを環境変数に追加) ボックスをオンにしてから **[Finish]** (完了) をクリックしてインストールを完了します。

2.2.3 Renesas Synergy™ソフトウェアパッケージ (SSP) のインストール (Installing the Renesas Synergy™ Software Package (SSP))

e² studio インストーラには、Renesas Synergy™ソフトウェアパッケージ (SSP) は付属していません。プラットフォームインストーラを使用していない場合、SSP は個別にインストールする必要があります。SSP パッケージインストーラには、ドライバライブラリ、SSP の評価用ライセンス、HTML 形式のユーザーズマニュアル、readme ファイルが付属しています。

SSP をインストールするには、以下の手順に従ってください。

1. Synergy Platform Web サイト (www.renesas.com/synergy) の **[Solutions Gallery]** (ソリューション Gallery) をクリックします。
2. **[ソリューションギャラリー]** ページで、**[ソフトウェア]** の、**[Synergy Software Package]** を選択します。
3. **[Synergy Software Package]** ページで、**[ダウンロード]** を選択し、`SSP_Distribution_<SSP-version>.zip` ファイルをダウンロードします。 (**[ダウンロード]** を有効にするには、MyRenesas アカウントにサインインする必要があります)。

SSP のリリースノートとユーザーズマニュアルも、Synergy Software Package (Synergy ソフトウェアパッケージ) ページからダウンロードできます。

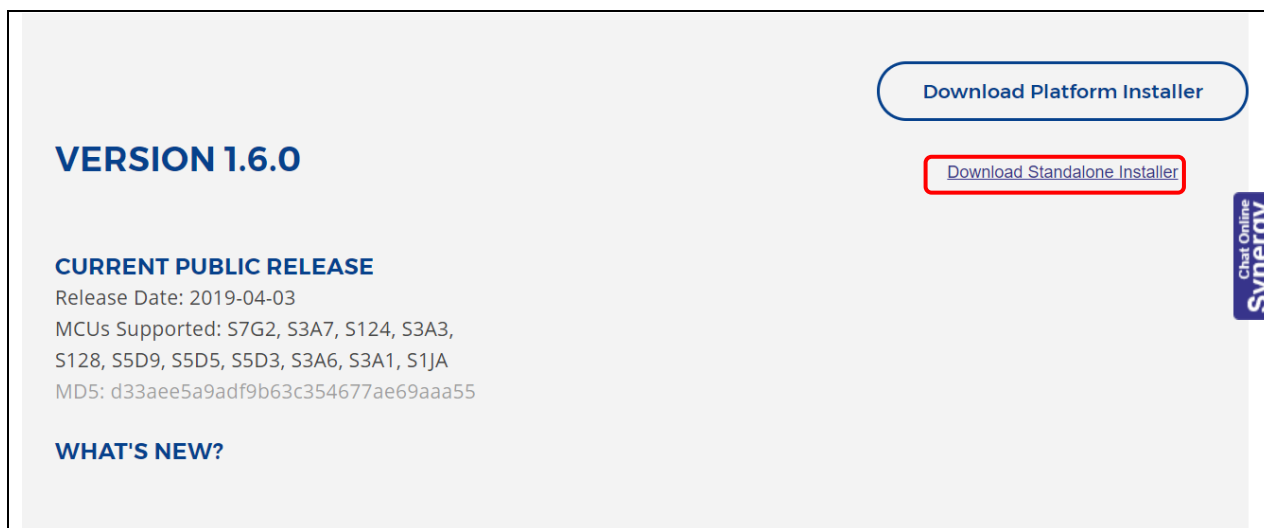


図 19. インストール - スタンドアロン SSP パッケージのダウンロード

4. 互換性のある e² studio が既にインストールされていること、またこのインストールの間は閉じていることを確認してください。
5. パッケージを解凍し、SSP_Distribution_<SSP-version>.exe インストーラを実行します。
6. インストールウィザードダイアログで **[Next]** (次へ) ボタンをクリックします。
7. ライセンス契約を読み、**[I Agree]** (同意する) をクリックしてインストールプロセスを続けます。

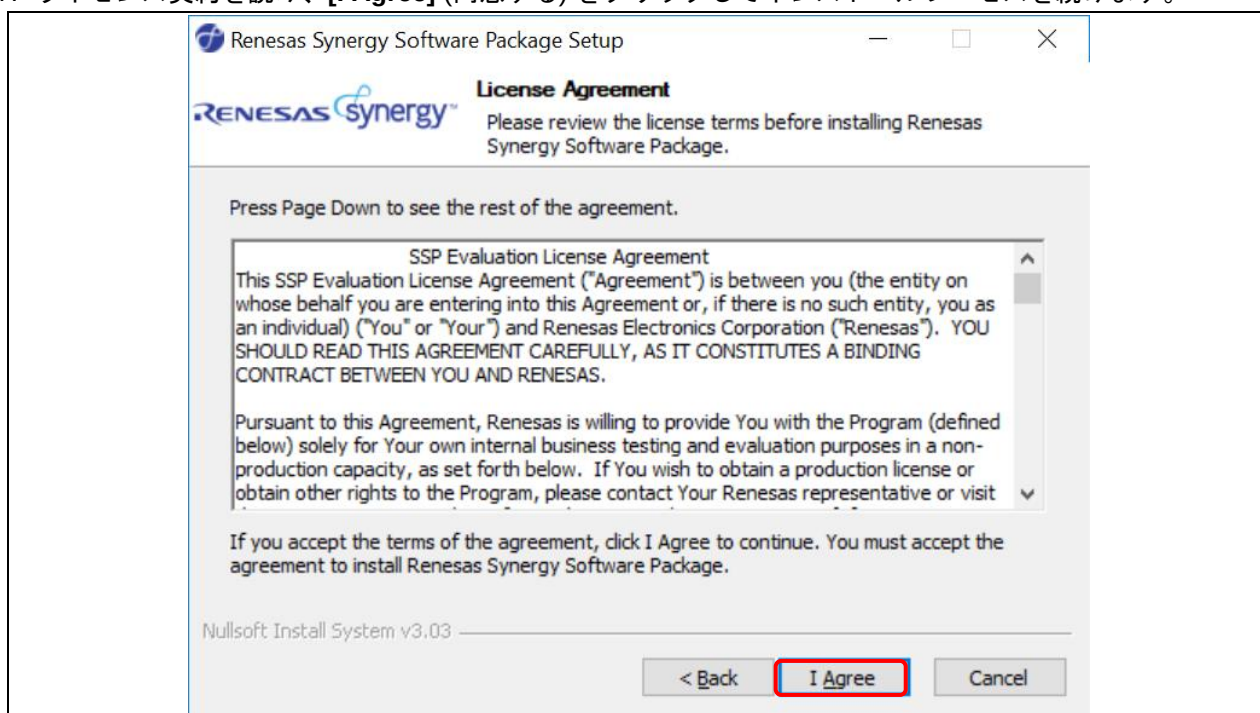


図 20. インストール - License Agreement (ライセンス契約)

8. **[Choose Components]** (コンポーネントの選択) ダイアログでデフォルト設定を使用し、**[Next]** (次へ) ボタンをクリックして続けます。

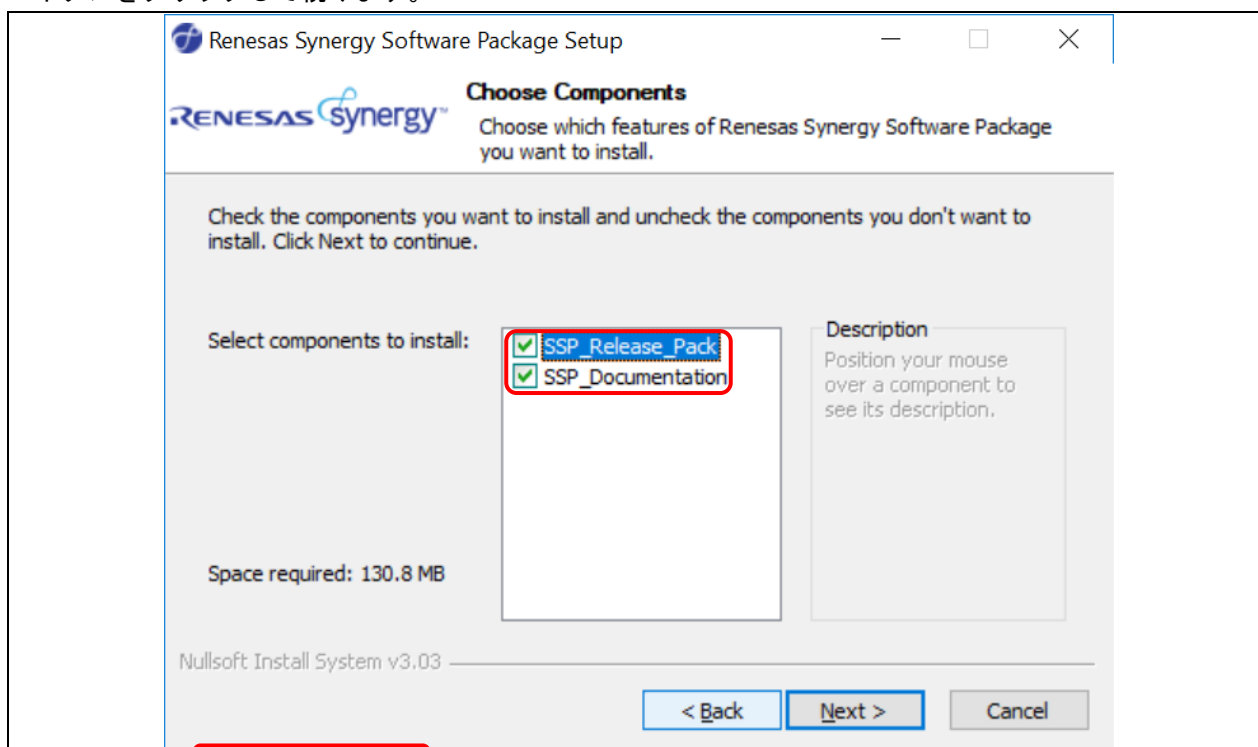


図 21. インストール - Choose components (コンポーネントの選択)

9. SSP を、e² studio のルートフォルダ (デフォルトのルートフォルダは C:\Renesas\le2_studio) にインストールします。この C:\Renesas\le2_studio フォルダは、SSP のデフォルトインストールフォルダです。**[Install]** (インストール) ボタンをクリックしてインストールを開始します。

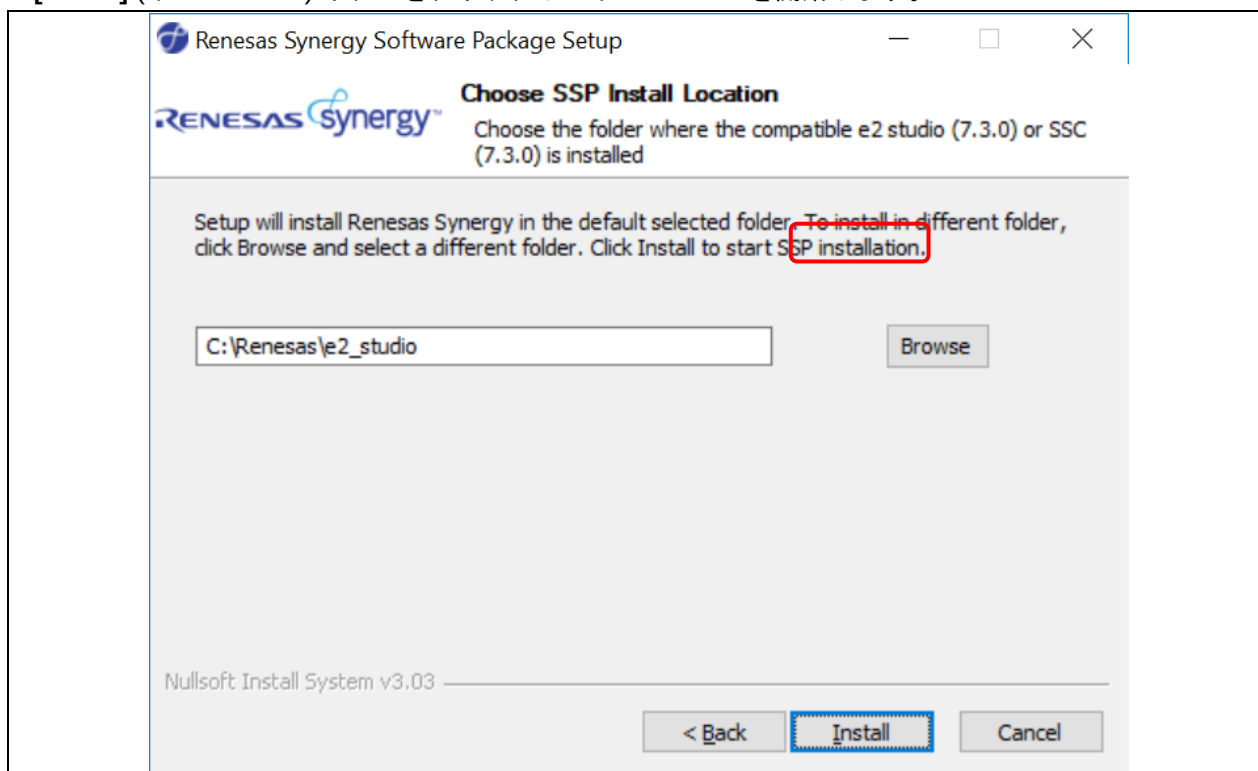


図 22. インストール - SSP Installation Folder Selection (SSP のインストール先フォルダの選択)

10. SSP インストールプロセスの実行中に、[SSP Documentation Installation] ウィザードでドキュメントのインストールについて確認を求められます。ドキュメントのデフォルトインストール場所は

C:\Renesas\Synergy です。ユーザはデフォルトの場所を変更することもできます。**[Install]** (インストール) ボタンをクリックしてインストールを開始します。

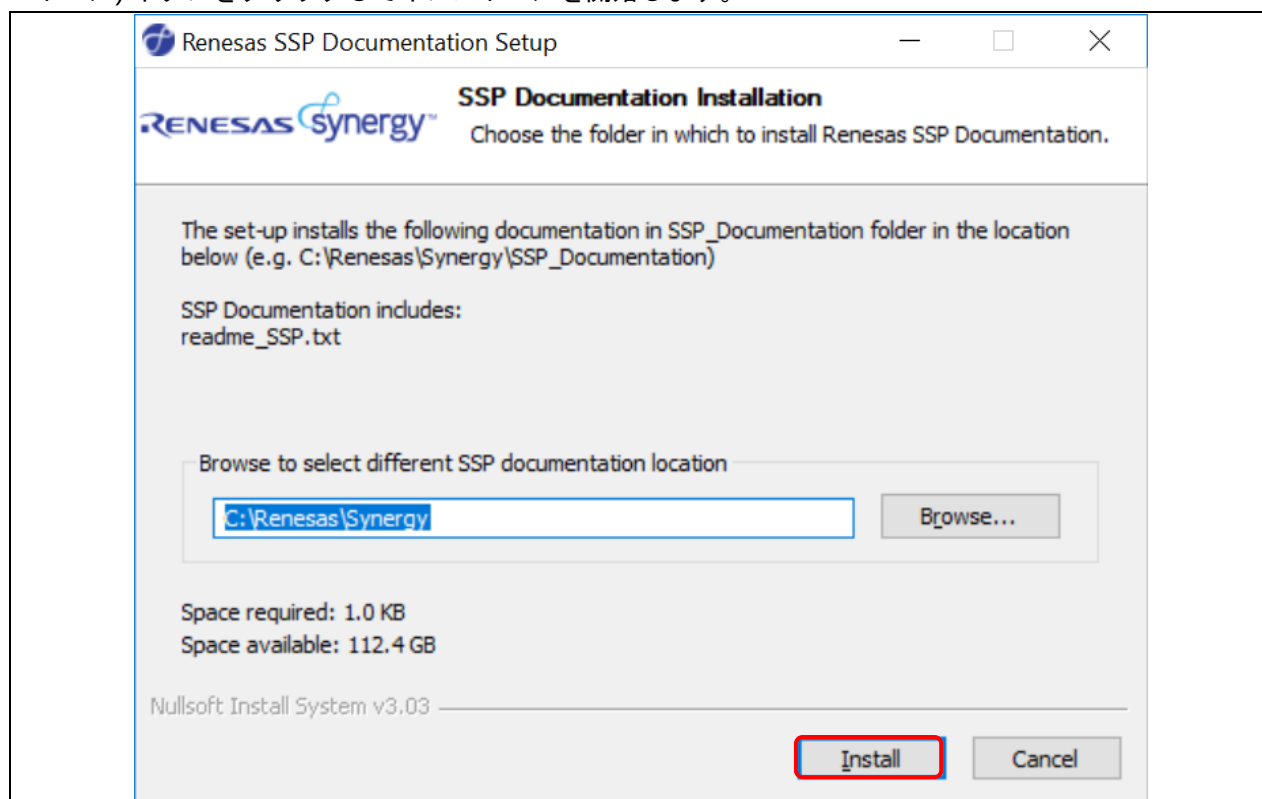


図 23. インストール - SSP Documentation Installation (SSP ドキュメントのインストール)

11. インストールが完了した時点で **[Close]** (閉じる) ボタンをクリックしてインストールウィザードを終了します。

SSP をインストールした後、評価用ライセンスファイルは

<e2_studio_base_dir>/internal/projectgen/arm/Licenses/ディレクトリに保存されます。

2.3 e² studio のアンインストール (Uninstalling e² studio)

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]** 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ます。

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

2.4 e² studio の更新 (Updating e² studio)

新しいバージョンに更新するには、Synergy Platform Web サイト (<https://www.renesas.com/products/synergy.html>) の[ソリューションギャラリー]からプラットフォームインストーラまたはスタンドアロンインストーラをダウンロードします。

既存のインストールを上書きしないでください。IDE をアップグレードする前に、e² studio の古いバージョンをアンインストールする必要があります。ただし、e² studio の古いバージョンを残したまま新しいバージョンを利用したい場合は、e² studio の新しいバージョンを新たに作成した別のフォルダにインストールすることもできます。

2.5 Synergy ライセンスの登録 (Register Synergy license)

プロジェクトをビルドして実行するには、e² studio で Synergy ライセンス (XML 形式) を登録する必要があります。登録情報は、workspace フォルダに保存されています。したがって、ユーザが新しい空のワークスペースに切り替えたときは必ず、登録を再度実行する必要があります。

Synergy ライセンスを登録するには、以下の手順に従ってください。

1. メニューバーで、[Help] (ヘルプ) → [Synergy License] (Synergy ライセンス) をクリックします。

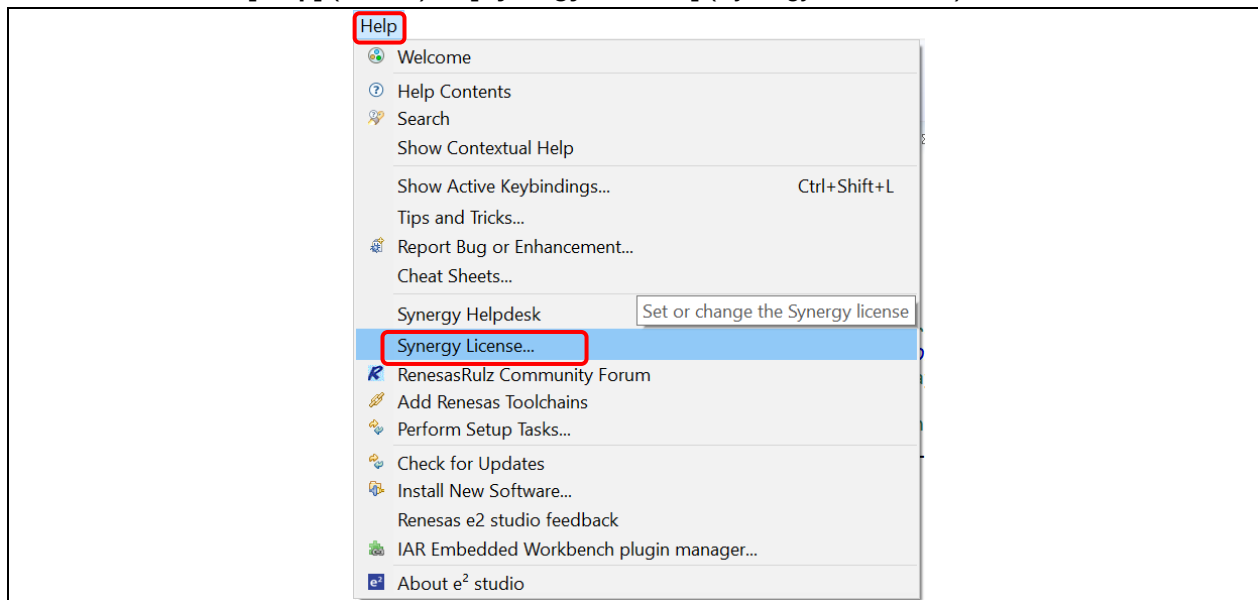


図 24. Synergy ライセンスの登録 (Register Synergy license)

2. [...] をクリックし、次に [Browse] (参照) をクリックしてライセンスファイルを参照します。
<e2_studio_base_dir>\internal\projectgen\arm\Licenses\に保存されているデフォルトの評価用ライセンスファイルが自動的に事前選択されています。ライセンスファイルを選択してから、[Open] (開く) をクリックし、次に [OK] を 2 回クリックしてライセンスを登録します。

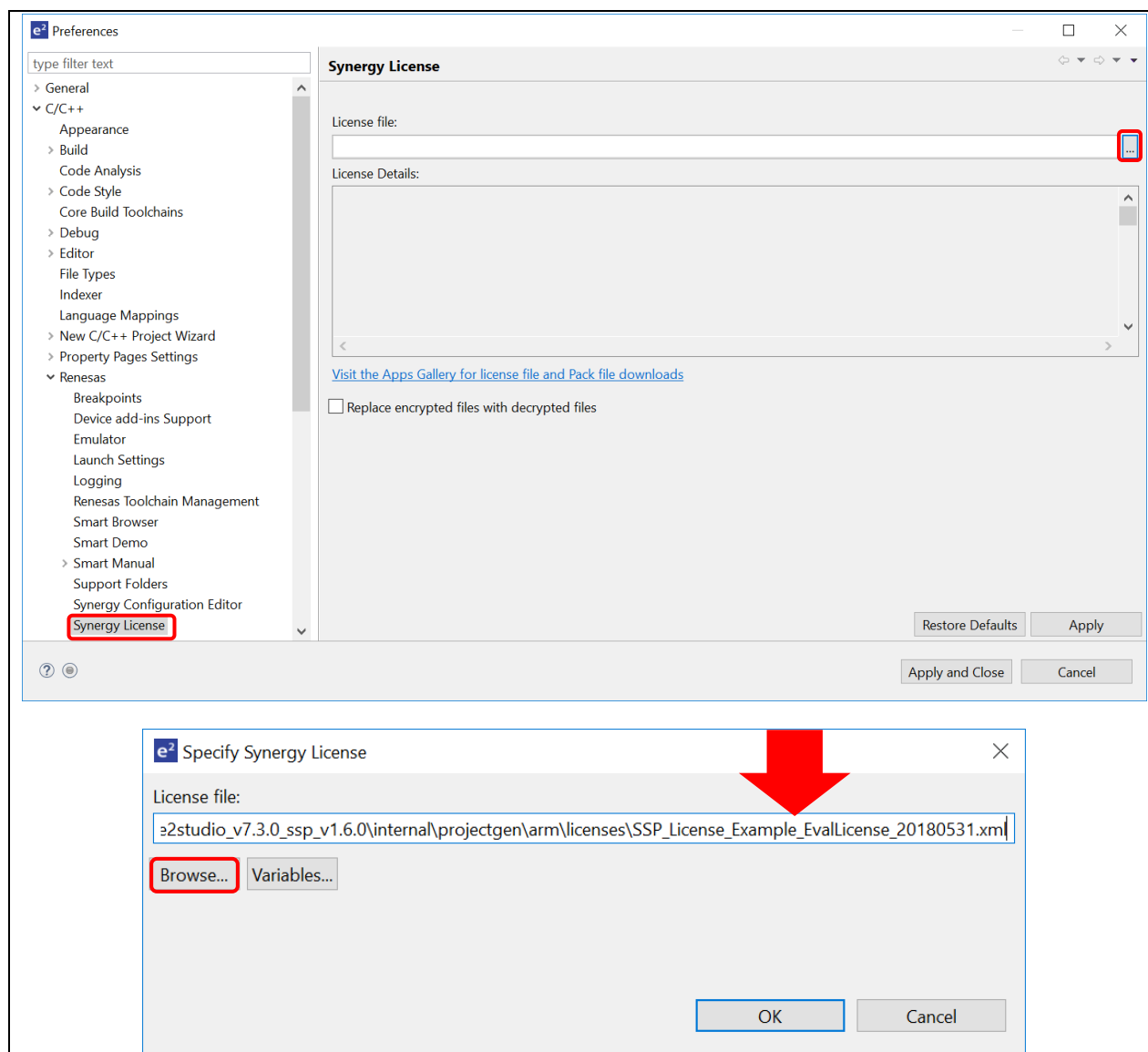


図 25. Synergy ライセンスファイルの場所を指定

3. プロジェクトの作成 (Project Generation)

この章では、新しい Synergy プロジェクトの作成について説明します。e² studio には、新しい Synergy プロジェクトの迅速な作成を支援するウィザードが付属しています。このような迅速さを実現しているのは、プロジェクトを特定の Synergy デバイスやボードに対応させるウィザードの機能です。

プロジェクトジェネレータを使用して、ピンの構成、割り込み、クロックの構成、さらに必要なドライバソフトウェアを設定することもできます。

前提条件として、SSP とツールチェーンをホストコンピュータにインストールする必要があります (第 2 章を参照)。

3.1 新しい Synergy プロジェクトの作成 (Generating a New Synergy Project)

e² studio のシンプルなプロジェクト生成ウィザードにより、新しい Synergy プロジェクトを生成し、プロジェクト名のほか、ボードレベルドライバも含め関連するデバイスとボードを指定することができます。

e² studio アプリケーションを起動し、Workspace Launcher でワークスペースフォルダを指定します。新しい Synergy プロジェクトフォルダを構成するには、以下の手順に従ってください。

1. **[File]** (ファイル) → **[New]** (新規作成) → **[Synergy C/C++ Project]** を選択します。

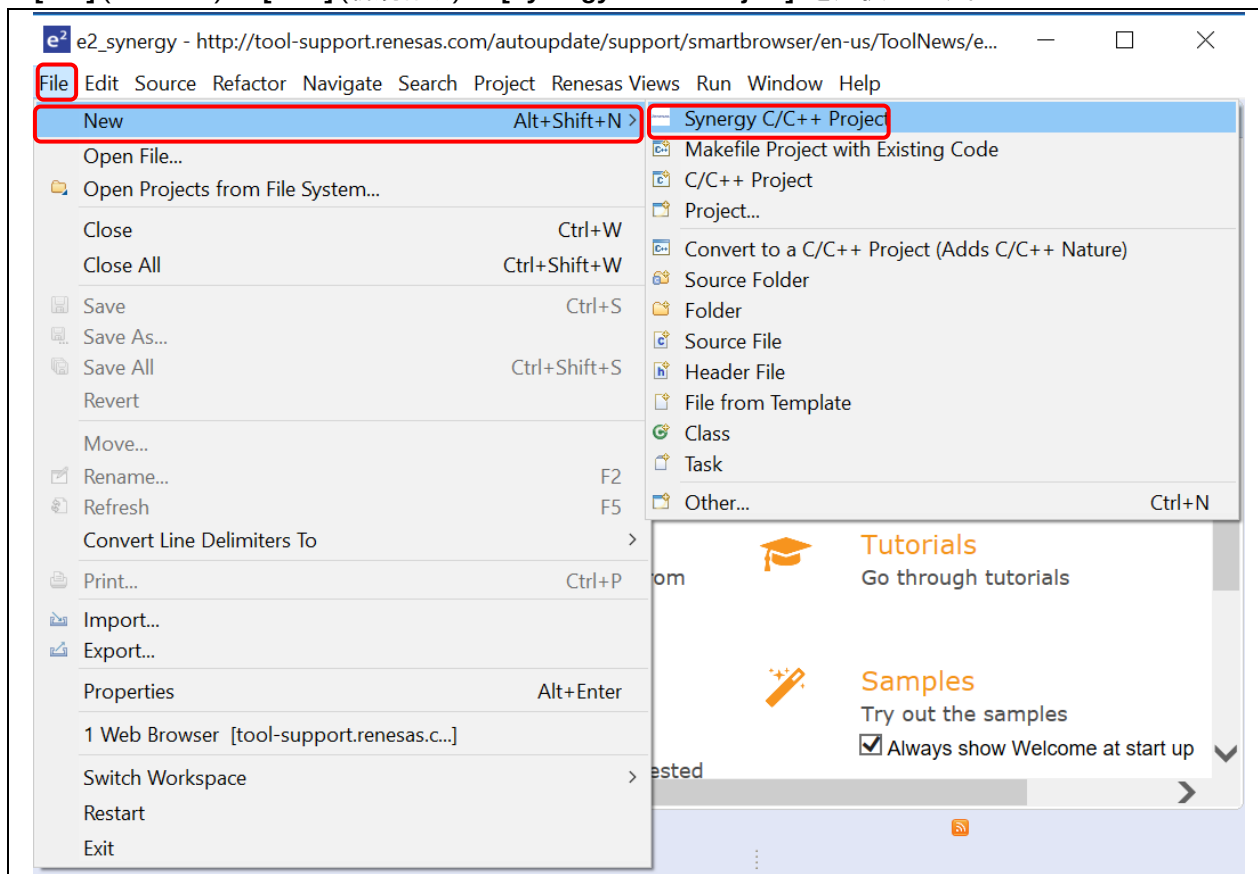


図 26. プロジェクトの生成 - 新しいプロジェクトの作成

2. **[Renesas Synergy C Executable Project]** (Renesas Synergy C 実行可能プロジェクト) テンプレートを
選択します。**[Next]** (次へ) をクリックして続けます。

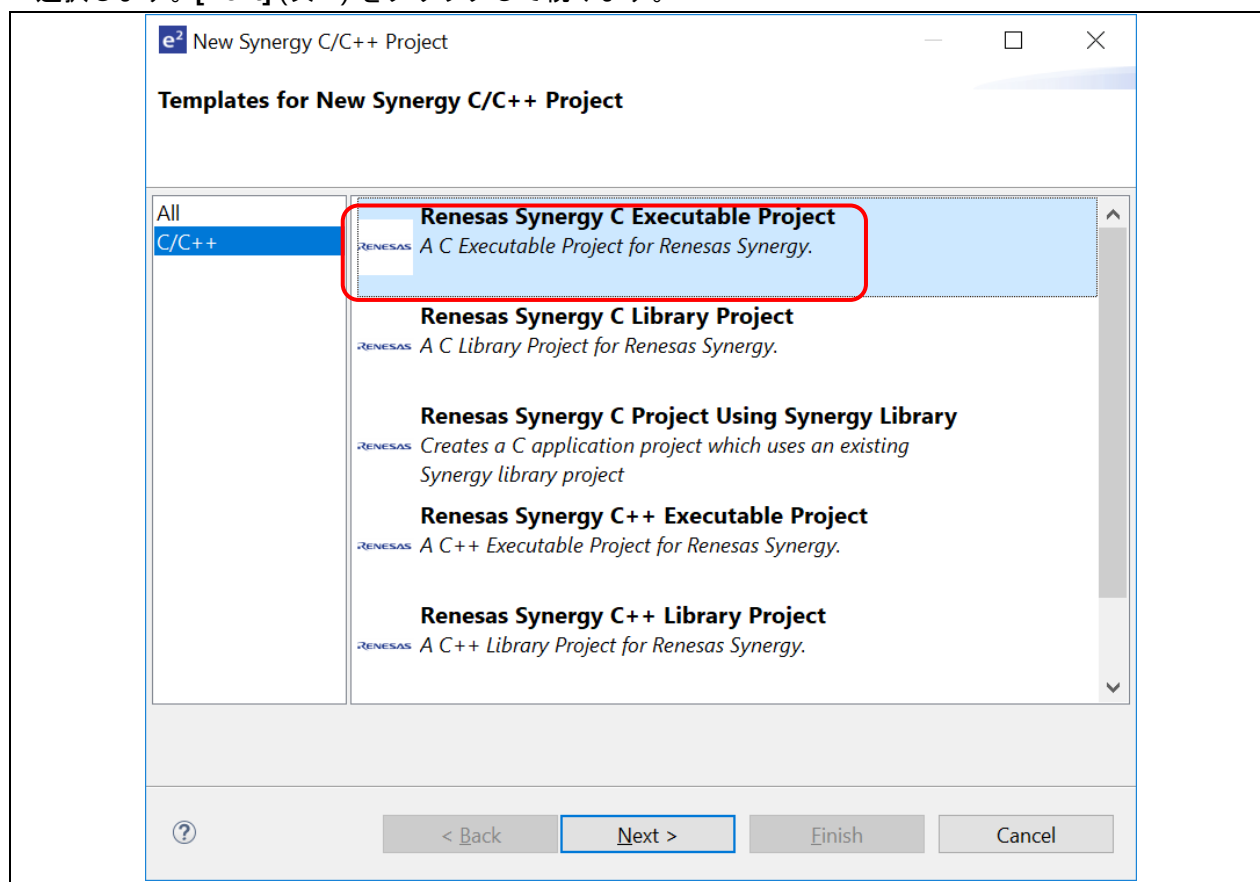


図 27. プロジェクトの生成 - 実行可能プロジェクトテンプレートの選択

3. プロジェクト生成ウィザードで、以下のプロジェクト情報を入力します。
- **[Project name]** (プロジェクト名): 「Synergy」など、名前を入力します。
 - **[Use default location]** (デフォルト・ロケーションの使用) チェックボックスをオンにします。ユーザが別の場所にプロジェクトを作成することを希望する場合、このボックスをオフにして、新しい場所を入力することもできます。
 - **[Toolchain]** (ツールチェーン): GCC Arm Embedded
 - **[License]** (ライセンス): 現在のワークスペースに Synergy ライセンスファイルが登録されていない場合、**[Change license file]** (ライセンスファイルの変更) をクリックして **[Synergy License]** (Synergy ライセンス) ダイアログボックスを開きます。[...] ボタンをクリックし、選択しようとする Synergy ライセンスファイルを参照します。Synergy ライセンスのデフォルトの場所は、以下のとおりです。
{e² studio のインストール先フォルダ}\internal\projectgen\arm\Licenses。
このライセンスファイルは、SSP をインストールした後でのみ使用可能です。
 - **[Next]** (次へ) ボタンをクリックして続けます。

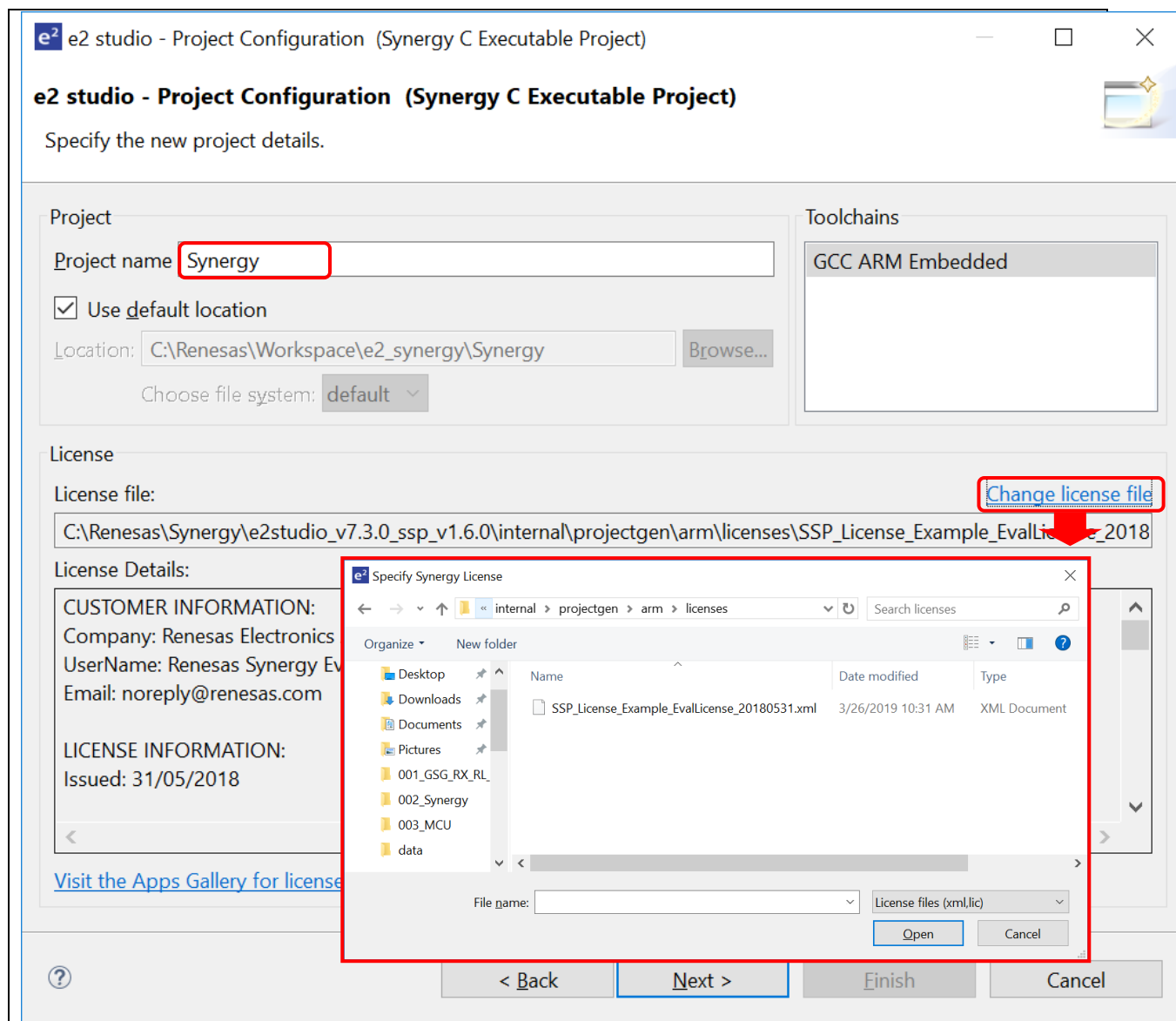


図 28. プロジェクト生成 - 新しい Synergy プロジェクトの生成ウィザード

4. デバイス選択ダイアログで、デバイスとツールの情報を入力します。
 - [Board] (ボード) (例 : S7G2 DK)
 - [Toolchain version] (ツールチェーンのバージョン): Renesas Synergy™での使用が承認されている最新の GNU コンパイラ (例えば、GCC Arm Embedded 7.2.1.20170904)
 - 他のすべてのフィールドは、デフォルト値のままにします。
 - [Next] (次へ) をクリックして続けます。

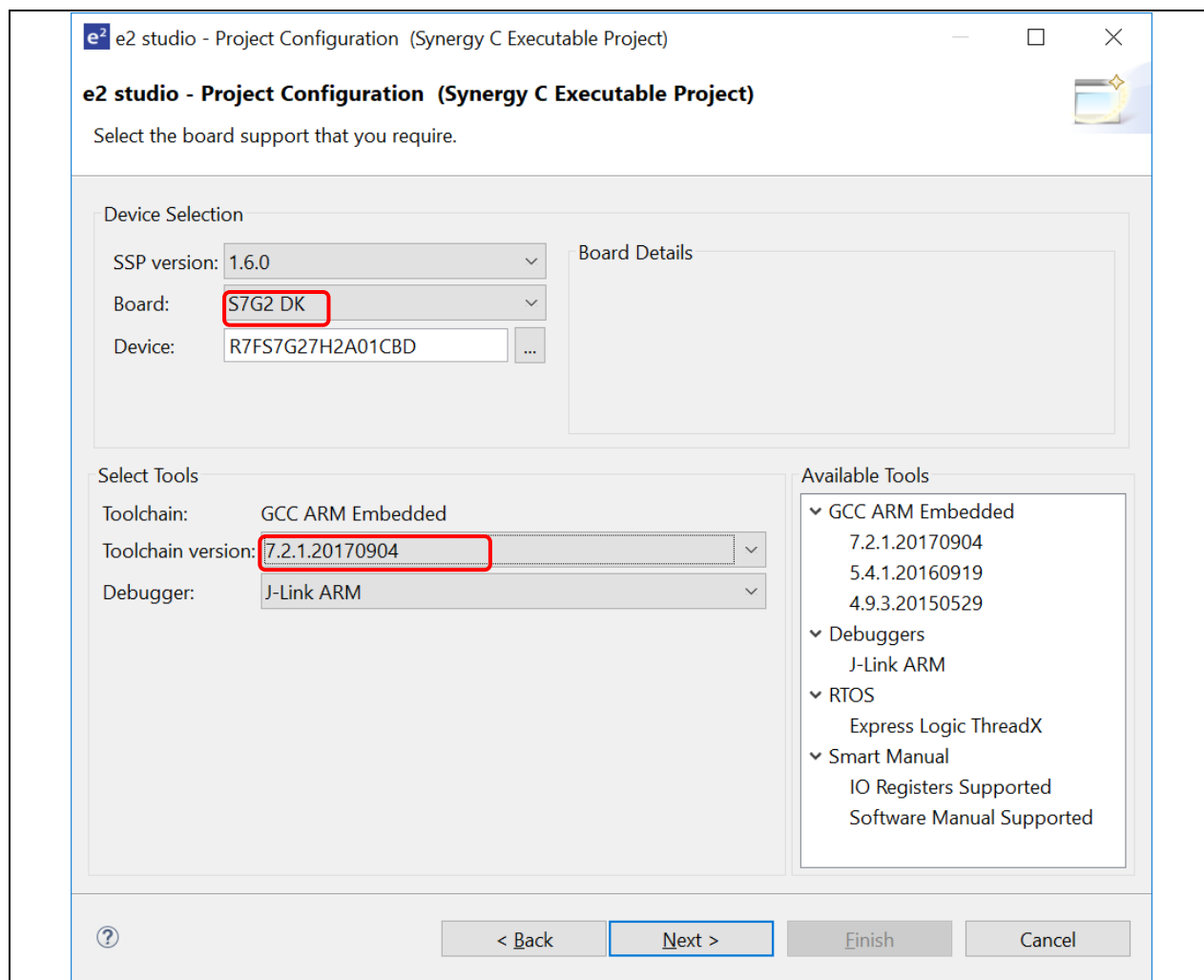


図 29. プロジェクトの生成 - デバイス選択

5. プロジェクトテンプレートダイアログで、Blinky のようなプロジェクトテンプレートを選択します。

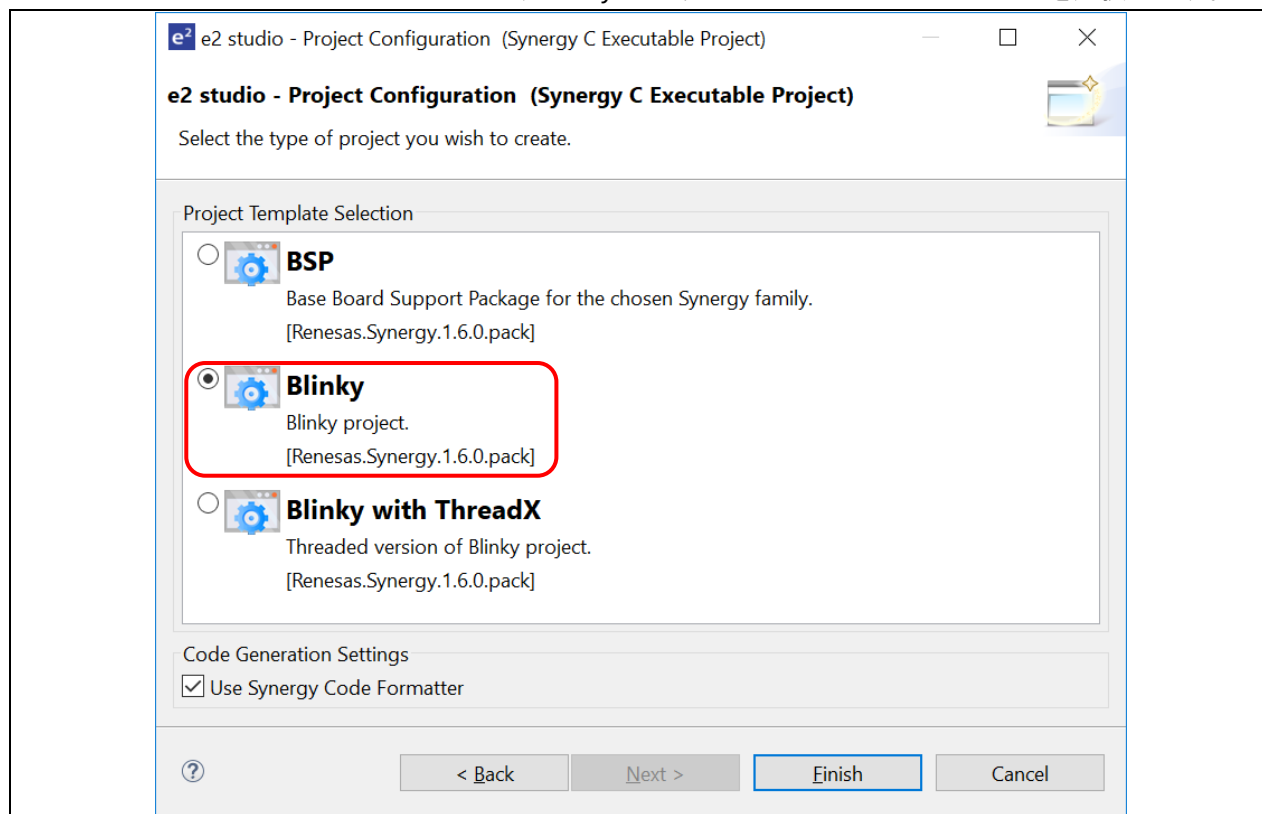


図 30. プロジェクトの生成 - プロジェクトテンプレート

6. **[Finish]** (完了) ボタンをクリックして新しいプロジェクトを作成します。

[Synergy Configuration] (Synergy 構成) パースペクティブを開くように確認を求められることがあります。

[Yes] (はい) をクリックして、パースペクティブを開きます。

(Eclipse で、「パースペクティブ」とは、事前定義されたペインやビューの配置のことです。)

e² studio で、新しいプロジェクトが作成され、[Project Explorer] (プロジェクトエクスプローラ) ビュー、Synergy Project Configuration Editor、[Package] (パッケージ) ビューなどのさまざまなビューが表示されます。

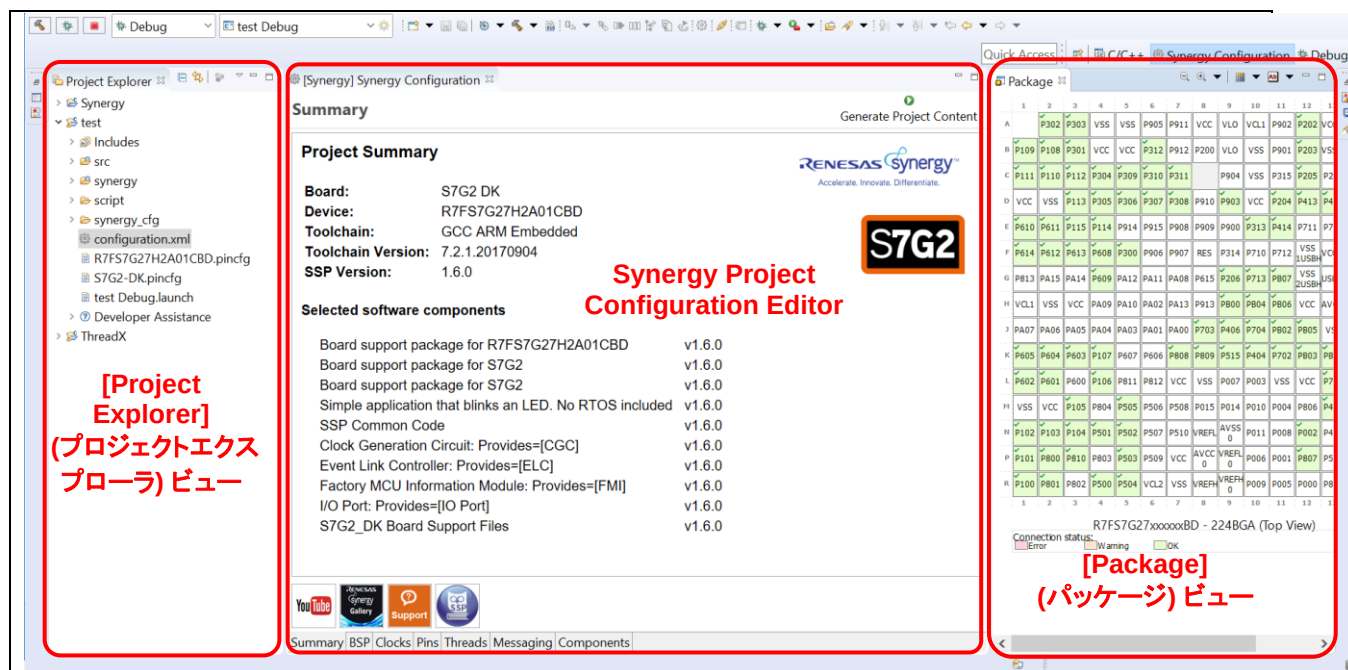


図 31. プロジェクトの生成 - 新しいプロジェクトの作成ビュー

3.2 既存のプロジェクトのインポート (Importing an Existing Project)

既存の Synergy プロジェクトをインポートするには、以下の手順に従ってください。

1. **[File]** (ファイル) → **[Import]** (インポート) をクリックします。

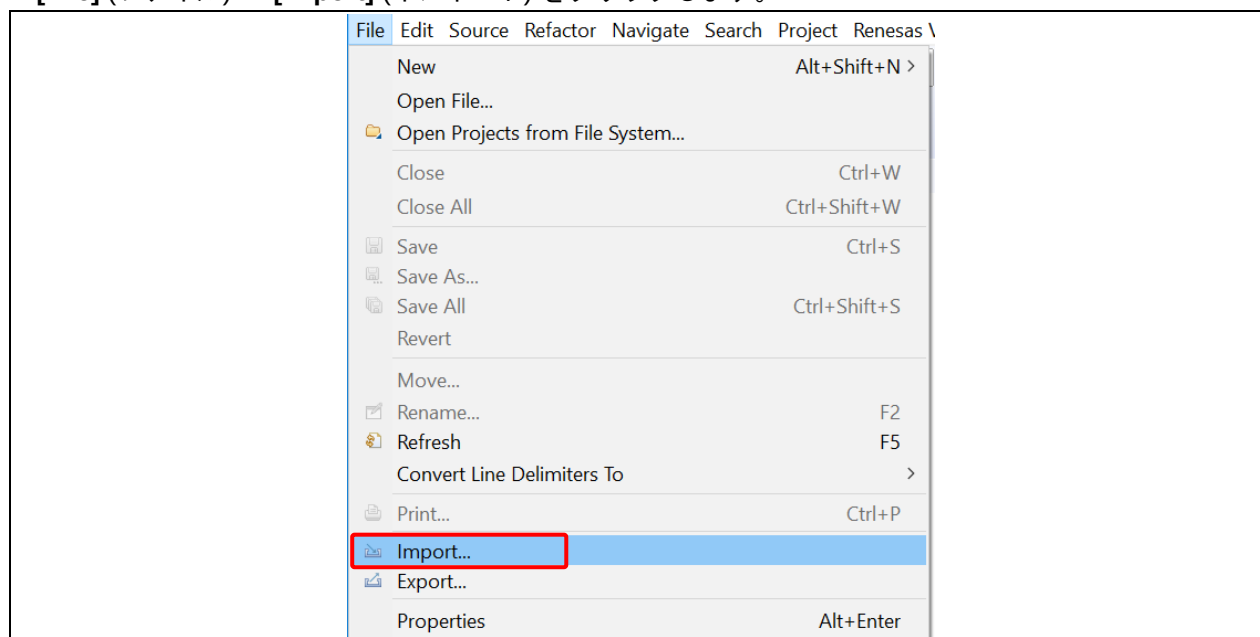


図 32. プロジェクトのインポート

2. **[Import]** (インポート) ダイアログで、**[General]** (一般) → **[Existing Projects into Workspace]** (既存のプロジェクトをワークスペースへ) を選択します。 **[Next]** (次へ) をクリックします。

注記：プロジェクトをインポートする際にプロジェクト名を変更するには、代わりに **[General]** (一般) → **[Rename & Import Existing Projects into Workspace]** を選択します。

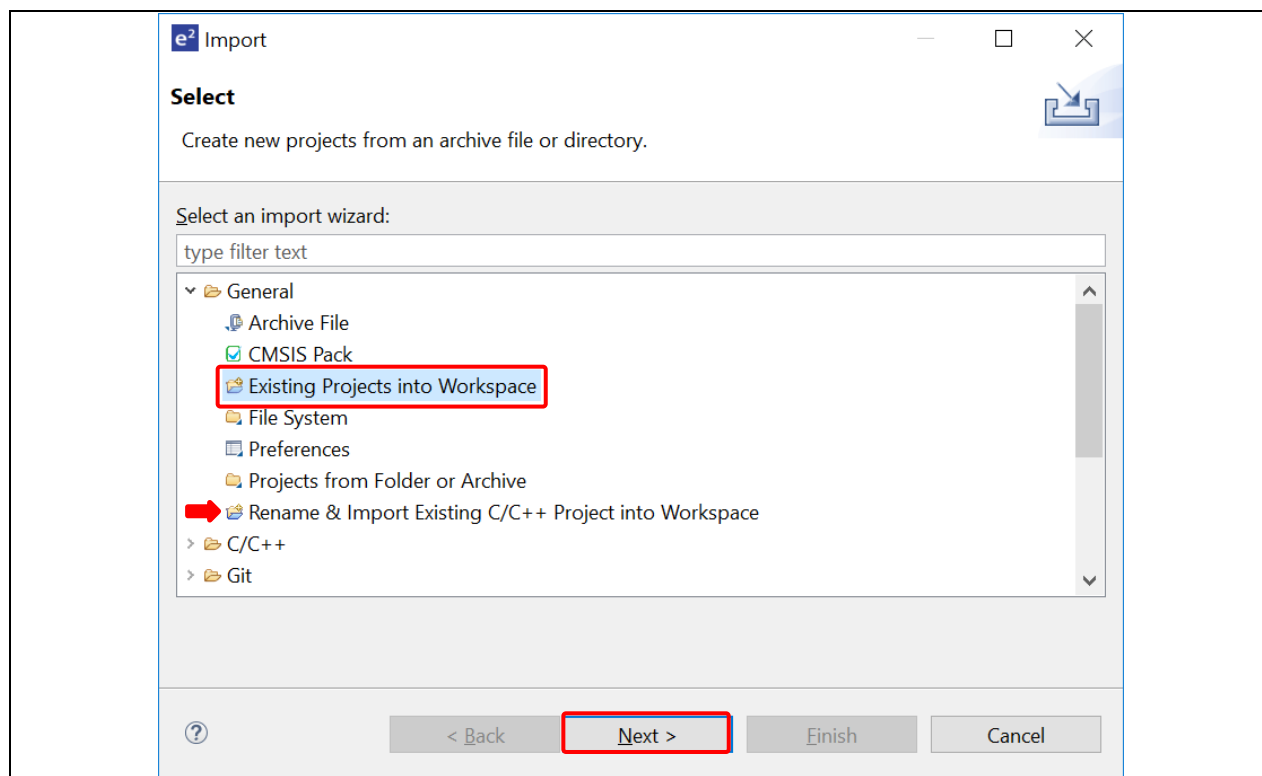


図 33. インポートの種類を選択

3. [Import Projects] (プロジェクトのインポート) ダイアログで、[Select archive file] (アーカイブファイルの選択) オプションをクリックした後、[Browse...] (参照...) をクリックし、プロジェクトを格納している圧縮ファイル (.zip) を見つけます。
既存のプロジェクトがフォルダ内に格納されている場合、[Select root directory] (ルートディレクトリの選択) オプションを選択します。

4. インポートするプロジェクトを選択し、[Finish] (完了) をクリックします。

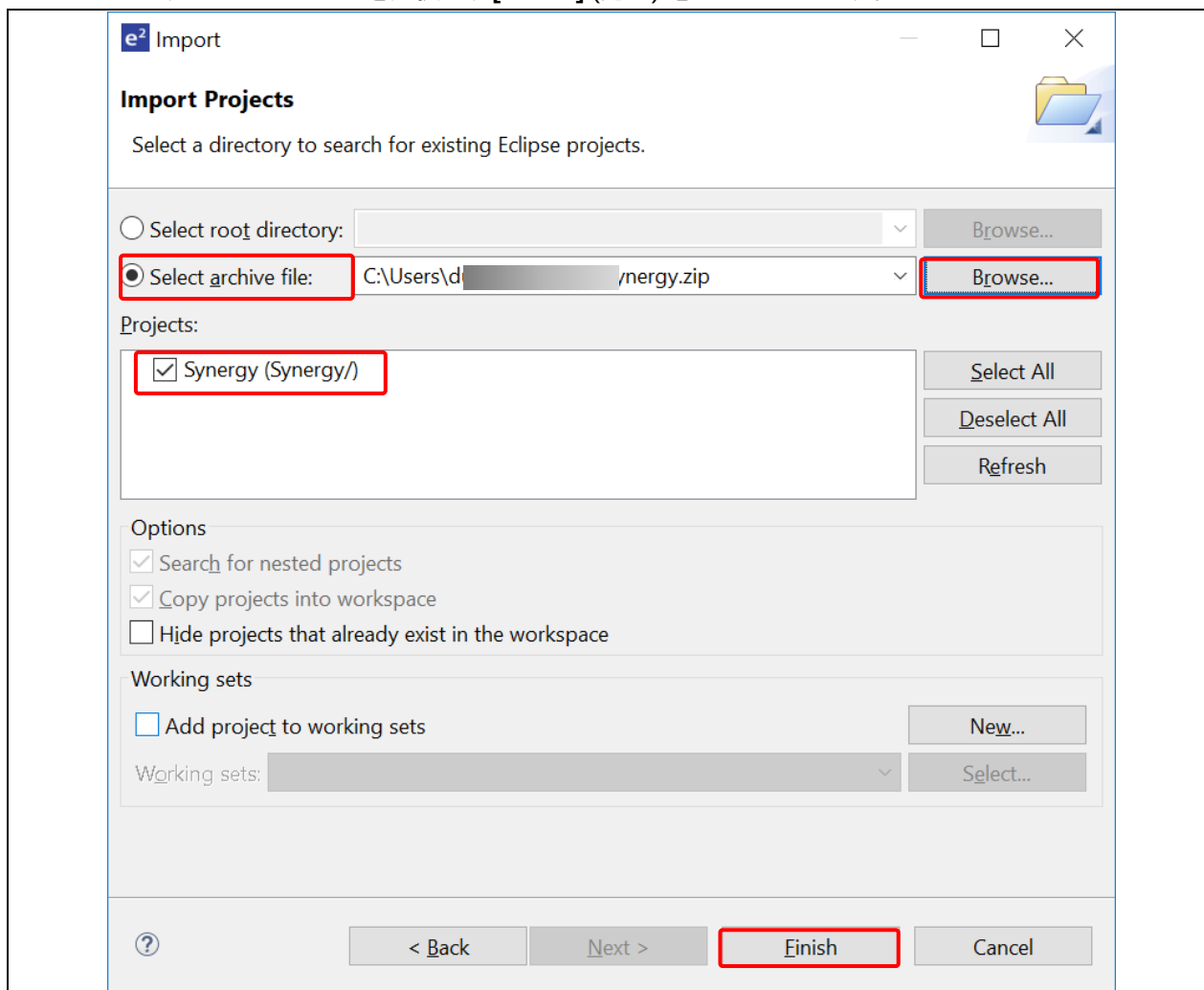


図 34. 圧縮ファイル内にあるプロジェクトの選択

5. プロジェクトが e² studio 内にインポートされます。

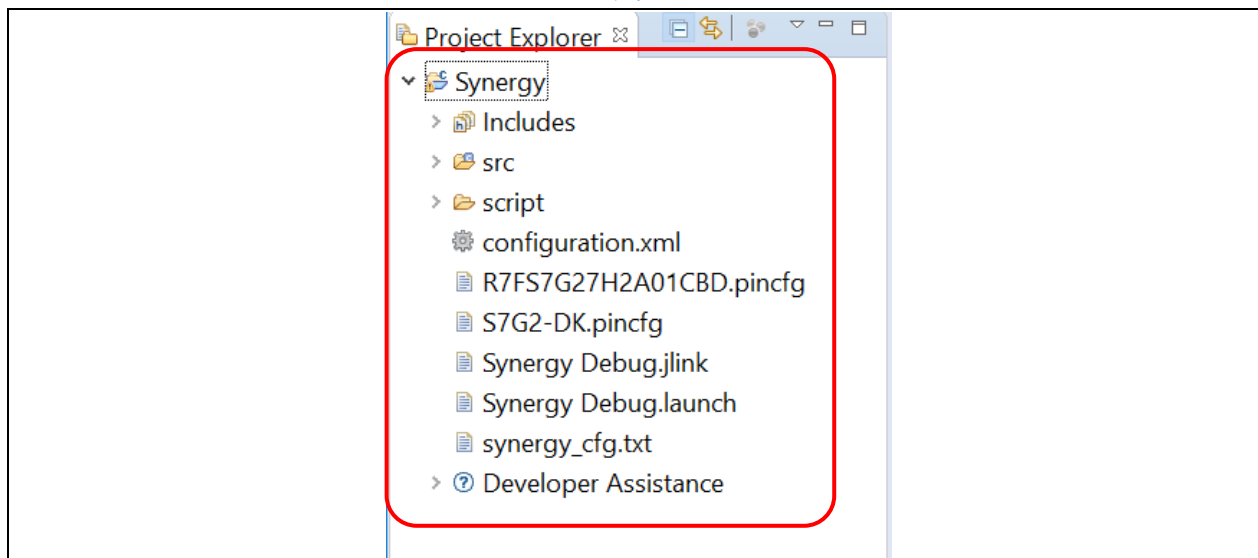


図 35. インポート済みプロジェクト

3.3 Synergy 静的ライブラリの生成と使用 (Generating and Using a Synergy Static Library)

この章では、Synergy 静的ライブラリプロジェクトと、ライブラリプロジェクトを参照する実行可能プロジェクトを生成する方法について説明します。

3.3.1 静的ライブラリプロジェクトの作成 (Creating the Static Library Project)

以下の手順で、Synergy 静的ライブラリプロジェクトを作成する方法の例を示します。

1. **[File]** (ファイル) → **[New]** (新規作成) → **[Synergy C/C++ Project]** を選択します。
2. **[Renesas Synergy C Library Project]** テンプレートを選択します。 **[Next]** (次へ) をクリックして続けます。

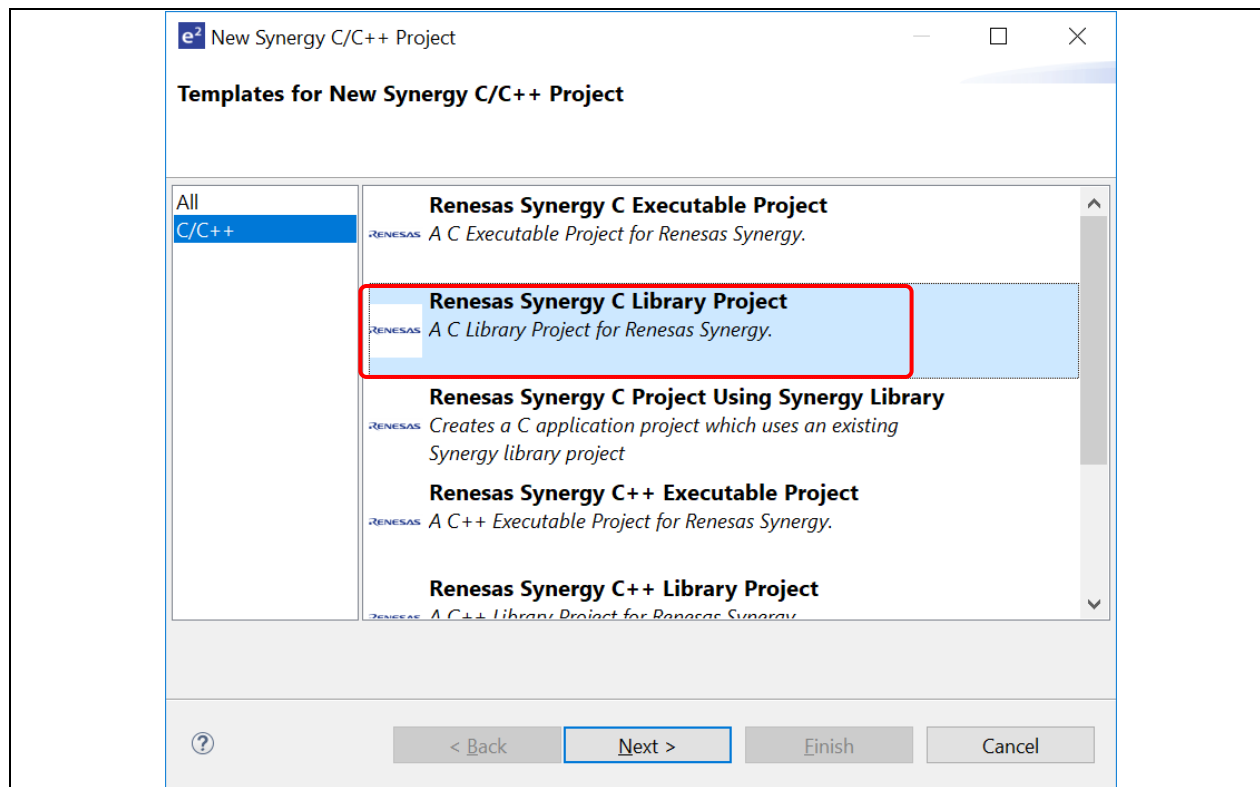


図 36. プロジェクトの生成 - ライブラリプロジェクトテンプレートの選択

3. プロジェクトの詳細ページで、静的ライブラリプロジェクトの名前 (synergylib など) を入力し、[Next] (次へ) をクリックします。

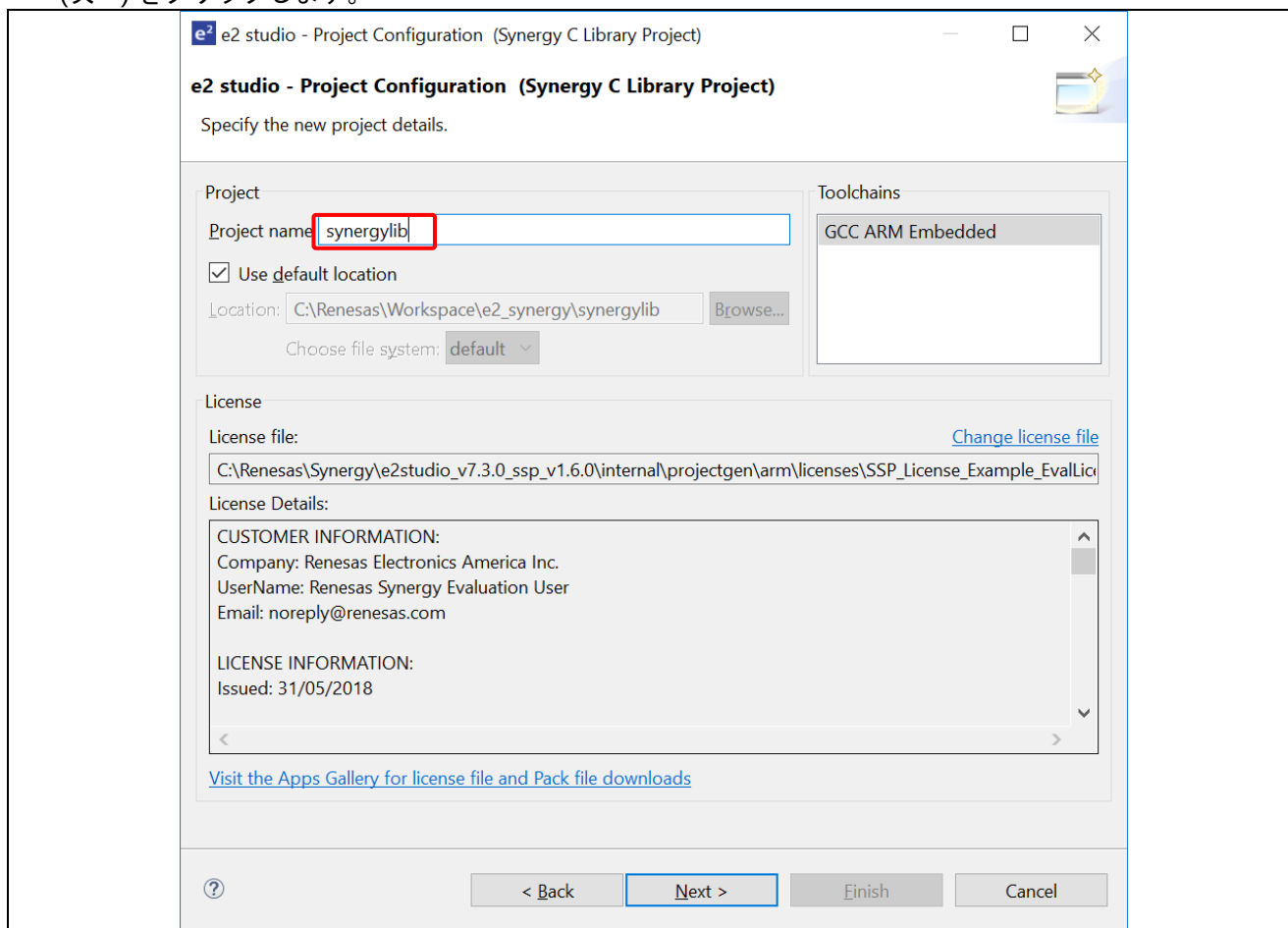


図 37. ライブラリプロジェクトの設定

4. デバイスとツールを選択するダイアログで、実行可能プロジェクトと同じデバイスおよび同じツールチェーンを選択し、[Next] (次へ) をクリックします。

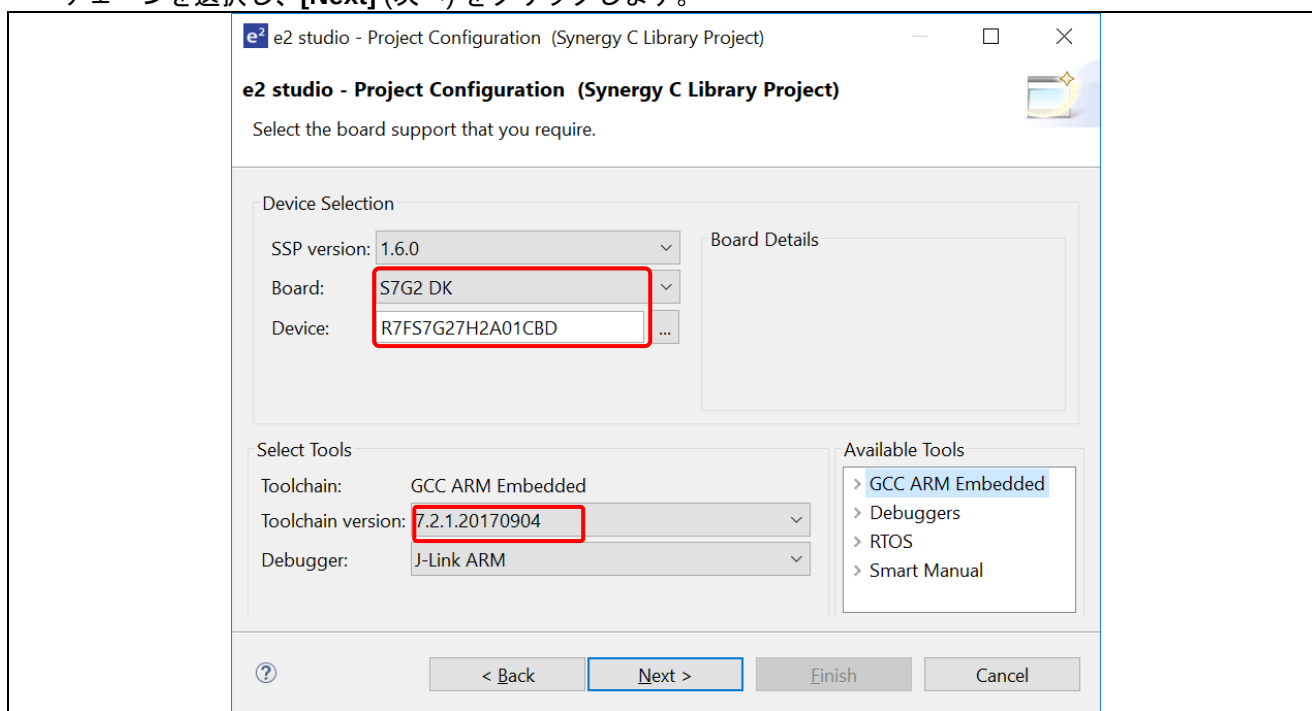


図 38. デバイスとツールチェーンの選択

5. プロジェクトテンプレートダイアログで、**[Blinky]** を選択します。**[Finish]** (完了) をクリックしてプロジェクトを作成します。

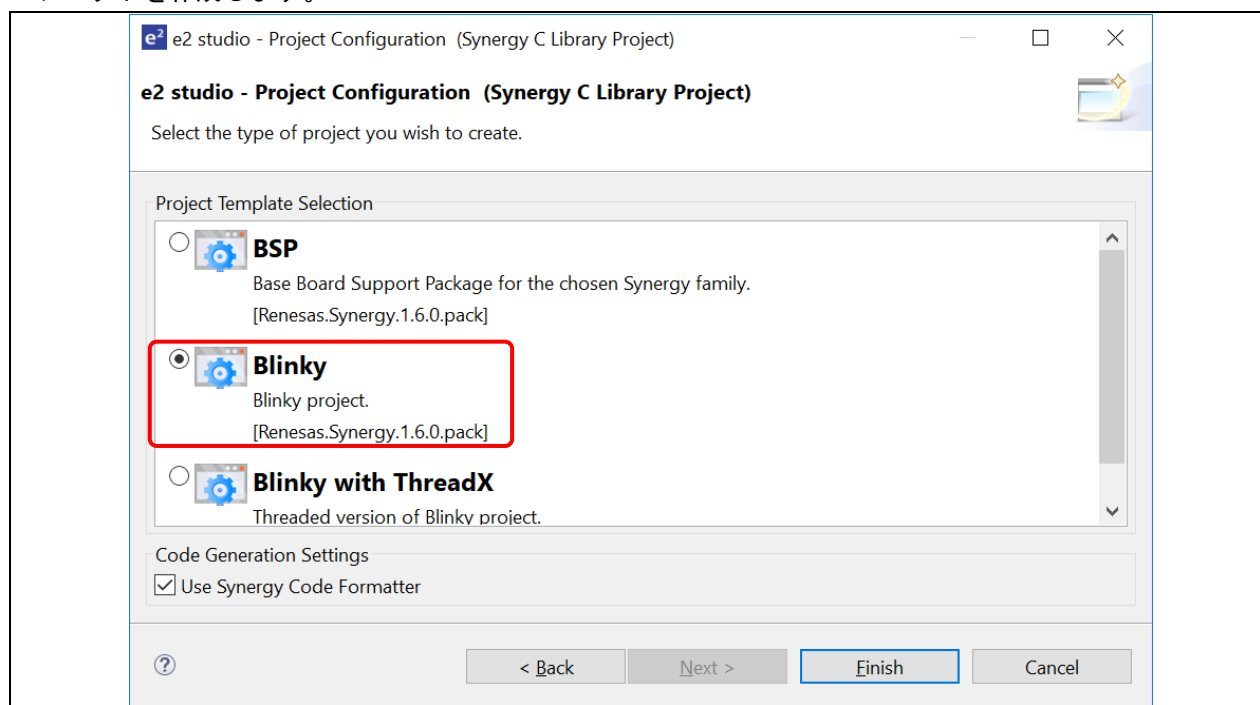


図 39. ライブラリに対応するプロジェクトテンプレートの選択

6. e² studio が、Synergy パースペクティブの切り替えをユーザに求めることがあります。その場合、**[Yes]** (はい) をクリックして、パースペクティブを開きます。
7. **[Generate Project Content]** (プロジェクトコンテンツの生成) をクリックします。

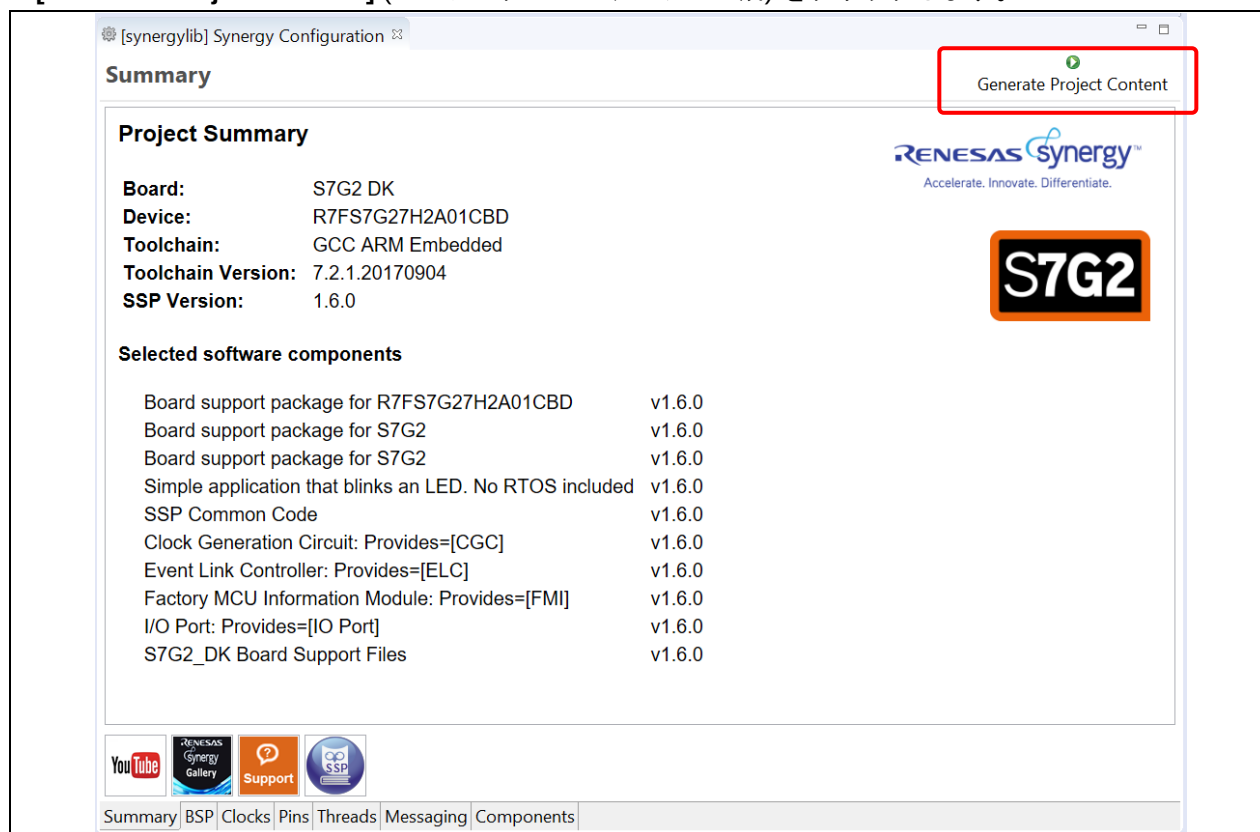
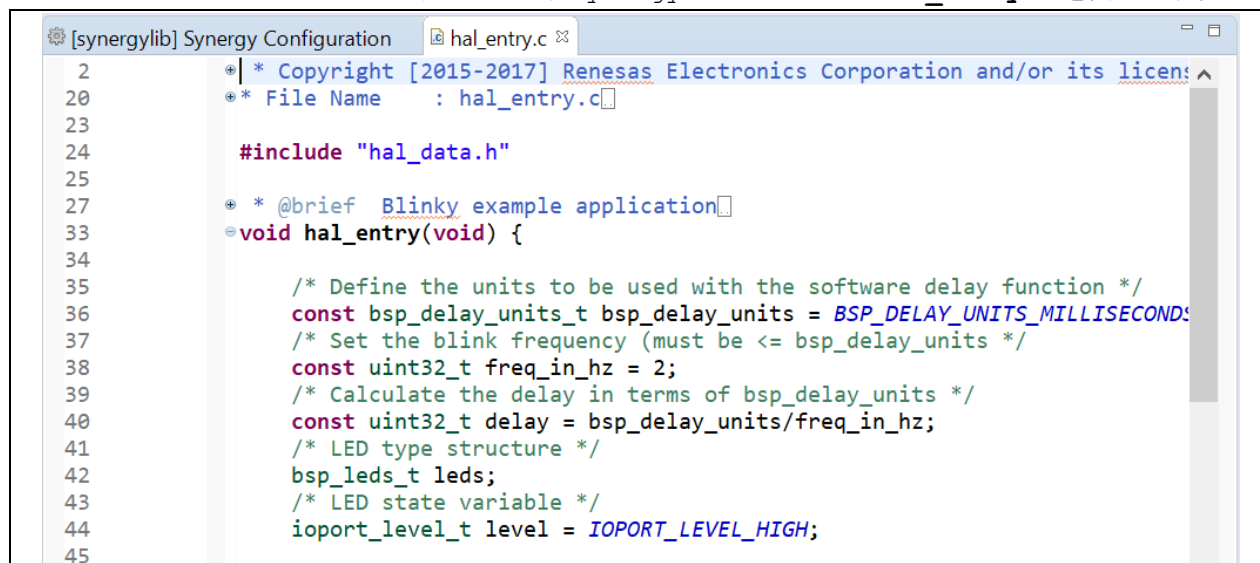


図 40. ライブラリプロジェクトコンテンツの生成

8. プロジェクトエクスプローラウィンドウで、synergylib\src\ 内の `hal_entry.c` を開きます。



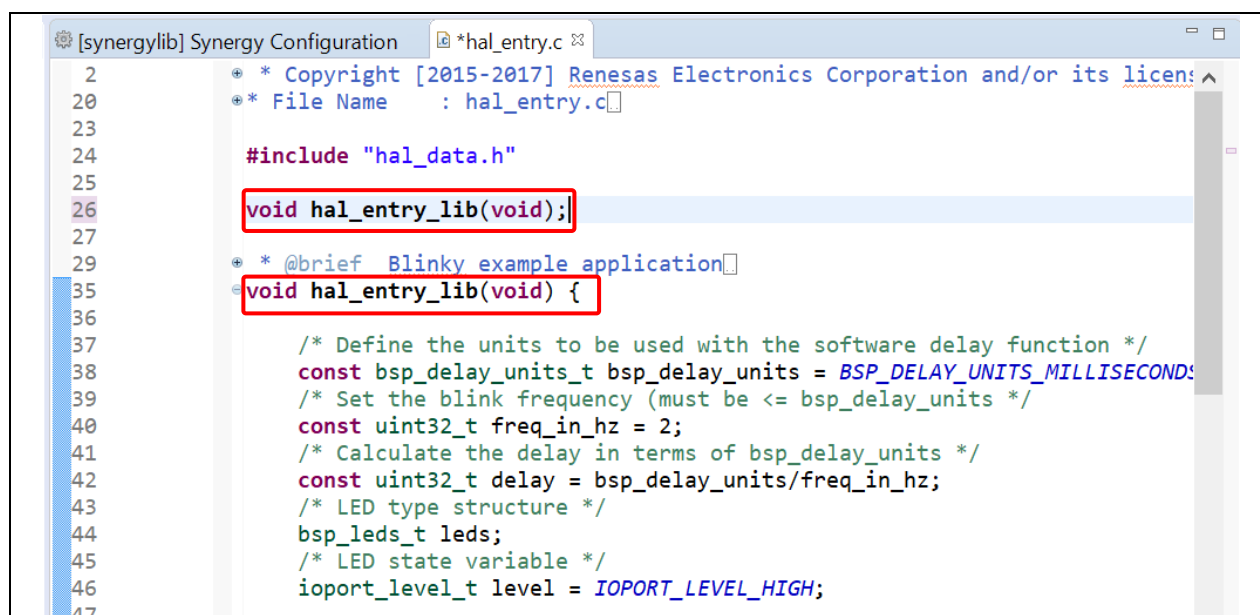
```

2      * Copyright [2015-2017] Renesas Electronics Corporation and/or its licensors
20     * File Name      : hal_entry.c
23
24     #include "hal_data.h"
25
26     * @brief Blinky example application
33     void hal_entry(void) {
34
35         /* Define the units to be used with the software delay function */
36         const bsp_delay_units_t bsp_delay_units = BSP_DELAY_UNITS_MILLISECONDS;
37         /* Set the blink frequency (must be <= bsp_delay_units) */
38         const uint32_t freq_in_hz = 2;
39         /* Calculate the delay in terms of bsp_delay_units */
40         const uint32_t delay = bsp_delay_units/freq_in_hz;
41         /* LED type structure */
42         bsp_leds_t leds;
43         /* LED state variable */
44         ioport_level_t level = IOPORT_LEVEL_HIGH;
45

```

図 41. 元の `hal_entry.c`

次に、`hal_entry()` 関数の名前を `hal_entry_lib()` に変更し、`hal_entry_lib()` に対応する宣言を追加します。



```

2      * Copyright [2015-2017] Renesas Electronics Corporation and/or its licensors
20     * File Name      : hal_entry.c
23
24     #include "hal_data.h"
25
26     void hal_entry_lib(void);
27
28     * @brief Blinky example application
35     void hal_entry_lib(void) {
36
37         /* Define the units to be used with the software delay function */
38         const bsp_delay_units_t bsp_delay_units = BSP_DELAY_UNITS_MILLISECONDS;
39         /* Set the blink frequency (must be <= bsp_delay_units) */
40         const uint32_t freq_in_hz = 2;
41         /* Calculate the delay in terms of bsp_delay_units */
42         const uint32_t delay = bsp_delay_units/freq_in_hz;
43         /* LED type structure */
44         bsp_leds_t leds;
45         /* LED state variable */
46         ioport_level_t level = IOPORT_LEVEL_HIGH;
47

```

図 42. 新しい `hal_entry.c`

9. ライブラリプロジェクトをビルドします。ビルドの結果、synergylib\Debug\libsynergylib.a という静的ライブラリファイルが出力されます。

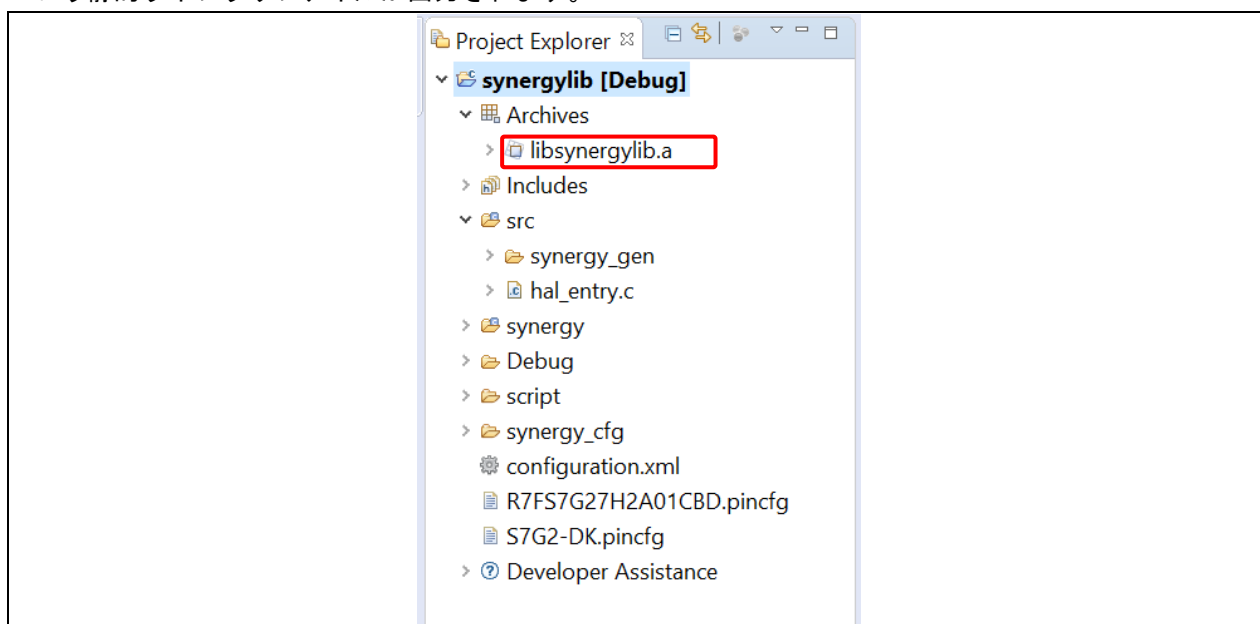


図 43. ビルド済みの静的ライブラリ

3.3.2 実行可能プロジェクト内での静的ライブラリプロジェクトの使用 (Using the Static Library in the Executable Project)

この章では以下の手順を実行し、Synergy 実行可能プロジェクトの中で、3.3.1 章で作成した静的ライブラリを使用する方法を示します。

- Synergy 実行可能プロジェクトを作成します。
- 静的ライブラリプロジェクト内で宣言した関数 (`hal_entry_lib()`) を呼び出すように、ソースコードを変更します。
- 静的ライブラリを追加するように、ビルド設定を変更します。
- Synergy 実行可能プロジェクトをビルドします。

以下の手順を適用します。

1. **synergyapp** という実行可能プロジェクトを作成し、**[BSP]** テンプレートを選択します。**[Finish]** (完了) をクリックしてプロジェクトを作成します。実行可能プロジェクトの作成に関する詳細は、3.1 章を参照してください。

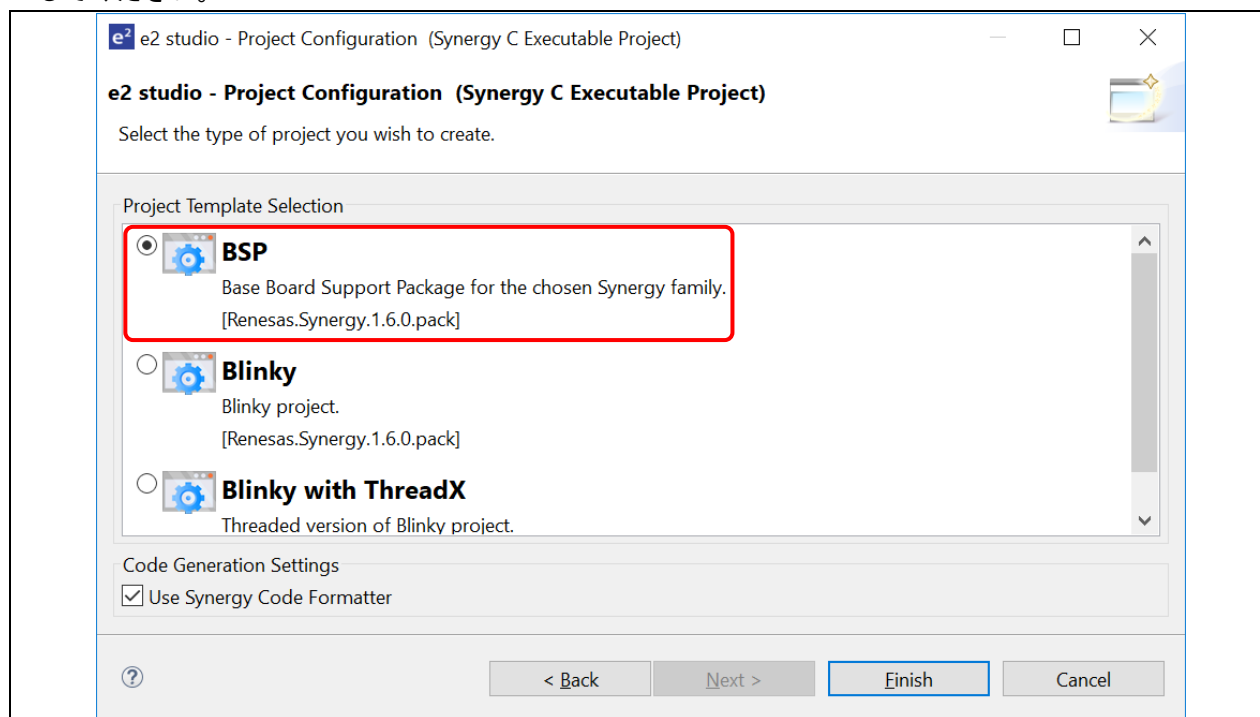


図 44. ライブラリを参照する実行可能プロジェクトに対応するプロジェクトテンプレートの選択

2. 実行可能プロジェクトを作成した後、**[Generate Project Content]** (プロジェクトコンテンツの生成) をクリックします。
3. プロジェクトエクスプローラウィンドウで、`synergylib\src\` 内の `hal_entry.c` を開きます。

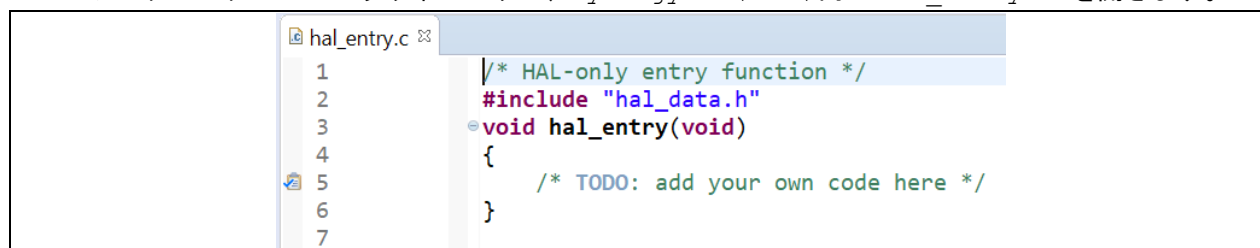


図 45. 元の hal_entry.c

`hal_entry()` 関数の中で、LED 点滅ライブラリ関数 `hal_entry_lib()` を呼び出すコードを追加し、このライブラリ関数に関する宣言を追加します。

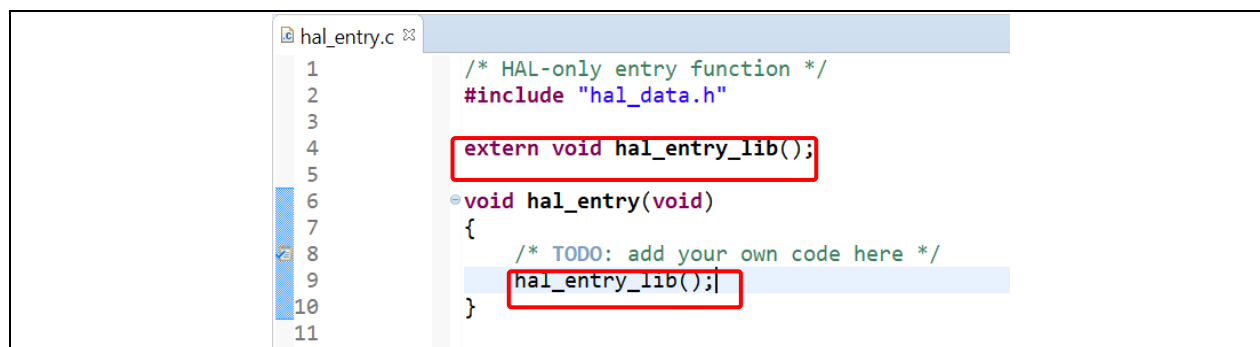


図 46. 新しい hal_entry.c

4. **[synergyapp]** プロジェクトを右クリックした後、**[Properties]** (プロパティ) を選択します。
5. **[Project Properties]** (プロジェクトのプロパティ) ダイアログで、**[C/C++ Build]** (C/C++ のビルド) → **[Settings]** (設定) を選択した後、**[Tool Settings]** (ツール設定) タブを選択します。**[GNU ARM Cross C Linker]** (GNU ARM クロス C リンカ) → **[Libraries]** (ライブラリ) を選択した後、ライブラリとして **[synergylib]** を追加します。

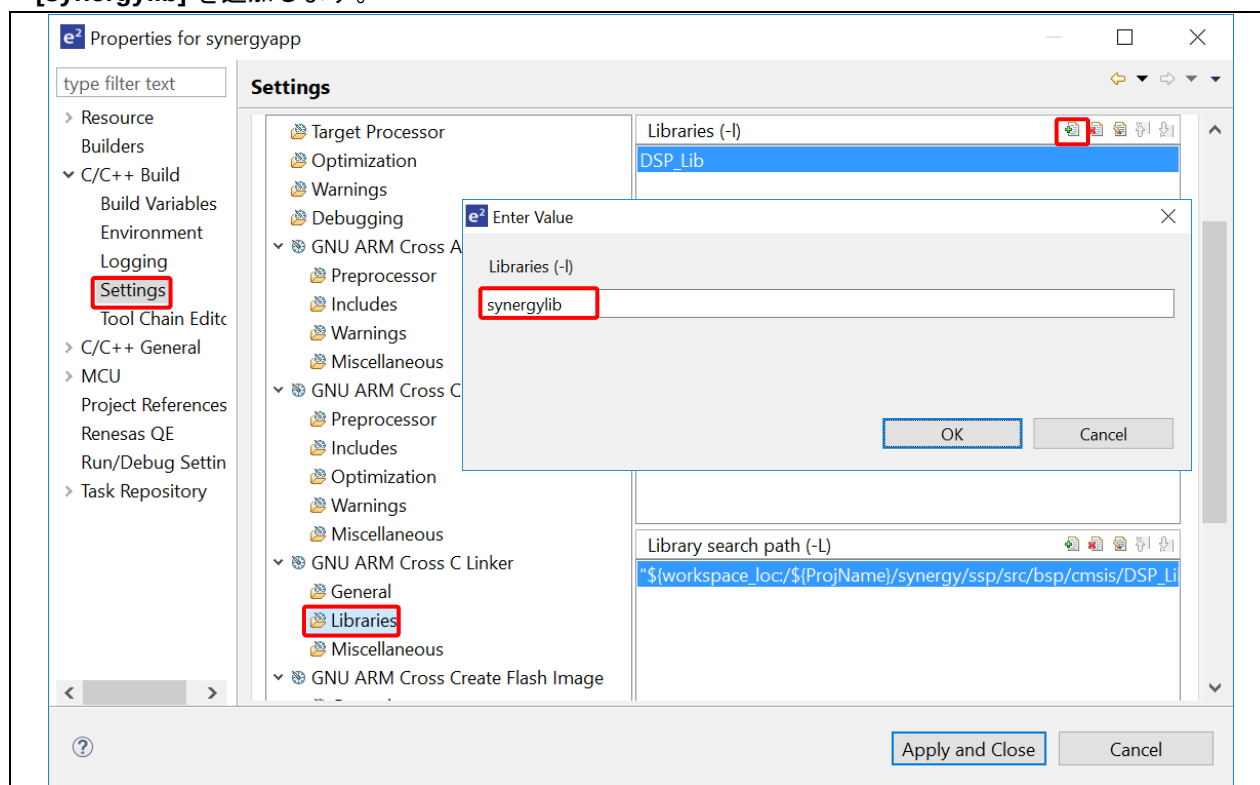


図 47. ライブラリプロジェクトを、実行可能プロジェクトに対応するライブラリとして追加

6. **[GNU ARM Cross C Linker]** (GNU ARM クロス C リンカ) → **[Libraries]** (ライブラリ) で、新しいライブラリ検索パスとして `${workspace_loc:/synergylib/Debug}` を追加します。

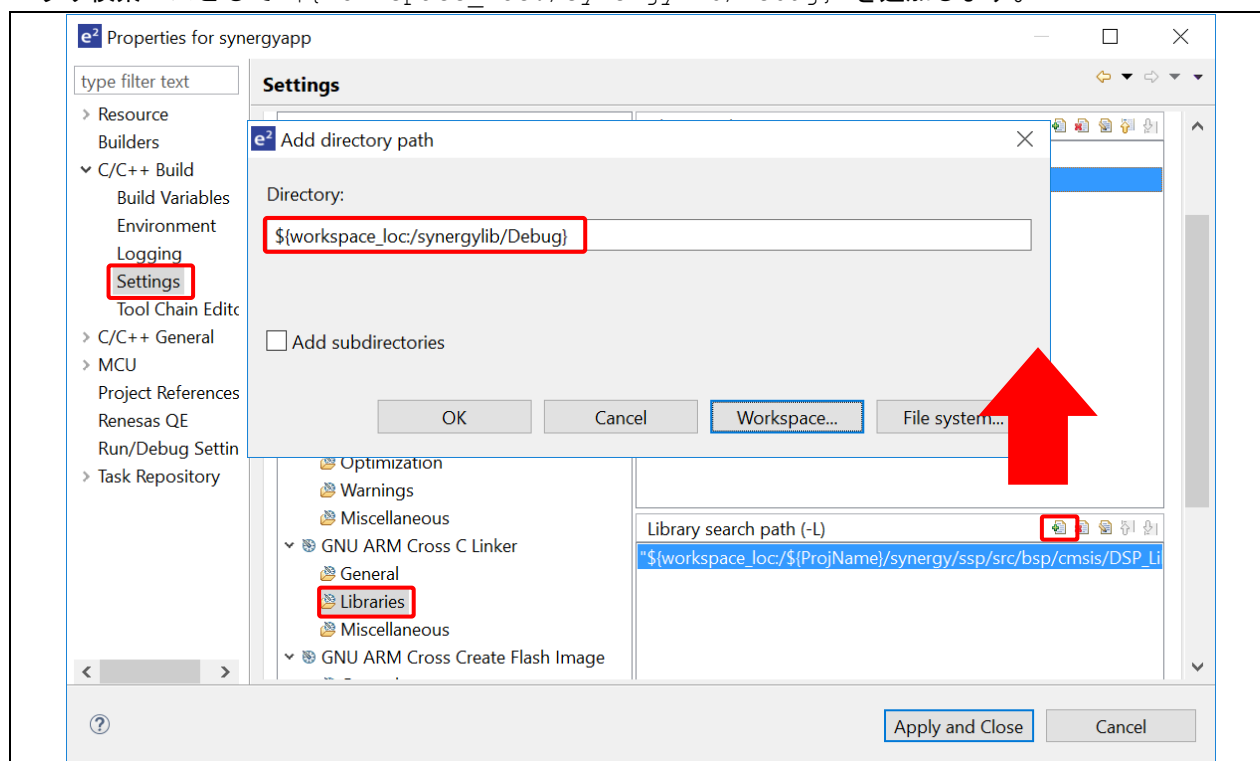


図 48. ライブラリ検索パスの更新

7. **[GNU ARM Cross C Linker]** (GNU ARM クロス C リンカ) → **[Libraries]** (ライブラリ) の下で、新しいパスに更新されていることを確認します。

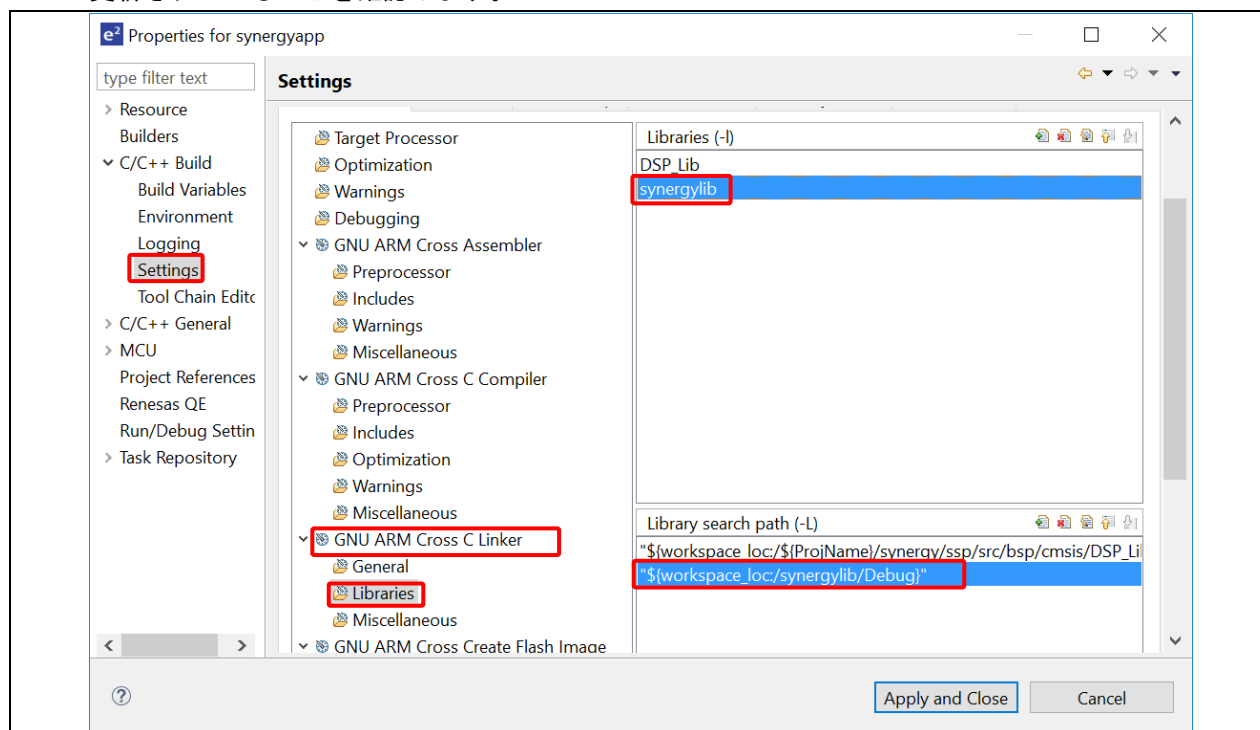


図 49. GNU ARM クロス C リンカのライブラリは既に更新済み

注記：直前の図で表示されている **[Libraries (-l)]** (ライブラリ) オプションは、`libxxx.a` というファイル名形式のライブラリファイルのみを追加する目的で使います。ライブラリ名の形式が `libxxx.a` ではない場合、**[GNU ARM Cross C Linker]** (GNU ARM クロス C リンカ) → **[Miscellaneous]** (その他) 設定を使用してそのライブラリを追加します。

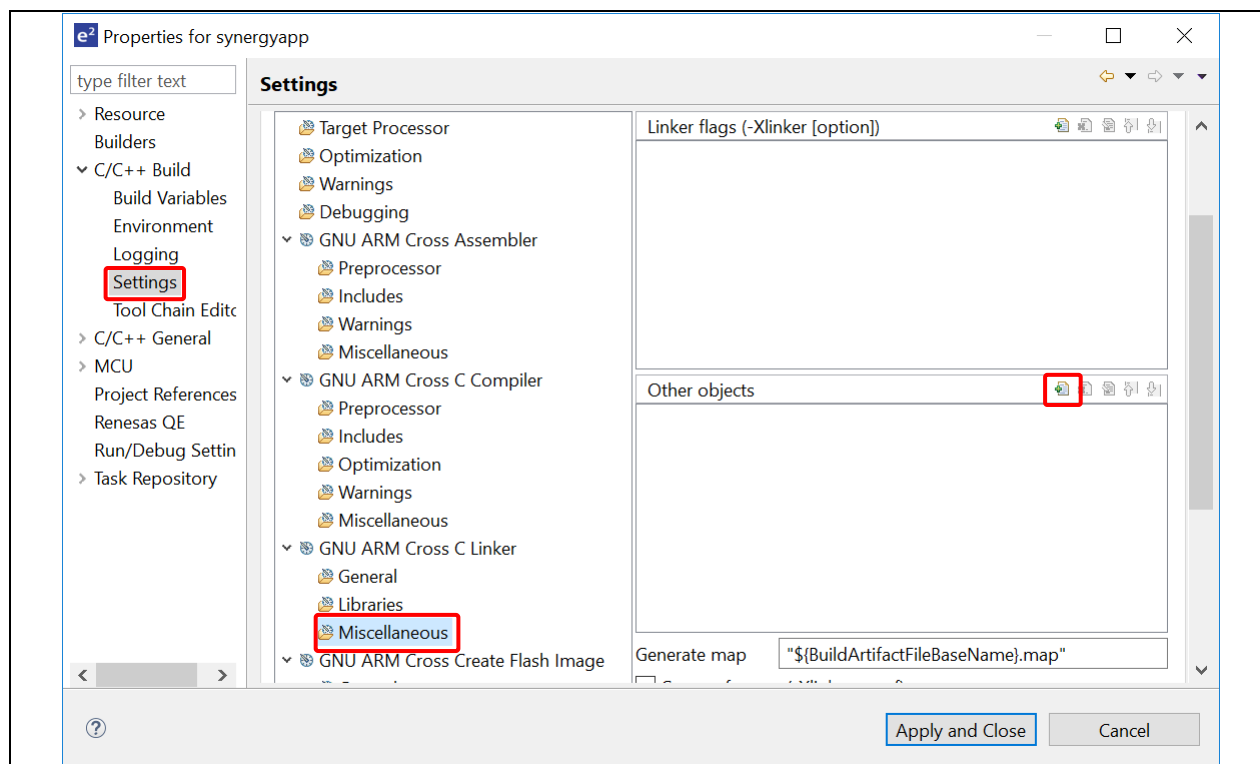


図 50. libxxx.a という形式の名前を使用していないライブラリの追加

8. [Properties] (プロパティ) ダイアログの左側のペインで、[Project References] (プロジェクト参照) を選択した後、[synergylib] チェックボックスをオンにし、その実行可能プロジェクトが静的ライブラリプロジェクトに依存しているというマークを付けます。[Apply and Close] (適用して閉じる) をクリックします。

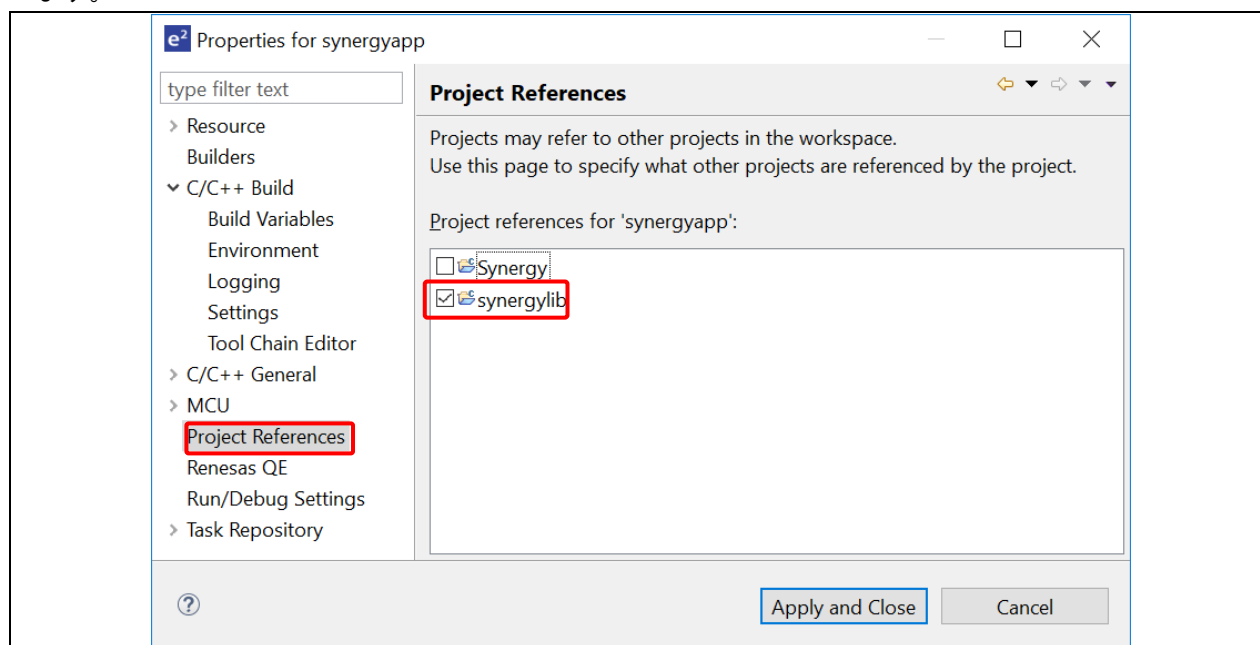


図 51. 実行可能プロジェクトが静的ライブラリプロジェクトに依存しているというマークを付ける

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()` を呼び出している場所にブレークポイントを設定します。
[synergyapp] 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が停止した時点で、そのプログラムを再開します。複数の LED を点滅させるライブラリ関数 (`hal_entry_lib()`) が実行されることを確認します。

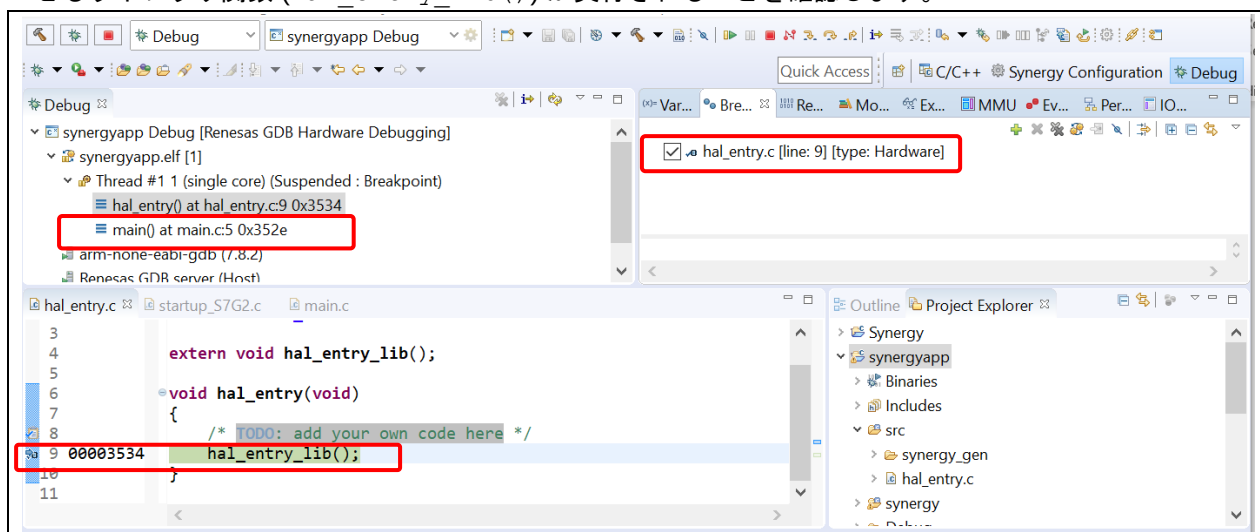


図 52. ライブラリ関数を実行するアプリケーションプロジェクト

3.4 Synergy Project Configuration Editor

[Synergy Project Configuration Editor] ビューで、現在のプロジェクト構成に関する設定が表示されます。これらの設定は、configuration.xml ファイルに保存されます。プロジェクト構成の設定は複数のページにグループ化されており、端子やクロックを設定する方法や、どのドライバをインクルードするかなど、プロジェクトに関して構成可能な複数の側面を設定することができます。ドライバは、シンプルなハードウェアレベルドライバから、RTOS 対応アプリケーションまで広い範囲にわたっています。ミューテックス、セマフォ、イベントなど、マルチスレッド固有のコンポーネントを構成することもできます。

プロジェクト構成を編集する場合、以下のことを確認してください。

- e² studio ウィンドウの右上隅の、[Synergy Configuration] パースペクティブを選択するか、[Window] (ウィンドウ) → [Perspective] (パースペクティブ) → [Open Perspective] (パースペクティブを開く) → [Other...] (その他...) → [Synergy Configuration] (Synergy 設定) をクリックします。
- 「configuration.xml」ファイルを開きます。



図 53. Synergy Project Configuration - [Synergy Project Configuration] (Synergy プロジェクト設定) ビュー

Synergy Project Configuration Editor 内に 7 つのページ（またはタブ）があります。

[Summary] (概要) ページには、プロジェクト固有の概要情報が表示されます。

[BSP] ページで、ユーザが SSP のバージョン、Synergy ボードのタイプ、デバイスを選択できます。

さらに、[Clocks] (クロック)、[Pins] (端子)、[Threads] (スレッド)、[Messaging] (メッセージング)、[Components] (コンポーネント) の各ページの詳細について、構成のステップとオプションも含め、以下の章で順に説明します。

3.4.1 Summary (概要) ページ

[Summary] (概要) ページには、プロジェクト固有の概要として、現在選択されているデバイスとボードの詳細、Synergy ソフトウェアコンポーネント、その他の詳細が表示されます。Synergy Platform Web サイト、「Renesas Presents」YouTube チャンネル、SSP ユーザーズマニュアルへの役立つリンクも表示されます。

ユーザが新しいスレッドを追加した場合や、モジュール/オブジェクトをスレッドに追加した場合、その情報も [Summary] (概要) ページに表示されます。

The screenshot shows the 'Summary' page of the Synergy Configuration tool. It contains several sections: Project Summary, Interrupts Summary, Threads Summary, and Selected software components. Red boxes and arrows highlight specific information:

- Project Summary:** A box highlights the Board (S7G2 DK), Device (R7FS7G27H2A01CBD), Toolchain (GCC ARM Embedded), Toolchain Version (7.2.1.20170904), and SSP Version (1.6.0). An arrow points to this box with the text: 'ボード、デバイス、ツールチェーン、SSP に関する情報'.
- Interrupts Summary:** A table shows interrupt properties. A box highlights the 'g_cac0 Clock Accuracy Circuit Driver on r_cac' section, with an arrow pointing to it from the text: 'スレッドと割り込みに関する情報'.
- Threads Summary:** A table shows thread properties. A box highlights the 'HMI Thread' with priority '1'.
- Selected software components:** A list of components and their versions. A box highlights the list, with an arrow pointing to it from the text: 'プロジェクトで使われているソフトウェアコンポーネント'.
- Footer:** A row of icons for YouTube, Synergy Gallery, Support, and SSP, with an arrow pointing to them from the text: '役立つリンク'.

At the bottom of the page, there is a navigation bar with links: Summary, BSP, Clocks, Pins, Threads, Messaging, Components.

図 54. Summary (概要) ページ

3.4.2 [BSP] ページ

[BSP] ページで、ユーザが SSP のバージョン、ボード、デバイスを選択できます。ユーザはこのページで、CMSIS パックをインポートすることもできます。

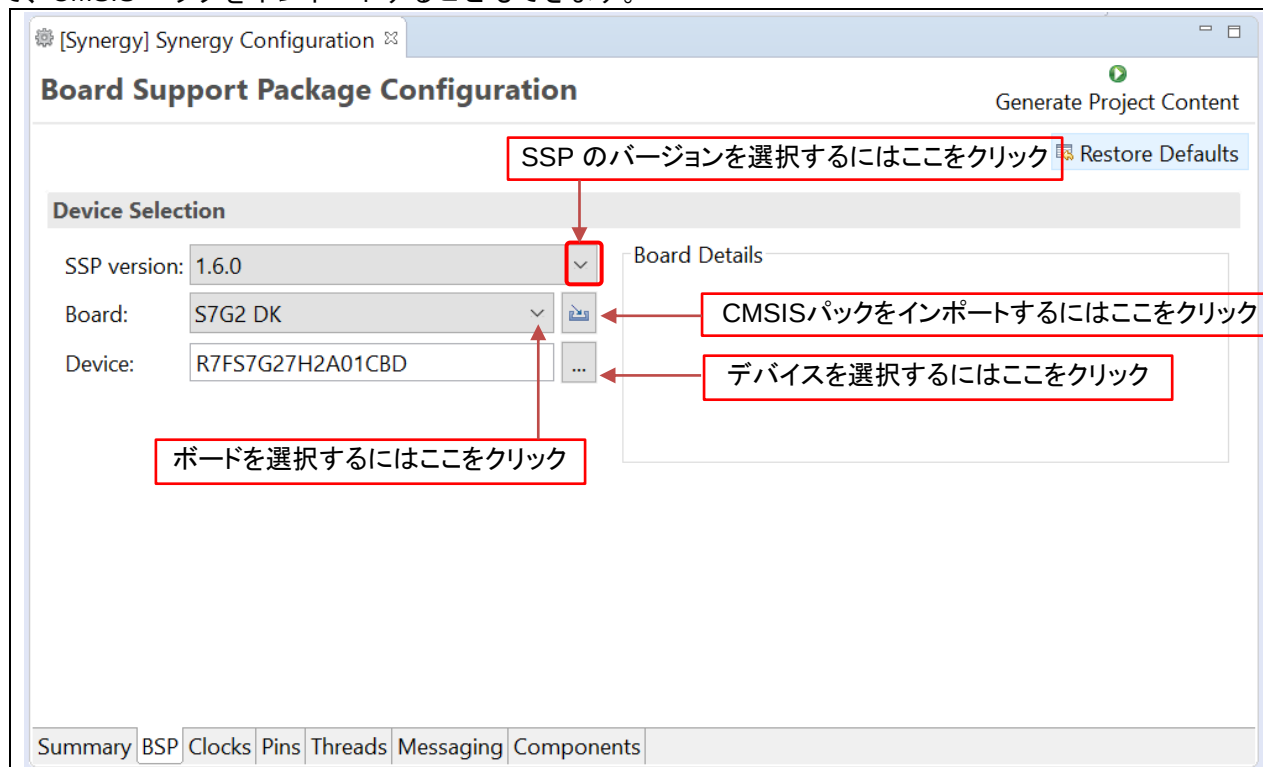


図 55. Synergy Project Configuration (Synergy プロジェクト設定) - [BSP] ページ

3.4.3 [Clocks Configuration] (クロック設定) ページ

[Clocks Configuration] (クロック設定) ページで、アプリケーションで使用する初期クロックを設定します。クロックソース、PLL 設定、クロックディバイダの設定を、出力クロックごとに選択できます。

クロック生成回路 (CGC) の詳細については、『Synergy MCU ユーザーズマニュアル』を参照してください。プロジェクトを更新するには、以下の手順に従ってください。

1. GUI を使用し、クロック設定に対応するプルダウンメニューでいずれかの値を選択します。

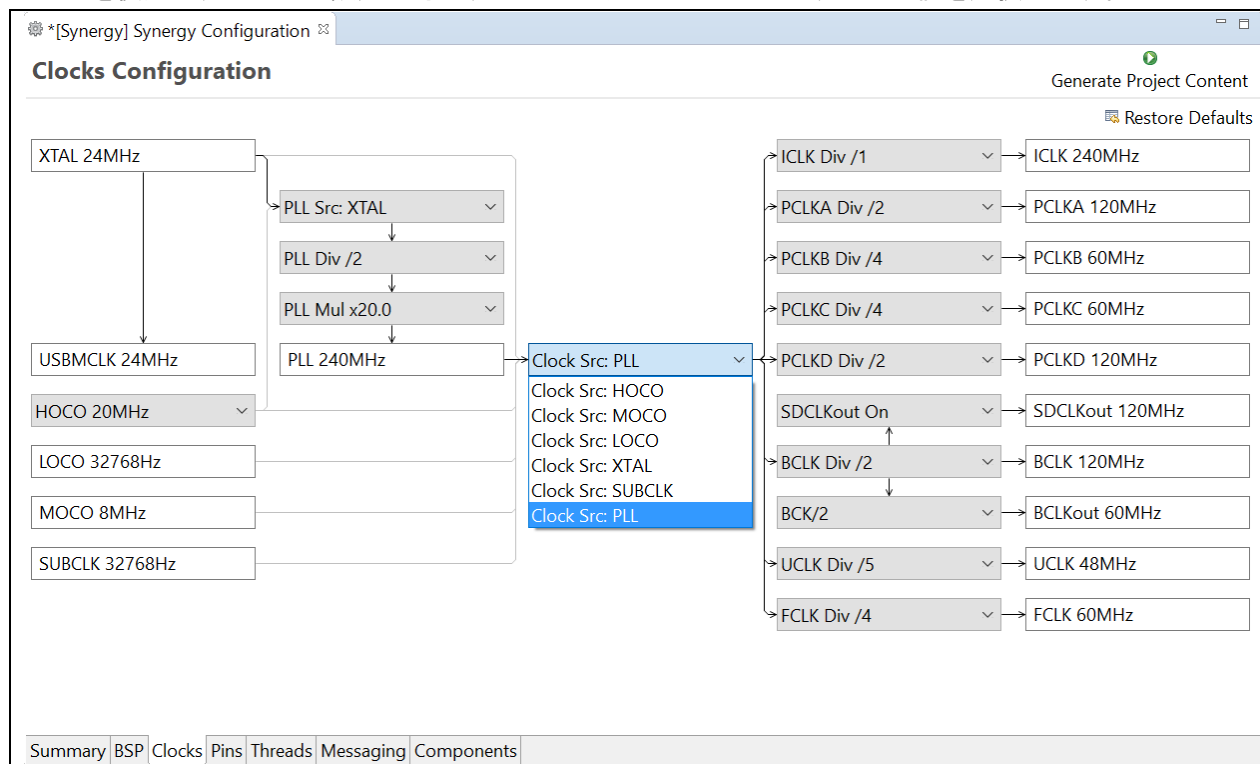


図 56. Synergy Project Configuration (Synergy プロジェクト設定) - Clocks Configuration (クロック設定)

2. Ctrl-S ショートカットを使用して、例えば、Project Configuration Settings (プロジェクト設定項目) を保存します。
3. **[Generate Project Content]** (プロジェクトコンテンツの生成) ボタン **Generate Project Content** をクリックします。
4. 選択したクロック設定を使用して、bsp_clock_cfg.h ファイルが更新されます。

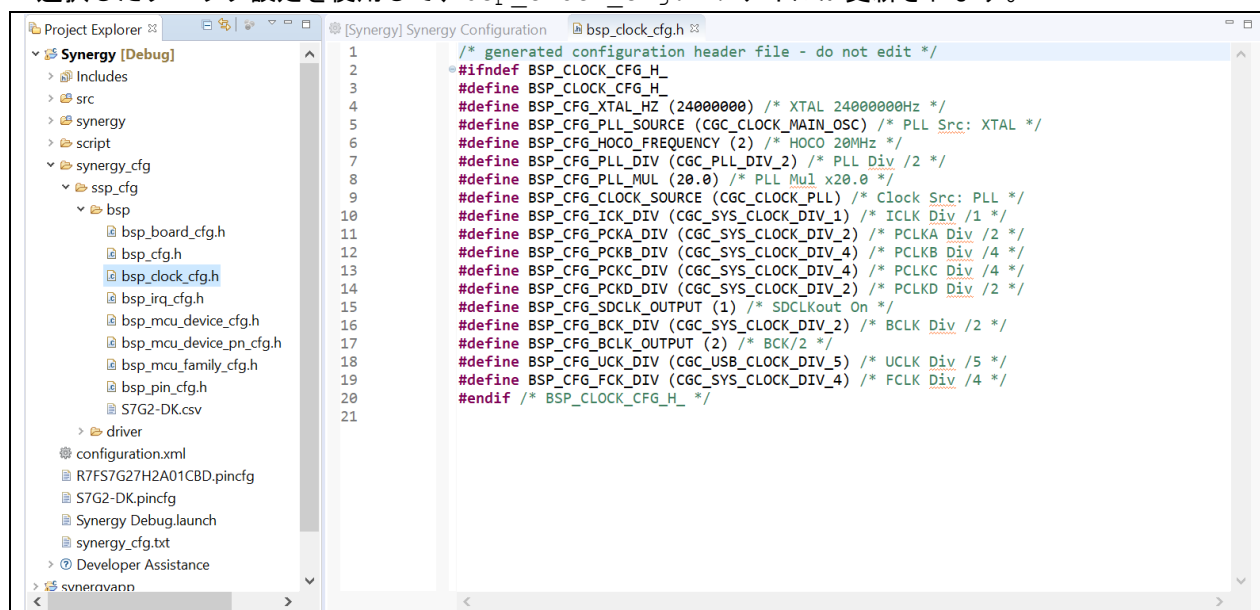


図 57. 更新済みの bsp_clock_cfg.h

3.4.4 [Pin Configuration] (端子設定) ページ

[Pin Configuration] (端子設定) ページには、プロジェクトの端子設定を生成するためのグラフィカルユーザインタフェース (GUI) があります。

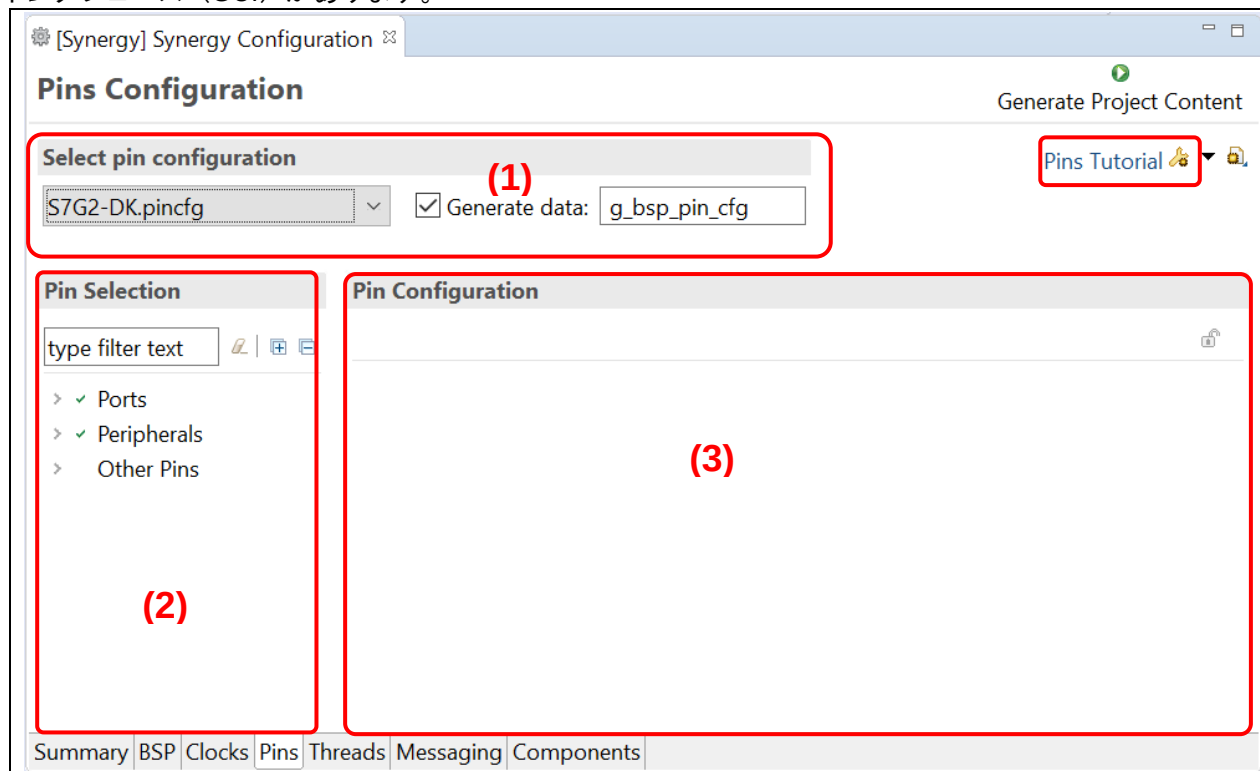


図 58. Synergy Project Configuration (Synergy プロジェクト設定) - Pin Configuration (端子設定) GUI

[Pin Configuration] (端子設定) ウィンドウには、3つのパートがあります。

1. **[Select Pin Configuration] (端子設定の選択)** : pin-configuration ファイルを選択し、関連付けるデータ構造名を指定します。複数の端子設定を、以下の方法で設定することもできます。
 - A. Project Explorer (プロジェクトエクスプローラ) で、既存のファイルをコピーして、新しい.pincfg ファイル (例えば、NewName.pincfg) を作成します。
 - B. **[Select Pin Configuration]** (端子設定の選択) ダイアログボックスで、新しい.pincfg ファイル (例えば、NewName.pincfg) を選択します。
 - C. **[Generate data]** (データの生成) チェックボックスをオンにし、テキストフィールドで、新しい端子設定に対して固有のデータ構造名を割り当てます。
 - D. 複数の端子設定は、異なるデータ構造で作成されます。
2. **[Pin Selection] (端子選択)** : 設定しようとする端子または周辺回路 (peripheral) を選択します。
3. **[Pin Configuration] (端子設定)** : 選択した端子/周辺回路の機能/プロパティを設定します。
[Pin Tutorial] (端子チュートリアル) をクリックして、端子設定に関するチュートリアルビデオを YouTube で参照することもできます。

以下の手順を使用して、プロジェクトで使用する周辺回路を構成すると、複数の端子を設定することができます。

1. **[Pin Selection]** (端子の選択) ペインで、いずれかの周辺回路を選択します。例えば、**[Connectivity:SCI] → [SCI1]**。この周辺回路に関する設定は、**[Pin Configuration]** (端子設定) ペインで表示されます。
2. 周辺回路で使用する **[Operation Mode]** (動作モード)、例えば、**[Simple SPI]** (簡易 SPI) を選択します。
3. 選択した周辺回路の選択したモードで、**[Input/Output]** (入出力機能) に使用する端子を選択します。

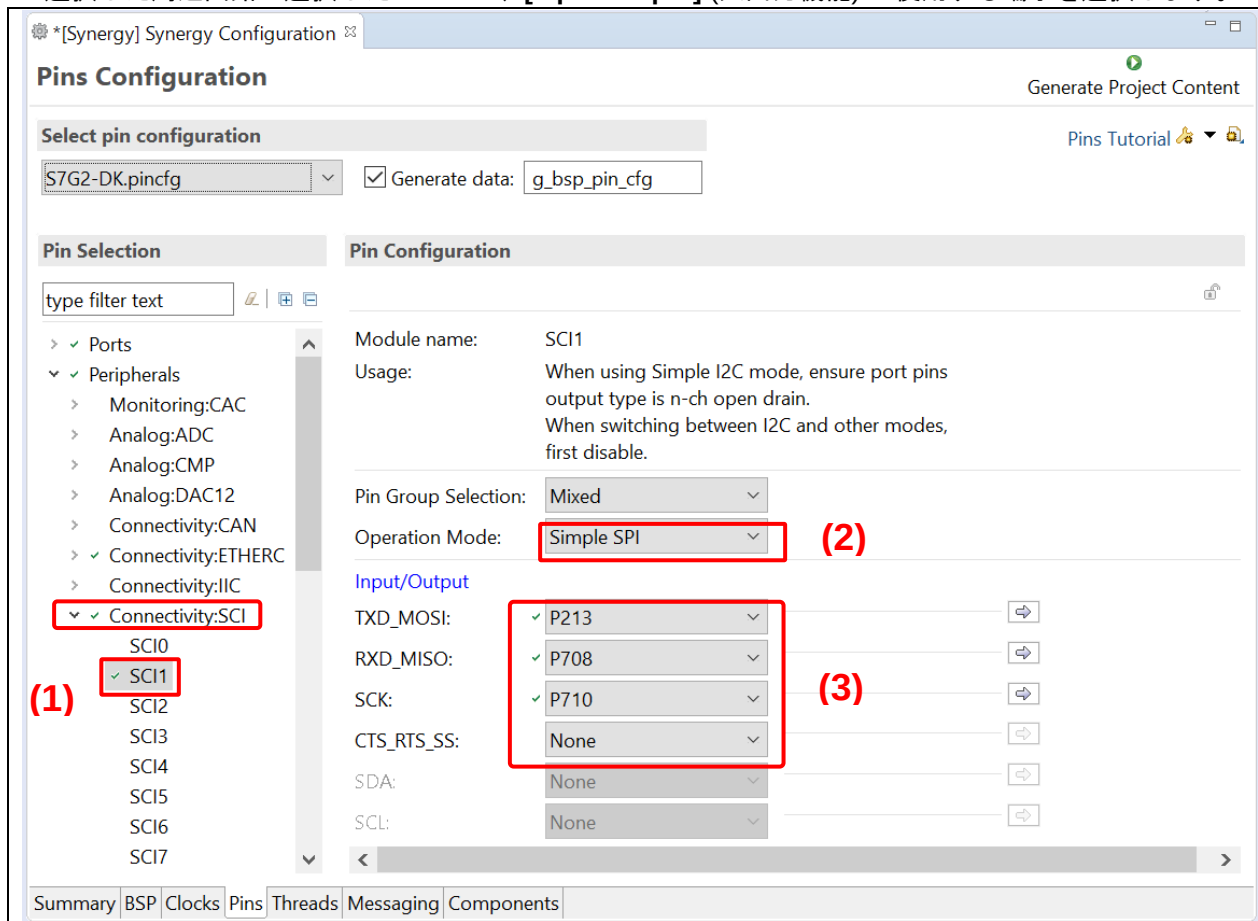


図 59. Synergy Project Configuration (Synergy プロジェクト設定) - Pin Configuration (端子設定) (周辺回路ごと)

1 本の端子を以下の手順で設定することもできます。

1. **[Pin Selection]** (端子の選択) ペインで、いずれかの端子を選択します。例えば、**[Ports]** (ポート) → **[P0]** → **[P003]**。この端子に関する設定は、**[Pin Configuration]** (端子の設定) ペインで表示されます。
2. この端子に関するプロパティを入力します。例：

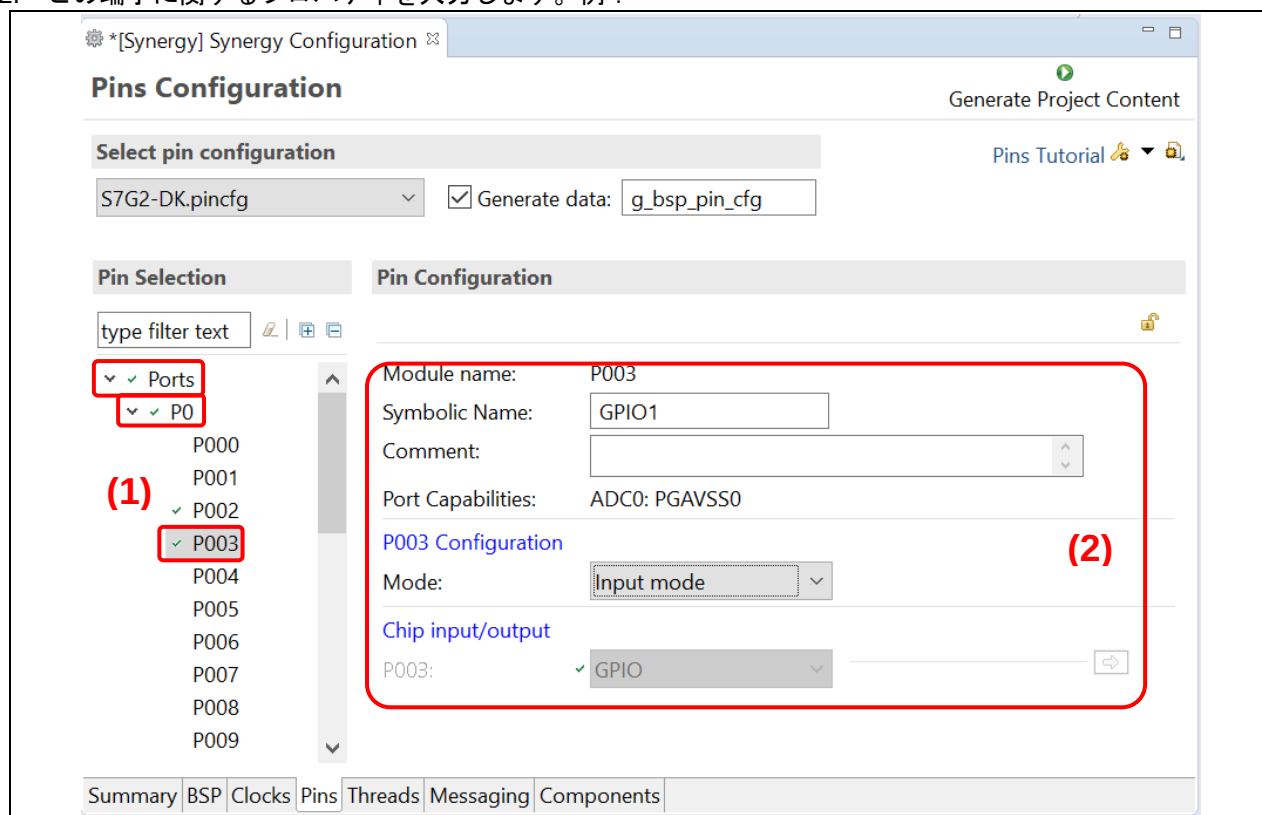


図 60. Synergy Project Configuration (Synergy プロジェクト設定) - Pin Configuration (端子設定)
(1 本のピンごと)

3. [Package] (パッケージ) ビューで、端子が変更されたことを表示します。

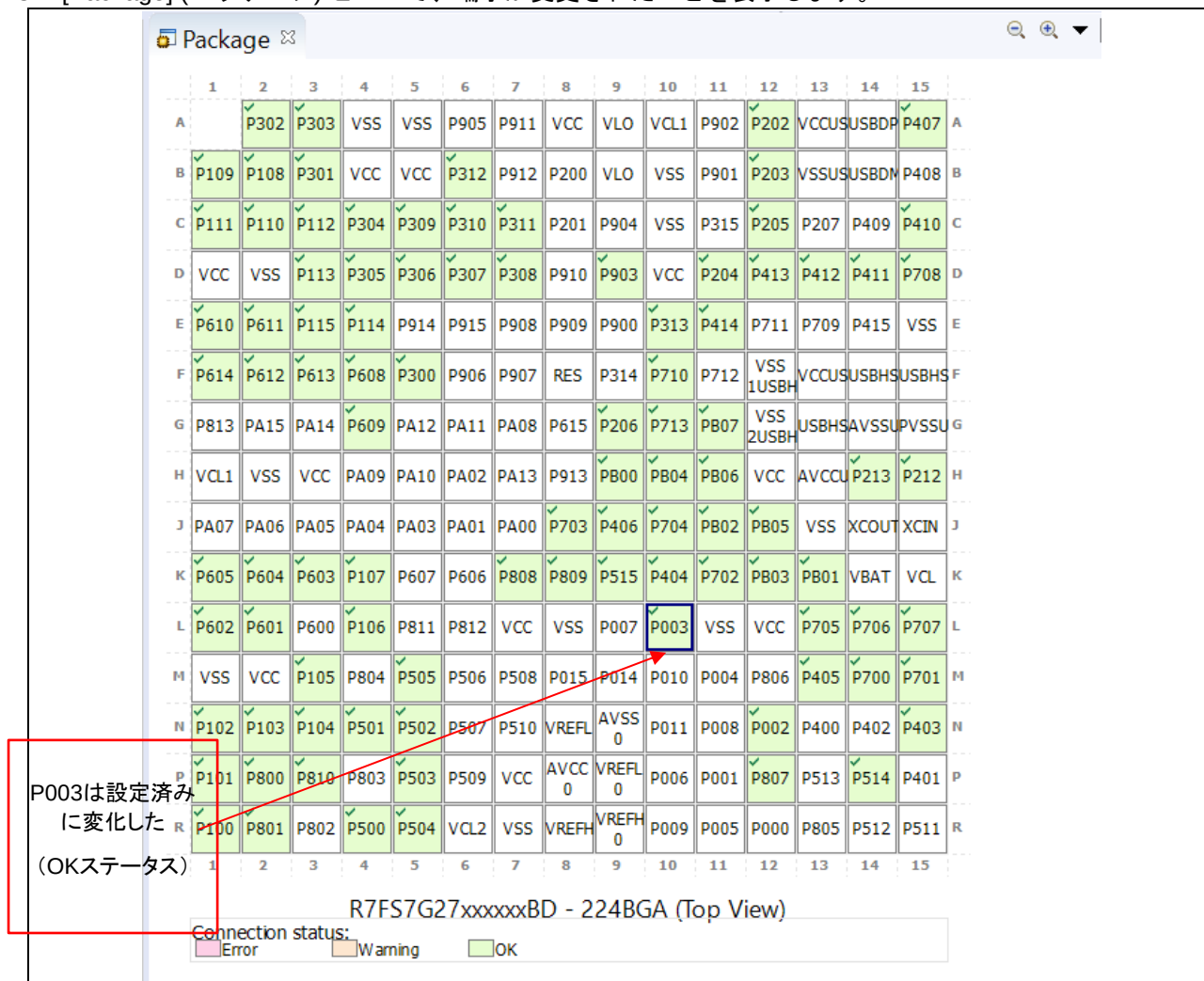


図 61. Synergy Project Configuration (Synergy プロジェクト設定) - [Package] (パッケージ) ビュー (接続ステータス)

このページで、端子設定を 1 つのデバイスから他のデバイスに移行することもできます。この移行を実施するには、ツールバーの **[Import a pin configuration]** (端子設定のインポート) ボタンを使用します。この機能により、ユーザ設定を維持したまま、端子設定を新しいデバイスに移行することができます。

既存の端子設定を現在のプロジェクトに移行するには、**[Import a pin configuration]** (端子設定のインポート) ボタン  をクリックした後、インポートする端子設定ファイルを選択します。

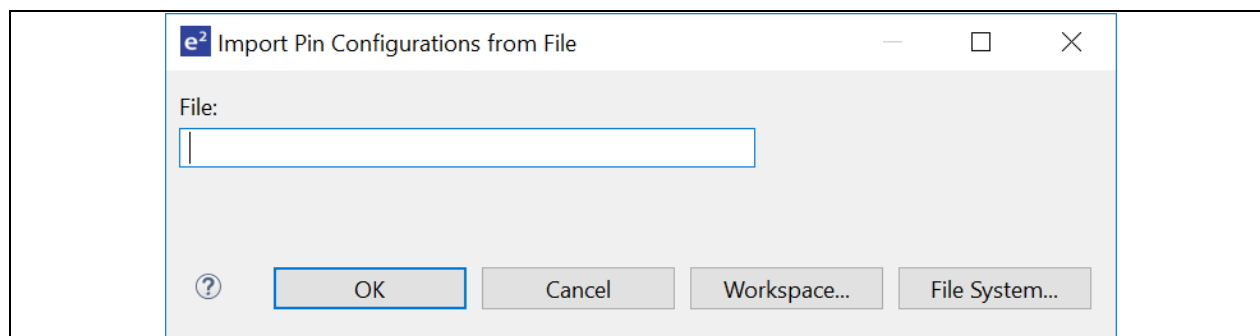


図 62. 既存の端子設定を現在のプロジェクトにインポート

インポート機能によって競合が示され、ユーザに以下の選択肢が提供されることがあります。

1. インポート操作をキャンセルする
2. 競合を無視して、競合している設定をインポートする
3. 競合している設定をインポートせずに、インポート操作を続行する。

3.4.5 [Threads Configuration] (スレッド設定) ページ

[Threads Configuration] (スレッド設定) ページで、ユーザは以下の操作を実行できます。

- Synergy プロジェクト内でスレッドを設定。
- Synergy モジュールとオブジェクトをスレッドに追加。
- **[Properties]** (プロパティ) ビューでモジュールとオブジェクトのプロパティを変更。

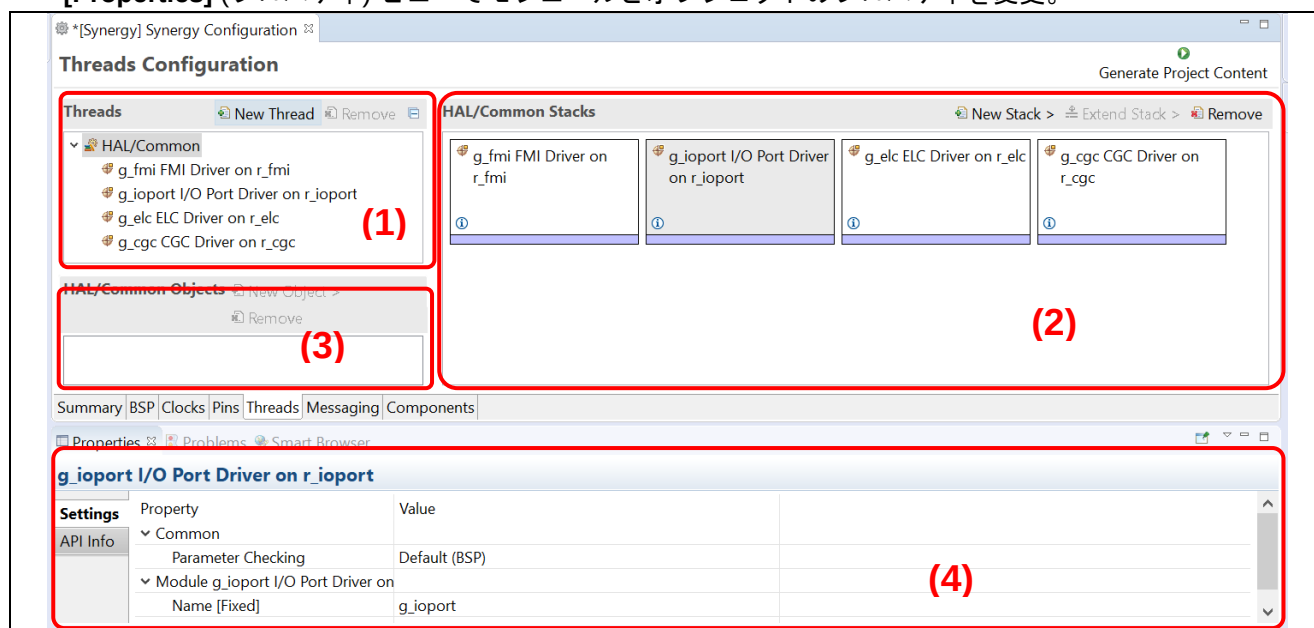


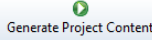
図 63. Synergy Project Configuration (Synergy プロジェクト設定) - Threads Configuration (スレッド設定) GUI

[Threads Configuration] (スレッド設定) ページには、3つのペインがあります。

1. **[Threads]** ペイン：スレッドの追加または削除を行います。詳細については、第 6 章で説明します。
2. **[Stacks]** (スタック) ペイン：SSP モジュールインスタンス。例えば、IO ポート、SCI、UART などの追加または削除を行います。
3. **[Objects]** (オブジェクト) ペイン：カーネルオブジェクトの追加または削除を行います。詳細については、第 6 章で説明します。

さらに、**[Properties]** (プロパティ) ビューは [Threads Configuration] (スレッド設定) をサポートしており、モジュールまたはオブジェクトのプロパティを変更する目的で使用できます。

以下の手順に従って、モジュールを既存のプロジェクトに追加することができます。

1. いずれかのスレッド、例えば、HAL/Common を選択します。このスレッド内にあるモジュールとオブジェクトが表示されます。
2. [Stacks] (スタック) ペインで、 **[New Stack]** (新しいスタック) をクリックし、いずれかのモジュールをスタックに追加します。例えば、**[New Stack]** (新しいスタック) → **[Driver]** (ドライバ) → **[Monitoring]** (モニタ) → **[Clock Accuracy Circuit Driver on r_cac]** (r_cac でのクロック高精度回路ドライバ) を追加します。
3. **[Generate Project Content]** (プロジェクトコンテンツの生成)  ボタンをクリックして、ソースコードの内容を生成します。
4. **[Properties]** (プロパティ) ビューで、選択したモジュールのプロパティが表示されます。ユーザは、モジュールの要件に従ってプロパティを変更することもできます。

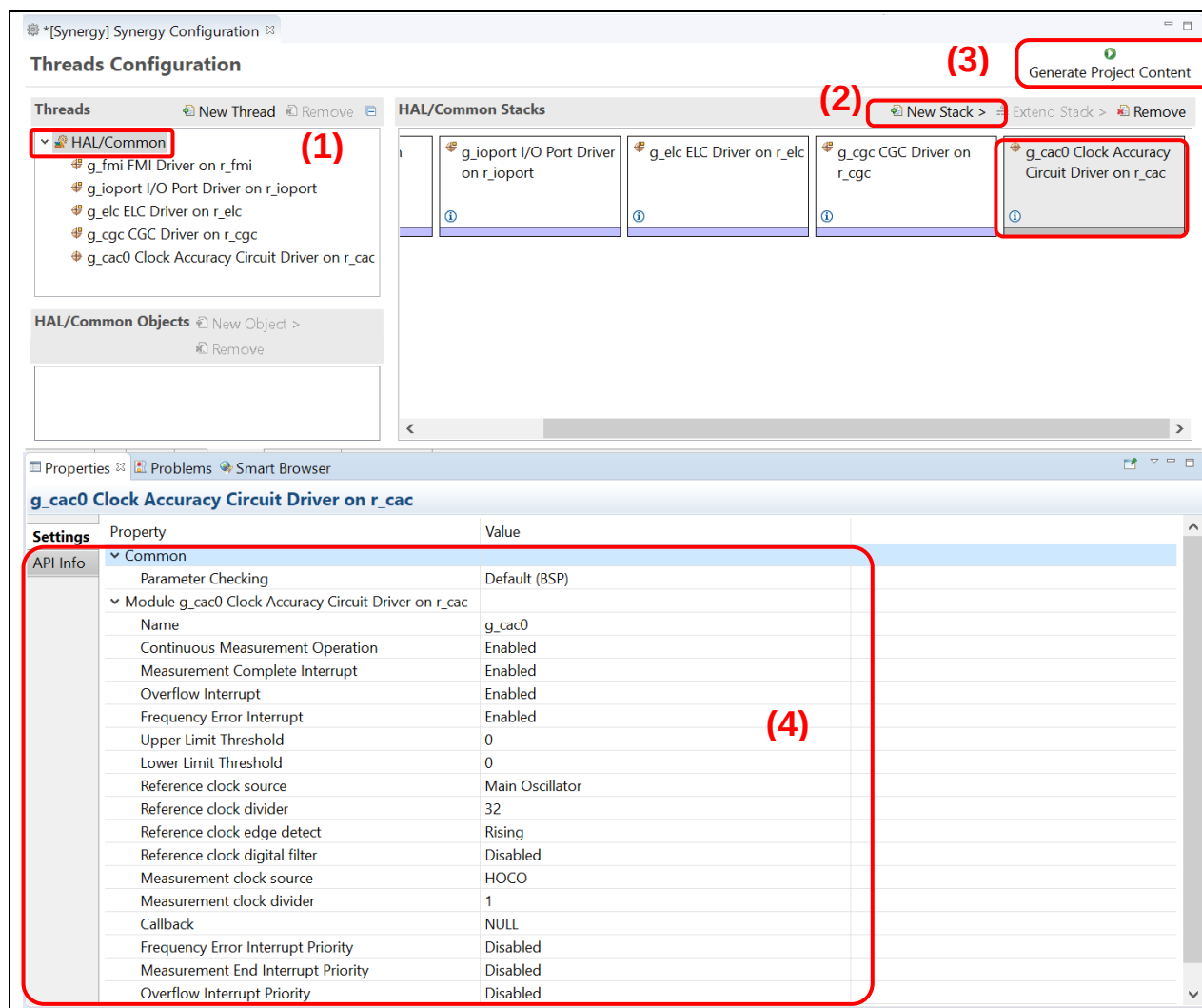


図 64. Synergy Project Configuration (Synergy プロジェクト設定) - 新しいモジュールをスレッドに追加

注記：別の例については、6.1 章「ThreadX 内の General Purpose Timer の例（General Purpose Timer Example in ThreadX）」を参照してください。この章では、GPT モジュールを **Blinky Thread** に追加する手順について説明します。

追加したモジュール (例えば、[UART Driver on r_sci_uart] (r_sci_uart での UART ドライバ)) は、依存先モジュールまたは設定項目を必要とすることがあります。必須の依存先モジュールは自動的に追加されます。オプションの依存先モジュールは、ユーザが手動で追加することを推奨します。このような場合、ユーザは推奨されたモジュールをクリックしてそのモジュールを追加し、そのモジュールのプロパティを設定する必要があります。

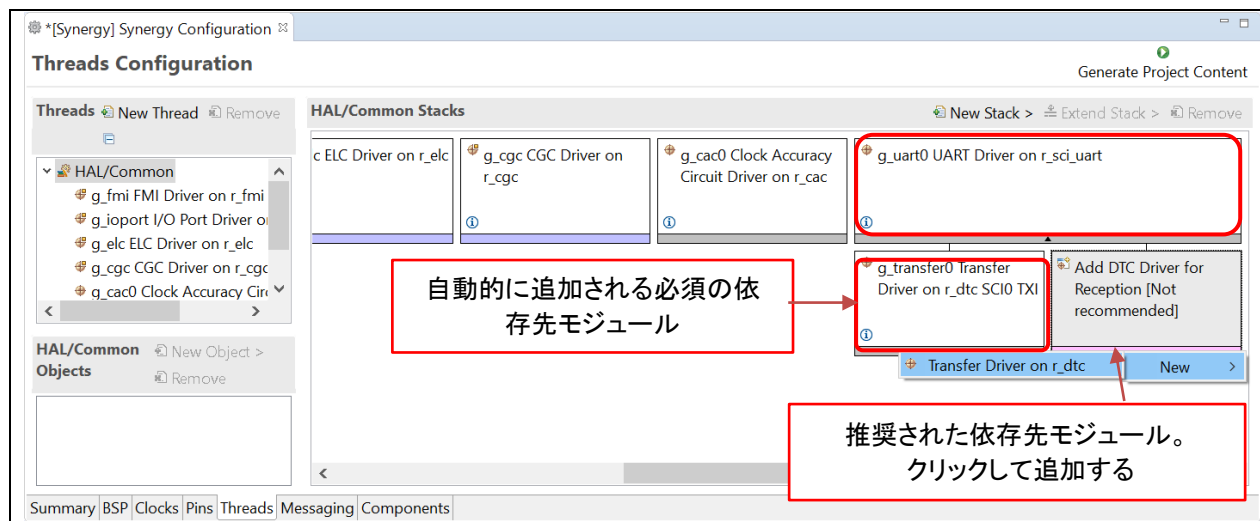


図 65. Synergy Project Configuration (Synergy プロジェクト設定) - 追加したモジュールの問題

[Threads Configuration] (スレッド設定) ページで「コピーアンドペースト」を行って、モジュールまたはモジュールスタックを追加することもできます。モジュールを右クリックし、**[Copy]** (コピー) を選択すると、そのモジュールをクリップボードにコピーできます。次に、同じプロジェクト内で、同じスレッドまたは別のスレッドのスタックペインを右クリックし、**[Paste]** (貼り付け) を選択します。同様に、「カットアンドペースト」操作もできます。

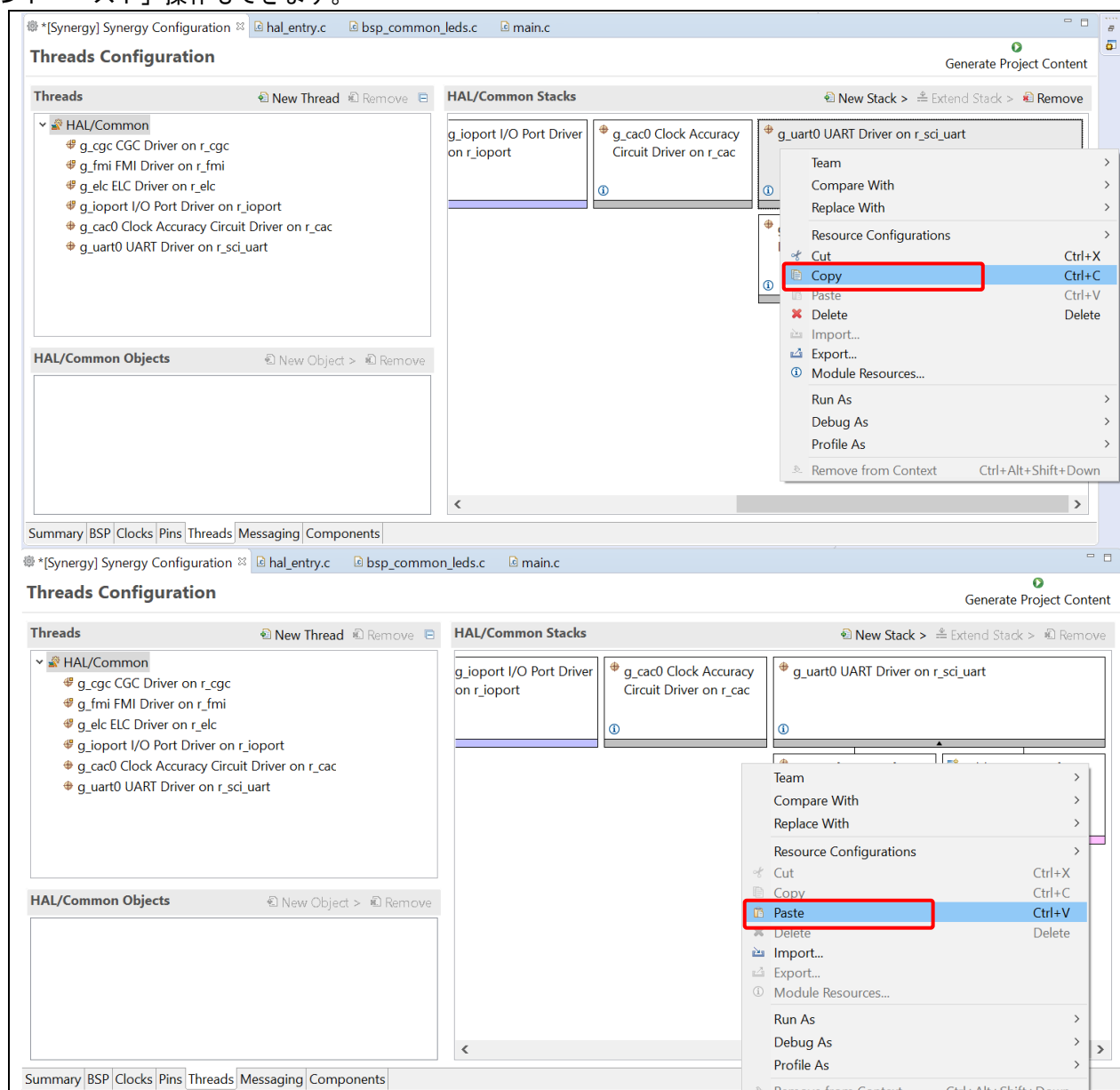


図 66. コピーアンドペースト

元のモジュールインスタンスと新しいモジュールインスタンスの間で名前の競合が発生します。どちらかのモジュールインスタンスの名前を変更すると、この問題を解決できます。

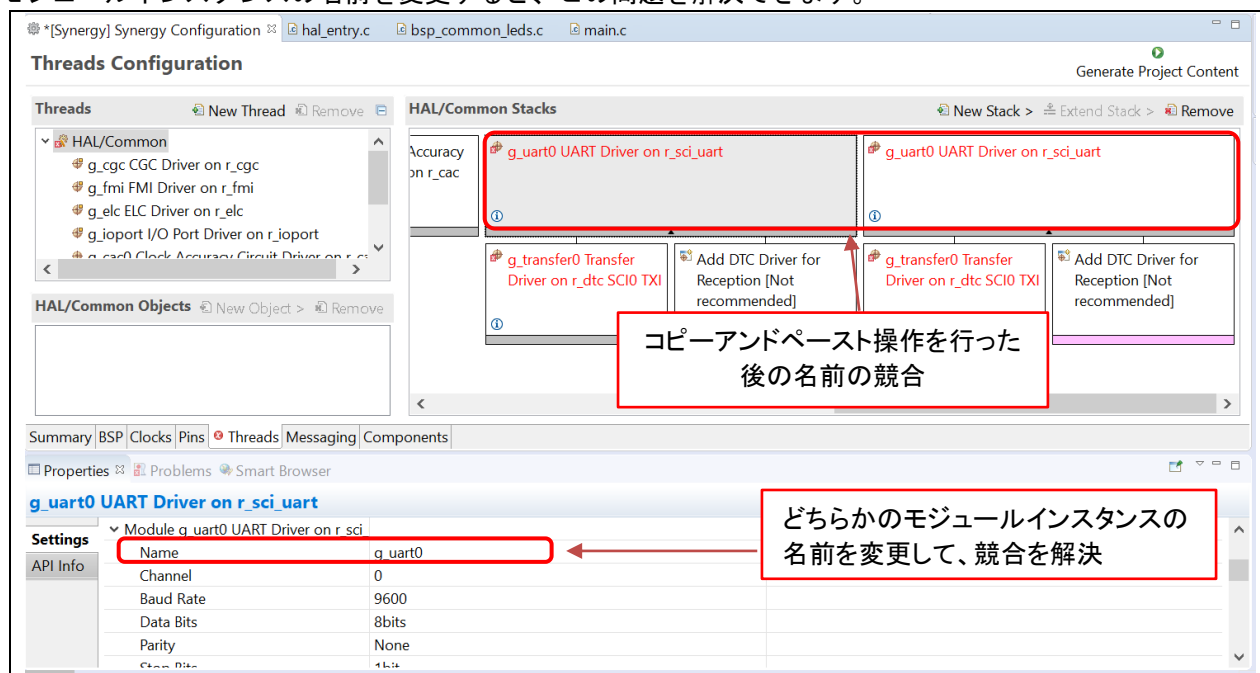


図 67. モジュールインスタンス名の競合

[Threads Configuration] (スレッド設定) ページでエクスポートとインポートの操作を行って、モジュールまたはモジュールスタックを追加することもできます。モジュールを右クリックし、**[Export...]** (エクスポート...) を選択すると、モジュールの設定を XML ファイルにエクスポートできます。次に、同じプロジェクト内で、同じスレッドまたは別のスレッドのスタックペインを右クリックし、**[Import...]** (インポート...) を選択すると、エクスポート先の XML ファイルから設定をインポートできます。どちらかのモジュールインスタンスの名前を変更すると、名前の競合を解決できます。

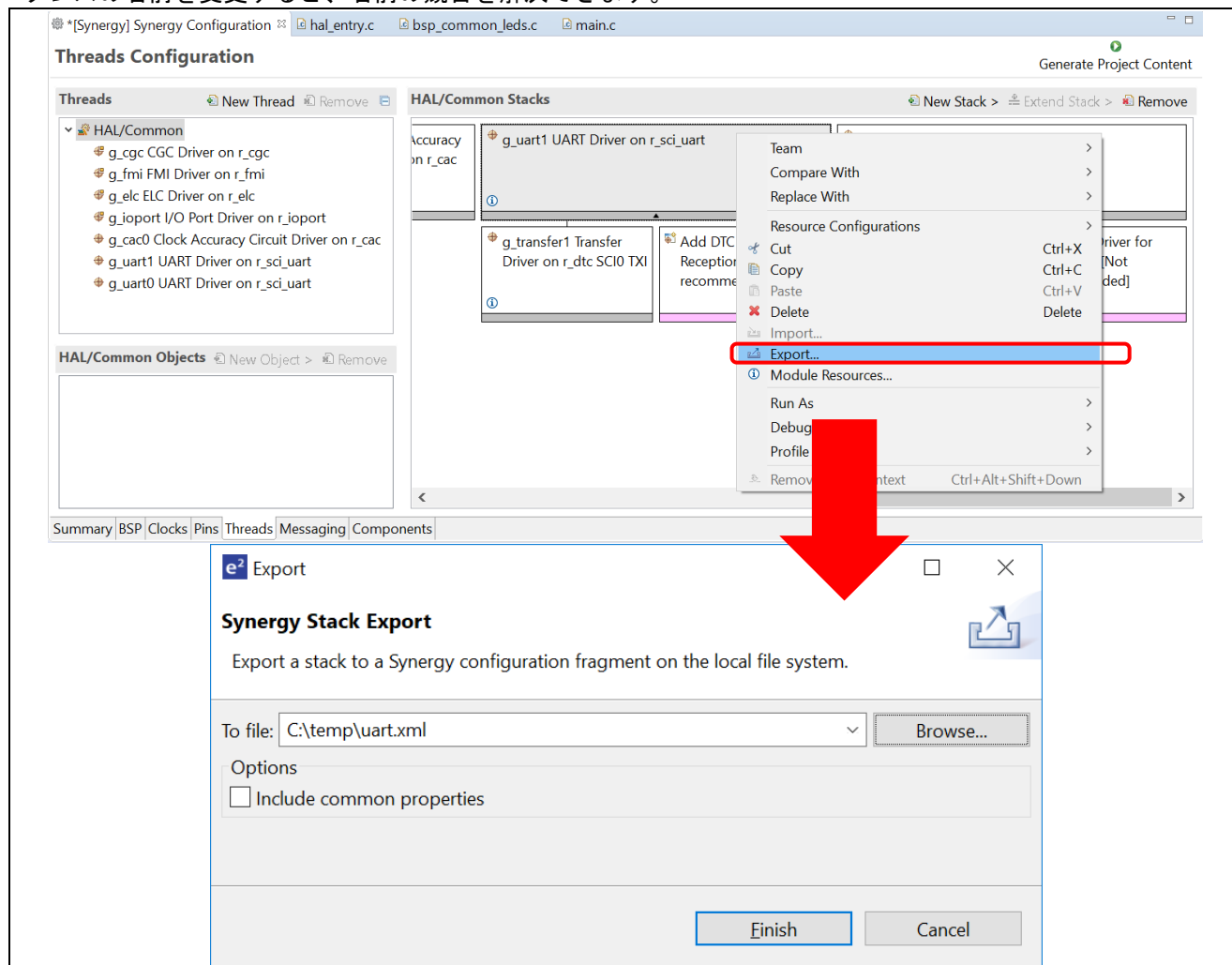


図 68. Synergy スタックのエクスポート

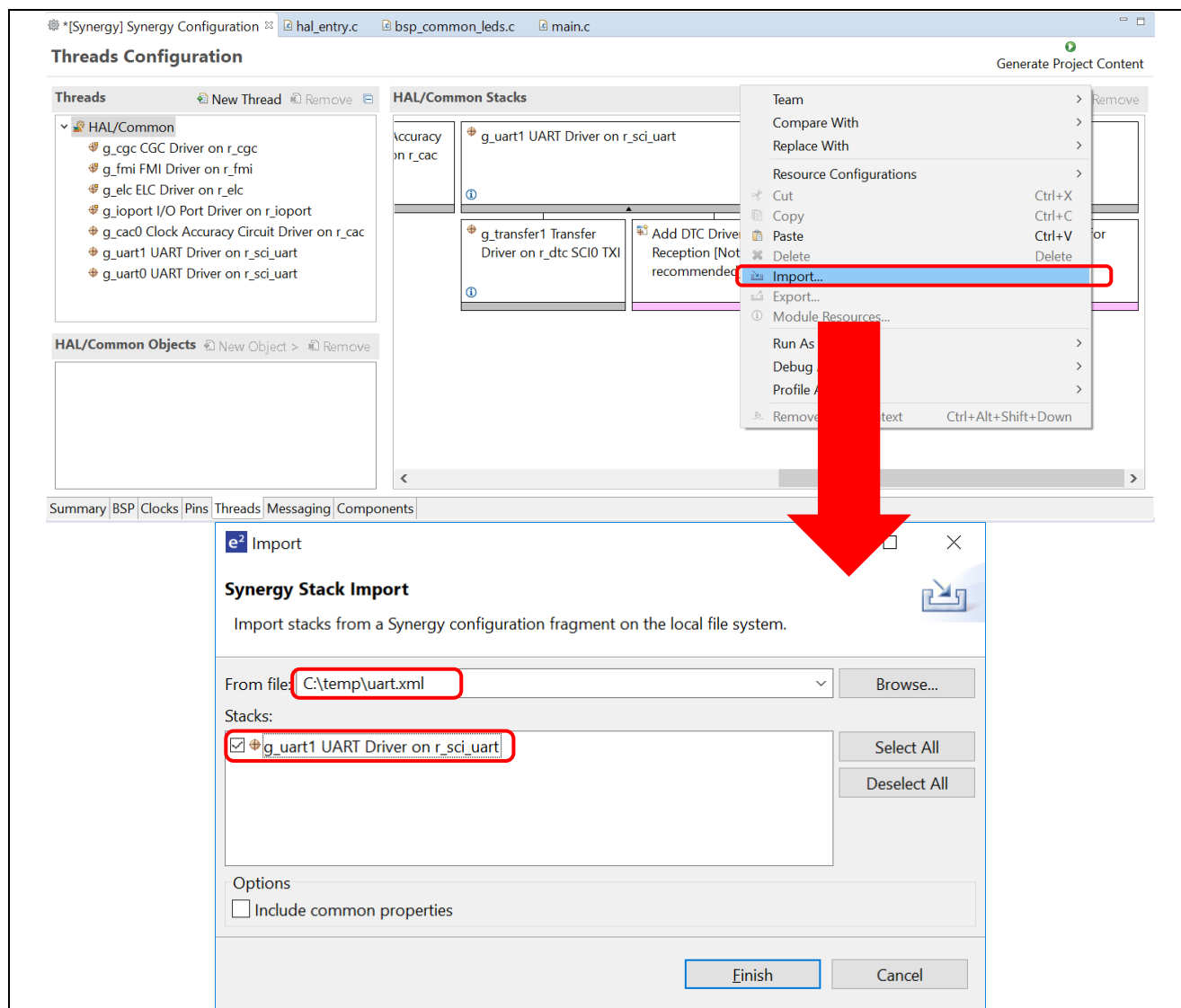


図 69. Synergy スタックのインポート

3.4.6 [Messaging] (メッセージング) ページ

[Messaging] (メッセージング) ページで、イベントクラス、イベント、およびサブスクライバを作成し、Synergy メッセージングフレームワークで 사용할 수 있습니다。

[Messaging] (メッセージング) ページには、3つのペインがあります。

1. **[Event Classes]** (イベントクラス) ペインには、Synergy モジュールをインスタンス化したときに提供されたイベントクラスや、手動で作成したイベントクラスのリストが表示されます。
2. **[Events]** (イベント) ペインには、Synergy モジュールをインスタンス化したときに提供されたイベントや、手動で作成したイベントが表示されます。
3. **[Subscribers]** (サブスクライバ) ペインには、作成済みのサブスクライバのリストが表示されます。各サブスクライバエントリの隣にあるチェックボックスは、そのサブスクライバが、現在選択されているイベントクラスに対応するメッセージを受信するかどうかを示します。チェックボックスをオンまたはオフにすると、現在選択されているイベントクラスに対応するメッセージの受信をサブスクライバごとに有効または無効にすることができます。

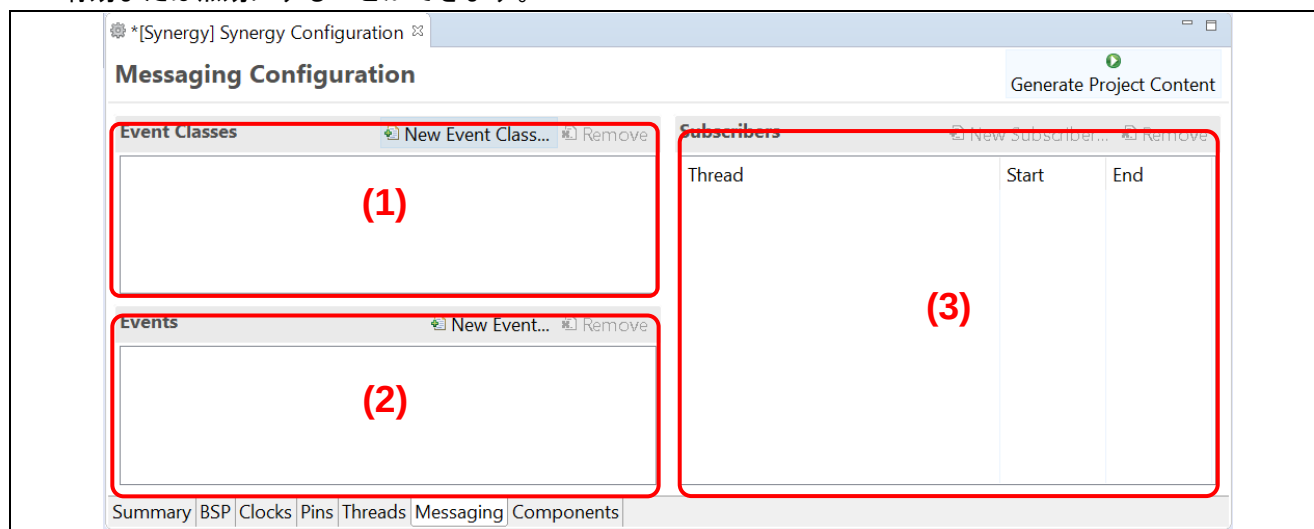


図 70. Synergy Project Configuration (Synergy プロジェクト設定) - [Messaging] (メッセージング) ページ

対応するセクションで適切なボタンをクリックして、イベントクラス、イベント、またはサブスクライバを手動で作成することもできます。

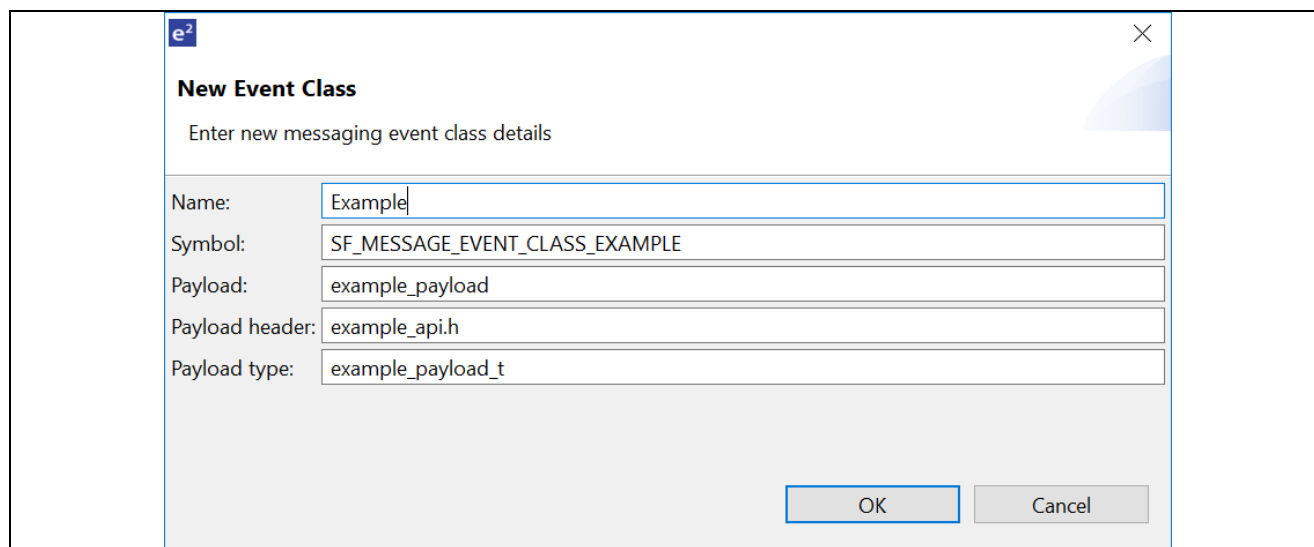


図 71. Messaging (メッセージング) ページ - 新しいイベントクラスの追加

手動で作成した項目を削除するには、その項目を選択し、対応するセクションで  ボタンをクリックします (Synergy モジュールのインスタンス化によって追加されたモジュールを削除することはできません)。

ユーザがいずれかの項目を選択すると、e² studio の **[Properties]** (プロパティ) ビューで、現在選択されているイベントクラス、イベント、またはサブスクリバに関連付けられているプロパティが表示されます。

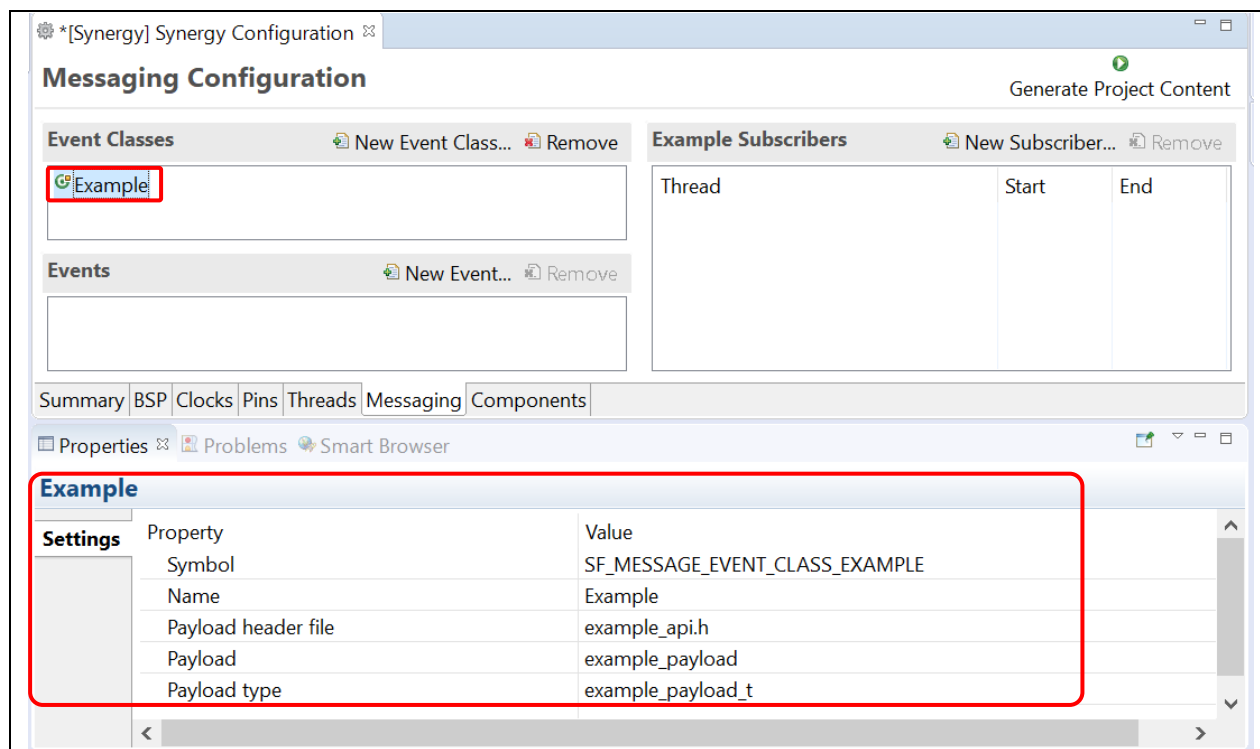


図 72. [Messaging] (メッセージング) のプロパティ

3.4.7 [Components Configuration] (コンポーネント設定) ページ

[Components Configuration] (コンポーネント設定) ページで、アプリケーションにインクルードまたは除外する必要のある個別のモジュールを設定することができます。

すべての Synergy プロジェクトに共通するモジュールは、事前選択されています（例えば、[HAL Drivers] (HAL ドライバ) → [all] (すべて) → [r_cgc]）。

[Threads] (スレッド) ページで選択されているドライバが必要とするすべてのモジュールは、自動的にインクルードされます。ユーザは、必要なコンポーネントの隣にあるボックスをチェックすることで、追加したモジュールをインクルードまたは除外することができます。

注記：アプリケーションにモジュールを追加する本来の方法は、**[Threads]** (スレッド) ページを使用することです。**[Components Configuration]** (コンポーネント設定) ページは主に、インストール済み SSP で利用可能なコンポーネントのリストを表示する目的で使用します。

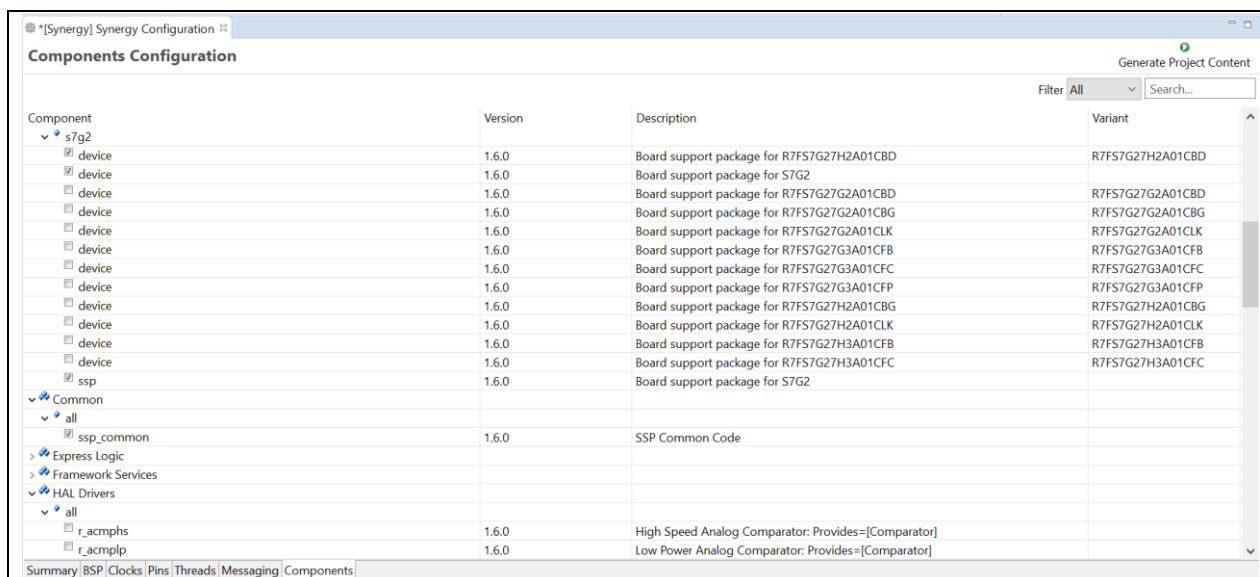


図 73. Synergy Project Configuration (Synergy プロジェクト設定) - Components Configuration (コンポーネント設定)

3.5 エディタのホバー (Editor hover)

e² studio は、テキストエディタでのホバー機能 (マウスカーソル合わせ) をサポートしています。この機能を有効または無効にするには、[Window] (ウィンドウ) → [Preferences] (環境設定) → [C/C++] → [Editor] (エディタ) → [Hovers] (ホバー) を使用します。

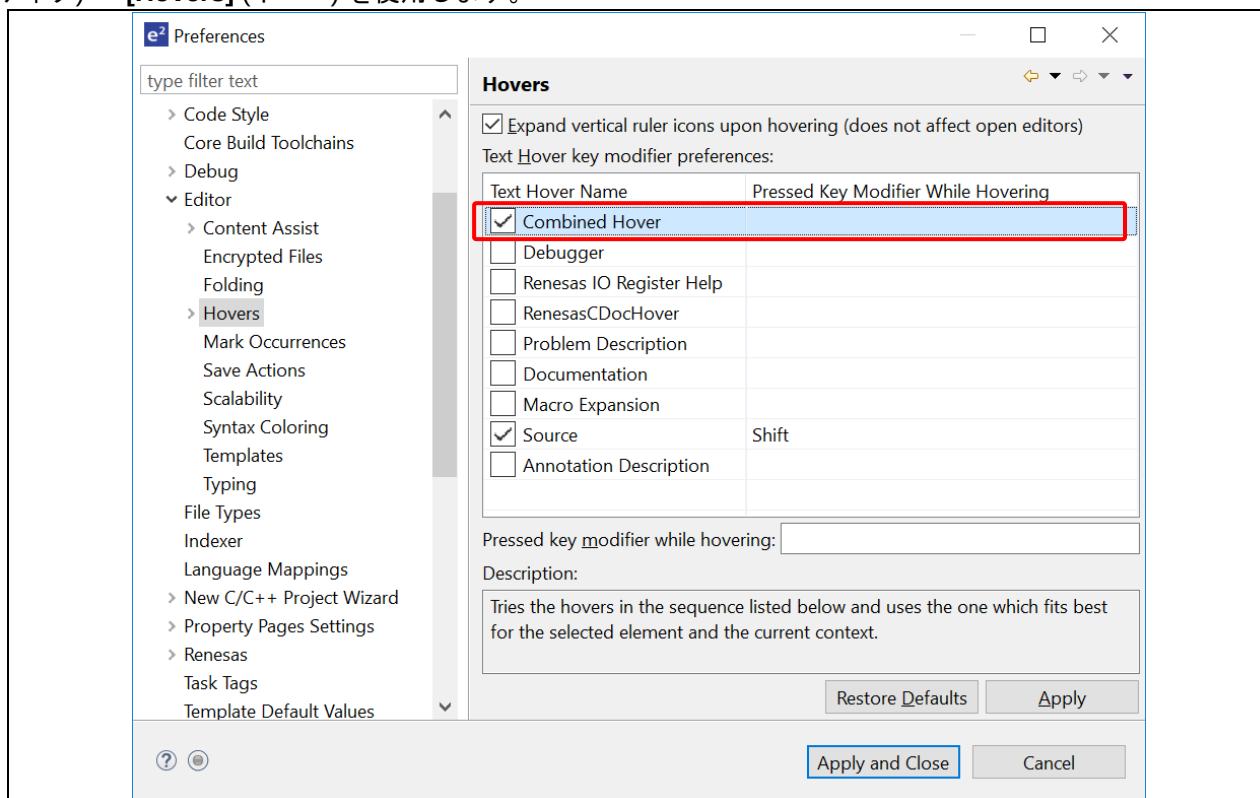


図 74. ホバーの設定

ホバーを有効にするには、[Combined Hover] (複合的なホバー) チェックボックスをオンにします。この機能を無効にするには、このチェックボックスをオフにします。この機能は、デフォルトで有効になっています。

Hover (ホバー) 機能を使用すると、識別子にマウスカーソルを合わせ、表示されたポップアップを参照することで、ソースコード内にある各識別子に関する詳細情報を表示できます。

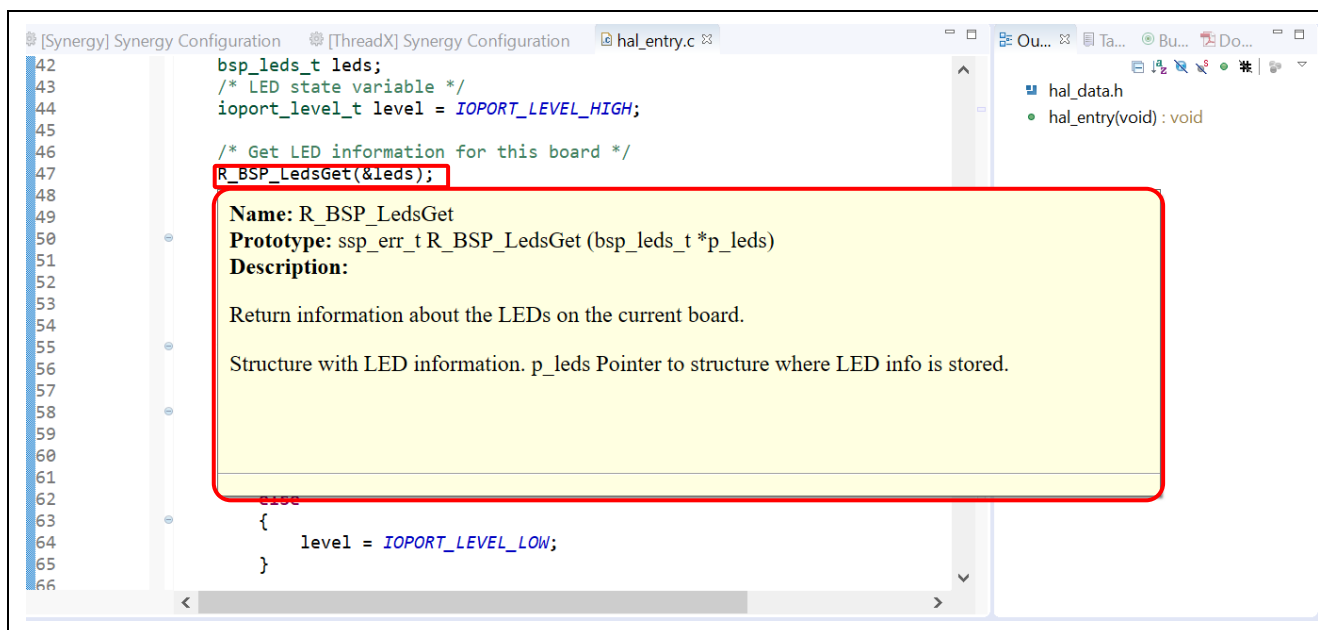


図 75. ホバー機能により表示される情報

3.6 [Developer Assistance] (開発者支援機能)

Synergy Developer Assistance (開発者支援機能) により、e² studio 内で、モジュールと API (アプリケーションプログラミングインタフェース) に関するリファレンスドキュメントを参照できます。Synergy Configuration Editor を使用して、Synergy プロジェクトに対応するスレッドとソフトウェアスタックを設定した後、設定済みのスタックモジュールを使用してそのプロジェクトに対応する C/C++ のアプリケーションコードの開発を開始できるように、Developer Assistance (開発者支援機能) がすぐに支援を行います。

1. プロジェクトエクスプローラを展開し、[Developer Assistance] (開発者支援機能) を表示します。

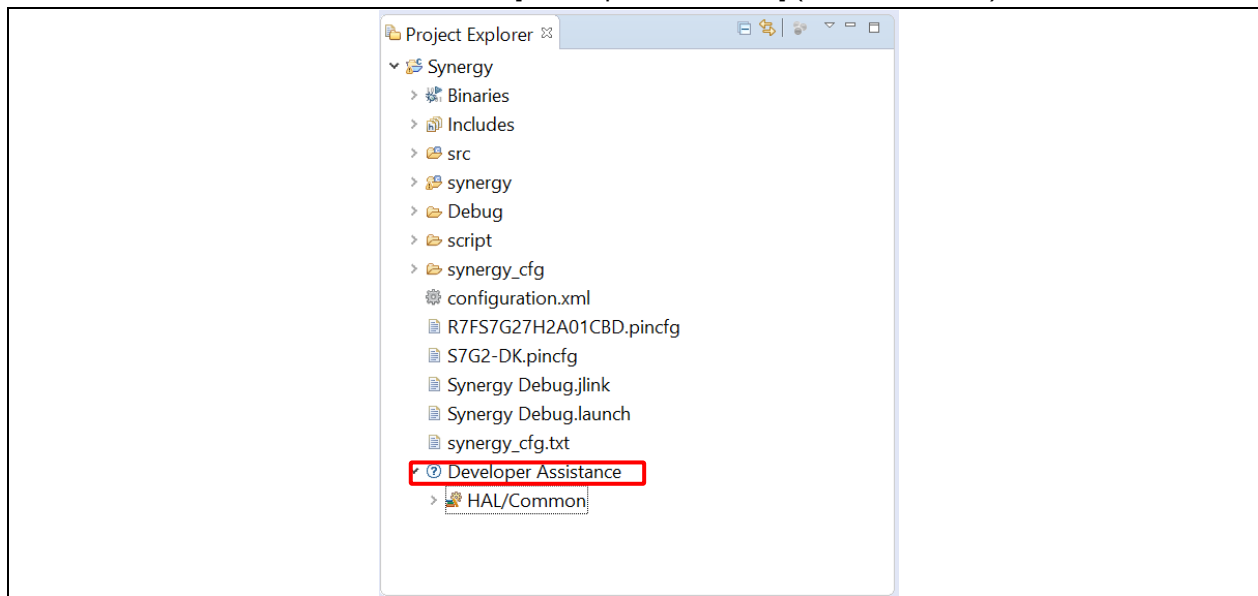


図 76. Synergy Developer Assistance (開発者支援機能)

2. スタックモジュールを展開し、その API を表示します

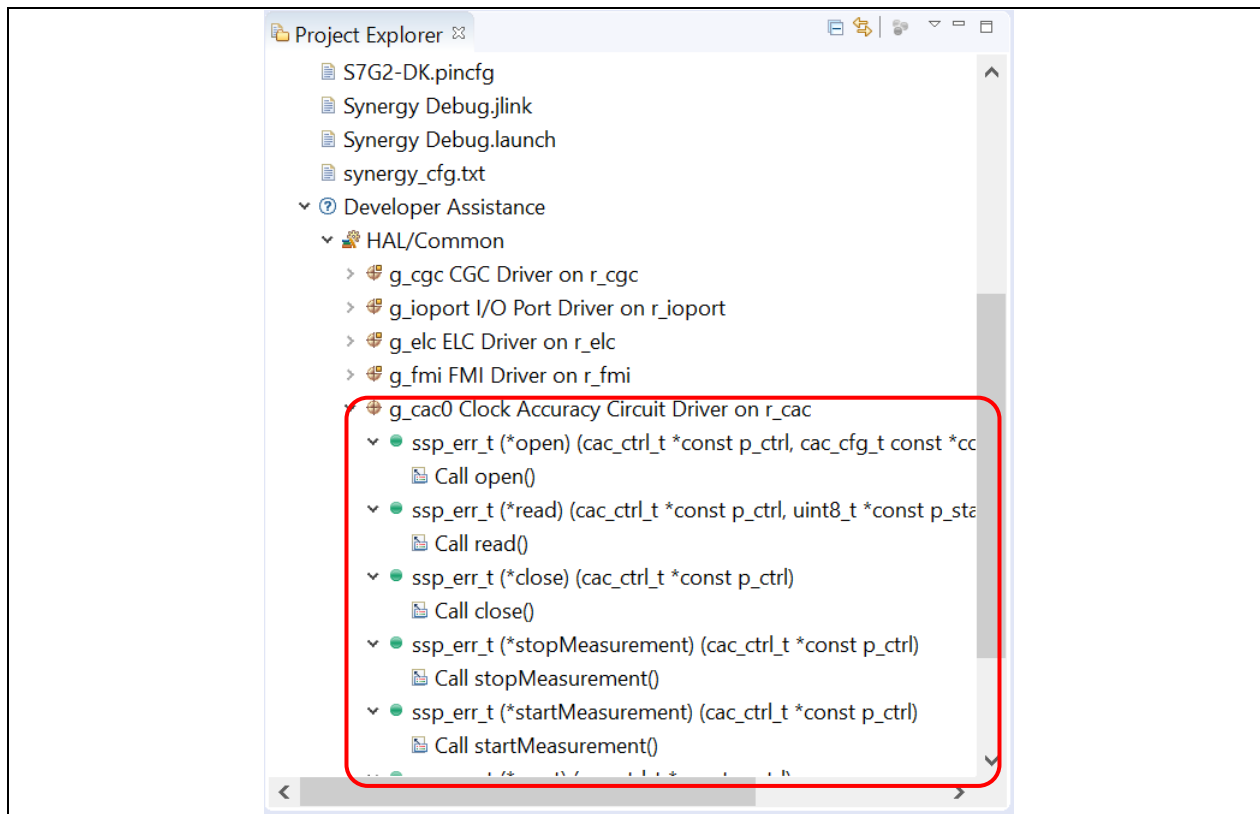


図 77. API 情報

3. [Develop Assistance] (開発者支援機能) から API をソースファイルにドラッグアンドドロップすると、ソースコードの迅速な入力に役立ちます。

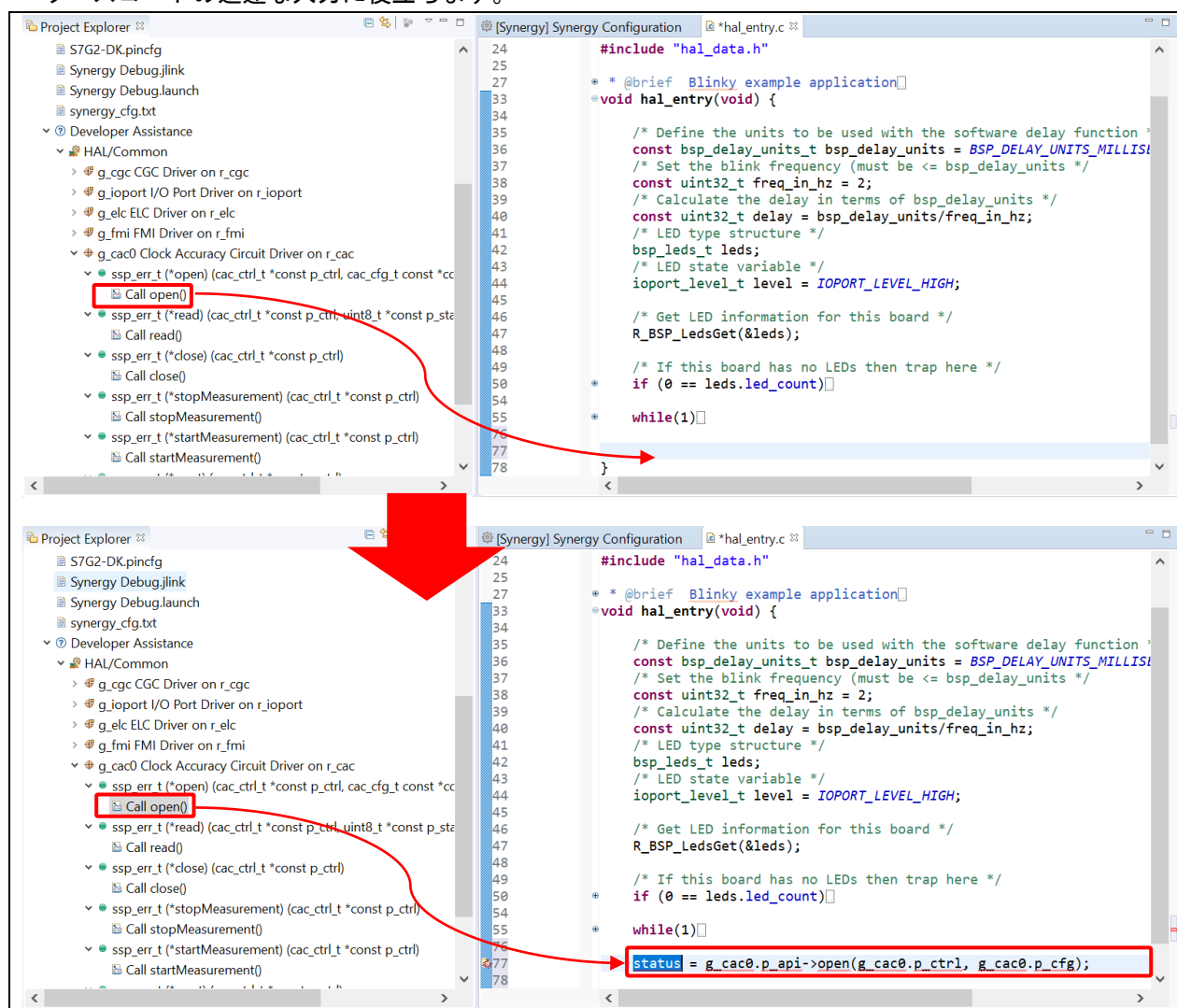


図 78. Developer Assistance (開発者支援機能) - API のドラッグアンドドロップ

4. ビルド (Building)

この章では、e²studio のビルド構成と主なビルド機能について説明します。

4.1 ビルド構成 (Build Configurations)

プロジェクトを作成するときにデフォルトのビルドオプションが生成され、通常プロジェクトをビルドするときに、そのオプションを使用することができます。

ただし、ビルドオプションを変更する必要がある場合（例えば、ツールチェーンのバージョン、最適化オプションなど）、プロジェクトをビルドする前に以下の手順に従ってください。

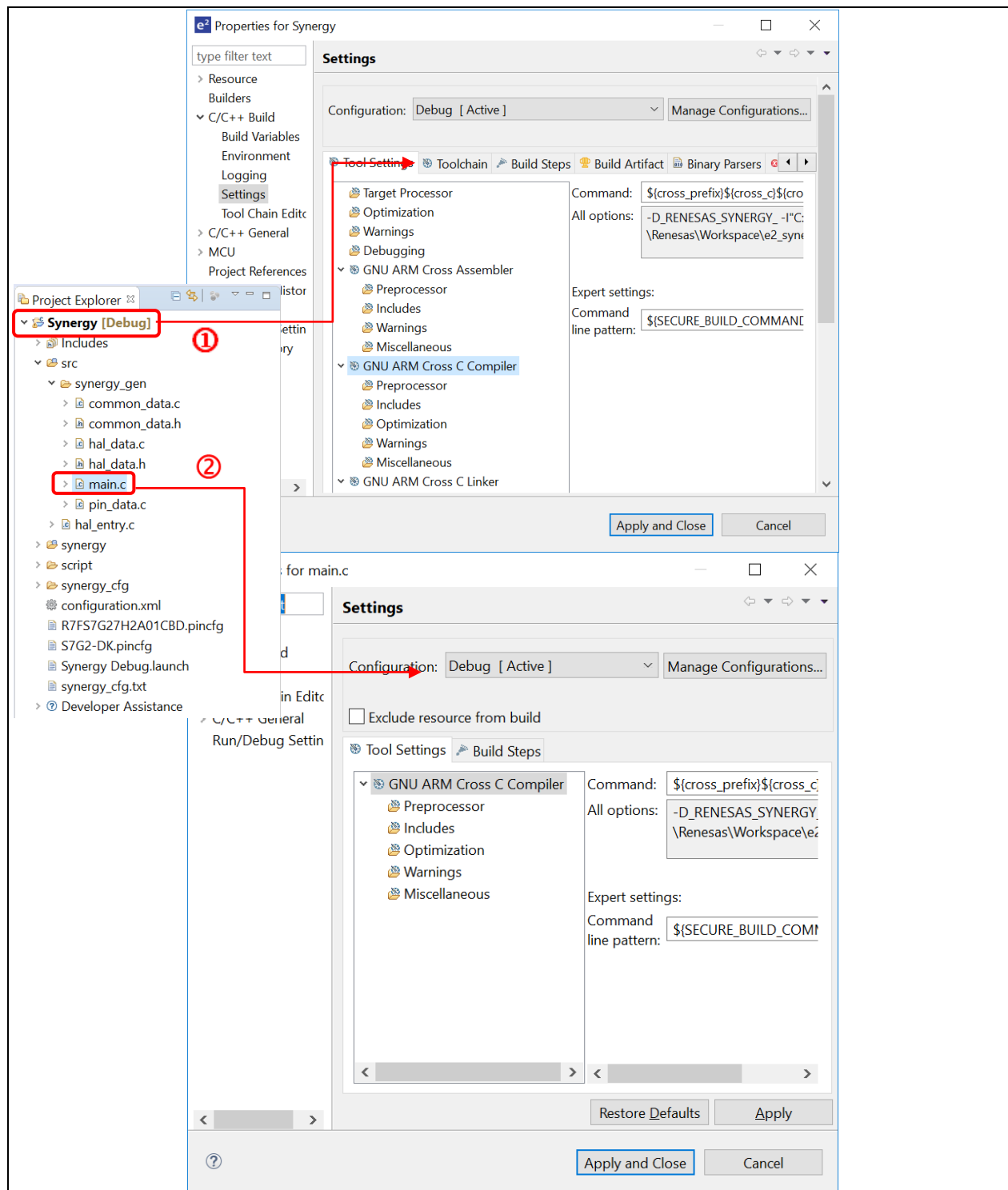


図 79. ビルド - Synergy プロジェクトと main.c ソースファイルのプロパティ

プロジェクトまたはソースファイルのプロパティウィンドウから、ビルドオプションにアクセスできます。

1. ① プロジェクト名か、② ソースファイル名のいずれかを対象として選択します。
2. 右クリックで **[Properties]** (プロパティ) を選択するか、ショートカットキー **[Alt]+[Enter]** を使用して、プロパティダイアログを開きます。
3. **[C/C++ Build]** オプションをクリックして、設定項目を表示または編集します。

[Properties] (プロパティ) ウィンドウは、プロジェクトレベルとソースレベルでサポートされています。プロジェクトに対応する [Properties] (プロパティ) ウィンドウは、より多くの構成をサポートしており、同じプロジェクト内にあるすべてのファイルに対してそれらの構成が適用されます。

4.2 サンプルプロジェクトのビルド (Building a Sample Project)

プロジェクトをビルドするには、以下の手順に従ってください。

1. **[Project Explorer]** (プロジェクトエクスプローラ) で、Synergy プロジェクトをクリックして選択します。
2. このプロジェクトをビルドするには、**[Project]** (プロジェクト) → **[Build Project]** (プロジェクトのビルド) をクリックするか、 アイコンをクリックします。
3. ビルドが完了した後に何もエラーが発生していないことを確認します。

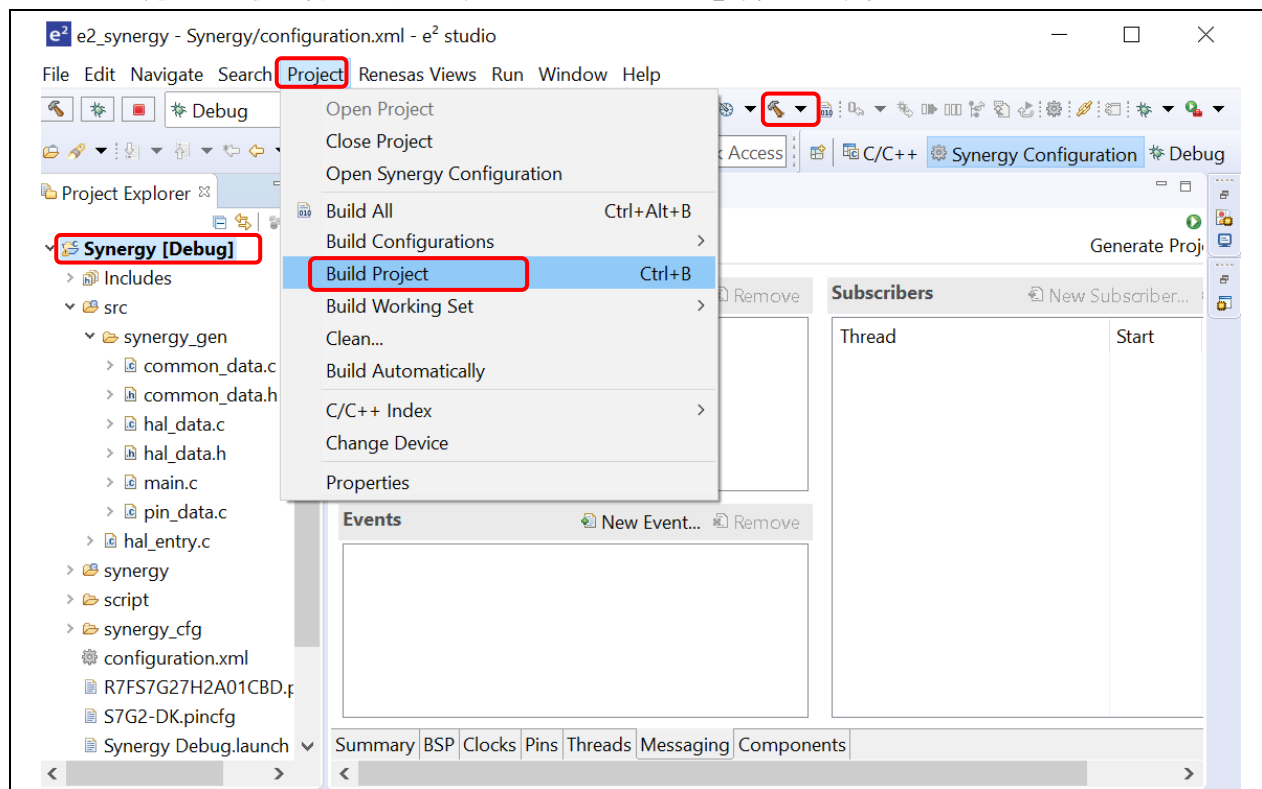


図 80. ビルド - サンプルプロジェクトのビルド

4.3 ビルド設定レポートの保存 (Saving the Build Settings Report)

[Project Reporter] (プロジェクト報告) 機能を使用して、e² studio IDE 内のプロジェクトのビルド設定をファイルに保存することができます。

1. **[Project Explorer]** (プロジェクトエクスプローラ) ビューを右クリックし、コンテキストメニューをポップアップします。
2. **[Save build settings report]** (ビルド設定レポートの保存) を選択し、ビルド設定レポートを保存します。

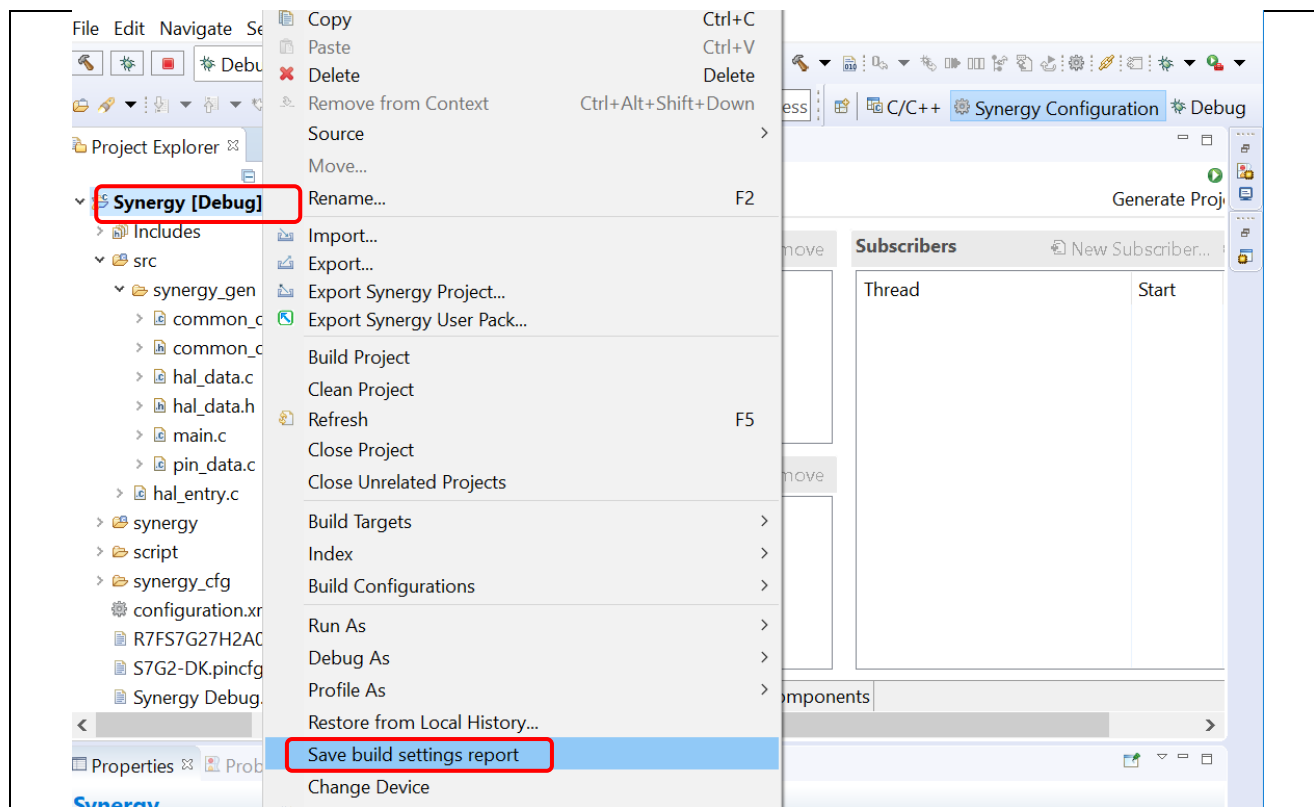


図 81. ビルド - ビルド設定レポートの保存

5. デバッグ (Debugging)

この章では、e² studio のデバッグ構成の使用方法と主なデバッグ機能について説明します。以下の図は、第 4.2 章 - 「サンプルプロジェクトのビルド (Building a Sample Project)」でビルドした「Synergy」プロジェクトを参照しており、以下のハードウェア構成をベースにしています。J-link Arm エミュレータと Synergy DK-S7G2 ボード。

TraceX®を使用して ThreadX ベースのプロジェクトをデバッグする方法については、Synergy Platform Web サイト (www.renesas.com/synergy) で入手できる『TraceX®ユーザーズマニュアル』を参照してください。TraceX を使用した ThreadX プロジェクトのデバッグは、このドキュメントで説明しません。

各アイコンの名前を表示するには、任意のパースペクティブアイコンを右クリックし、[Show Text] (テキストの表示) を選択します。

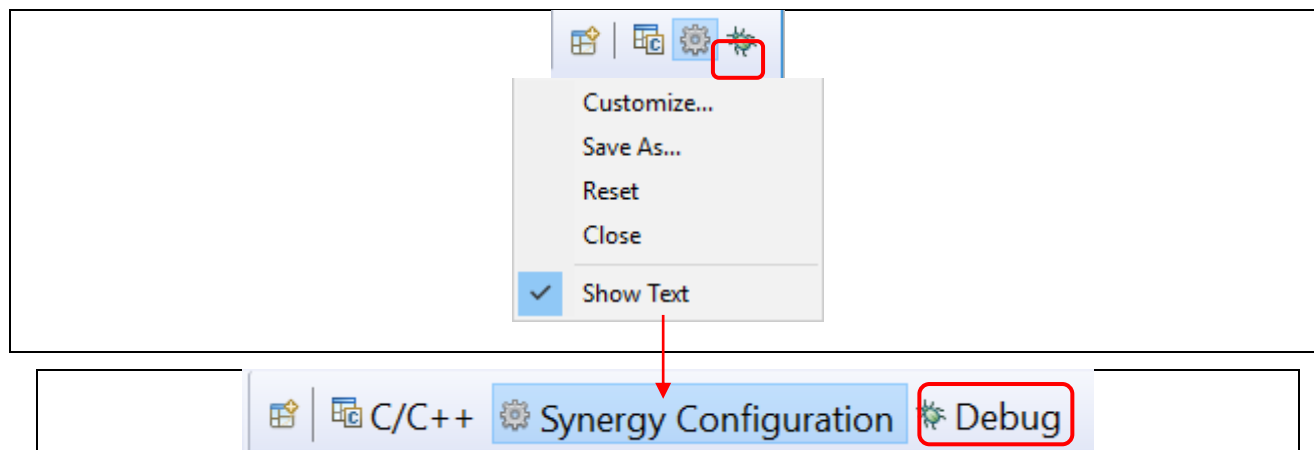


図 82. デバッグ - [Debug] パースペクティブへの切り替え

e² studio で Synergy プロジェクトを開き、[Debug] (デバッグ) をクリックして [Debug] (デバッグ) パースペクティブに切り替えます。

既に説明したように、Eclipse の「パースペクティブ」とは、[Workbench] ウィンドウ内のペインやビューに関するレイアウトのことです。各パースペクティブは、ビュー、メニュー、ツールバーの組み合わせで形成されており、ユーザはそれらを使用して個別のタスクを実行できます。

例えば、

- **[Debug] (デバッグ)** パースペクティブには、ユーザがプログラムをデバッグするための複数のビューがあります。
- プロジェクト構成設定に対応する [Package] (パッケージ) ビューと [Properties] (プロパティ) ビューと同様に、**[Synergy Configuration]** (Synergy 設定) パースペクティブは、エディタウィンドウにある `configuration.xml` を使用して、Synergy 設定を開くことができます。
- **[C/C++]** パースペクティブには、ユーザが C/C++ プログラムを開発するための複数のビューがあります。

[Debug] (デバッグ) パースペクティブを使用していない状態で、ユーザがデバッグに接続しようとした場合、e² studio はユーザに、**[Debug] (デバッグ)** パースペクティブに切り替えるように求めます。

単一のワークベンチ設定内に、1 つまたは複数のパースペクティブが存在することも可能です。ユーザはこれらのパースペクティブをカスタマイズするか、新しいパースペクティブを追加することもできます。

5.1 既存のデバッグ構成の変更 (Changing an Existing Debug Configuration)

個別の Synergy プロジェクトを最初にビルドするときに、デフォルトのデバッグ構成が自動的に作成されます。既存のデバッグ構成を、以下の手順で変更することもできます。

1. **[Project Explorer]** (プロジェクトエクスプローラ) ビューで、プロジェクト名をクリックして絞り込みます。
2. **[Run]** (実行) → **[Debug Configurations...]** (デバッグ構成...) をクリックするか、 アイコン (プルダウンボタン) → **[Debug Configurations ...]** (デバッグ構成...) をクリックして、**[Debug Configurations]** (デバッグ構成) ウィンドウを開きます。

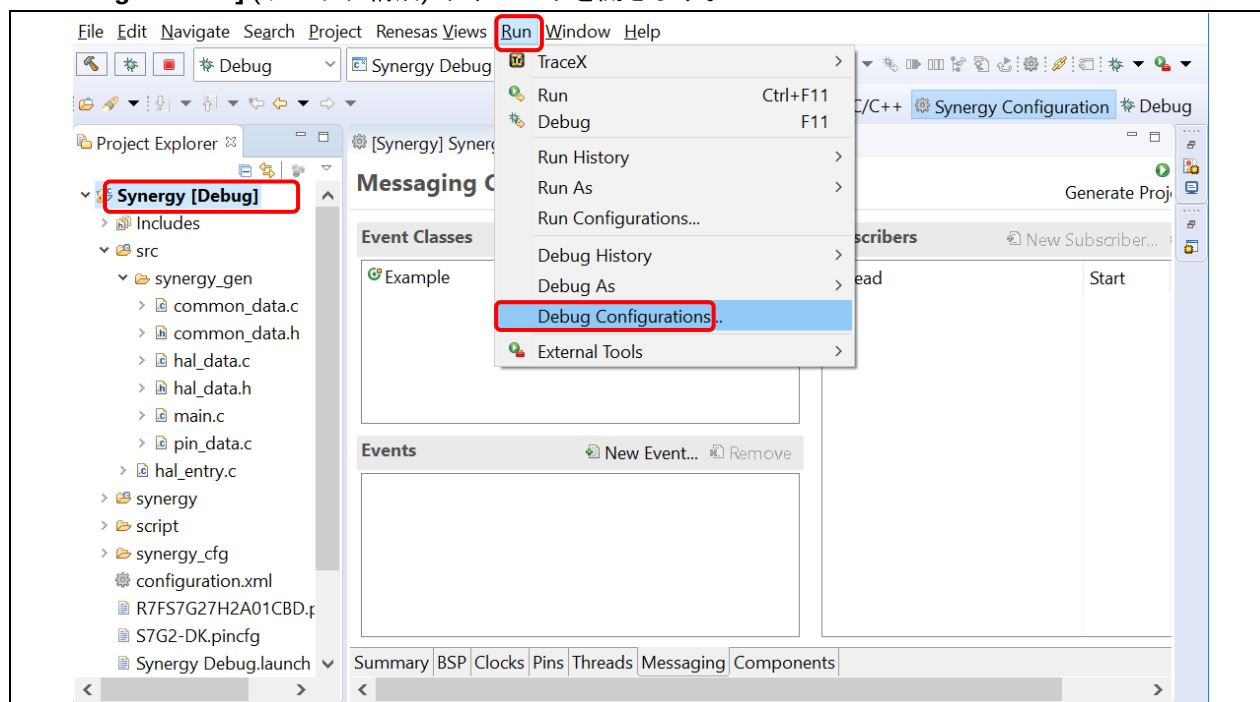


図 83. デバッグ - [Debug Configurations] (デバッグ構成) ウィンドウを開く

3. [Debug Configurations] (デバッグ構成) ウィンドウで、[Renesas GDB Hardware Debugging] (Renesas GDB ハードウェアデバッグ) デバッグ構成を開き、既存のデバッグ構成のいずれか（例えば、[Synergy Debug] (Synergy デバッグ)) をクリックします。
4. **[Main]** (メイン) タブに移動し、プロジェクトビルドフォルダ内にある、ロードモジュール（例えば、Synergy.elf）を参照して追加します。

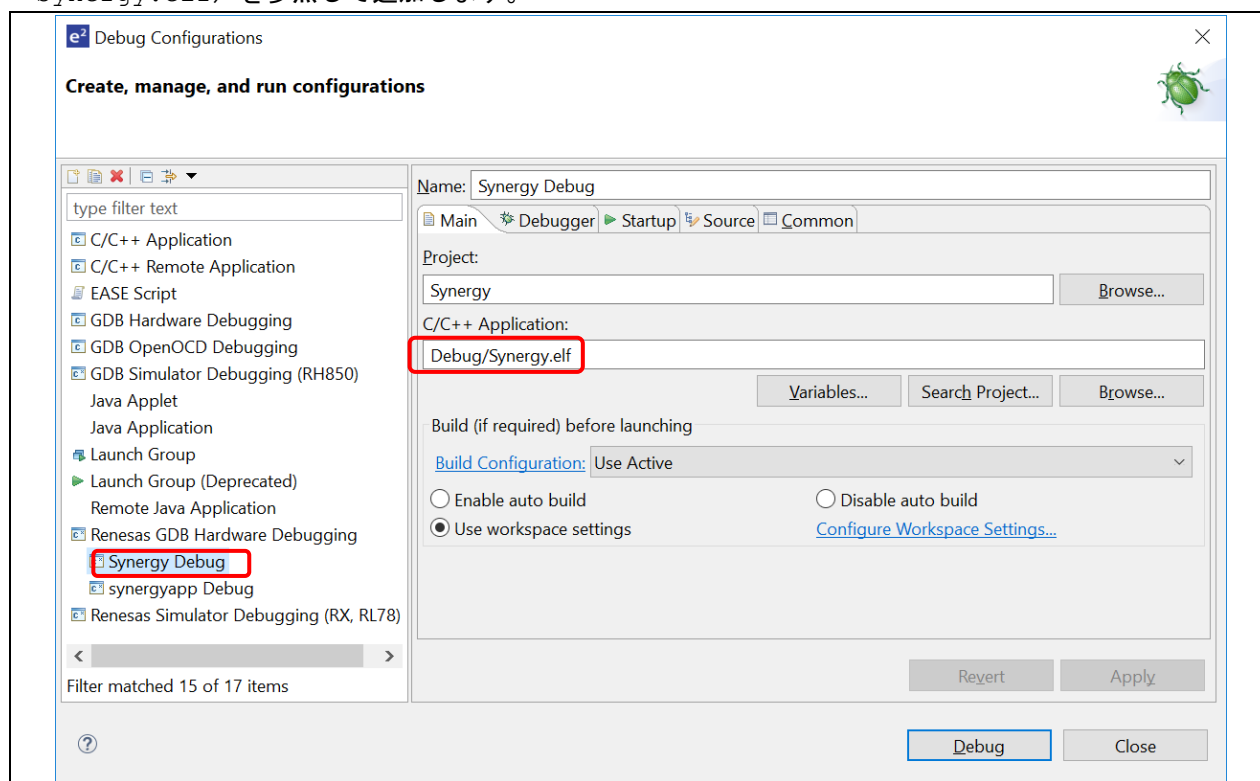


図 84. デバッグ - ロードモジュールの選択

5. **[Debugger]** (デバッガ) タブに切り替え、J-Link ARM と R7FS7G2 を対象デバイスとして設定します。
 - [Debug hardware] (デバッグハードウェア): **J-link ARM**
 - [Target Device] (対象デバイス): **R7FS7G27H**
6. **[Apply]** (適用) ボタンをクリックして、設定を確認します。
7. **[Debug]** (デバッグ) ボタンをクリックしてデバッグ起動構成を実行し、J-Link と Synergy ボードに接続します。

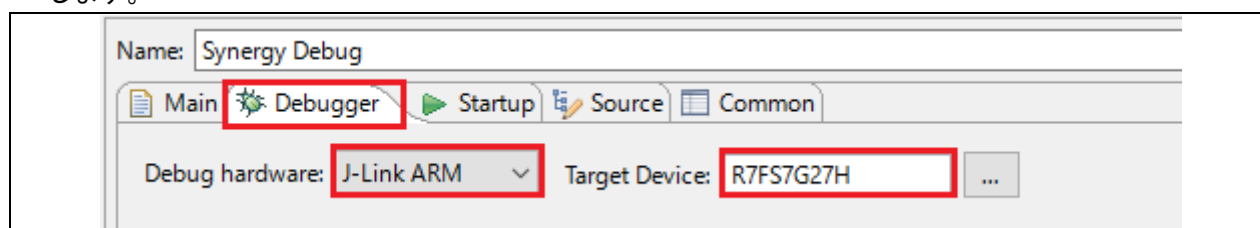


図 85. デバッグ - 接続設定の変更

8. 接続を成功させるために、[Debug] (デバッグ) ビューではターゲットのデバッグ情報がツリー階層内で表示されています。プログラムのエントリポイントは、startup_S7G2.c 内の Reset_Handler() に設定されています。

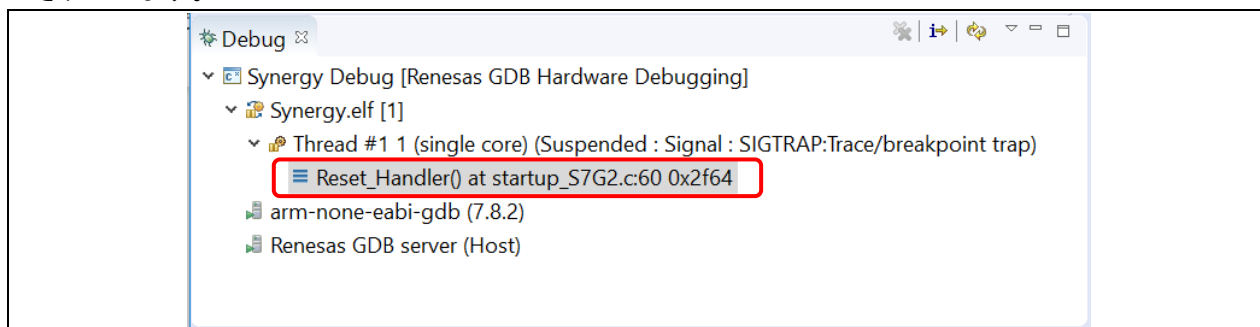


図 86. デバッグ - [Debug] (デバッグ) ビュー内で表示されているユーザのターゲット接続

5.2 新しいデバッグ構成の作成 (Creating a New Debug Configuration)

新しいデバッグ構成を作成する最も簡単な方法は、既存の構成を複製することです。以下の手順で実施できます。

1. [Debug Configuration] (デバッグ構成) ウィンドウを開きます（「図 83」を参照）。
2. **[Debug Configurations]** (デバッグ構成) ウィンドウで、いずれかのデバッグ構成（例えば、[Synergy Debug] (Synergy デバッグ)) を選択し、現在選択されている起動構成を複製する  アイコンをクリックします。新しいデバッグ起動構成（例えば、Synergy Debug (1)）が作成されます。
3. 新しいデバッグ構成は、5.1 章で説明した方法で構成できます。

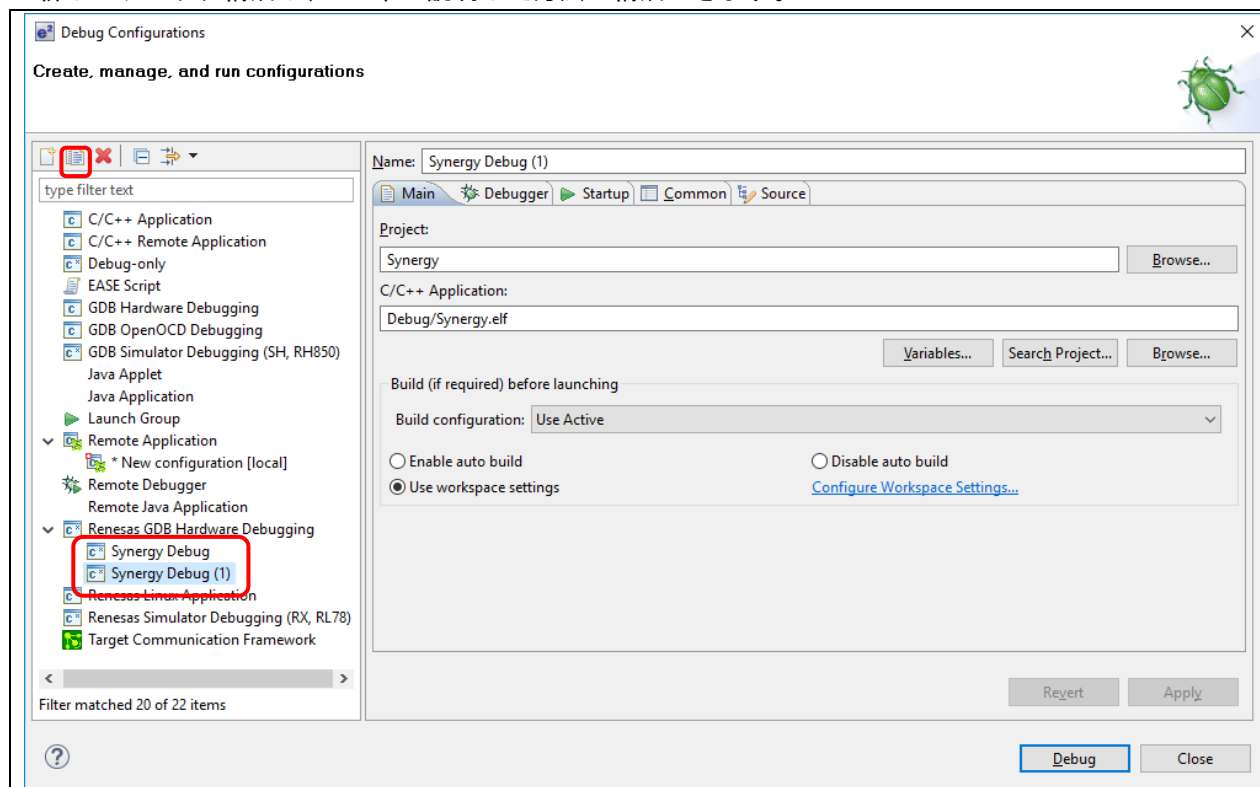


図 87. デバッグ - 選択されているデバッグ起動構成の複製

5.3 基本的なデバッグ機能 (Basic Debugging Features)

本章では、e² studio がサポートする典型的な [Debug] (デバッグ) ビューを説明します。

- 標準的な GDB デバッグ (Eclipse IDE フレームワークによるサポート) : [Breakpoints] (ブレークポイント)、[Expressions] (数式)、[Registers] (レジスタ)、[Memory] (メモリ)、[Disassembly] (逆アセンブル)、および [Variables] (変数)[MMU] ビューは Synergy のサポート対象外であることに注意してください。
- 標準的な GDB Debug に対するルネサスの拡張 : [IO Registers] (IO レジスタ)、[Eventpoints] (イベントポイント)、[Trace] (トレース)、および [Fault Status] (障害ステータス)。

[Debug Toolbar] (デバッグツールバー) を開くには、プルダウンメニューをクリックし、[Show Debug Toolbar] (デバッグツールバーの表示) オプションをオンにします。以下のスクリーンショットは、[Debug] (デバッグ) ビューの便利なツールバーです。

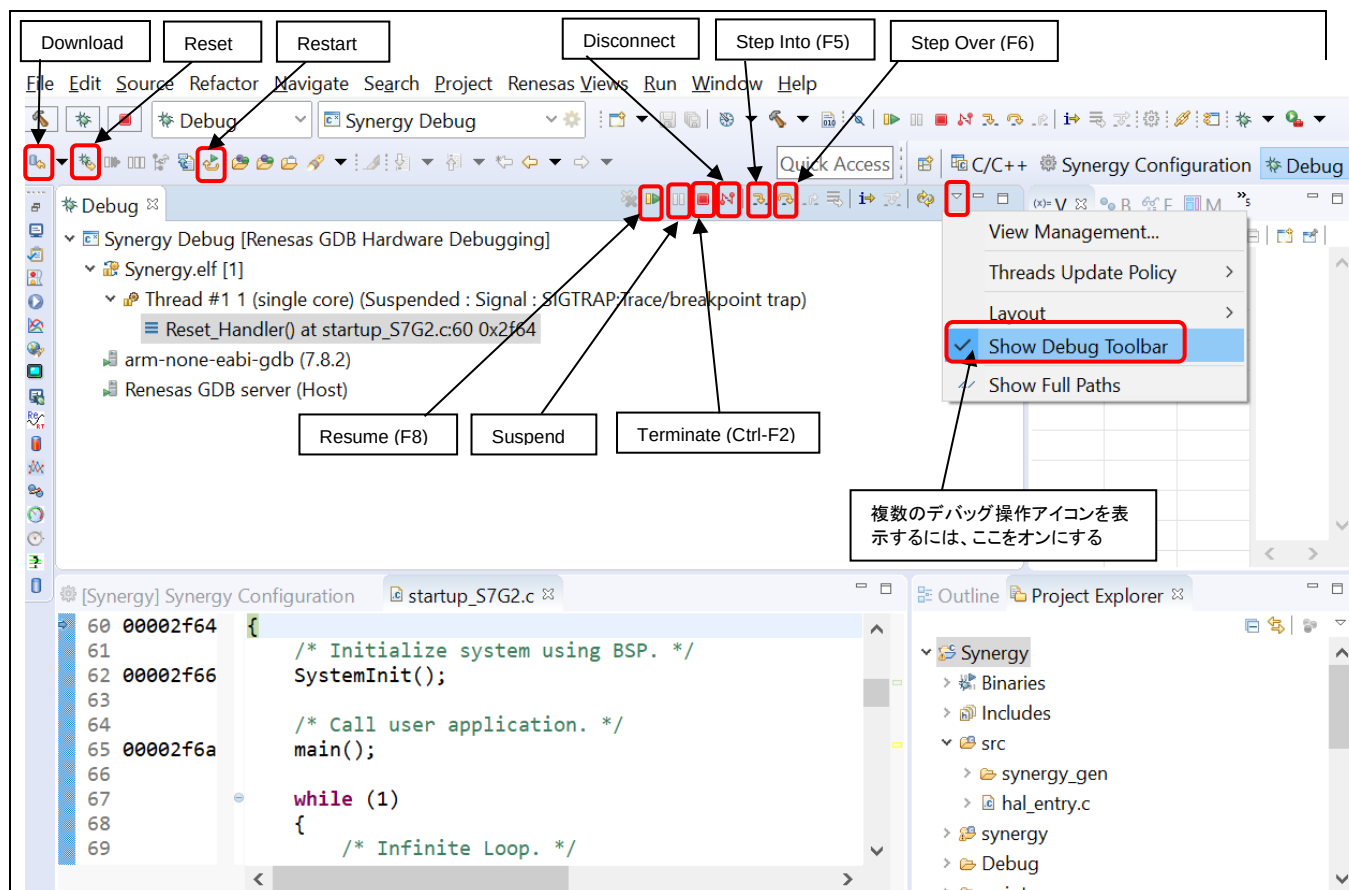


図 88. デバッグ - [Debug] (デバッグ) ビューの便利なツールバー

プログラムを実行するには、 ボタンをクリックするか、[F8] を押します。

ブレークポイントが見つかったとき、または ボタンをクリックしたときに、プログラムの実行が中断されます。プログラムの実行が中断された時点で、ユーザは以下の操作を実行できます。

- ボタンまたは [F5] を使用して、現在実行されているコード行で次のメソッド呼び出しにステップインします。
- ボタンまたは [F6] を使用して、現在実行されているコード行で次のメソッド呼び出しにステップオーバーします（実行しますが、メソッド内に入りません）。
- ボタンをもう一度クリックして、プログラムの実行を再開することもできます。

デバッグプロセスを停止するには、 ボタンをクリックして選択されているデバッグセッションやデバッグプロセスを終了するか、 ボタンをクリックして選択されているプロセスからデバッグを接続解除することができます。

他の使用可能な操作は、以下の通りです。

-  ボタンをクリックしてリセットし、プログラムを実行します。**[Debug]** (デバッグ) 構成の中でブレークポイントが構成されている場合、main() で停止する可能性があります。
-  ボタンをクリックして、パワーオンリセット時のエントリポイントにプログラムをリセットします。
-  ボタンを使用して、バイナリファイルをターゲットシステムに再ダウンロードします。

5.3.1 [Breakpoints] (ブレークポイント) ビュー

[Breakpoints] (ブレークポイント) ビューは、プログラムの実行可能な行に設定済みのブレークポイントを保存します。デバッグの実行中にブレークポイントが有効になっている場合、そのコード行を実行する前に実行は中断されます。e² studio では、IDE 内でソフトウェアブレークポイントとハードウェアブレークポイントを明確に設定できます。マーカーバーをダブルクリックして追加したブレークポイントは、デフォルトではハードウェアブレークポイントになります。ハードウェアリソースがない場合、ブレークポイントの設定に失敗します。ハードウェアブレークポイントの設定に失敗した場合、エラーメッセージが表示され、ソフトウェアブレークポイントに変更することをユーザに求めます。

ハードウェアブレークポイントかソフトウェアブレークポイントを選択するには、

1. マーカーバーを右クリックし、コンテキストメニューをポップアップします。
2. ハードウェアブレークポイントにする場合、**[Breakpoint Types]** (ブレークポイントの種類) → **[e² studio Breakpoint]** (e² studio ブレークポイント) を選択します。
3. ソフトウェアブレークポイントにする場合、**[Breakpoint Types]** (ブレークポイントの種類) → **[C/C++ Breakpoints]** (C/C++ ブレークポイント) を選択します。

ブレークポイントを設定するには、

1. 例えば、startup_S7G2.c の 62 行目で、**[C/C++ Editor]** (C/C++ エディタ) ペインの左端に表示されているマーカーバーをダブルクリックし、ブレークポイントを設定します。選択した **[Breakpoint Type]** (ブレークポイントの種類) に応じて、ドット  (ハードウェアブレークポイント) または  (ソフトウェアブレークポイント) がマーカーバーに表示されます。**[Breakpoint Type]** (ブレークポイントの種類) は、デフォルトでハードウェアブレークポイントになっています。
2. 代わりに、マーカーバーを右クリックし、**[Toggle Hardware Breakpoint]** (ハードウェアブレークポイントの切り替え) または **[Toggle Software Breakpoint]** (ソフトウェアブレークポイントの切り替え) を選択して、ハードウェアブレークポイント  またはソフトウェアブレークポイント  を設定することもできます。
3. **[Windows]** (ウィンドウ) → **[Show View]** (ビューの表示) → **[Breakpoints]** (ブレークポイント) または  アイコンをクリックする (代わりに、ショートカットキー **[ALT]+[Shift]+[Q]**、**[B]** を使用する) 方法で **[Breakpoints]** (ブレークポイント) ビューを開き、対応するソフトウェアブレークポイントセットを表示することができます。**[Breakpoints]** (ブレークポイント) ビューで、ブレークポイントを有効または無効にすることができます。

ブレークポイントを無効にする場合、ユーザは特定のブレークポイントを無効にするか、すべてのブレークポイントをスキップするか、選ぶことができます。

1. 特定のブレークポイントを無効にするには、**[C/C++ Editor]** (C/C++ エディタ) ペインの左端に配置されているソフトウェアブレークポイント  またはハードウェアブレークポイント  を右クリックし、**[Disable Breakpoint]** (ブレークポイントの無効化) を選択するか、**[Breakpoints]** (ブレークポイント) ビューで関連する行のチェックをオフにします。無効になったブレークポイントは、白いドット (○ または ) の形式で表示されます。
2. すべてのブレークポイントをスキップするには、**[Breakpoints]** (ブレークポイント) ビューで、 アイコンをクリックします。エディタペインと **[Breakpoints]** (ブレークポイント) ビューの両方に、バックスラッシュの付いた青いドットが表示されます。

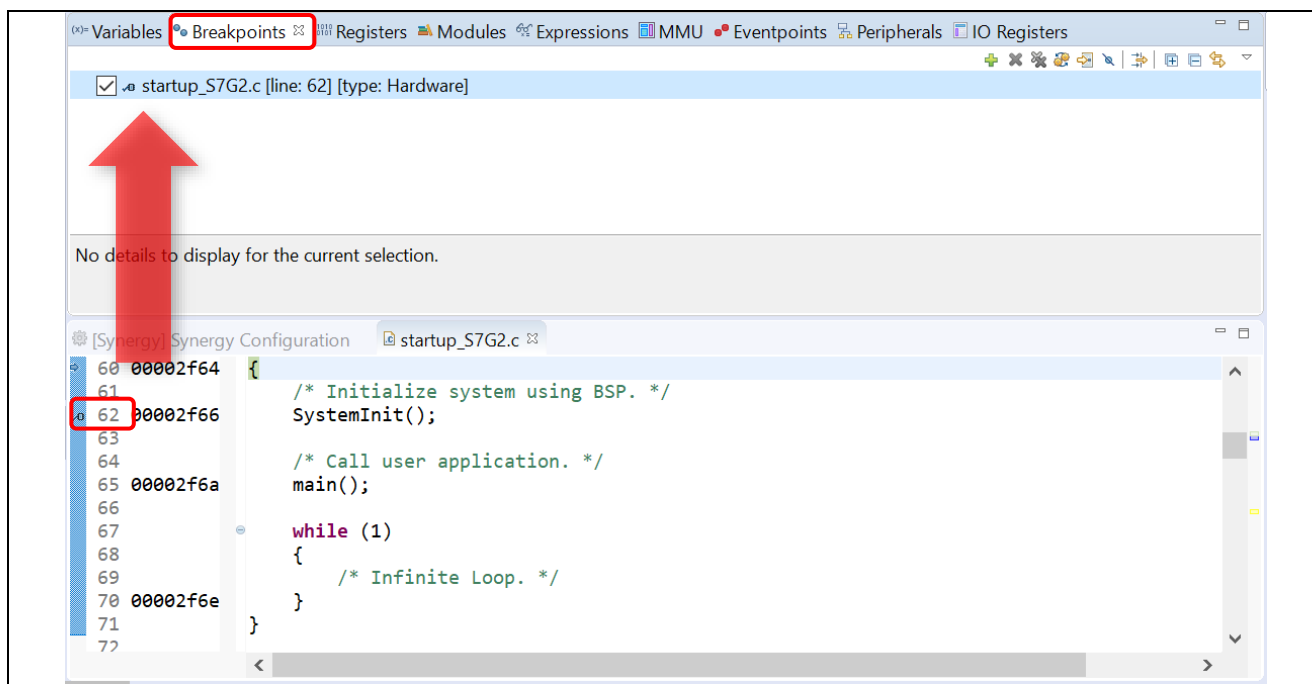


図 89. [Debug] (デバッグ) - [Breakpoints] (ブレイクポイント) ビュー

5.3.2 [Expressions] (式) ビュー

[Expressions] (式) ビューで、デバッグ中のグローバル変数、ローカル変数、静的変数の値をモニタできます。

変数を表示するには、以下の手順に従ってください。

1. [Windows] (ウィンドウ) → [Show View] (ビューの表示) → [Expressions] (式) または  アイコンをクリックして、[Expressions] (式) ビューを開きます。
2. 変数 (例えば、r_cgc.c 内の g_cgc_version) を [Expressions] (式) ビューにドラッグアンドドロップします。
3. 代わりに、変数を右クリックし、[Add Watch Expression...] (新しい式を追加...) メニュー項目を選択して、その変数を [Expressions] (式) ビューに追加することもできます。

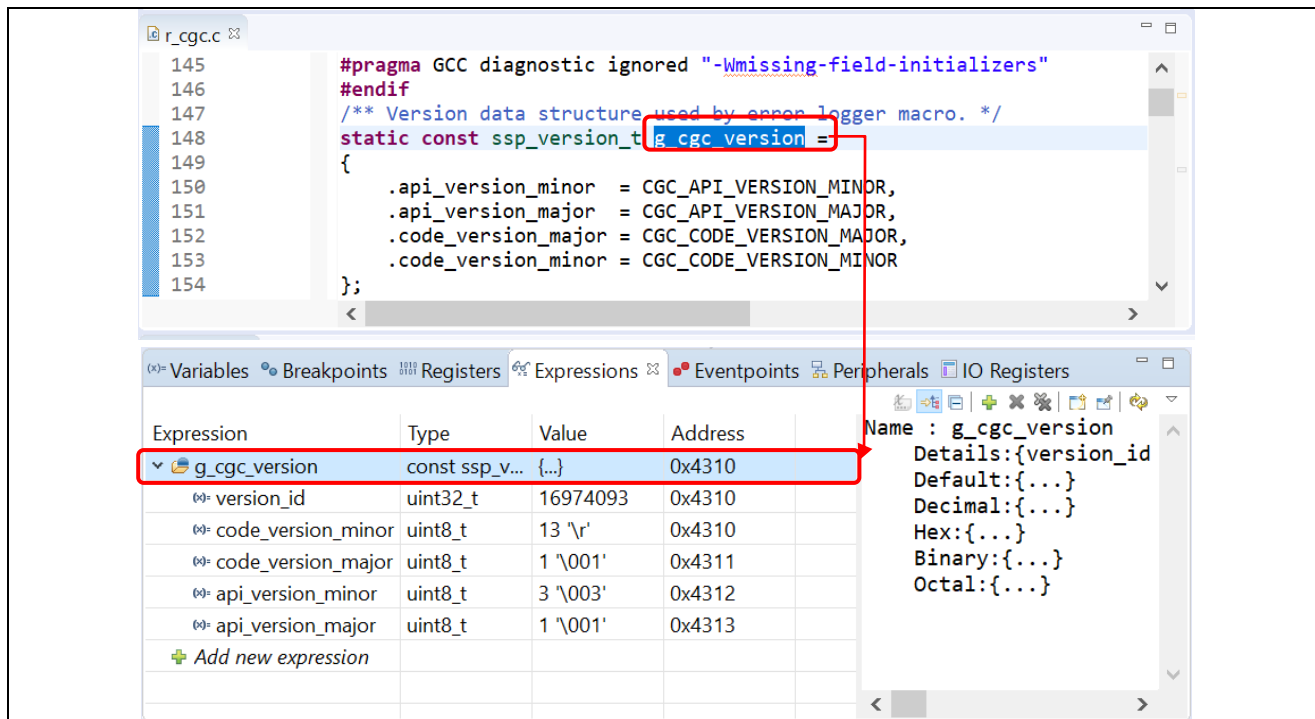


図 90. デバッグ - [Expressions] (式) ビュー

5.3.3 [Registers] (レジスタ) ビュー

[Registers] (レジスタ) ビューで、Synergy 内の汎用レジスタに関する情報がリスト表示されます。プログラムが停止した時点で、変更された値は強調して表示されます。

1. **[Windows] (ウィンドウ) → [Show View] (ビューの表示) → [Registers] (レジスタ)** をクリックするか、 アイコンをクリックして、**[Registers] (レジスタ) ビュー**を開きます。
2. いずれかのレジスタをクリックすると、さまざまな基数フォーマットで値が表示されます。

プログラムが停止した時点で、変更された値は **[Registers] (レジスタ) ビュー**内で強調して表示されます (例：黄色)。

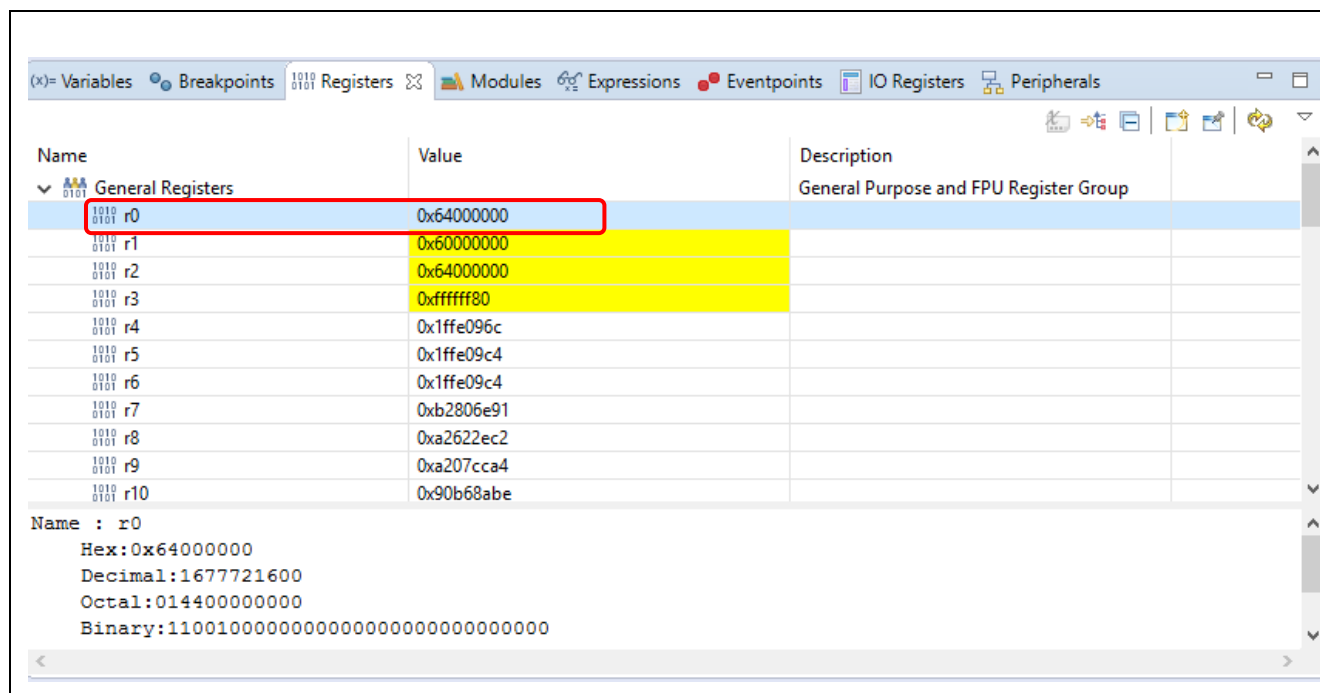


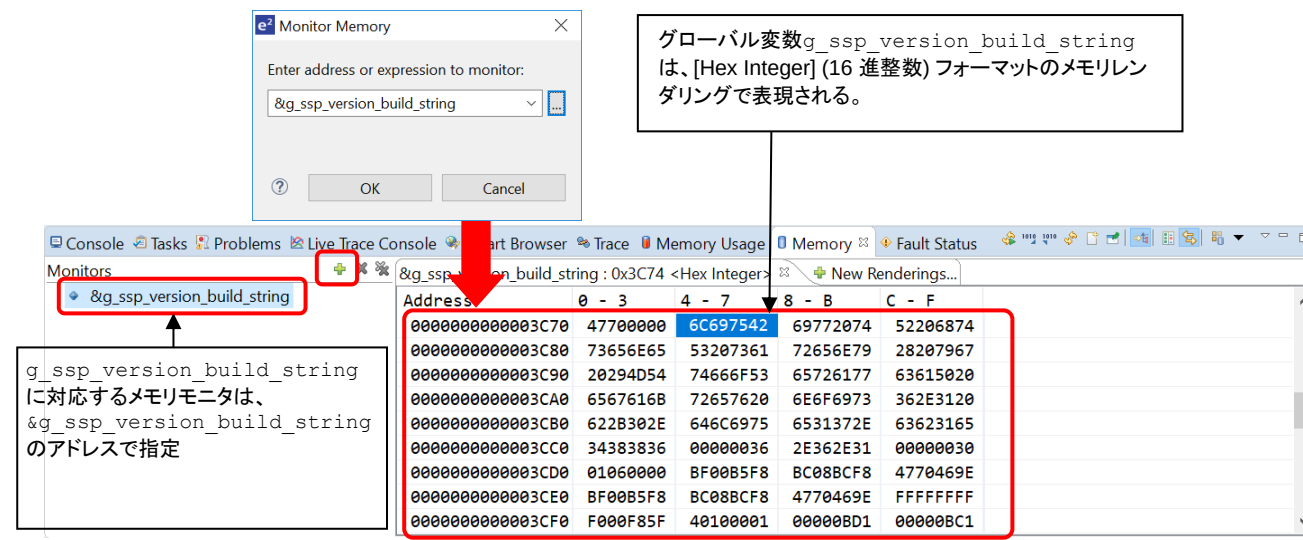
図 91. デバッグ - [Registers] (レジスタ) ビュー

5.3.4 [Memory] (メモリ) ビュー

[Memory] (メモリ) ビューで、ユーザはメモリモニタ内に存在している変数の表示と編集を行うことができます。各モニタは、「ベースアドレス」と呼ばれる場所で指定されたメモリのセクションを表しています。各メモリモニタ内のメモリデータは、さまざまな「メモリレンダリング」(メモリ表現形式)で表現されます。これらは、事前定義されたデータフォーマット(例えば、16進整数、符号付き整数、符号なし整数、ASCII、イメージなど)です。

変数(例えば、g_ssp_version_build_string)のメモリを表示するには、以下の手順に従ってください。

1. [Windows] (ウィンドウ) → [Show View] (ビューの表示) → [Memory] (メモリ) または  アイコンをクリックして、[Memory] (メモリ) ビューを開きます。
2.  アイコンをクリックすると、[Monitor Memory] (メモリのモニタ) ダイアログボックスが表示されます。変数のアドレス &g_ssp_version_build_string を入力します。



The screenshot shows the e² studio interface with the Memory view open. A dialog box titled "Monitor Memory" is displayed, with the input field containing "&g_ssp_version_build_string". Below the dialog, the Memory view shows a table of memory addresses and their corresponding data. The variable g_ssp_version_build_string is highlighted in the Monitors list, and its address is used to filter the memory dump.

グローバル変数g_ssp_version_build_stringは、[Hex Integer] (16進整数) フォーマットのメモリレンダリングで表現される。

Address	0 - 3	4 - 7	8 - B	C - F
0000000000003C70	47700000	6C697542	69772074	52206874
0000000000003C80	73656E65	53207361	72656E79	28207967
0000000000003C90	20294D54	74666F53	65726177	63615020
0000000000003CA0	6567616B	72657620	6E6F6973	362E3120
0000000000003CB0	622B302E	646C6975	6531372E	63623165
0000000000003CC0	34383836	00000036	2E362E31	00000030
0000000000003CD0	01060000	BF00B5F8	BC08BCF8	4770469E
0000000000003CE0	BF00B5F8	BC08BCF8	4770469E	FFFFFFFF
0000000000003CF0	F000F85F	40100001	00000BD1	00000BC1

g_ssp_version_build_string
に対応するメモリモニタは、
&g_ssp_version_build_string
のアドレスで指定

図 92. デバッグ - [Memory] (メモリ) ビュー

変数 `g_ssp_version_build_string` に対して新しいレンダリングフォーマット (例えば、ASCII) を追加するには、以下の手順に従ってください。

1. タブ **New Renderings...** をクリックし、**[ASCII]** を選択してレンダリングを追加します。

この結果、`[&g_ssp_version_build_string <Hex Integer>]` タブの隣に、`[&g_ssp_version_build_string <ASCII>]` という新しいタブが作成されます。

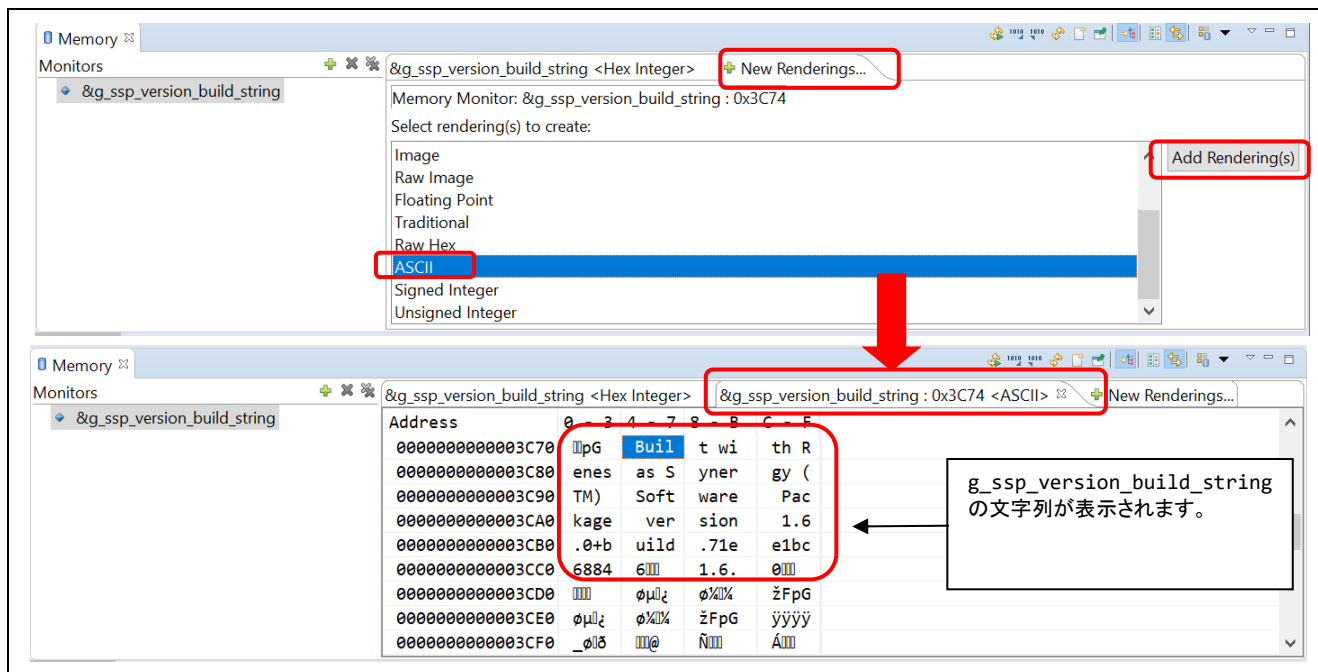


図 93. デバッグ - [Memory] (メモリ) ビュー内の新しいレンダリング

5.3.5 Memory Usage (メモリ使用状況) ビュー

[Memory Usage] (メモリ使用状況) を使用して、プロジェクトから (*.map) ファイルまたはライブラリリストファイル (*.lbp) の情報を取得することができます。このビューは、合計メモリサイズ、ROM と RAM の使用比率、およびプロジェクト内で使用しているセクション、オブジェクト、シンボル、モジュール、ベクタ、クロスリファレンスの詳細情報を表示します。

バージョン 7.3 およびそれ以降の e² studio は、ROM と RAM の各メモリ領域内での使用状況を表示するグラフィカルビューをサポートしています。

[Memory Usage] (メモリ使用状況) ビューを開くには、[Window] (ウィンドウ) → [Show View] (ビューの表示) → [Others...] (その他...) メニューをクリックします。[Show View] (ビューの表示) ダイアログで、[Memory Usage] (メモリ使用状況) を選択し、[Open] (開く) をクリックします。

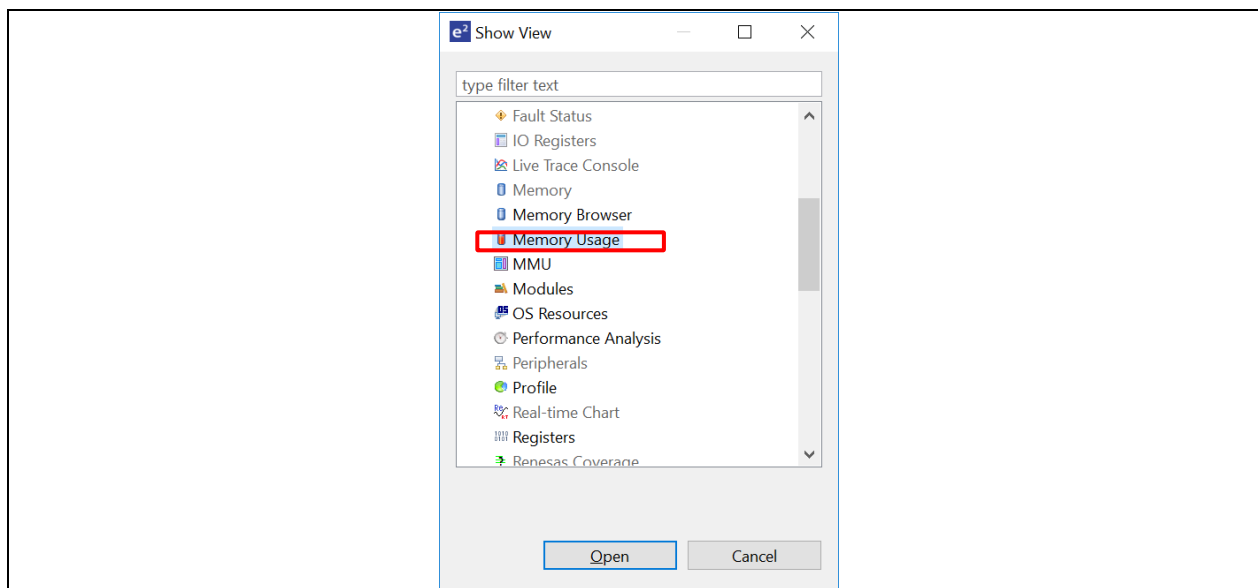


図 94. [Memory Usage] (メモリ使用状況) ビューの表示

[Memory Usage] (メモリ使用状況) ビューには 3 つの領域があります。

1. [Group size] (グループサイズ) 領域
2. [Memory Region Usage] (メモリ領域の使用状況) 領域 ([Device Memory Usage] (デバイスメモリ領域の使用状況) 領域は今のところサポートされていません)
3. [Detail table] (詳細表) 領域

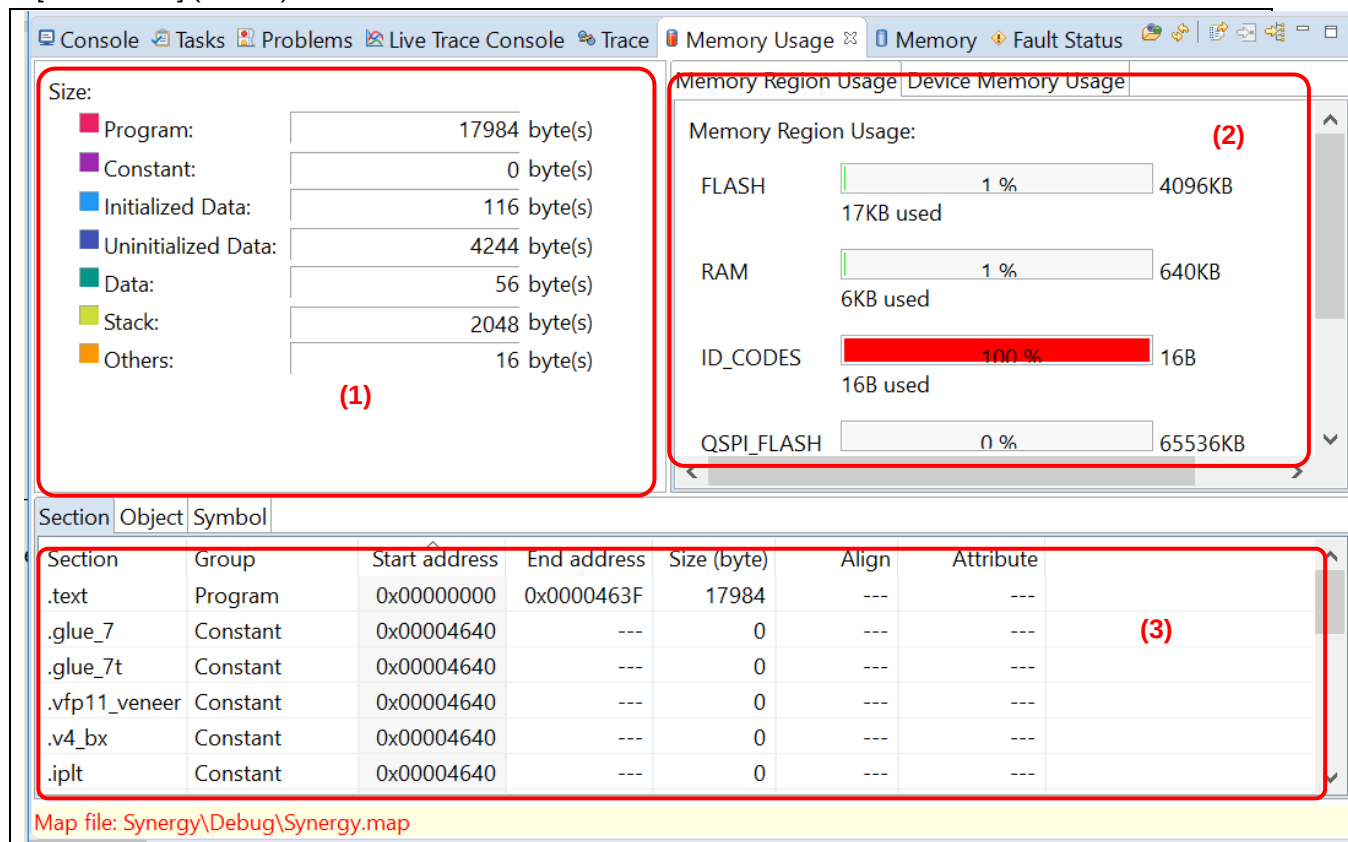


図 95. [Regions of Memory Usage] (メモリ領域の使用状況) ビュー

[Memory Usage] (メモリ使用状況) ビューは、以下の操作をサポートしています。

: マップファイルまたはライブラリリストファイルを選択し、[Memory Usage] (メモリ使用状況) で表示します。

: [Memory Usage] (メモリ使用状況) ビューのすべての情報を更新します。

: エディタ内で *.map ファイルまたは *.lbp ファイルを開きます (Synergy ライブラリプロジェクト内にはライブラリリストファイルは存在しません)。

: 選択したプロジェクトの [Map file output] (マップファイル出力) ページを開きます。

: 選択したプロジェクトの [Section] (セクション) ページを開きます。

5.3.6 [Disassembly] (逆アセンブル) ビュー

[Disassembly] (逆アセンブル) ビューには、ロード済みのプログラムが、比較の目的でアセンブリ命令とソースコードを組み合わせた形式で表示されます。現在実行されている行は、ビュー内で矢印マーカーにより強調表示されます。[Disassembly] (逆アセンブル) ビューで、ユーザはアセンブラ命令に対するブレークポイントの設定、それらのブレークポイントの有効化または無効化、逆アセンブル命令のステップ実行、さらにプログラム内にある特定の命令へのジャンプを実行できます。

C とアセンブリ両方のコードを組み合わせたモードで表示するには、

1. [Windows] (ウィンドウ) → [Show View] (ビューの表示) → [Disassembly] (逆アセンブル) または  アイコンをクリックして、[Disassembly] (逆アセンブル) ビューを開きます。
2. アセンブリソースと C ソース (アクティブなデバッグコンテキスト) の間で同期を有効にするには、 アイコンをクリックします。
3. [Disassembly] (逆アセンブル) ビューで、アドレス列を右クリックし、[Show Opcodes] (オペコードの表示) と [Show Function Offsets] (関数オフセットの表示) を選択します。

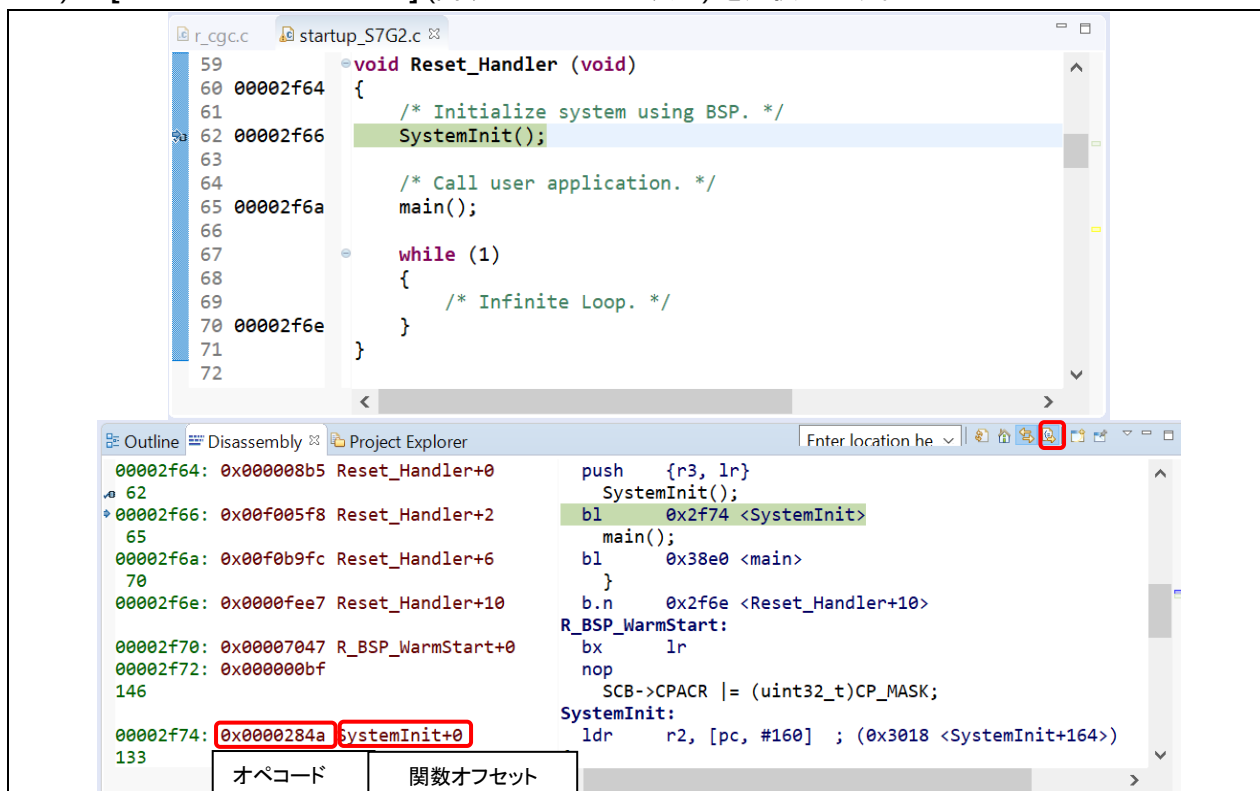


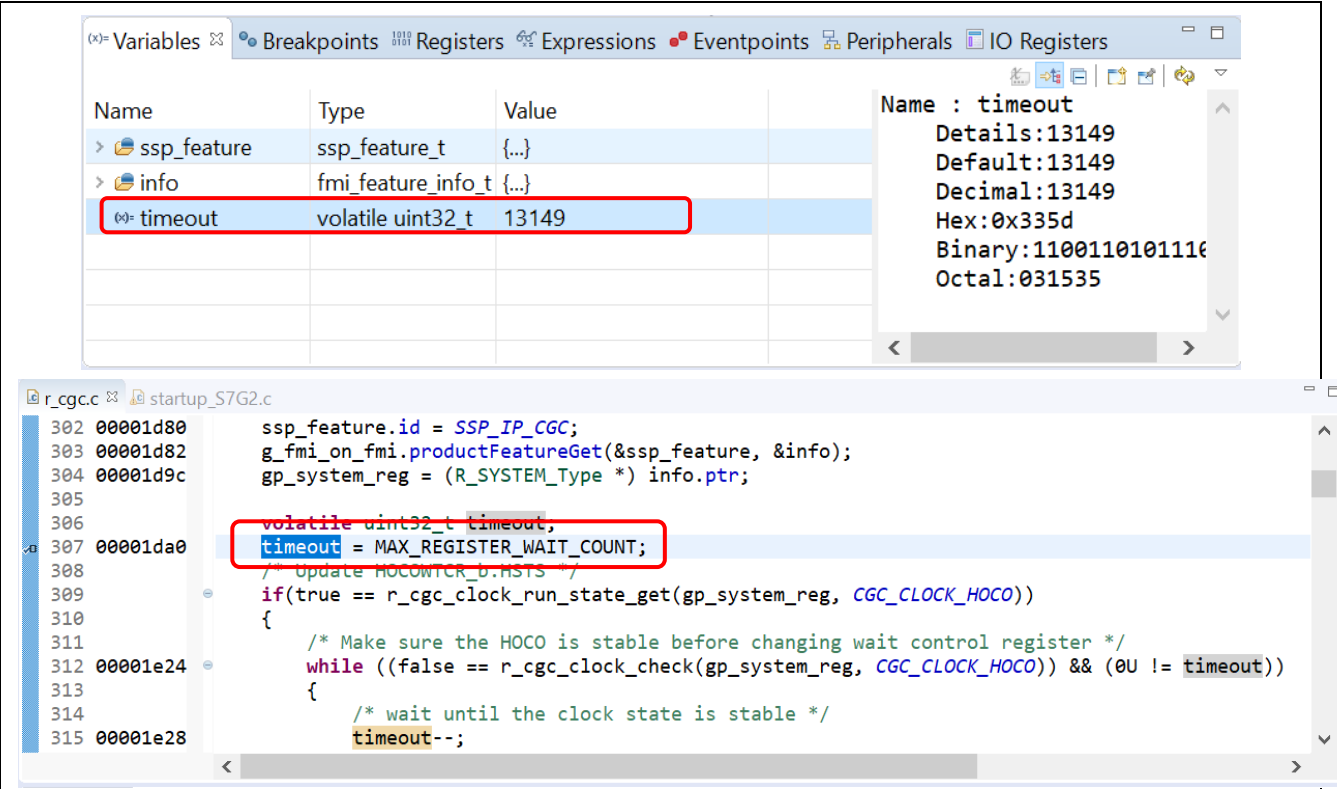
図 96. デバッグ - [Disassembly] (逆アセンブル) ビュー

5.3.7 [Variables] (変数) ビュー

[Variables] (変数) ビューでは、現在のプログラムスコープ内にあるすべての有効なローカル変数が表示されます。

ローカル変数（例えば、R_CGC_Init() 関数の timeout）を表示するには、以下の手順に従ってください。

1. [Windows] (ウィンドウ) → [Show View] (ビューの表示) → [Variables] (変数) または  アイコンをクリックして、[Variables] (変数) ビューを開きます。
2. 関数 R_CGC_Init () にステップインし、ローカル変数「timeout」の値を表示します。



The screenshot displays the e² studio IDE interface. At the top, the 'Variables' view is active, showing a table of local variables. The 'timeout' variable is highlighted with a red box. To the right of the table, the details for the 'timeout' variable are shown, including its value (13149) and various representations (Hex, Binary, Octal).

Name	Type	Value
ssp_feature	ssp_feature_t	{...}
info	fmi_feature_info_t	{...}
timeout	volatile uint32_t	13149

Below the table, the details for the 'timeout' variable are listed:

```

Name : timeout
Details:13149
Default:13149
Decimal:13149
Hex:0x335d
Binary:1100110101110
Octal:031535
  
```

The source code window shows the function R_CGC_Init in startup_S7G2.c. The line where 'timeout' is initialized is highlighted with a red box:

```

307 00001da0 volatile uint32_t timeout,
          timeout = MAX_REGISTER_WAIT_COUNT;
  
```

図 97. デバッグ - [Variables] (変数) ビュー

5.3.8 [IO Registers] (IO レジスタ) ビュー

[IO Registers] (IO レジスタ) は、Special Function Registers (SFR、特殊機能レジスタ) とも呼ばれています。[IO Registers] (IO レジスタ) ビューでは、ターゲット固有の IO ファイル内で定義されているすべてのレジスタが表示されます。ユーザは、その固有の IO レジスタを **[Selected Registers]** (選択済みレジスタ) ペインに追加する方法で、[IO Registers] (IO レジスタ) ビューをさらにカスタマイズすることもできます。

選択した IO レジスタを表示するには、

1. **[Renesas Views]** (ルネサスビュー) → **[Debug]** (デバッグ) → **[IO Registers]** (IO レジスタ) または アイコンをクリックして、[IO Registers] (IO レジスタ) ビューを開きます。
2. **[All Registers]** (すべてのレジスタ) タブで、[IO Registers] (IO レジスタ) ビュー内にあるいずれかのモジュール (CAC など) を表示します。そのモジュールの IO レジスタリストを展開します。
3. そのレジスタ (CAICR と CASTR など) を、**[Selected Registers]** (選択済みレジスタ) ペインにドラッグアンドドロップします。IO レジスタの隣にある緑のドット ● は、選択されているレジスタのステータスを表します。
4. **[Selected Registers]** (選択済みレジスタ) タブに切り替え、選択されている **[IO Registers]** (IO レジスタ) を表示します。

展開された IO レジスタリストを **[All Registers]** (すべてのレジスタ) ペインにロードするまでに、多少時間がかかることがあります。したがって、**[Selected Registers]** (選択済みレジスタ) ペインでカスタマイズを行い、選択された複数の IO レジスタを表示することを推奨します。

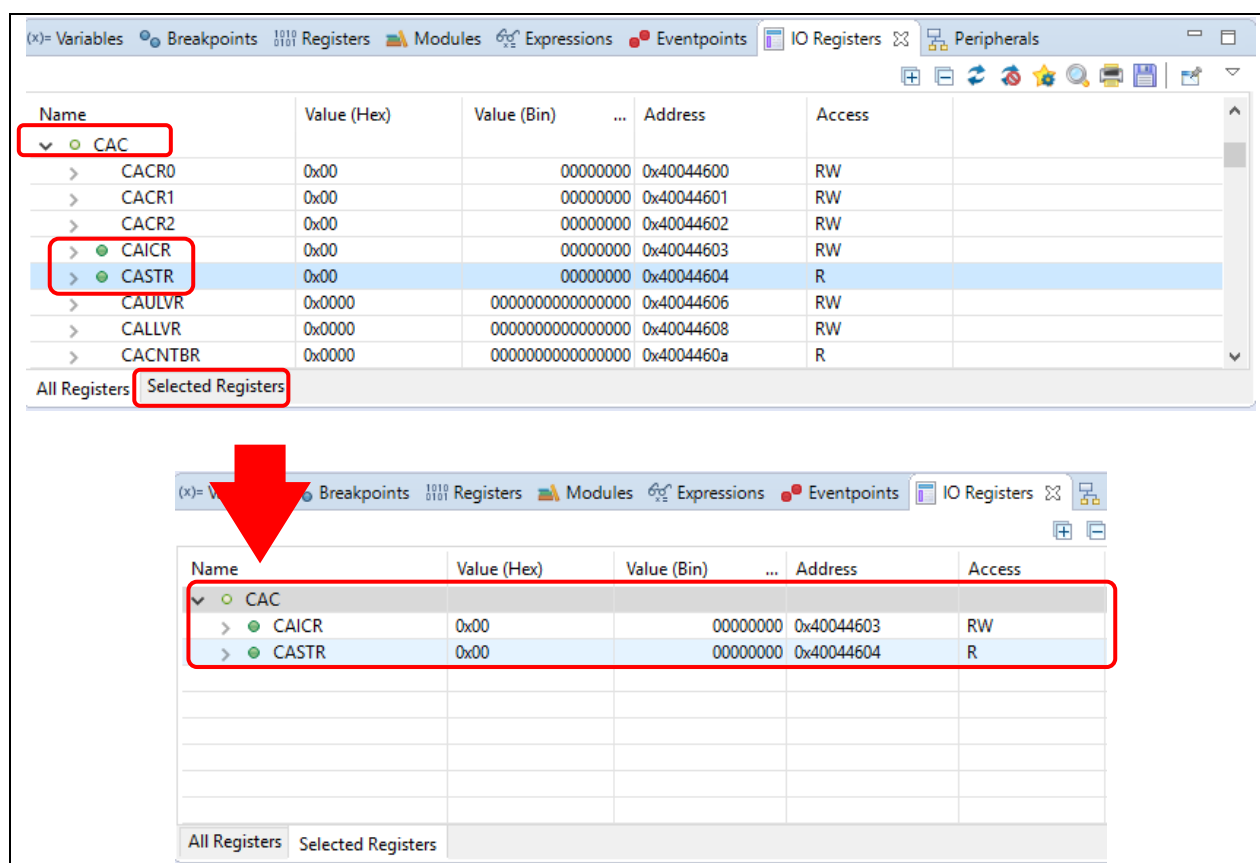


図 98. デバッグ - [IO Registers] (IO レジスタ) ビュー

5.3.9 [Eventpoints] (イベントポイント) ビュー

「イベント」とは、プログラムの実行中に、ブレーク機能またはトレース機能を実行する目的で設定された条件の組み合わせです。[Eventpoints] (イベントポイント) ビューでは、さまざまなカテゴリ（例えば、trace start（トレース開始）、trace stop（トレース停止）、または event break（イベントブレーク））の定義済みイベントの設定、および表示を行うことができます。

Synergy プロジェクトでは、データアクセスイベントのブレークがサポートされています。エミュレータは、指定した単一アドレスまたは指定したアドレス範囲内で、指定した条件下で発生したアクセスを検出します。この結果、アドレスとデータの複雑な組み合わせ条件を設定できます。

イベントの組み合わせ（例えば、OR、AND（累積型）、および Sequential（シーケンシャル））を複数のイベントに適用することもできます。

表 2. イベントの組み合わせに関する説明

イベントの組み合わせ	説明
OR	指定した複数のイベントのうちいずれか 1 つまたは複数が発生したときに、条件が満たされます。
AND (累積型)	タイミングにかかわらず、指定した複数のイベントのすべてが発生したときに、条件が満たされます。
Sequential (シーケンシャル)	指定した複数のイベントが指定した順序で発生したときに、条件が満たされます。

1. アドレスやデータが一致したとき（例えば、`g_bsp_leds` がアクセスされたとき）にグローバル変数に対してイベントブレークを設定するには、以下の手順に従ってください。
 - A. [Renesas Views] (ルネサスビュー) → [Debug] (デバッグ) → [Eventpoints] (イベントポイント) または  アイコンをクリックして、[Eventpoints] (イベントポイント) ビューを開きます。
 - B. [Event Break] (イベントブレーク) オプションをダブルクリックし、[Edit Event Break] (イベントブレークの編集) ダイアログボックスを開きます。
 - C. [Add...] (追加...) ボタンをクリックして続けます。
デフォルトで、[Data Access] (データアクセス) イベントポイントタイプが選択されています。

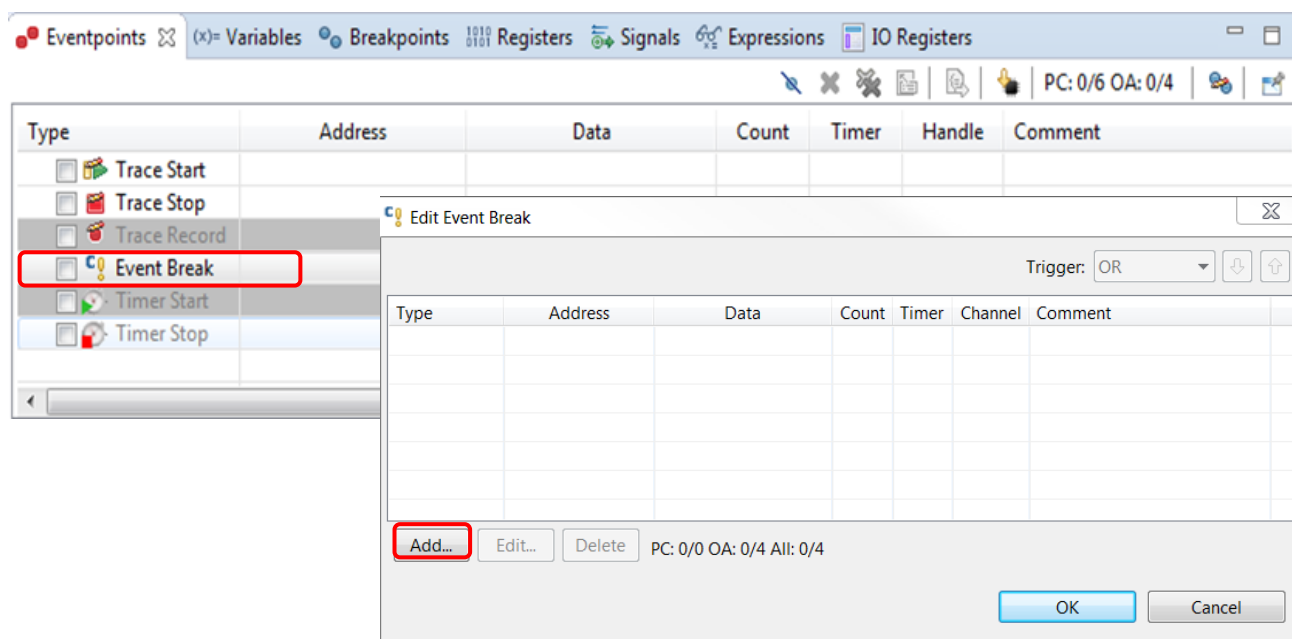


図 99. デバッグ - [Eventpoints] (イベントポイント) ビュー (1/2)

2. **[Address Settings]** (アドレス設定) タブに移動し、[...] アイコンをクリックして、g_bsp_leds シンボルを参照します。(このグローバル変数のアドレスは &g_bsp_leds です。)
3. 次に、**[Data Access Settings]** (データアクセス設定) タブに切り替え、[Read/Write] (読み取り/書き込み) の選択肢を **[Read]** (読み取り) に設定します。
4. **[OK]** をクリックして先に進みます。

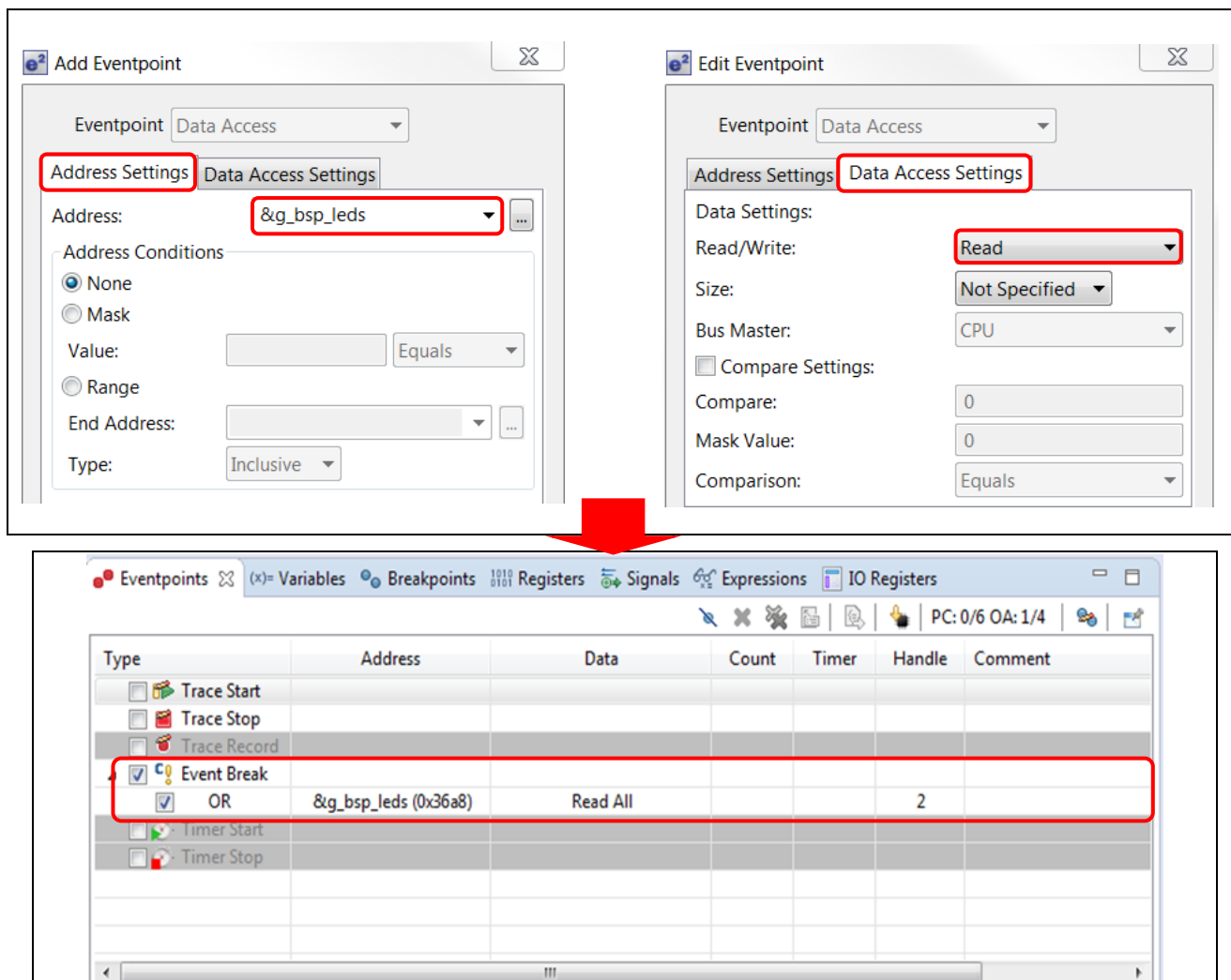


図 100. デバッグ - [Eventpoints] (イベントポイント) ビュー (2/2)

5. リセットを実行し、最初からプログラムを実行します。
6. 以下の図に、変数 `g_bsp_leds` がアクセス（読み取り）されたときに、プログラムが停止した状態を示します。

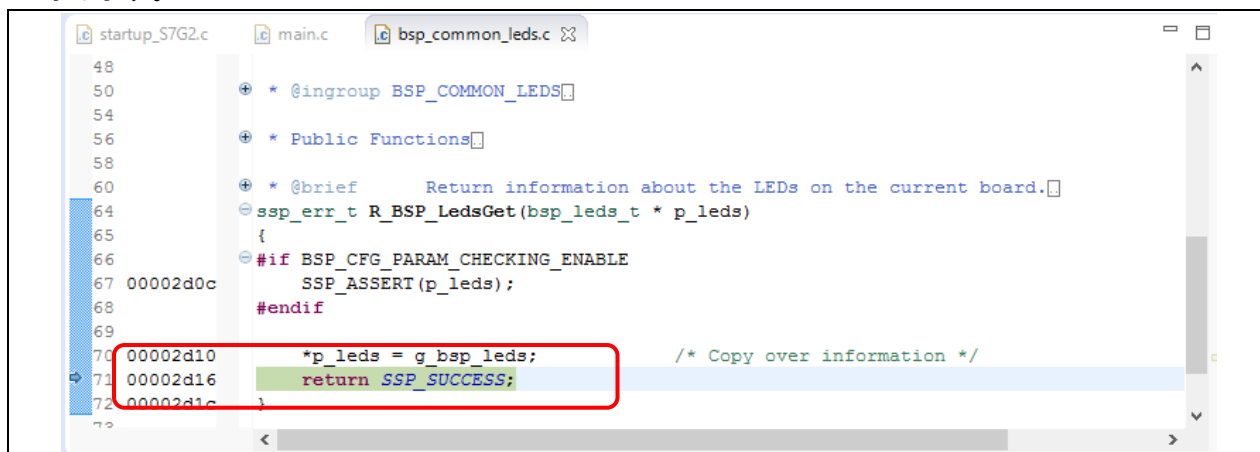


図 101. デバッグ - イベントブレークの実行

5.3.10 [Trace] (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。取得したトレース情報は **[Trace]** (トレース) ビューに表示されます。その結果、ユーザはプログラムの実行フローを追跡し、問題が発生した場所を探することができます。

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。

プログラムが中断されるまでのトレースを設定するには、ユーザは以下の手順を実行できます。

1. **[Renesas Views]** (ルネサスビュー) → **[Debug]** (デバッグ) → **[Trace]** (トレース) または  アイコンをクリックして、**[Trace]** (トレース) ビューを開きます。
2.  アイコンを選択して、**[Trace]** (トレース) ビューを有効にします。

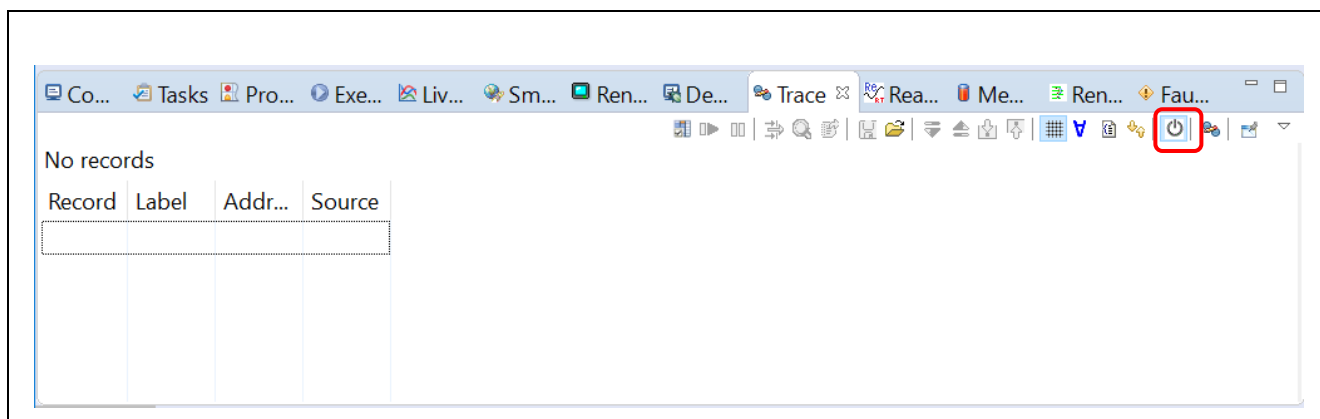


図 102. デバッグ - [Trace] (トレース) ビューの有効化

- プログラムの実行を開始した後、ブレークポイントを使用するか、[Debug] (デバッグ) ツールバーの **[Suspend]** (中断) ボタンをクリックしてプログラムの実行を停止します。この時点でトレースメモリに格納された内容は、トレース結果として表示されます。
- 表示モードを選択するには、対応するボタンをクリックします。

以下の図に、main()関数が実行される前のトレース結果を示します。

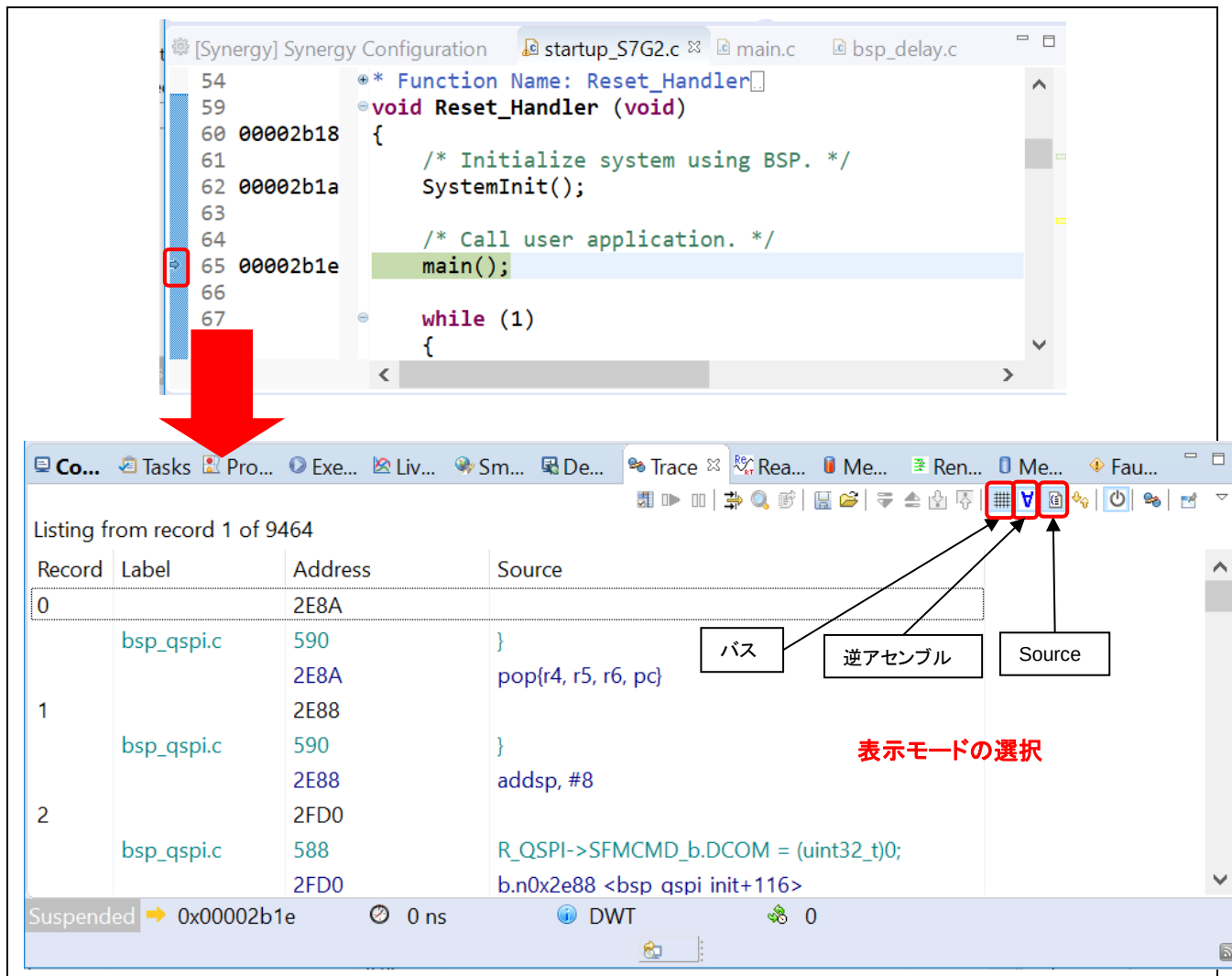


図 103. デバッグ - [Trace] (トレース) ビューの表示モード選択

5. デフォルトで、トレースレコードは最も古いデータから最も新しいデータの順に表示されます。表示順序を変更するには、 ボタンをクリックします。

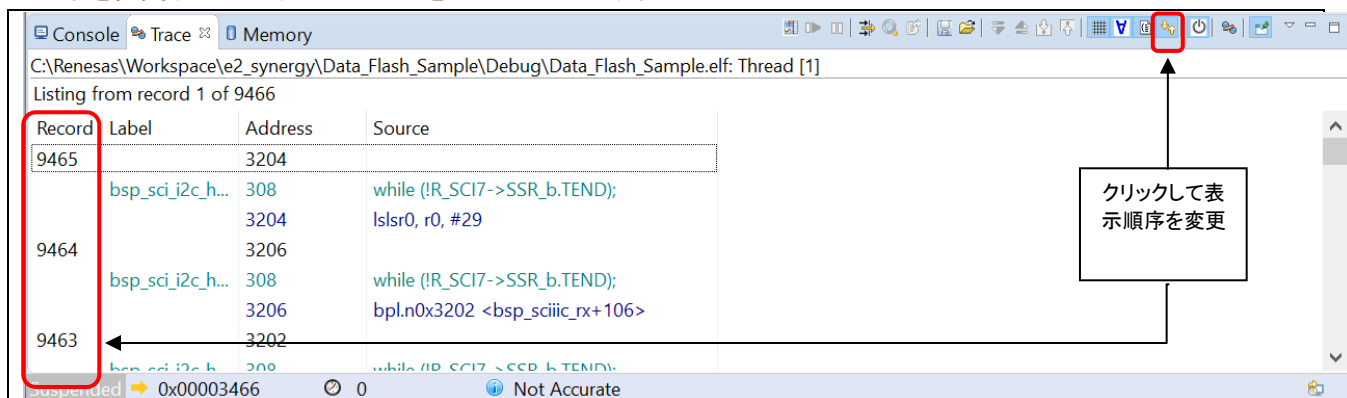


図 104. 表示順序の変更

6. トレース結果をフィルタ処理するには、 ボタンをクリックします。ユーザは [Record] (レコード) と [Address] (アドレス) の一方または両方を選択してフィルタ処理を実行することができます。

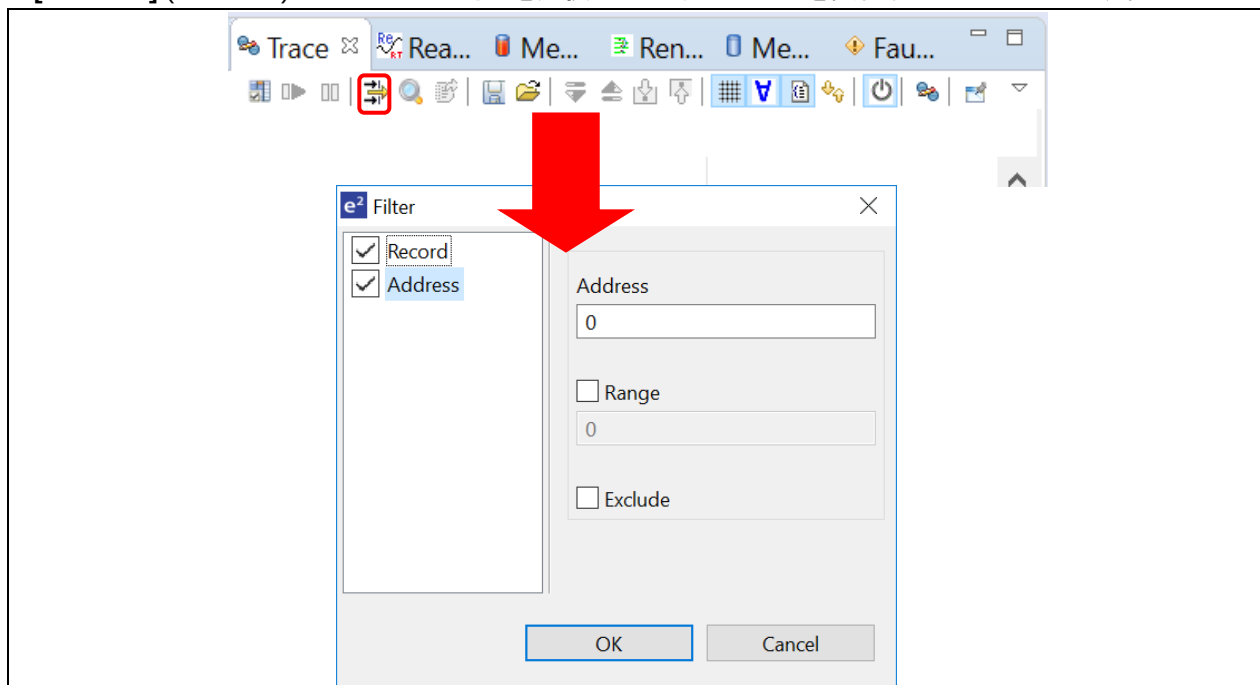


図 105. デバッグ - トレース結果のフィルタ処理

7. (バス、アセンブリ、ソースの情報を含め) .csv ファイルへの保存が可能です。また、[Trace] (トレース) ビューで、.csv ファイルからトレース結果をロードすることもできます。

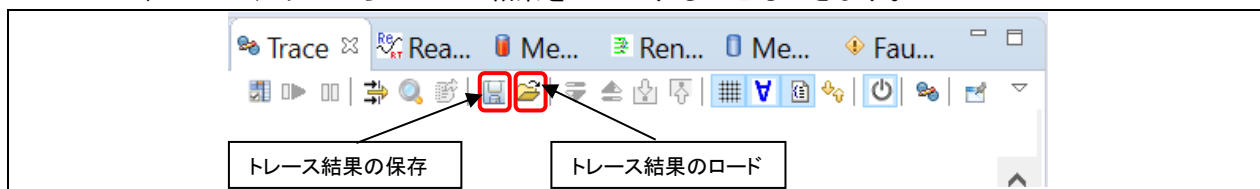


図 106. デバッグ - トレース結果の保存とロード

5.3.11 [Fault Status] (障害ステータス) ビュー

このビューでは、ハードウェア障害によるクラッシュが発生したときに、複数の障害レジスタのビットステータスと、主なレジスタの値をユーザに表示します。ハードウェア障害が発生したときに、その障害の原因に関連するレジスタのビットがチェックされ、r0、r1、r2、r3、r12、lr、pc、および psr の各レジスタの値が表示されます。以下の図に、この表示を示します。この機能は、e² studio v5.2 およびそれ以降のバージョンで利用できます。

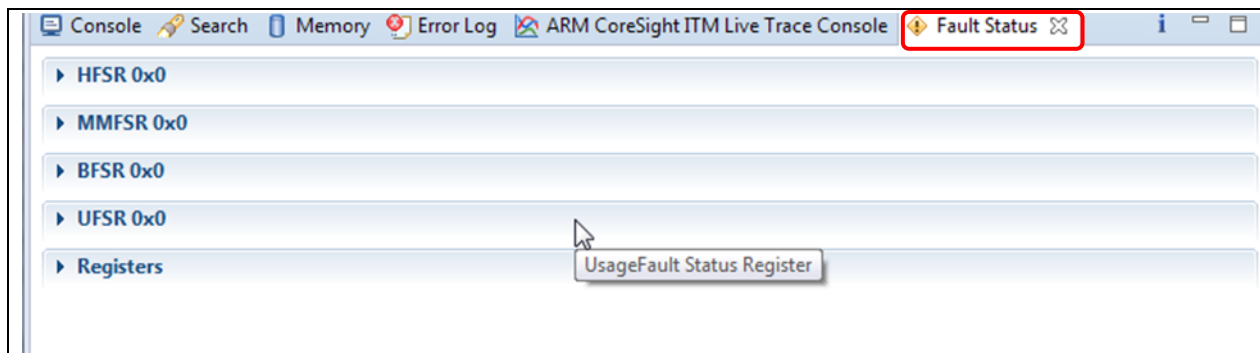


図 107. [Fault Status] (障害ステータス) でハードウェア障害が発生していない場合

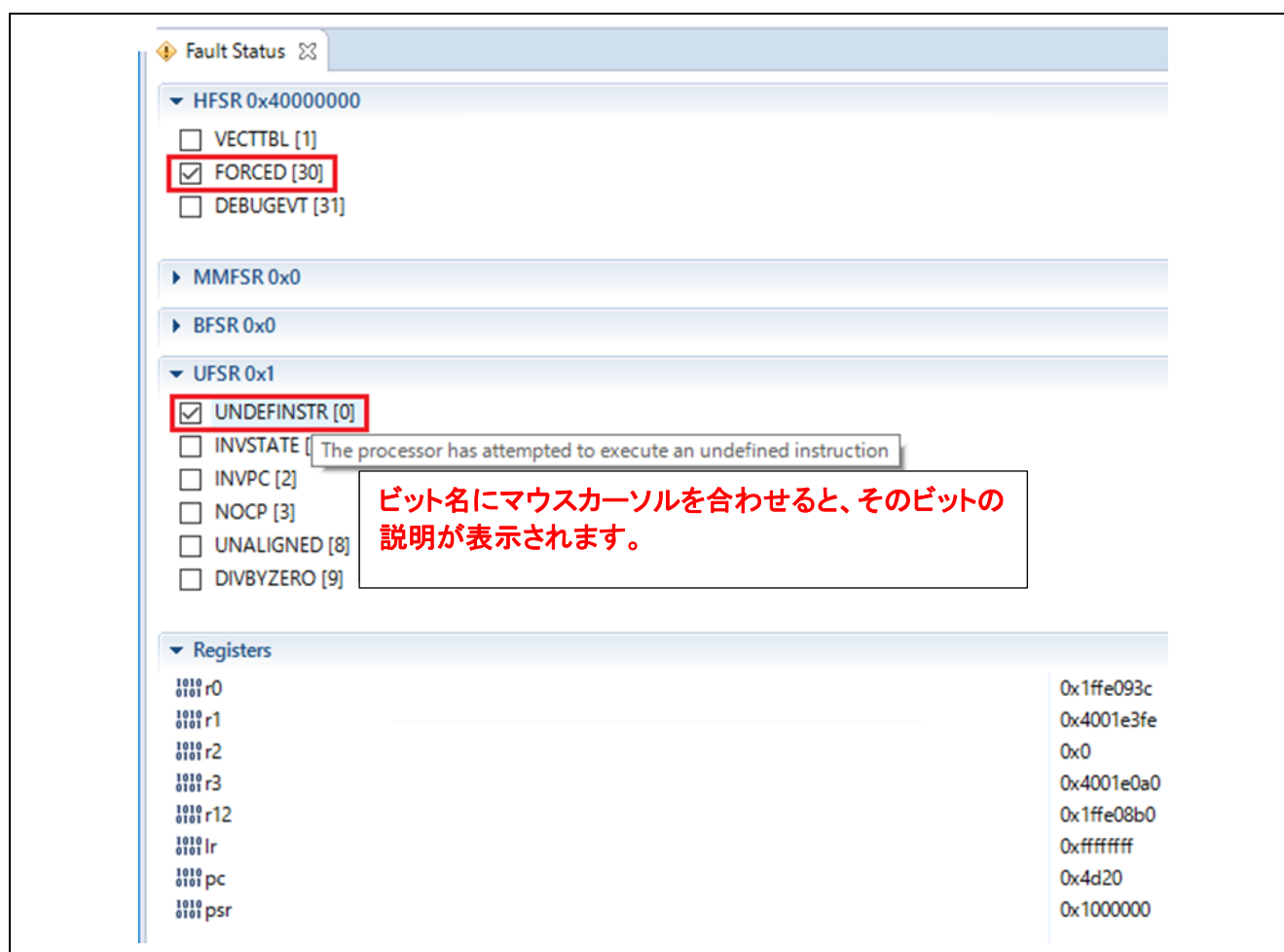


図 108. [Fault Status] (障害ステータス) でハードウェア障害が発生した場合

5.3.12 実行ブレイクタイマ (Run Break Timer)

実行ブレイクタイマ (Run Break Timer) 機能を使用すると、直前の実行結果をステータスバーで表示できます。プログラムが中断された時点で、ユーザは現在のプログラムカウンタ (PC)、時刻または CPU サイクルのいずれかで表記した直前の実行タイミング、および精度または使用している測定方法を確認することができます。

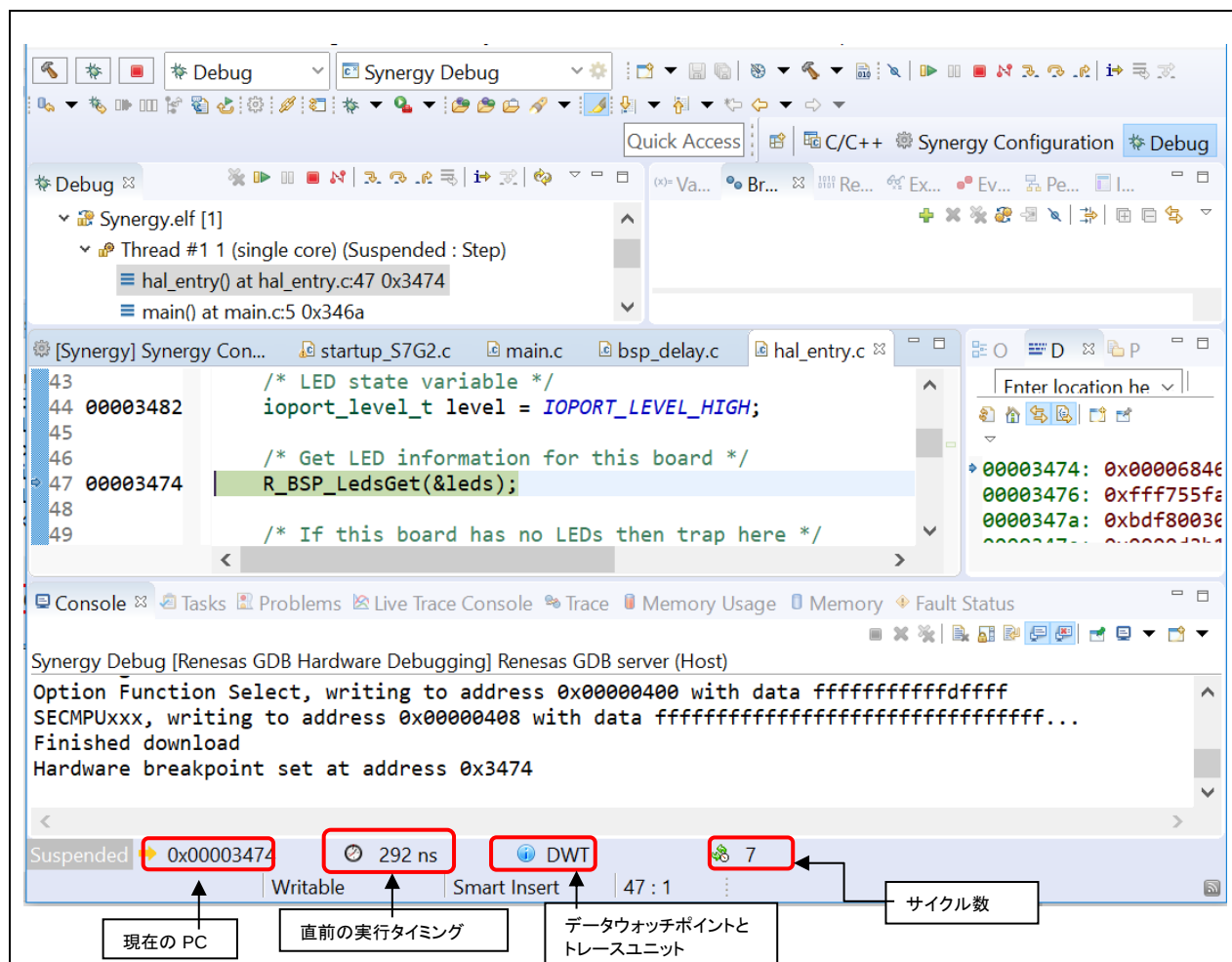


図 109. 実行ブレイクタイマは直前の実行結果を表示

以下の表に、各種 Synergy デバイスで利用できる実行ブレイクタイマ機能のサポート状況を示します。

表 3. 実行ブレイクタイマのサポート

デバイス	デバugg	サポート
Synergy S1 シリーズ (Cortex M0+/M23)	J-Link	システム時刻
Synergy S3、S5、S7 シリーズ	J-Link	データウォッチポイントとトレースユニット (DWT) - システム時刻を使用して計算した、サイクル数とオーバーフロー数

実行ブレイクタイマ機能は、e² studio v7.3.0 およびそれ以降のバージョンでサポートされています。仕様の更新については、www.renesas.com/synergy/e2studio を参照してください。ここには、e² studio のリリースノートが掲載されています。

6. ThreadX アプリケーションの設定 (Setting up a ThreadX Application)

この例では、Synergy プロジェクトの生成とビルドを行い、プロジェクトテンプレート **Blinky with ThreadX** を使用して ThreadX オブジェクトと General Purpose Timer (GPT、汎用タイマ) モジュールをインクルードする方法を示します。

6.1 ThreadX 内の General Purpose Timer の例 (General Purpose Timer Example in ThreadX)

[Project Template] (プロジェクトテンプレート) の選択肢である **Blinky with ThreadX** Synergy プロジェクトでは、タスクを短時間スリープ状態に移行しているときに LED が点滅し、その後、LED の状態をトグルします。

この例で、スリープによる遅延の代わりに、Blinky Thread がセマフォとタイマの割り込み (GPT によって生成) を待ち、1 秒ごとにセマフォを配置して、スレッドが再開できるように変更を加えます。

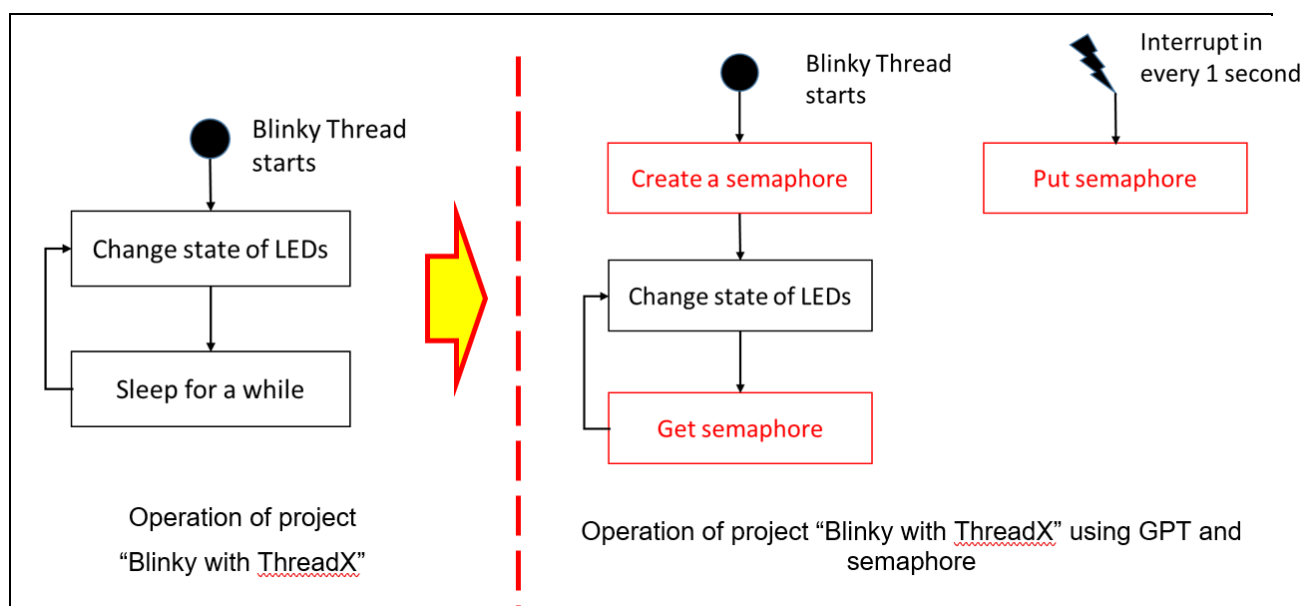


図 110. ThreadX アプリケーションの設定 - 導入

6.2 サンプルプロジェクトの作成 (Creating the Sample Project)

GPT とセマフォを使用するサンプルの ThreadX プロジェクトを作成するために、Synergy プロジェクトを以下のように構成します。

1. [New Project] (新しいプロジェクト) に対応するエディタを起動し、3.1 章 (新しい Synergy プロジェクトの作成 (Generating a New Synergy Project)) の手順に従って新しいプロジェクトを生成します。ただし、最後のダイアログ ([Project Template] (プロジェクトテンプレート) ダイアログ) で、**[Blinky with ThreadX]** を選択します。

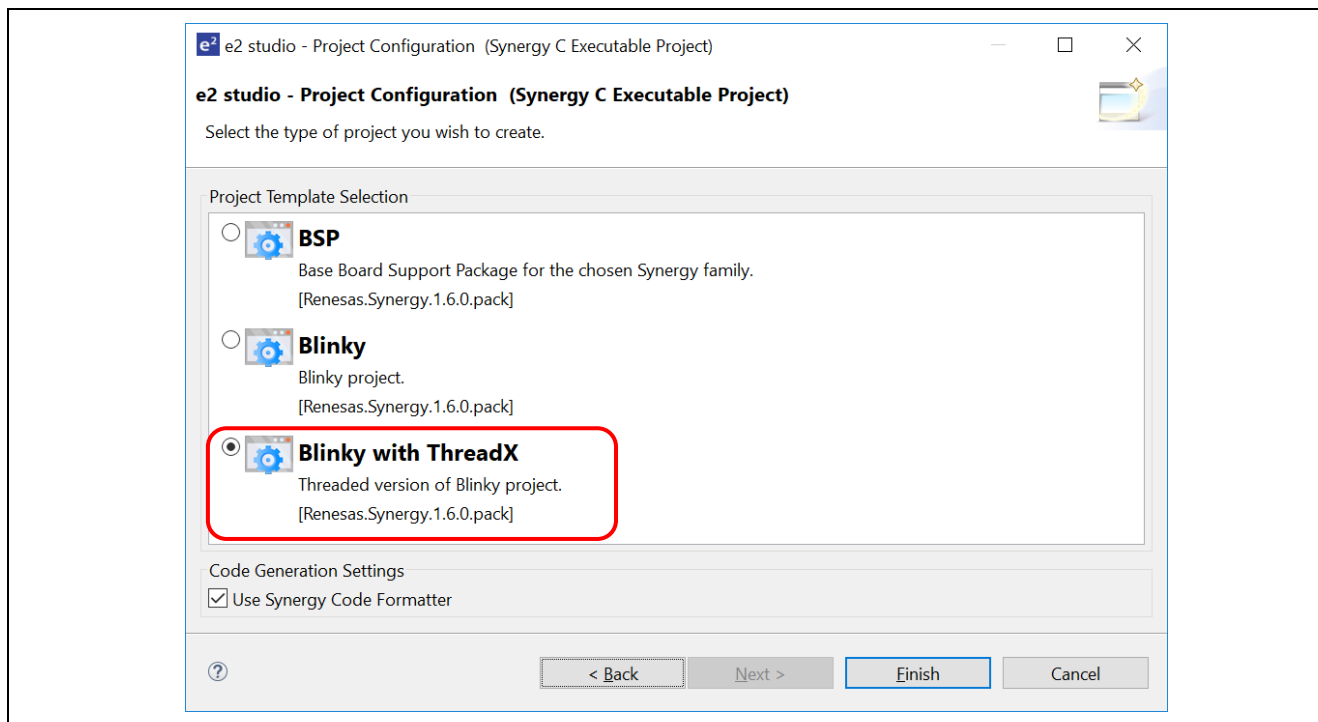


図 111. ThreadX アプリケーションの設定 - [Blinky with ThreadX] テンプレートの選択

2. **[Synergy Project Configuration]** (Synergy プロジェクト設定) で、**[Threads Configuration]** (スレッド設定) ページを開きます。3.4.5 章を参照してください。
3. GPT モジュールを Blinky Thread に追加するために、**[Threads]** (スレッド) パネルで **[Blinky Thread]** (Blinky スレッド) を選択し、**[Stacks]** (スタック) パネルで **[New Stack]** (新規スタック) → **[Driver]** (ドライバ) → **[Timers]** (タイマ) → **[Timer Driver on r_gpt]** (r_gpt でのタイマドライバ) を選択します。

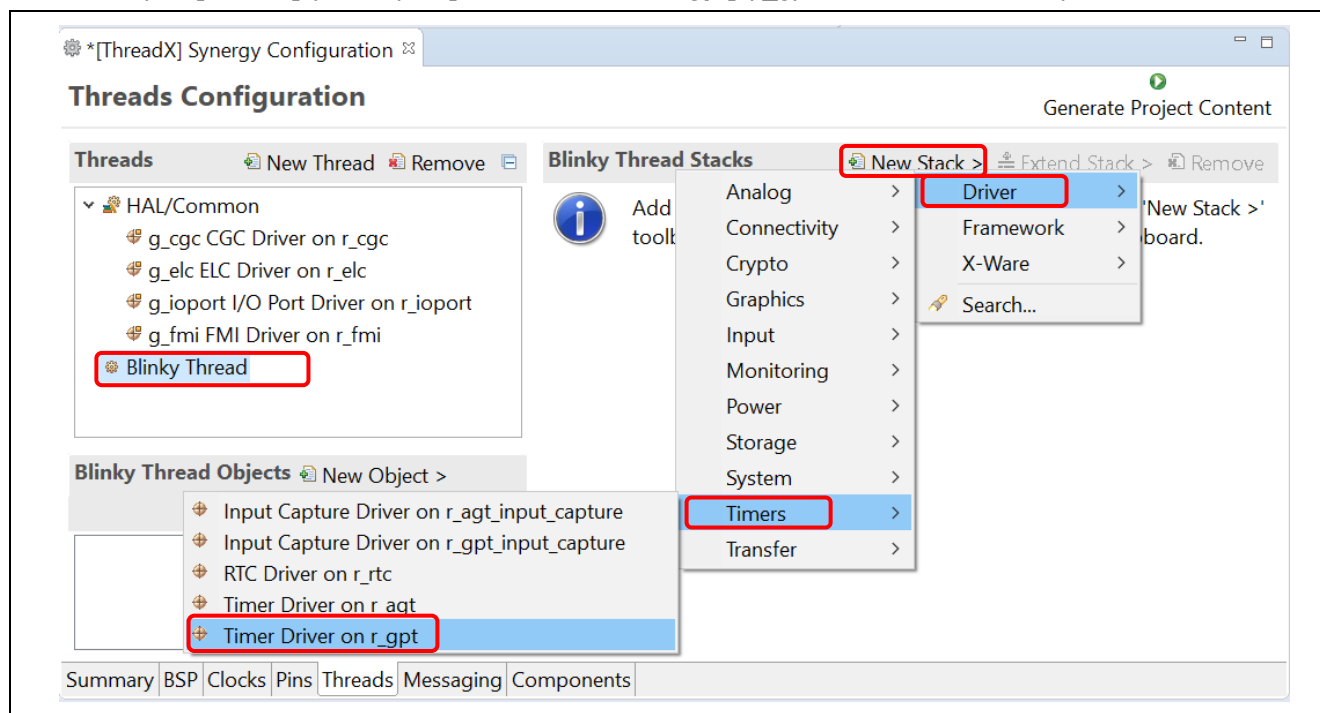


図 112. ThreadX アプリケーションの設定 - GPT モジュールの追加

4. 以下のように GPT モジュールを構成します。

- [Name] (名前): g_timer
- [Mode] (モード): Periodic (周期的)
- [Period Value] (期間の値): 1
- [Period Unit] (期間の単位): Seconds (秒)
- [Callback] (コールバック): gpt_callback
- [Overflow Interrupt priority] (オーバーフロー割り込みの優先順位): 2

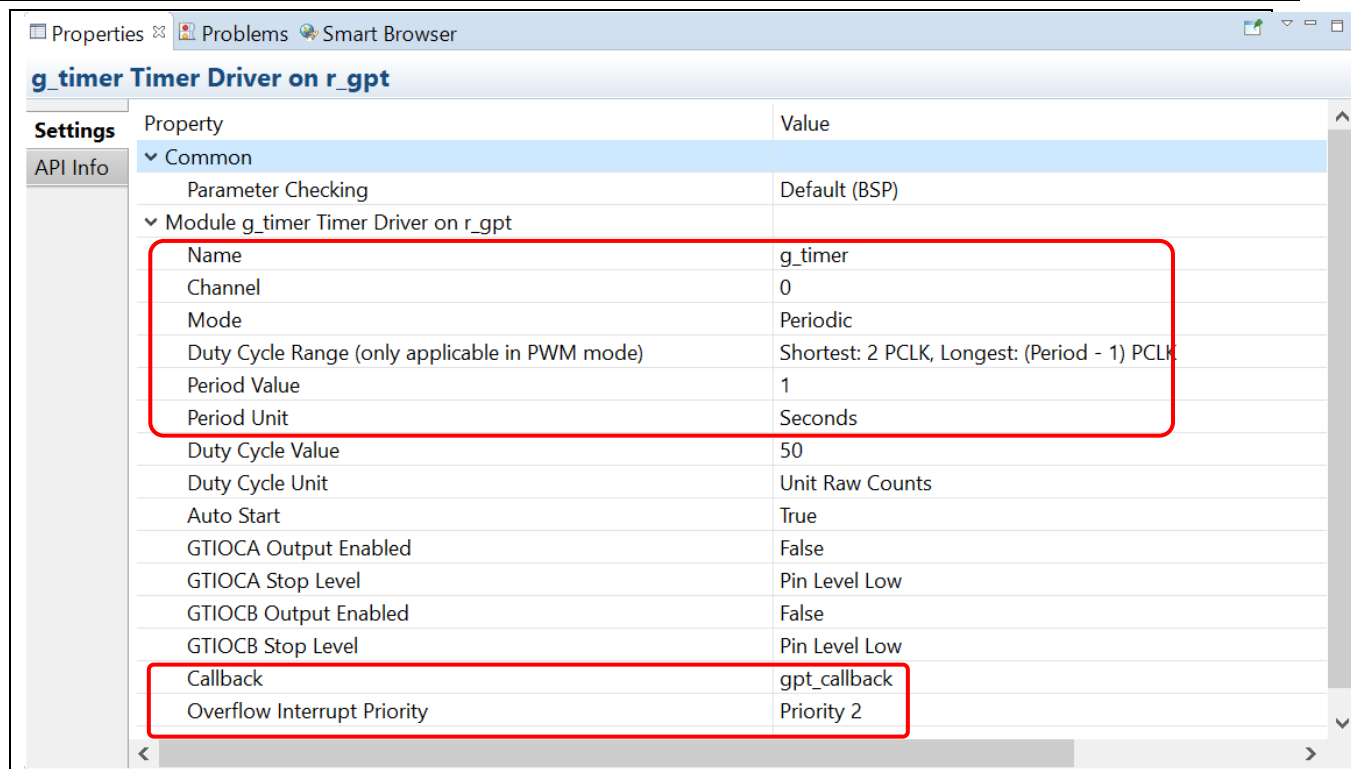
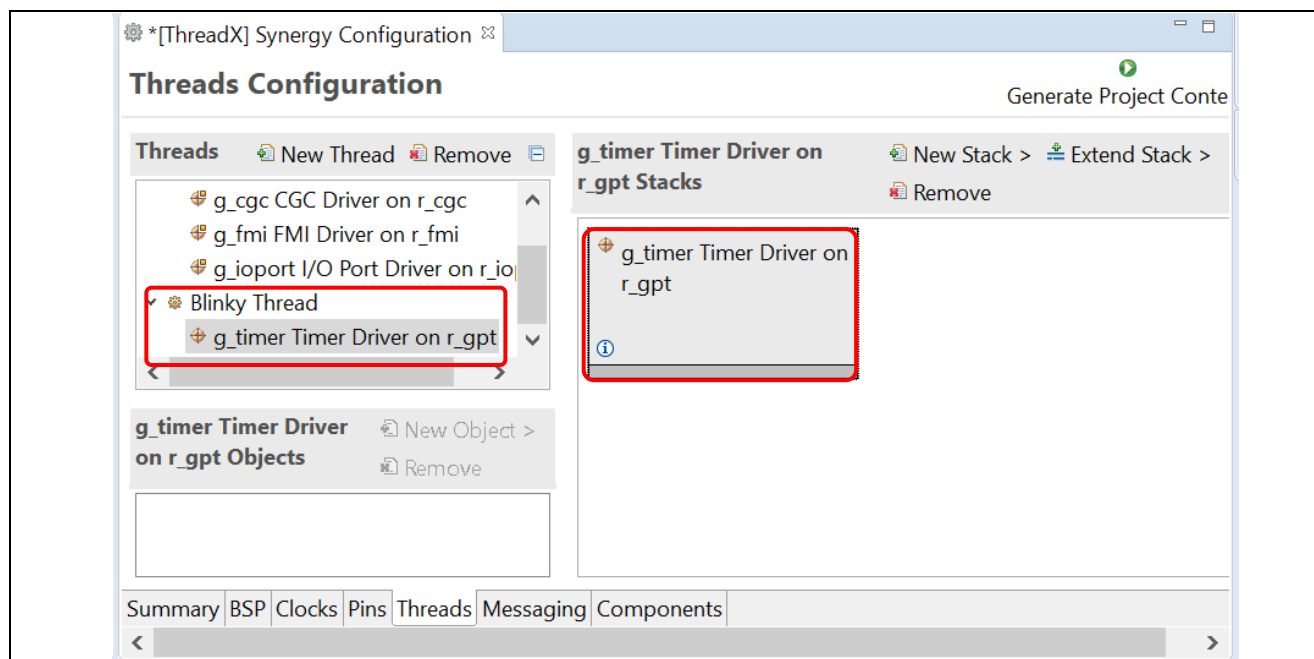


図 113. ThreadX アプリケーションの設定 - GPT モジュールの設定

5. セマフォオブジェクトを Blinky Thread に追加するために、[Threads] (スレッド) パネルで **[Blinky Thread]** (Blinky スレッド) を選択し、[Objects] (オブジェクト) パネルで  → **[Semaphore]** (セマフォ) を選択します。

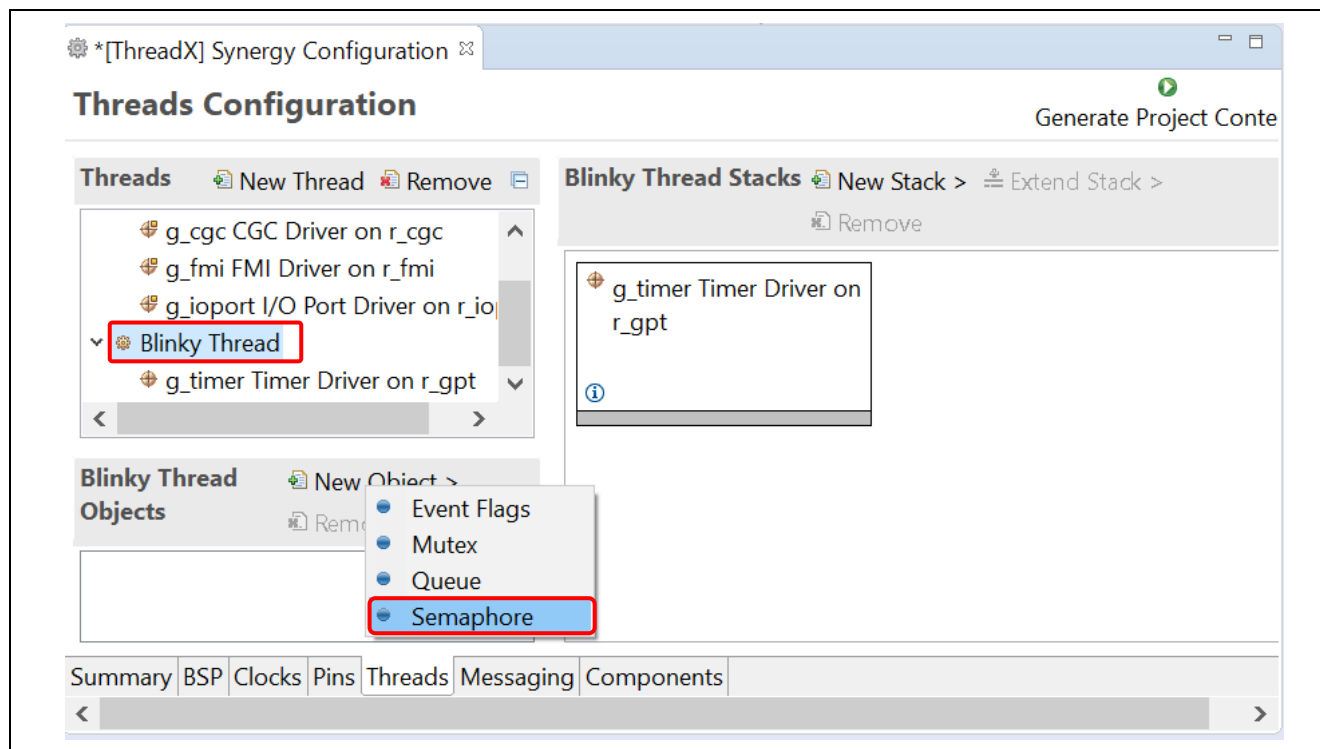


図 114. ThreadX アプリケーションの設定 - セマフォオブジェクトの追加

6. 新しく作成したセマフォを以下のように構成します。

- [Name] (名前): Blinky Semaphore
- Symbol (シンボル): g_blinky_semaphore
- [Initial count] (初期カウント): 0

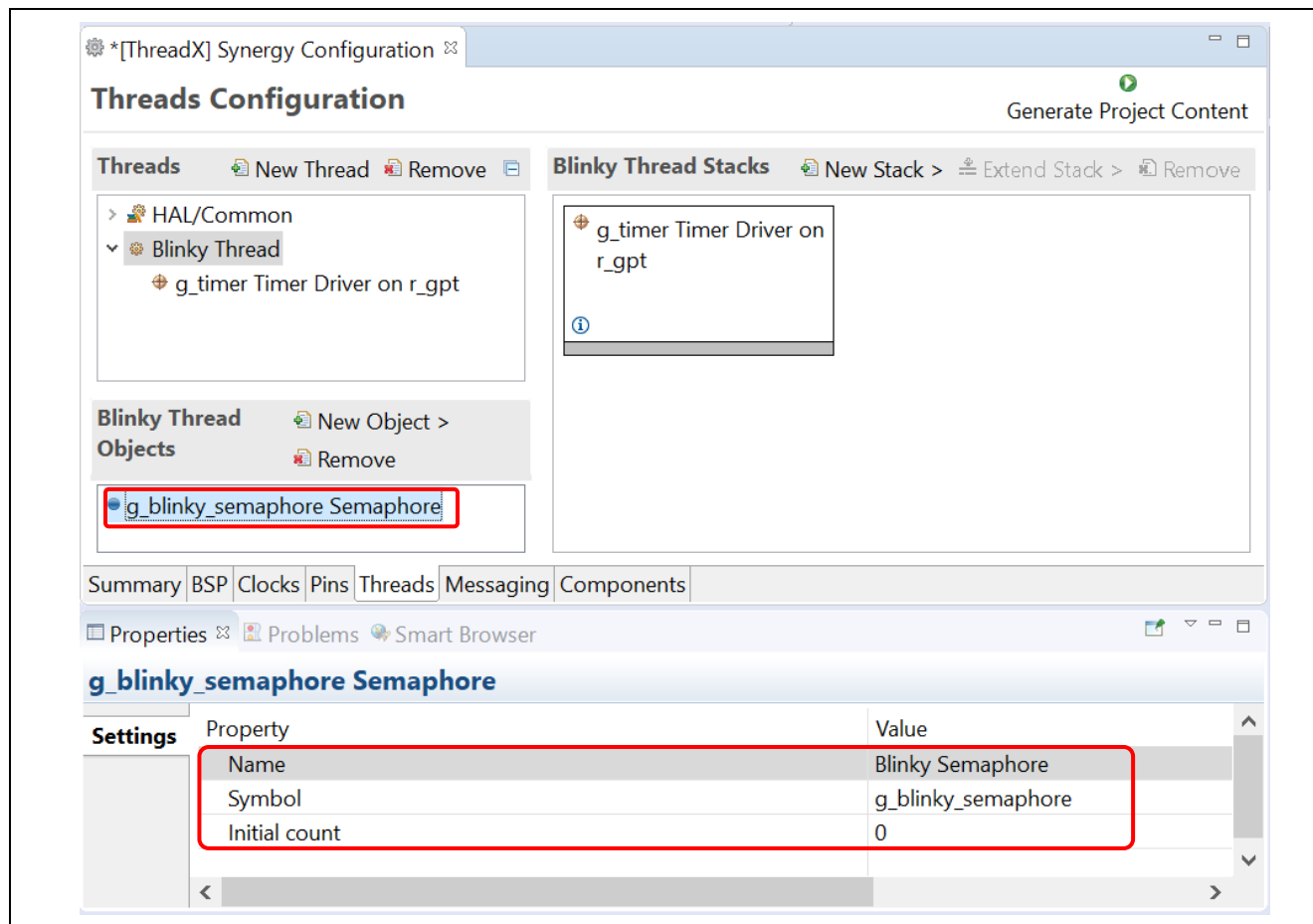


図 115. ThreadX アプリケーションの設定 - セマフォオブジェクトの設定

Ctrl+S をクリックして設定を保存し、[Generate Project Content] (プロジェクトコンテンツの生成)

Generate Project Content ボタンをクリックして、ソースコードの内容を生成します。

7. blinky_thread_entry.c を開き、以下の内容を加えます。

— blinky_thread_entry()内で、while(1) ループの前に、GPT モジュールを初期化するソースコードを追加します。

```
g_timer.p_api->open(g_timer.p_ctrl, g_timer.p_cfg);
```

— スレッドの sleep 命令を削除し、blinky_thread_entry()内でセマフォを待つコードを追加します。

```
tx_semaphore_get(&g_blinky_semaphore, TX_WAIT_FOREVER);
```

— gpt_callback()関数を加えて、Blinky スレッドに対してセマフォを生成します。

```
void gpt_callback(timer_callback_args_t * p_args){
    tx_semaphore_put(&g_blinky_semaphore);
}
```

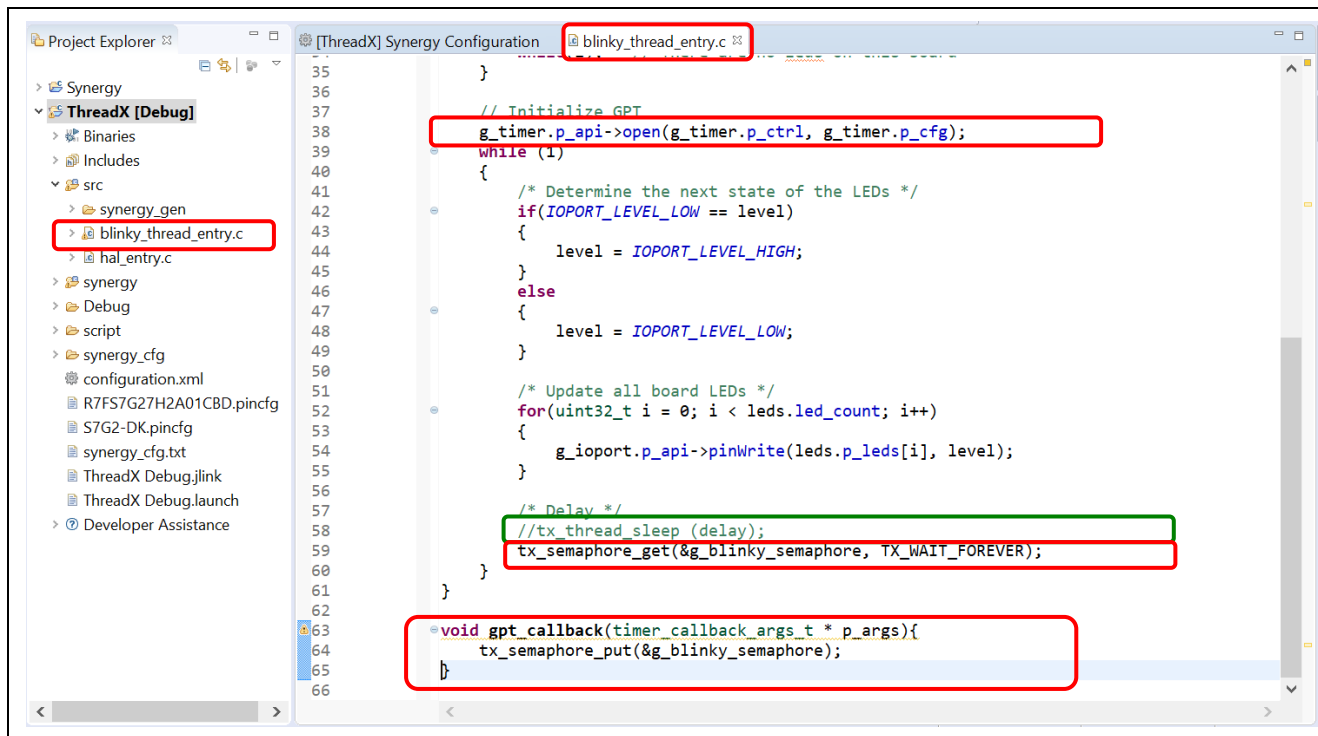


図 116. ThreadX アプリケーションの設定 - ユーザソースコードの追加

8. プロジェクトをビルドし、Synergy DK-S7G2 ボードで実行します。LED が 1 秒ごとに点灯および消灯することを確認します。

7. [Help] (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 [Help] (ヘルプ) ウィンドウや [Help] (ヘルプ) 画面から、ヘルプドキュメントの参照、検索、ブックマーク、印刷を実行できます。また、このメニューから、e² studio 専用のオンラインフォーラムにアクセスすることもできます。

[Help] (ヘルプ) タブをクリックして、[Help] (ヘルプ) メニューを開きます。

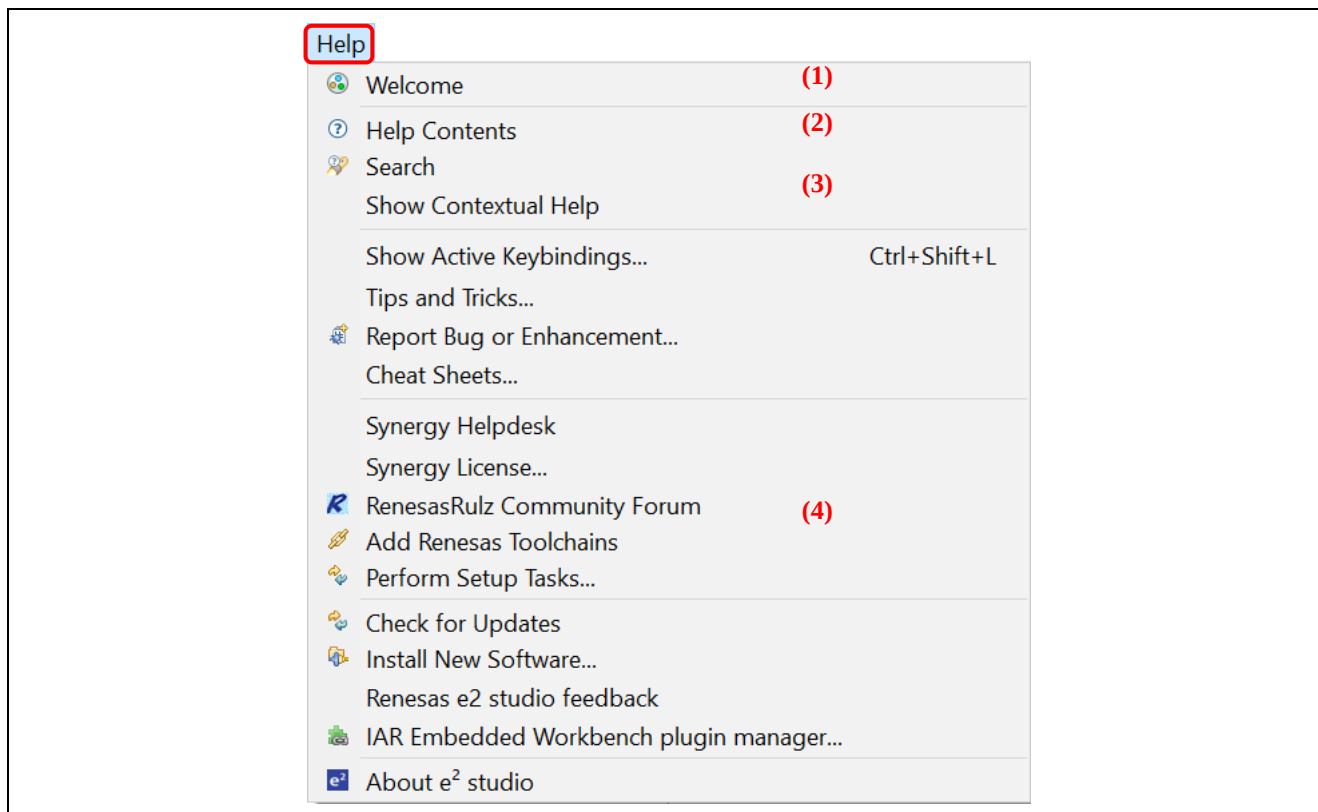


図 117. [Help] (ヘルプ) - [Help] (ヘルプ) メニュー

ヘルプに関する便利なヒント：

1. **[Welcome]** (ようこそ) をクリックすると、e² studio の概要やリリースノートを表示できます。
2. **[Help Contents]** (ヘルプコンテンツ) をクリックすると、検索機能が利用できる個別の [Help] (ヘルプ) ウィンドウが開きます。
3. **[Show Contextual Help]** (コンテキストヘルプの表示) をクリックすると、ワークベンチ内で [Help] (ヘルプ) ビューが開きます。
4. **[RenesasRulz Community Forum]** (RenesasRulz コミュニティフォーラム) をクリックすると、e² studio に関連するトピックとディスカッション専用のオンラインフォーラムにアクセスできます (インターネット接続が必要です。)

[Help Contents] (ヘルプコンテンツ) ウィンドウには、以下のような役立つトピックが多数あります。

- **[e² studio Debug Help]** (e² studio デバッグヘルプ) トピックは、デバッグ構成や、サポートされているブレークポイントの数など、役立つ情報を掲載しています。
この機能を起動するには、**[Help] (ヘルプ) メニュー → [Help Contents] (ヘルプコンテンツ) → [e² studio Debug Help] (e² studio デバッグヘルプ)** をクリックします。
- **[Synergy Contents]** (Synergy コンテンツ) トピックは、Synergy Configuration Editor を使用して Synergy プロジェクトを作成する方法や、FAQ などの情報を掲載しています。

この機能を起動するには、**[Help] (ヘルプ) メニュー → [Help Contents] (ヘルプコンテンツ) → [Synergy Contents] (Synergy コンテンツ)** をクリックします。

Web サイトおよびサポート

以下のさまざまな URL にアクセスすると、Synergy プラットフォームの主要要素に関する詳細を確認し、それらに関連するドキュメントをダウンロードし、サポートを活用することができます。

Synergy ソフトウェア	www.renesas.com/synergy/software
Synergy ソフトウェアパッケージ	www.renesas.com/synergy/ssp
ソフトウェアアドオン	www.renesas.com/synergy/addons
ソフトウェア用語集	www.renesas.com/synergy/softwareglossary
開発ツール	www.renesas.com/synergy/tools
Synergy ハードウェア	www.renesas.com/synergy/hardware
マイクロコントローラ	www.renesas.com/synergy/mcus
MCU 用語集	www.renesas.com/synergy/mcuglossary
パラメトリック検索	www.renesas.com/synergy/parametric
キット	www.renesas.com/synergy/kits
Synergy ソリューション Gallery	www.renesas.com/synergy/solutionsgallery
パートナープロジェクト	www.renesas.com/synergy/partnerprojects
アプリケーションプロジェクト	www.renesas.com/synergy/applicationprojects
セルフサービスサポートリソース :	
ドキュメント	www.renesas.com/synergy/docs
ナレッジベース	www.renesas.com/synergy/knowledgebase
フォーラム	www.renesas.com/synergy/forum
トレーニング	www.renesas.com/synergy/training
ビデオ	www.renesas.com/synergy/videos
チャットと Web チケット	www.renesas.com/synergy/resourcelibrary

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.21	2019.08.08	－	● 初版 英語版（R20UT4204EU0120, Rev.1.20, 2019.08.08）を翻訳

e² studio 入門ガイド

発行年月日 2019年8月8日 Rev.1.21

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

Renesas Synergy™ プラットフォーム e² studio 入門ガイド