

統合開発環境  
e<sup>2</sup> studio 2021-10 以上  
ユーザーズマニュアル クイックスタートガイド  
Renesas Synergy™

本資料に記載の全ての情報は本資料発行時点のものであり、ルネサス エレクトロニクスは、予告なしに、本資料に記載した製品または仕様を変更することがあります。  
ルネサス エレクトロニクスのホームページなどにより公開される最新情報をご確認ください。

## 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、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、これに限られません。）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た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、販売または転売等する場合は、「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したします。
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（豊洲フォレストシア）

[www.renesas.com](http://www.renesas.com)

## 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。

## 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/](http://www.renesas.com/contact/)

## 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を、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しないでください。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、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、 $V_{IL}(\text{Max.})$  から  $V_{IH}(\text{Min.})$  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、 $V_{IL}(\text{Max.})$  から  $V_{IH}(\text{Min.})$  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っていると、フラッシュメモリ、レイアウトパターンなどの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が異なる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	1
1.1	システム構成	3
1.2	システム要件	3
1.3	サポートするツールチェーン	3
1.4	サポートするエミュレータ	3
1.5	Synergyプロジェクト開発の概要	4
2.	インストール	5
2.1	プラットフォームインストーラを用いたインストール	5
2.2	e <sup>2</sup> studioとSSPの個別インストール	11
2.2.1	e <sup>2</sup> studioのインストール	11
2.2.2	GNU Armコンパイラの設定	17
2.2.3	Renesas Synergy™ Software Package (SSP) のインストール	17
2.3	e <sup>2</sup> studioのアンインストール	19
3.	プロジェクトの作成	20
3.1	新規Synergyプロジェクトの作成	20
3.2	既存Synergyプロジェクトのインポート	24
3.3	Synergyスタティックライブラリの作成と使用	27
3.3.1	スタティックライブラリプロジェクトの作成	27
3.3.2	スタティックライブラリを実行プロジェクトで使用する	32
3.4	Synergyプロジェクトコンフィグレーションエディタ	36
3.4.1	概要 (Summary) ページ	37
3.4.2	BSPページ	38
3.4.3	クロック構成 (Clocks Configuration) ページ	39
3.4.4	端子構成 (Pins Configuration) ページ	40
3.4.5	スレッド構成 (Threads Configuration) ページ	44
3.4.6	メッセージング (Messaging) ページ	51
3.4.7	コンポーネント構成 (Components Configuration) ページ	53
3.5	エディタ画面の吹き出し機能	54
3.6	開発者支援機能 (Developer Assistance)	55
4.	ビルド	58
4.1	ビルドオプションの設定	58
4.2	サンプルプロジェクトのビルド	59
4.3	プロジェクトレポート機能 (ビルドオプション一覧の出力)	60
5.	デバッグ	61
5.1	既存デバッグ構成の変更	62
5.2	新規デバッグ構成の作成	65
5.3	基本的なデバッグ機能	66
5.3.1	ブレークポイントビュー	67
5.3.2	式ビュー	68
5.3.3	レジスタビュー	69
5.3.4	メモリービュー	70
5.3.5	メモリー使用量ビュー	72
5.3.6	逆アセンブルビュー	74
5.3.7	変数ビュー	75
5.3.8	IOレジスタ (IO Registers) ビュー	76
5.3.9	イベントポイントビュー	78
5.3.10	トレースビュー	81

5.3.11	Fault Statusビュー .....	83
5.3.12	Run Break Timer .....	84
6.	ThreadXアプリケーションの設定 .....	85
6.1	ThreadXでの汎用PWMタイマ使用例 .....	85
6.2	サンプルプロジェクトの作成 .....	86
7.	ヘルプ .....	92

# 1. 概説

e<sup>2</sup> studio は、Renesas Synergy™ 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。e<sup>2</sup> 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) の拡張インターフェースにより実現されます。

e<sup>2</sup> 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、e<sup>2</sup> 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。

e<sup>2</sup> 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、既存ドライバの組み込みと設定、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。

Renesas Synergy™ は e<sup>2</sup> studio のリリース 4.1 以上でサポートしています。Arm® Cortex®-M ベースの Renesas Synergy™ マイクロコントローラとオープンソース GNU Arm Toolchain 用の各種ビューやエディタが用意されています。

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™ の豊富なソフトウェアライブラリの管理に活用できます。

注：本マニュアルでは、“Azure RTOS ThreadX” を “ThreadX”、“Azure RTOS TraceX” を “TraceX” と記します。

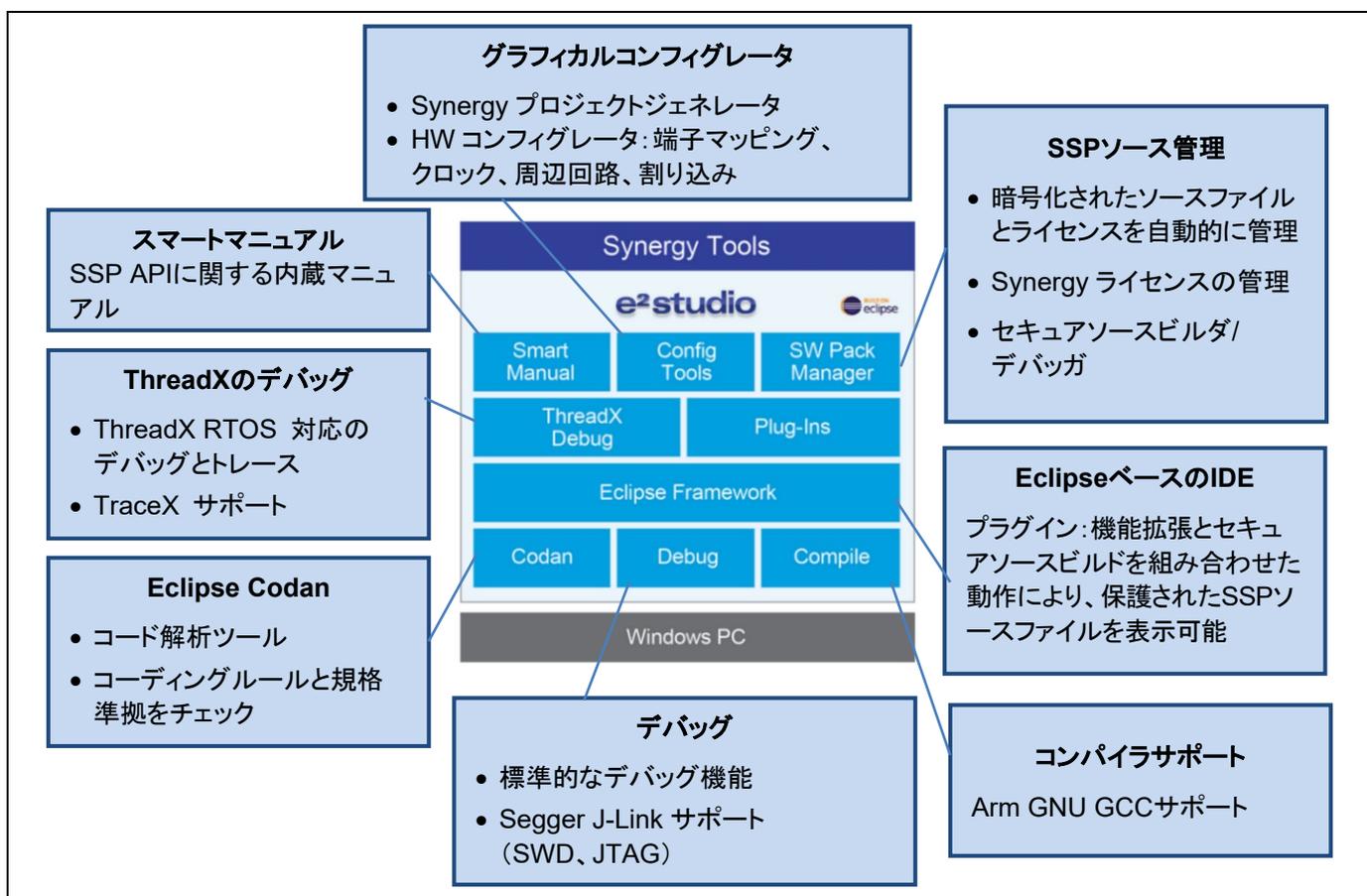


図 1-1 Renesas Synergy™ 向け e<sup>2</sup> studio

e<sup>2</sup> 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 および Arm 向け IAR ツールチェーンのサポートと、Renesas Synergy™ プロジェクトジェネレータ、および Renesas Synergy™ コンフィグレーションエディタです。

表 1-1 e<sup>2</sup> 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 コンフィグレーションエディタ	コード生成ツール/スマートコンフィグレータ
ツールチェーン	GCC Arm Embedded Arm 向け IAR ツールチェーン	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 CC-RL 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
対象デバッグ	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チップ	サポート対象 (SSP API)	サポート対象
コード解析 (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	一部の RX および RL ファミリのデバイスをサポート
デバッグ	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	Azure RTOS 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
メモリー使用状況ビュー	サポート対象	サポート対象
ビジュアル・エクスプレッション・ビュー	サポート対象	サポート対象

## 1.1 システム構成

一般的なシステム構成では、下図のようにホストコンピュータとターゲットボードを使用します。



図 1-2 システム構成

## 1.2 システム要件

- 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： 4GB 以上 (8GB 以上を推奨)
  - ハードディスク容量： 2GB 以上
  - 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)。ハイスピードを推奨。
- オペレーティングシステム  
ホストコンピュータ上の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 8.1 (64 ビット OS)
  - Windows 10 (64 ビット OS)

## 1.3 サポートするツールチェーン

GNU Arm®コンパイラ (バージョン : GCC v9.2.1 および GCC v7.2.1)

**注** : IAR Embedded Workbench® for Renesas Synergy™ (EWSYN) は、ルネサスWebサイトからのダウンロード提供を終了しました。入手についてはIARシステムズ社にお問い合わせください。また、EWSYNのライセンスもIARシステムズ社が発行しますので、ライセンスに関する情報やライセンスキーの新規発行や更新についても[IARシステムズ社](#)にお問い合わせください。

## 1.4 サポートするエミュレータ

Segger J-Link

## 1.5 Synergy プロジェクト開発の概要

本ドキュメントでは、Renesas Synergy™プラットフォームを用いた開発方法の入門について詳しく説明します。手順の概要を以下に示します。この手順を把握することで、3章や4章の説明が理解しやすくなります。

- (1) Synergy プロジェクトを作成する
- (2) ハードウェア仕様（クロック、ICU、端子機能など）に適合した Synergy プロジェクトを構成する
- (3) Azure RTOS ThreadX を構成する
- (4) BSP を構成する（HAL ドライバモデルを選択）
- (5) ユーザコードを追加する
- (6) プロジェクトをビルドする
- (7) デバッガを設定し、デバッグを開始する

## 2. インストール

プラットフォームインストーラ（FSP with e<sup>2</sup> studio Installer）またはスタンドアロンインストーラのいずれかを用いてインストールできます。

最新のインストーラパッケージは、Renesas Synergy™ プラットフォーム Web サイト（<https://www.renesas.com/products/synergy.html>）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を用いたインストール

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）
- e<sup>2</sup> studio ISDE
- GCC Arm Embedded
- Segger J-Link ドライバ

以下の手順で、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してください。

- (1) [Renesas Synergy™ プラットフォーム](https://www.renesas.com/products/synergy.html) Web サイトにアクセスします。[Synergy Software Package (SSP)] → [ダウンロード] を選択してください。[Synergy Download] の [DOWNLOAD PLATFORM INSTALLER] を選択して My Renesas アカウントへログインし、プラットフォームインストーラをダウンロードします。

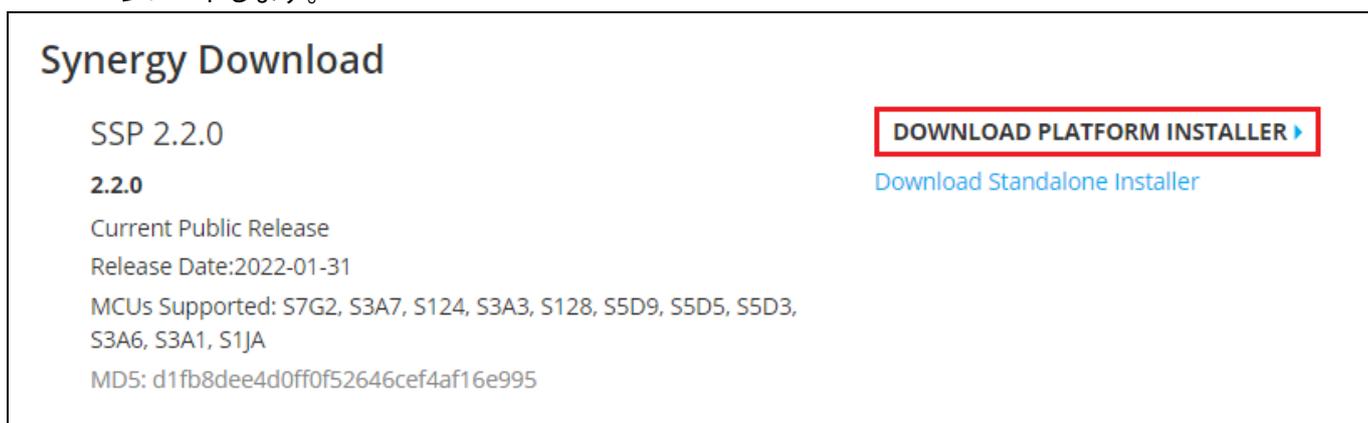


図 2-1 インストール – プラットフォームインストーラのダウンロード

- (2) 開発環境として、e<sup>2</sup> studio を選択します。

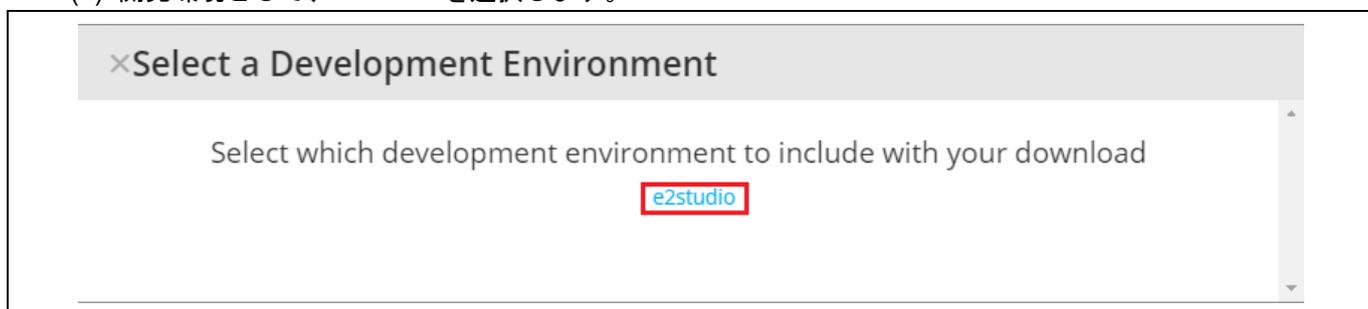


図 2-2 インストール – e<sup>2</sup> studio 開発環境の選択

- (3) [License Agreement] の [I AGREE] 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。

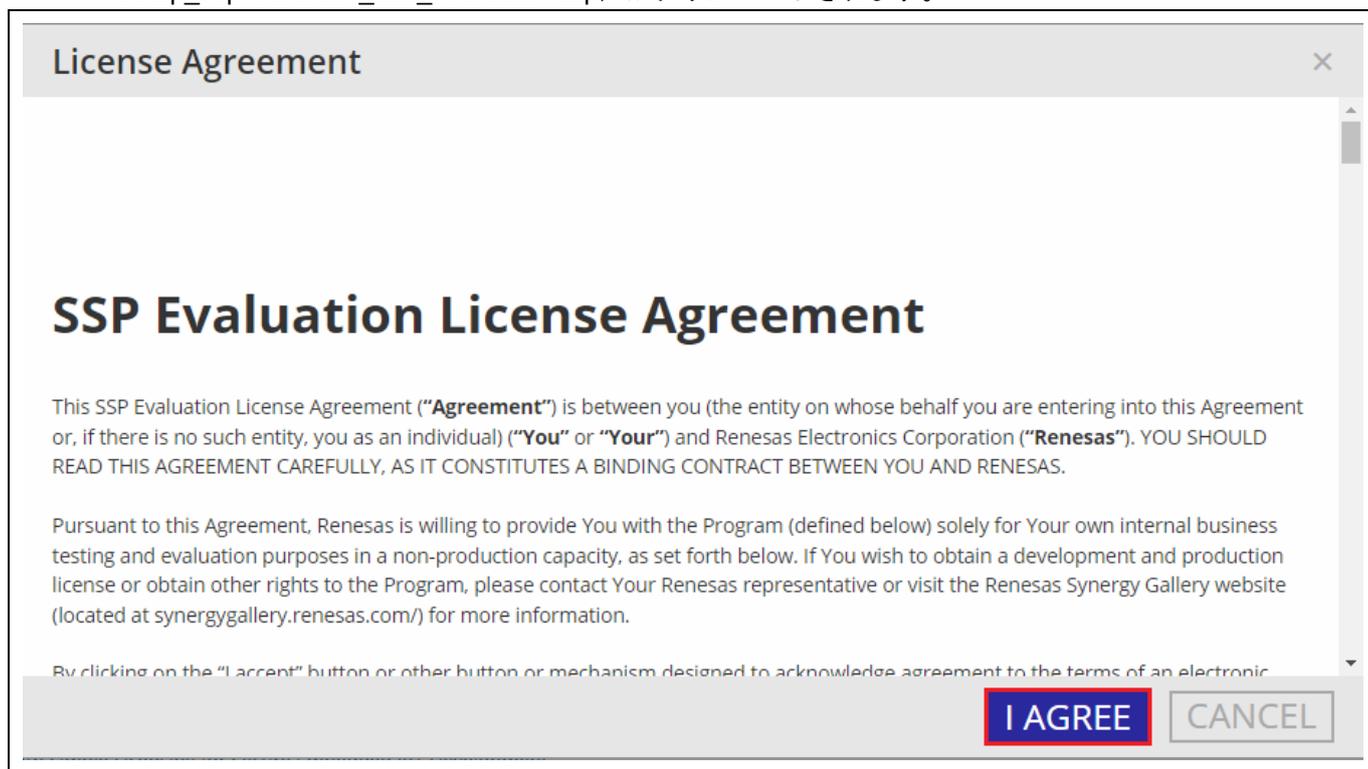


図 2-3 インストール – ライセンス契約の同意

- (4) インストールファイルを解凍し、実行します。

- (5) インストールするコンポーネントをカスタマイズしたい場合は、[インストール・タイプ] ページで “Custom Install” を選択し、[Next] ボタンを押してください。

初めて e<sup>2</sup> studio を使うユーザは、“Quick Install” オプションで簡単な手順を選択することをお勧めします。“Quick Install” オプションでは、デフォルトで e<sup>2</sup> studio、SSP、GCC Arm Embedded Toolchain をインストールします。“Quick Install” を選択した場合は、手順(7)の画面は表示されません。

注：“Quick Install”では日本語フォントをインストールしません。日本語フォントをインストールする場合は“Custom Install”を選択してください。

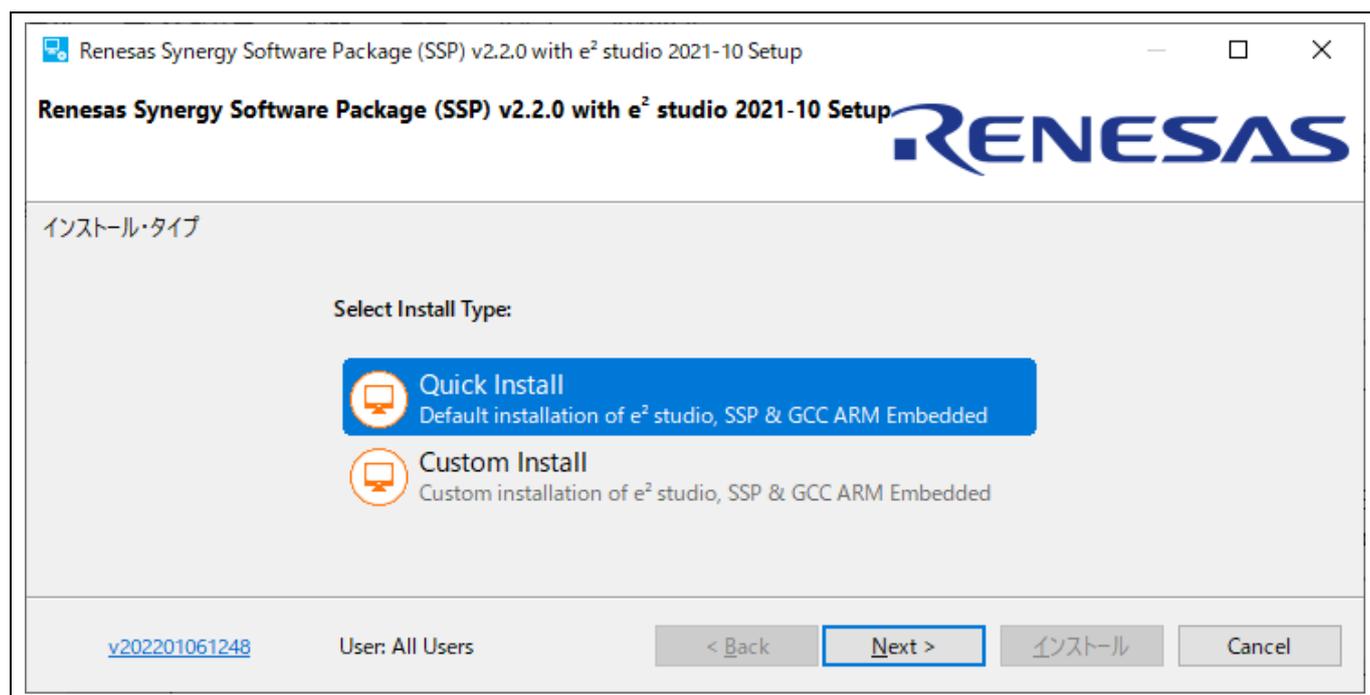


図 2-4 インストール – インストール・タイプの選択

(6) [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。



図 2-5 インストール - [ようこそ] ページ

- (7) “Custom Install” を選択した場合は、図 2-11 から図 2-13 の設定を行なってください。
- (8) ソフトウェア契約に同意いただけましたらチェックを入れてください。[Next] で次に進みます。

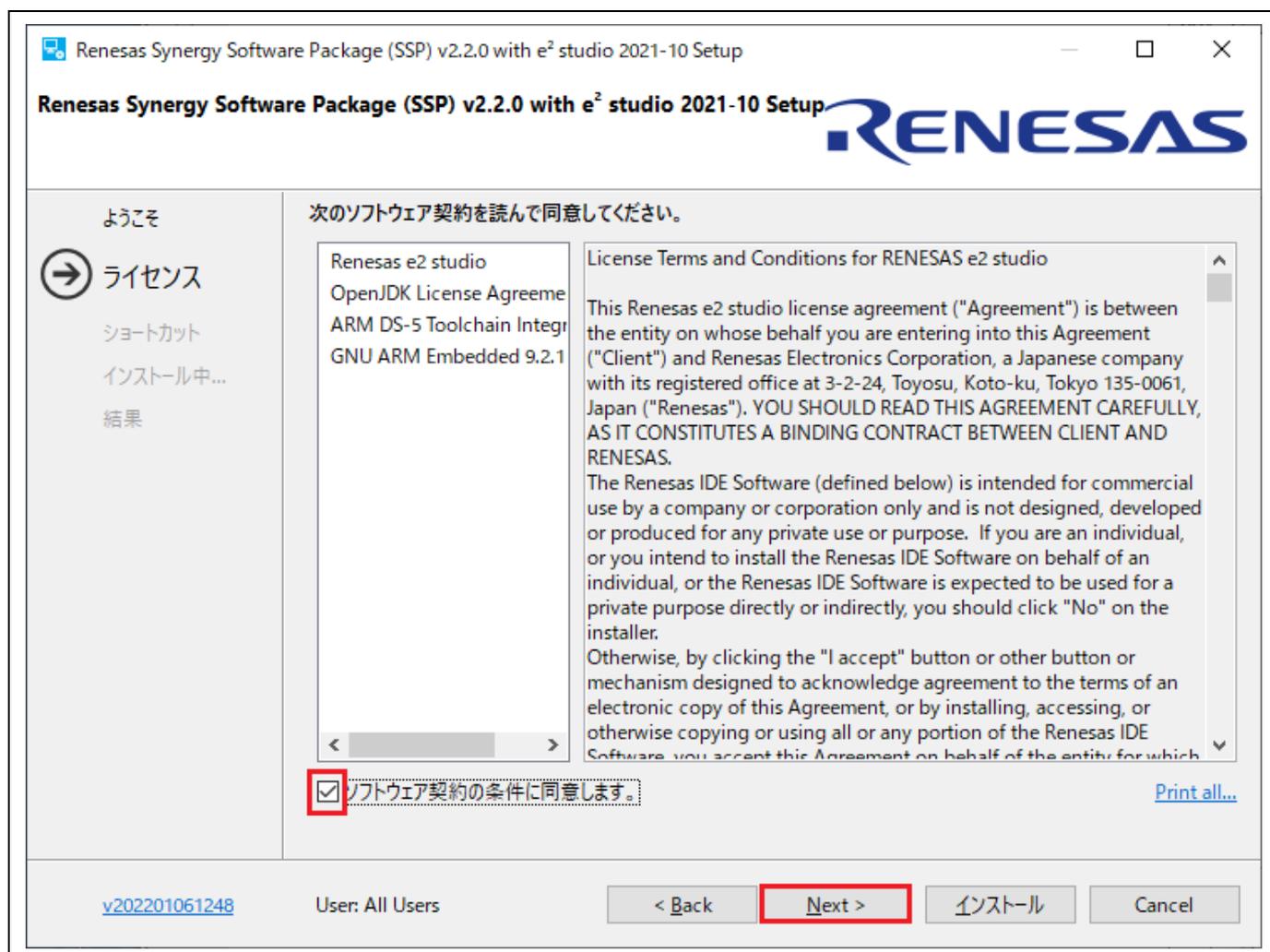


図 2-6 インストール – ソフトウェア契約

(9) [OK] をクリックしてインストールを完了します。

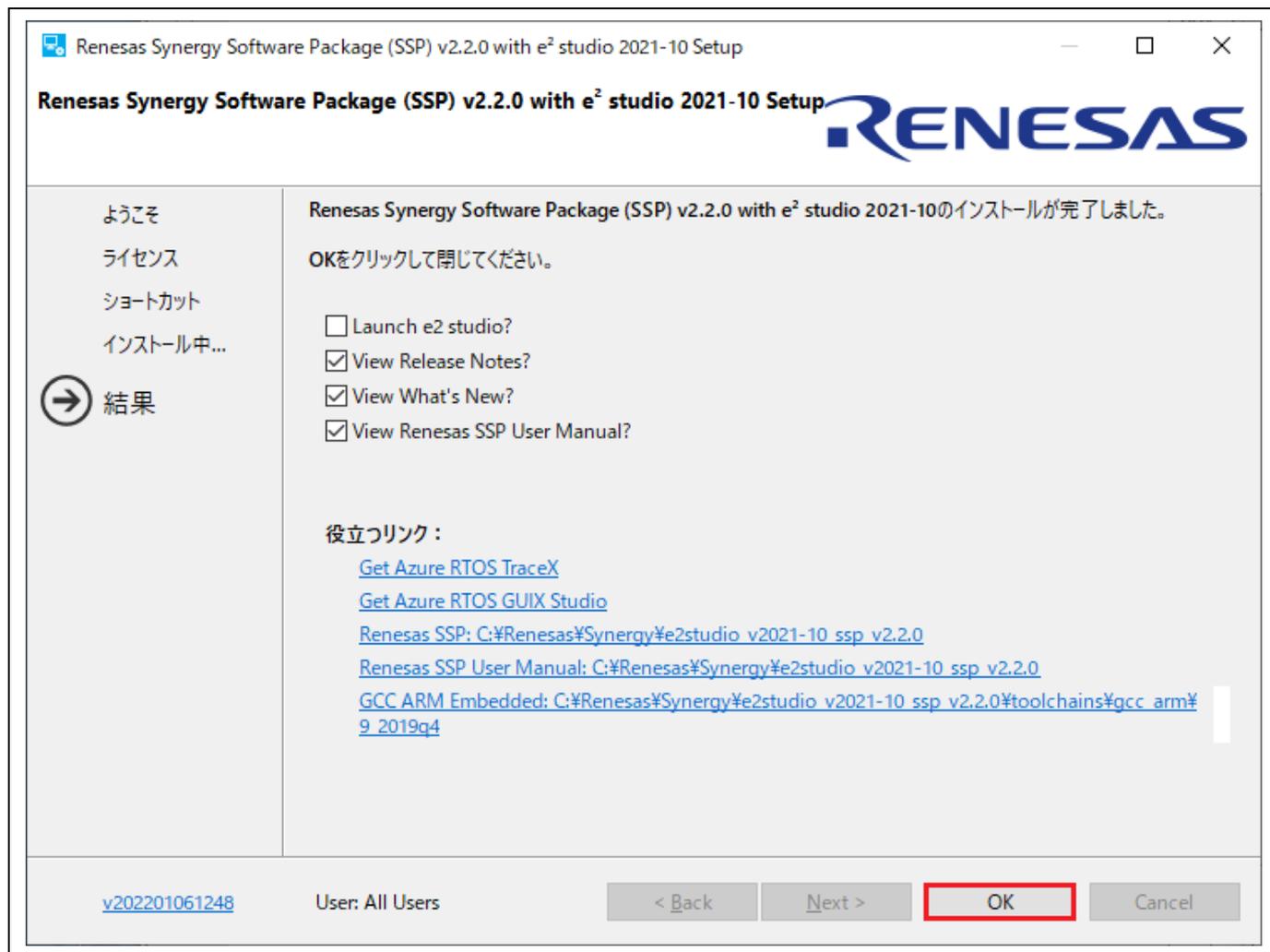


図 2-7 インストール – インストールの完了

## 2.2 e<sup>2</sup> studio と SSP の個別インストール

この章では、以下のコンポーネントを個別にインストールする場合の手順を説明します。

- e<sup>2</sup> studio
- GCC Arm Embedded
- Renesas Synergy™ Software Package (SSP)

### 2.2.1 e<sup>2</sup> studio のインストール

以下の手順で、Synergy 向けの e<sup>2</sup> studio をインストールしてください。

- (1) 使用する FSP に対応した e<sup>2</sup> studio (64 ビットバージョン) 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/jp/ja/software-tool/e-studio>
- (2) ダウンロードしたファイルを解凍し、e<sup>2</sup> studio インストーラを起動すると e<sup>2</sup> studio インストールウィザードページが開きます。
- (3) すでに e<sup>2</sup> 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 e<sup>2</sup> 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] ボタンで次に進みます。

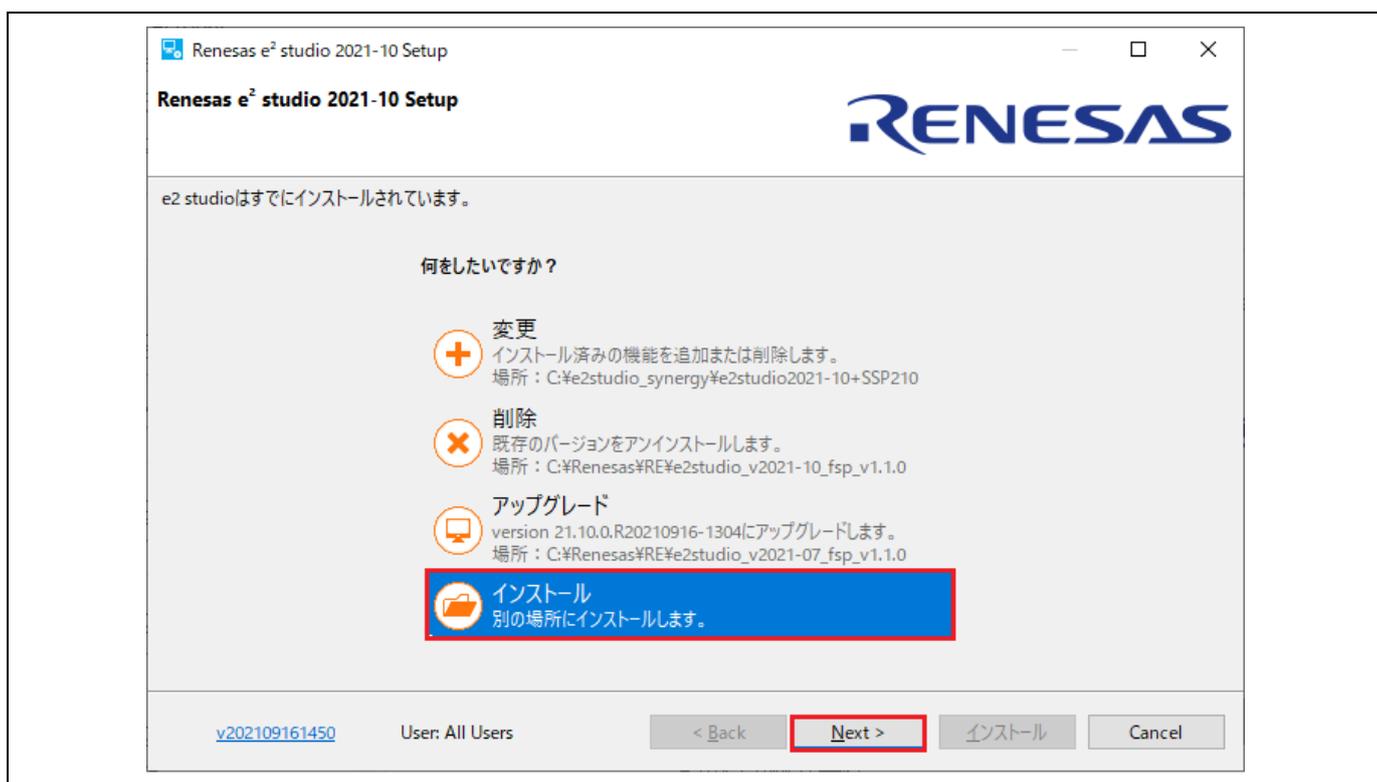


図 2-8 e<sup>2</sup> studio 複数バージョンのインストール

(4) [ようこそ] ページ

デフォルトのインストール先である “C:\%Renesas%e2\_studio” が設定されています。  
 [変更...] をクリックすると、インストール先を変更できます。[Next] ボタンで次に進みます。



図 2-9 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

“Renesas Synergy” 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] ボタンで次に進みます。

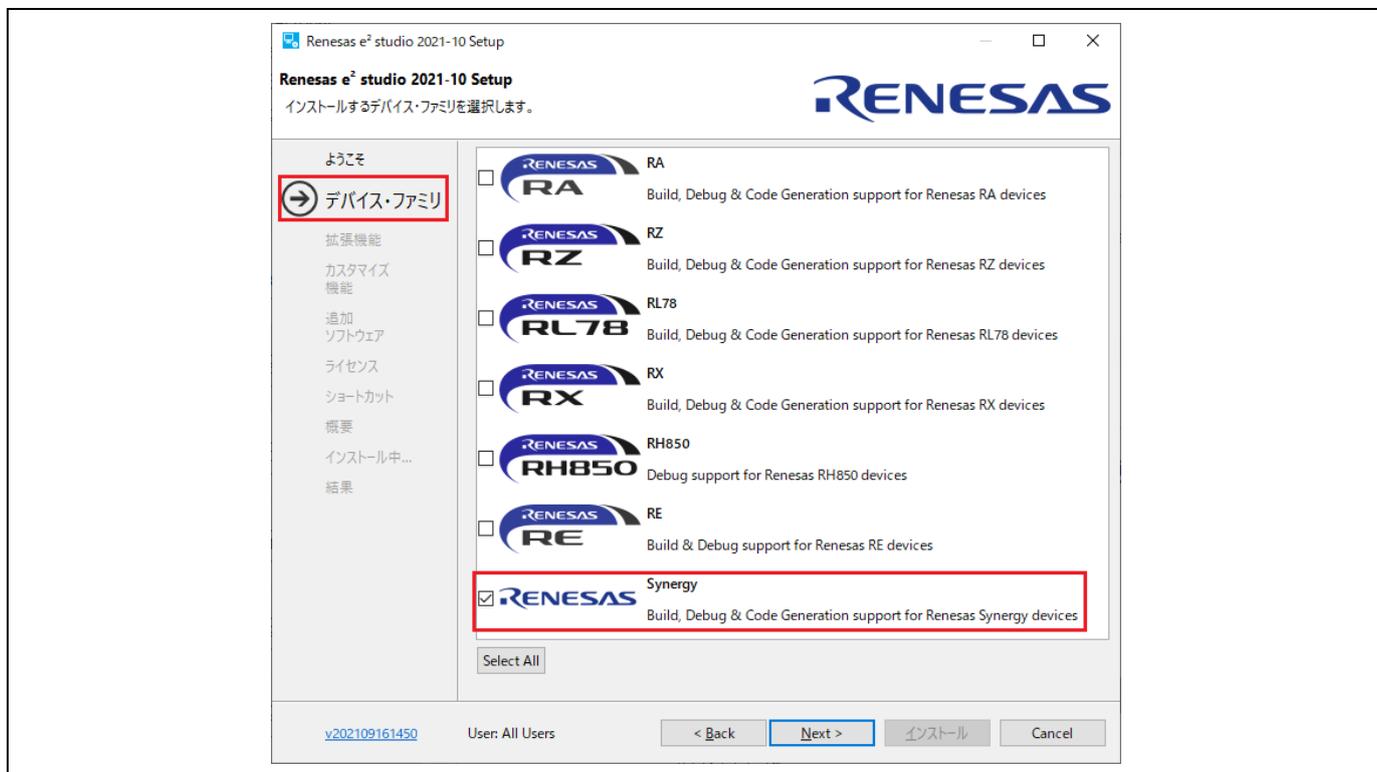


図 2-10 インストール - [デバイス・ファミリ] ページ

## (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、Git、RTOS のサポートプラグイン等）を選択してください。

英語以外の言語を利用する場合はここで言語パックを選択しておく必要があります。

[Next] ボタンで次に進みます。

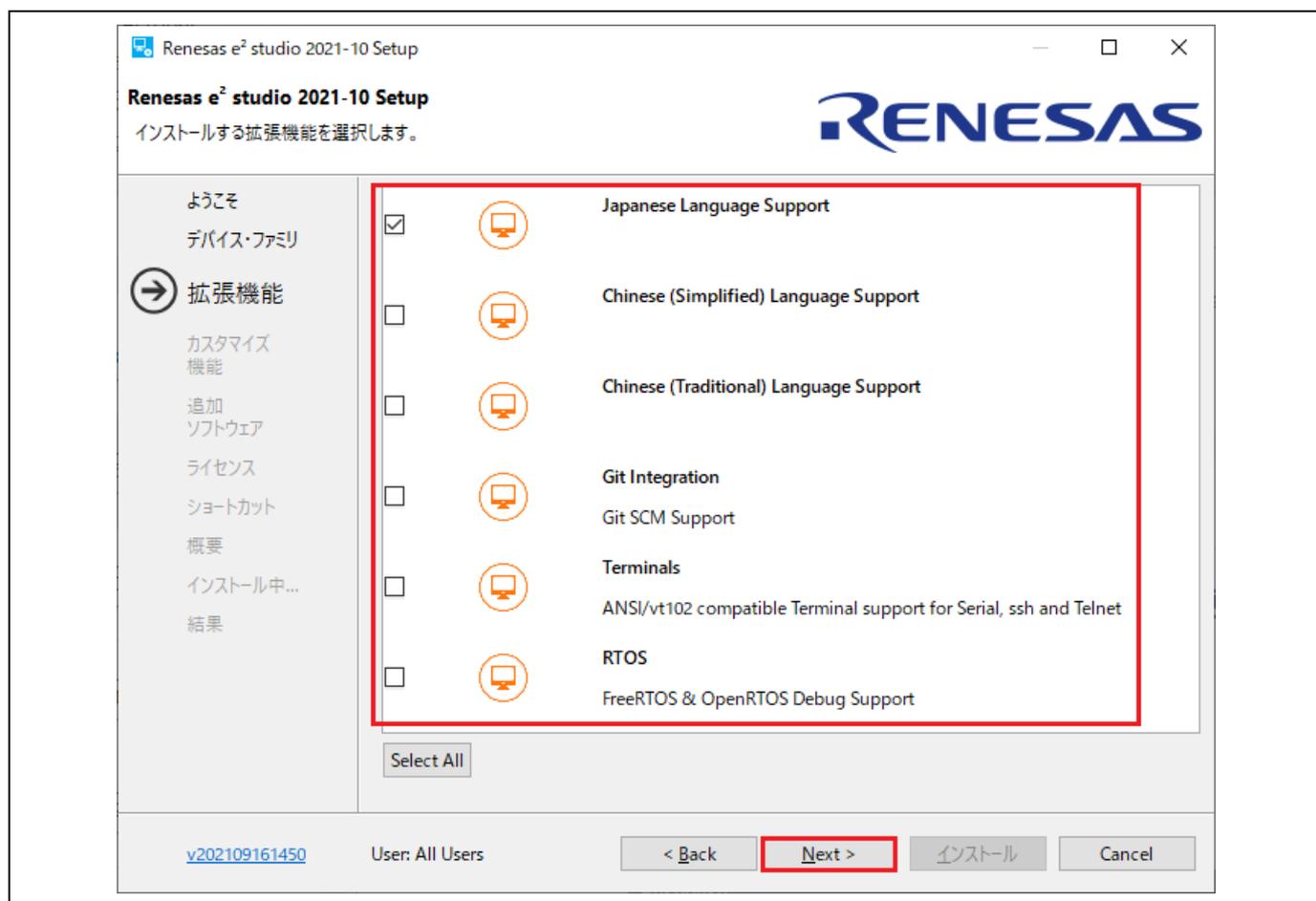


図 2-11 インストール - [拡張機能] ページ

(7) [カスタマイズ機能] ページ

必ず “Renesas Synergy Family Support” を選択してください。[Next] ボタンで次に進みます。

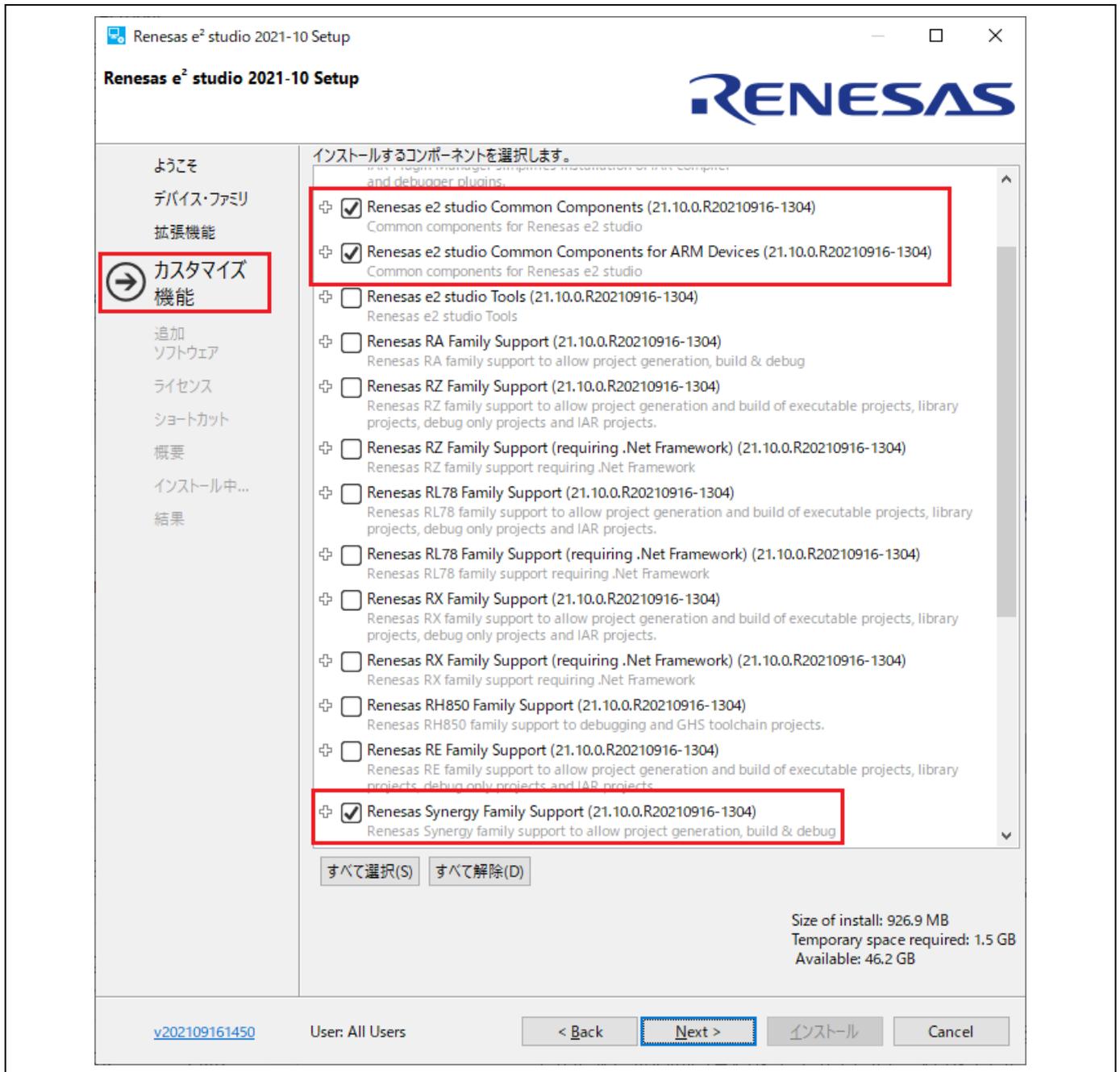


図 2-12 インストール – [カスタマイズ機能] ページ

## (8) [追加ソフトウェア] ページ

[追加ソフトウェア] ページで、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選択します（この環境でサポートする GCC ツールチェーンを追加します）。

[Next] ボタンで次に進みます。

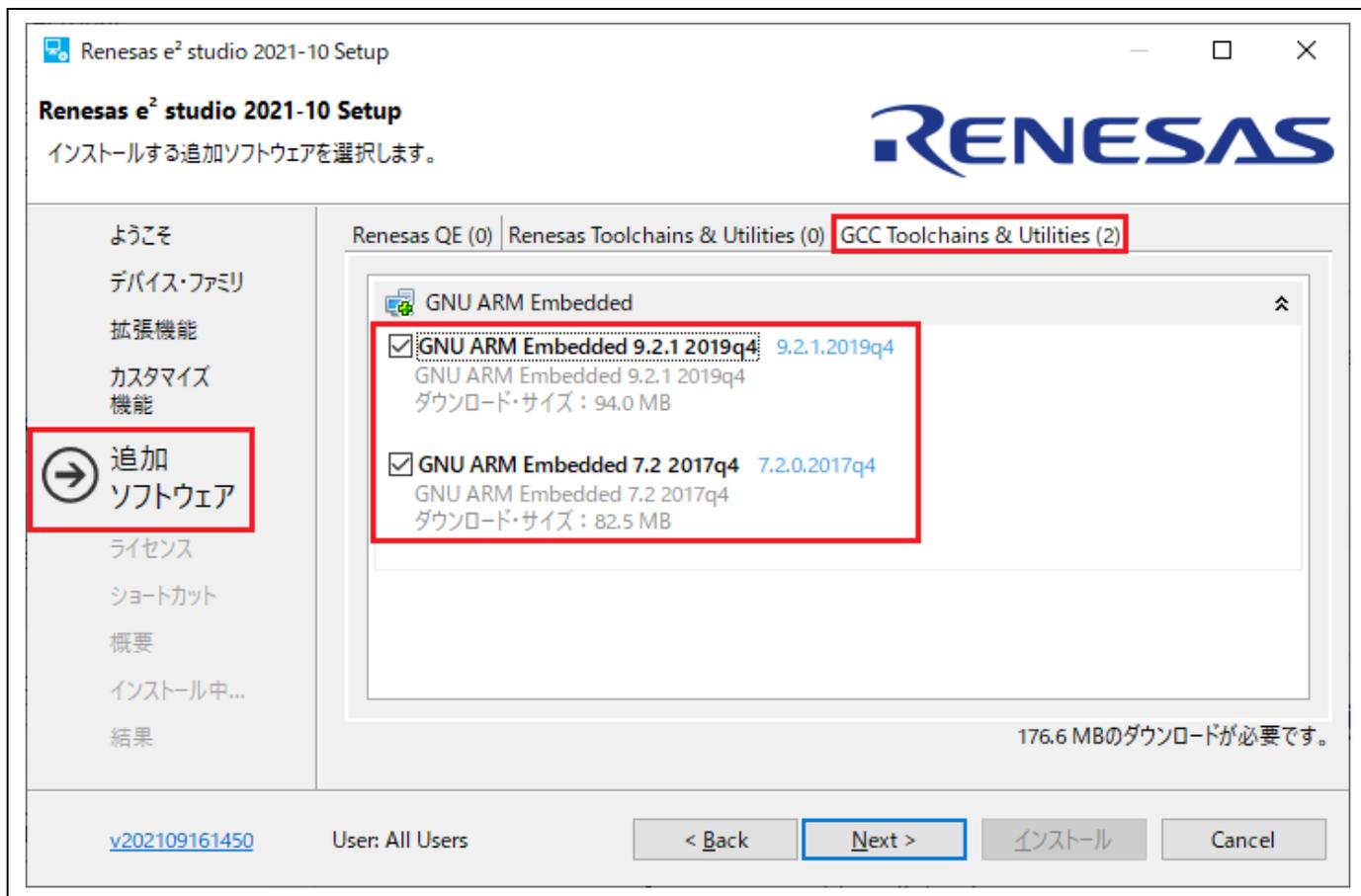


図 2-13 インストール - [追加ソフトウェア] ページ

## (9) ライセンス

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

## (10) ショートカット

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

## (11) 概要

[インストール] ボタンを押して e<sup>2</sup> studio をインストールしてください。

## (12) インストール中...

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

## 2.2.2 GNU Arm コンパイラの設定

GNU Arm Toolchain は e<sup>2</sup> studio のインストールと同時にインストールできます。

GNU Arm コンパイラを単独で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) Renesas Synergy™ がサポートする SSP で動作確認したバージョン（例えば v9.2.1）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 コンパイラ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” を選択し、[Finish] ボタンを押してインストールを完了します。

## 2.2.3 Renesas Synergy™ Software Package (SSP) のインストール

e<sup>2</sup> 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、ドライブライブラリ、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、以下の手順を実行してください。

- (1) Synergy プラットフォーム Web サイトにアクセスします。
- (2) "Synergy Software Package (SSP)" を選択して SSP ページを開いてください。
- (3) [Synergy Download] の [ダウンロード] を選択して My Renesas アカウントへログインします。
- (4) [Download Standalone Installer] を選択し、SSP\_Distribution\_<SSP-version>.zip ファイルをダウンロードします。（MyRenesas アカウントにサインインしていると [Download Standalone Installer] が有効になります）。

SSP のリリースノートとユーザーズマニュアルも、[Synergy Software Package] ページからダウンロードできます。

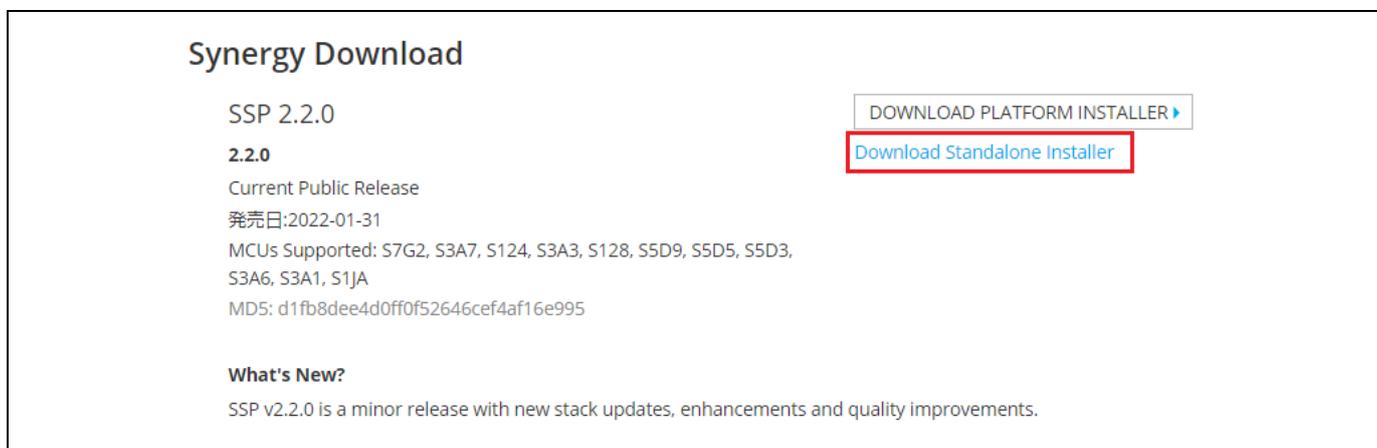


図 2-14 インストール - スタンドアロン SSP パッケージのダウンロード

- (5) 互換性のある e<sup>2</sup> studio がインストール済みであること、e<sup>2</sup> 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 e<sup>2</sup> studio を実行しないでください。
- (6) 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。
- (7) インストールウィザードダイアログで [Next] ボタンをクリックします。
- (8) ライセンス契約を確認し、[I AGREE] をクリックしてインストールを続けます。

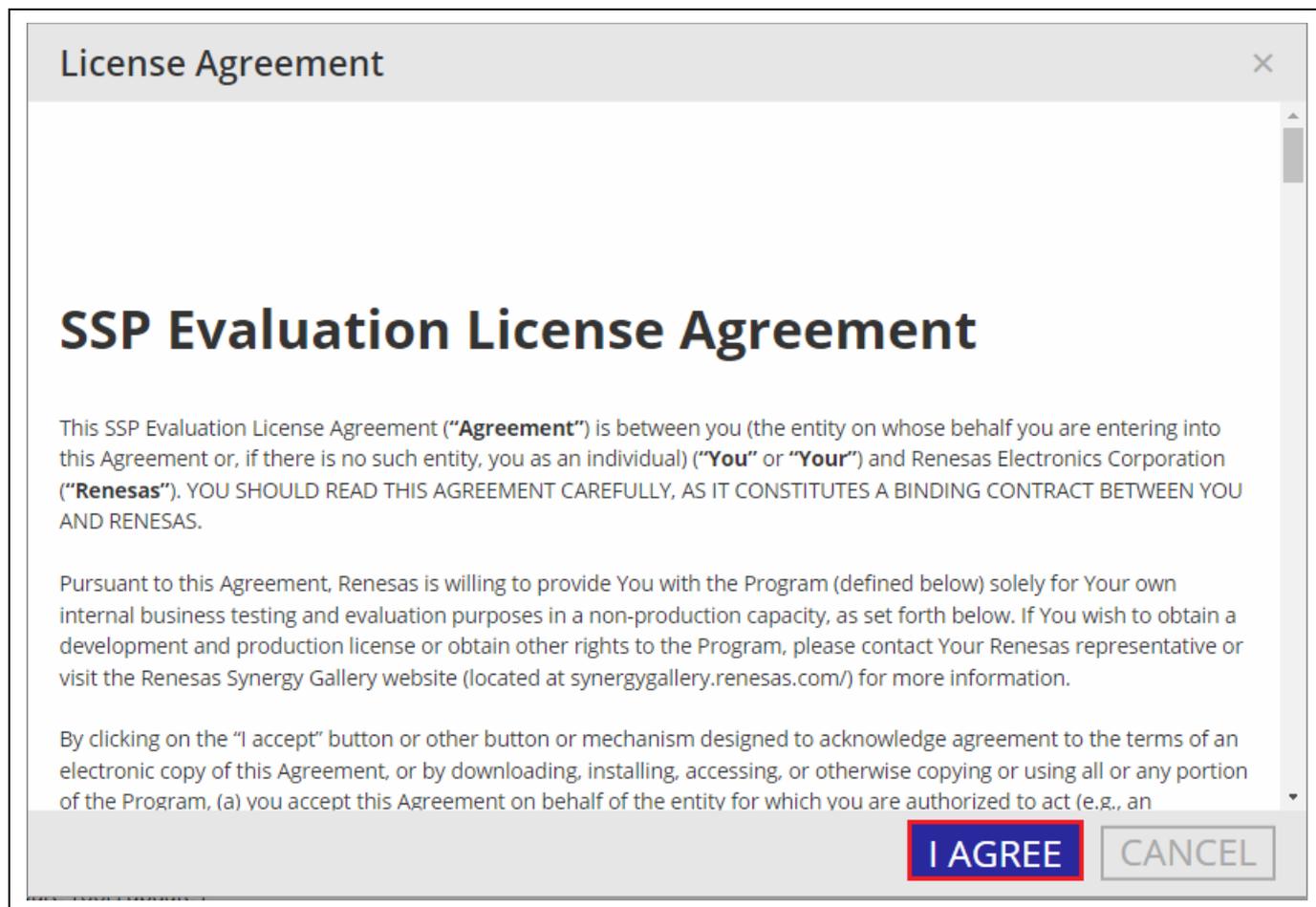


図 2-15 インストール - ルネサス Web サイト上のライセンス契約

- (9) SSP を、e<sup>2</sup> 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\%e2\_studio）にインストールします。この C:\Renesas\%e2\_studio フォルダが、SSP のデフォルトインストールフォルダです。[Install] ボタンをクリックしてインストールを開始してください。

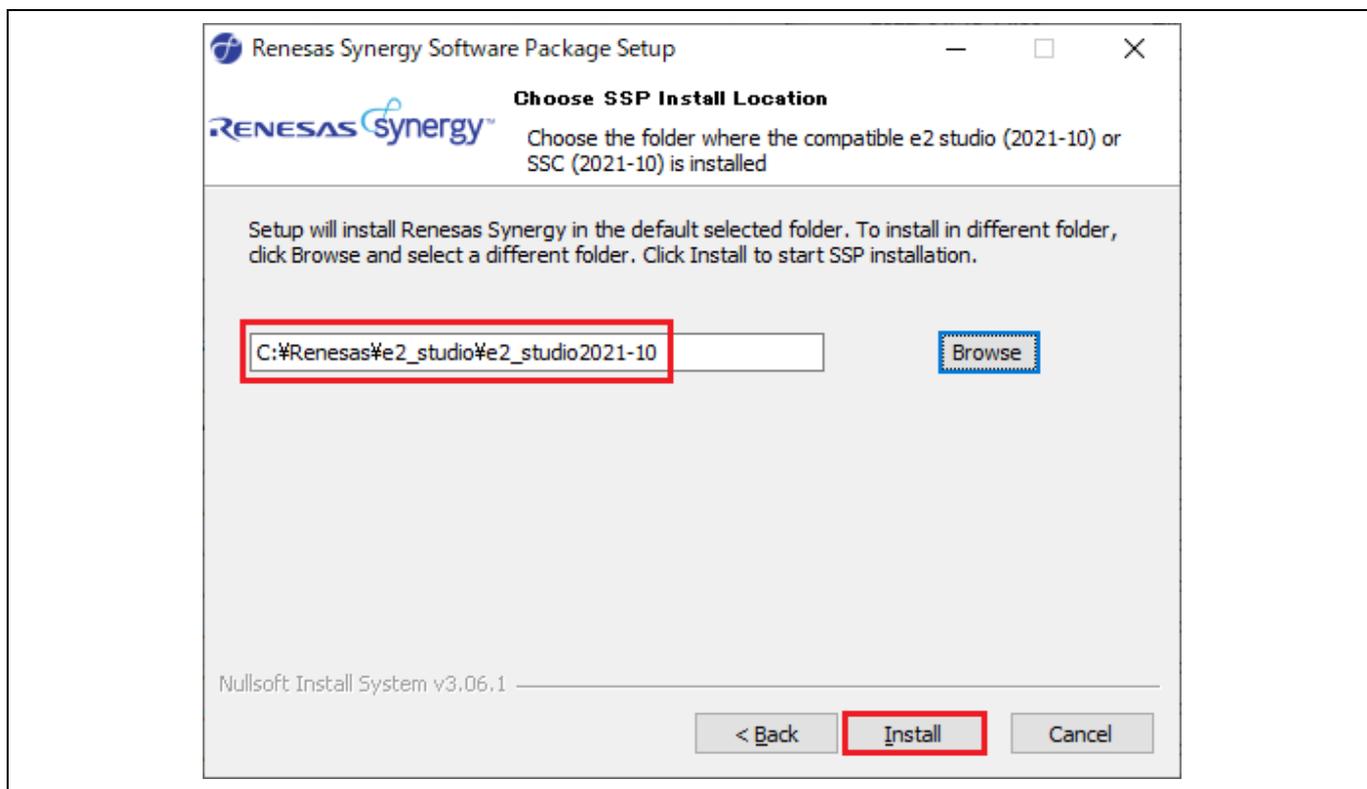


図 2-16 インストール – SSP のインストール先フォルダの選択

- (10)インストールが完了した時点で [Close] 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 のインストール後、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 e<sup>2</sup> studio のアンインストール

e<sup>2</sup> 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] → [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 e<sup>2</sup> 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、e<sup>2</sup> 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 e<sup>2</sup> 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

### 3. プロジェクトの作成

この章では、新規 Synergy プロジェクトの作成について説明します。e<sup>2</sup> studio は、新規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 Synergy 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 SSP 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。

#### 3.1 新規 Synergy プロジェクトの作成

e<sup>2</sup> studio ではプロジェクト作成用に簡単なウィザードが用意されています。プロジェクト名、対応するデバイスとボード、プロジェクトタイプ、出力オブジェクトタイプ、プロジェクトテンプレートを指定して、新規 Synergy プロジェクトを作成できます。

e<sup>2</sup> 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 Synergy プロジェクトを作成してください。

- (1) [ファイル] → [新規] → [Synergy C/C++ Project] の順に選択します。

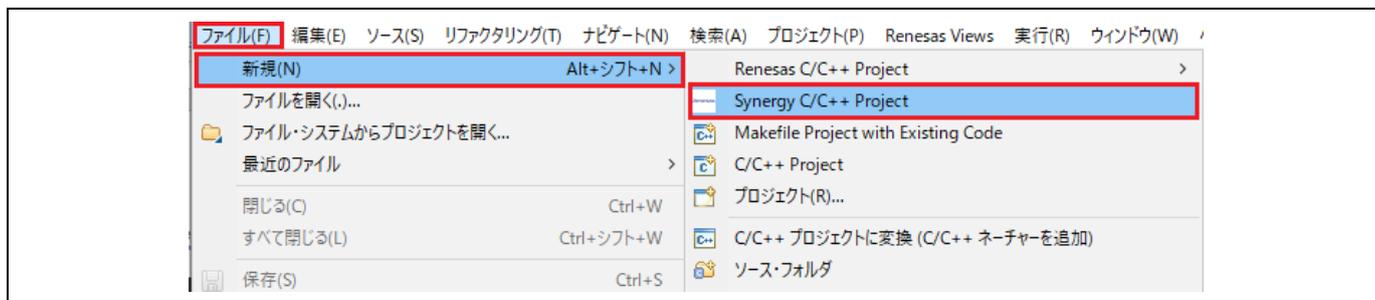


図 3-1 プロジェクトの作成 — 新規プロジェクトの作成

(2) “Renesas Synergy 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。

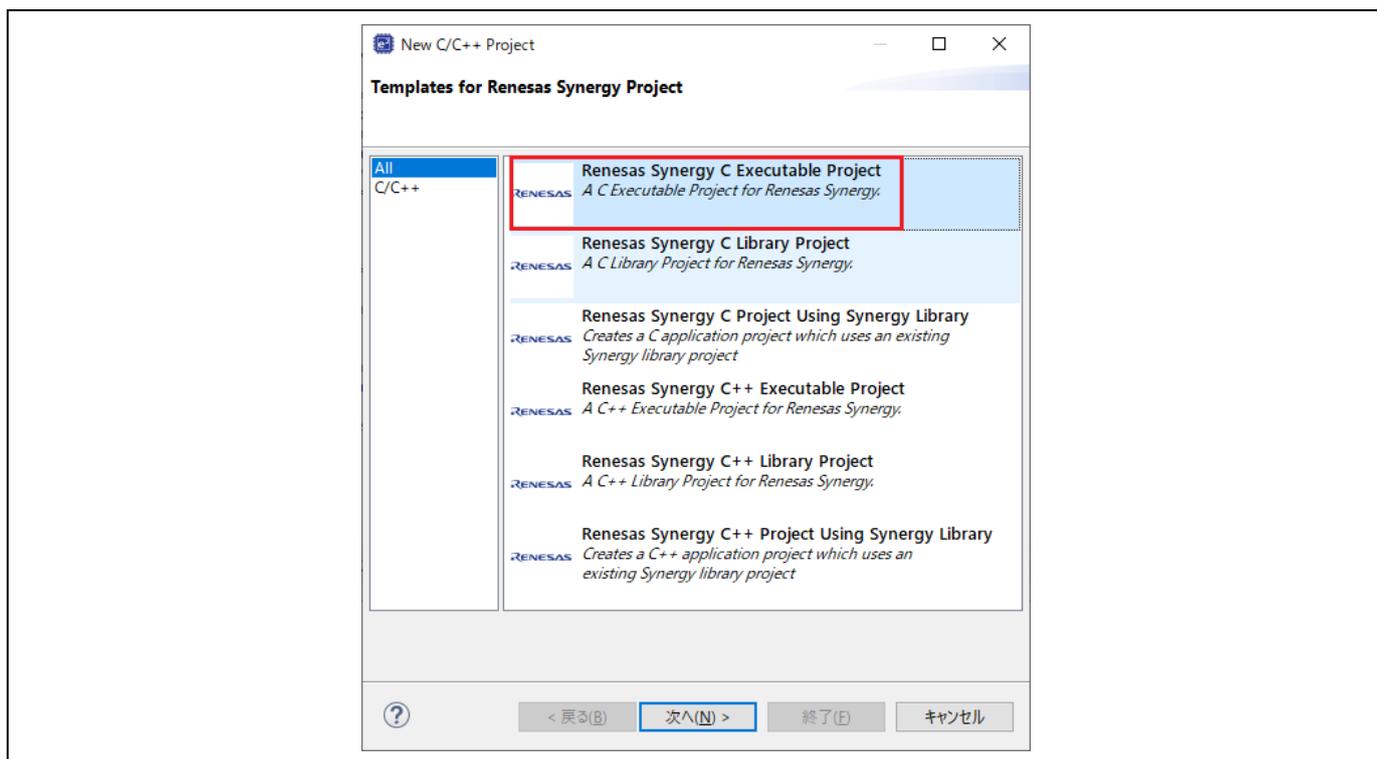


図 3-2 プロジェクトの作成 – 実行可能プロジェクトテンプレートの選択

(3) 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 : ボードを選択します (例 : S7G2 SK)
- Toolchain version : Renesas Synergy™ 向けに認定された最新の GNU コンパイラを選択します (例 : GCC ARM Embedded 9.2.1.20191025)。  
注 : IAR Embedded Workbench® for Renesas Synergy™ (EWSYN) は、ルネサス Web サイトからのダウンロード提供を終了しました。入手については IAR システムズ社にお問い合わせください。また、EWSYN のライセンスも IAR システムズ社が発行しますので、ライセンスに関する情報やライセンスキーの新規発行や更新についても [IAR システムズ社](#)にお問い合わせください。
- その他はすべてデフォルト設定のままにします。
- [次へ] で次に進みます。

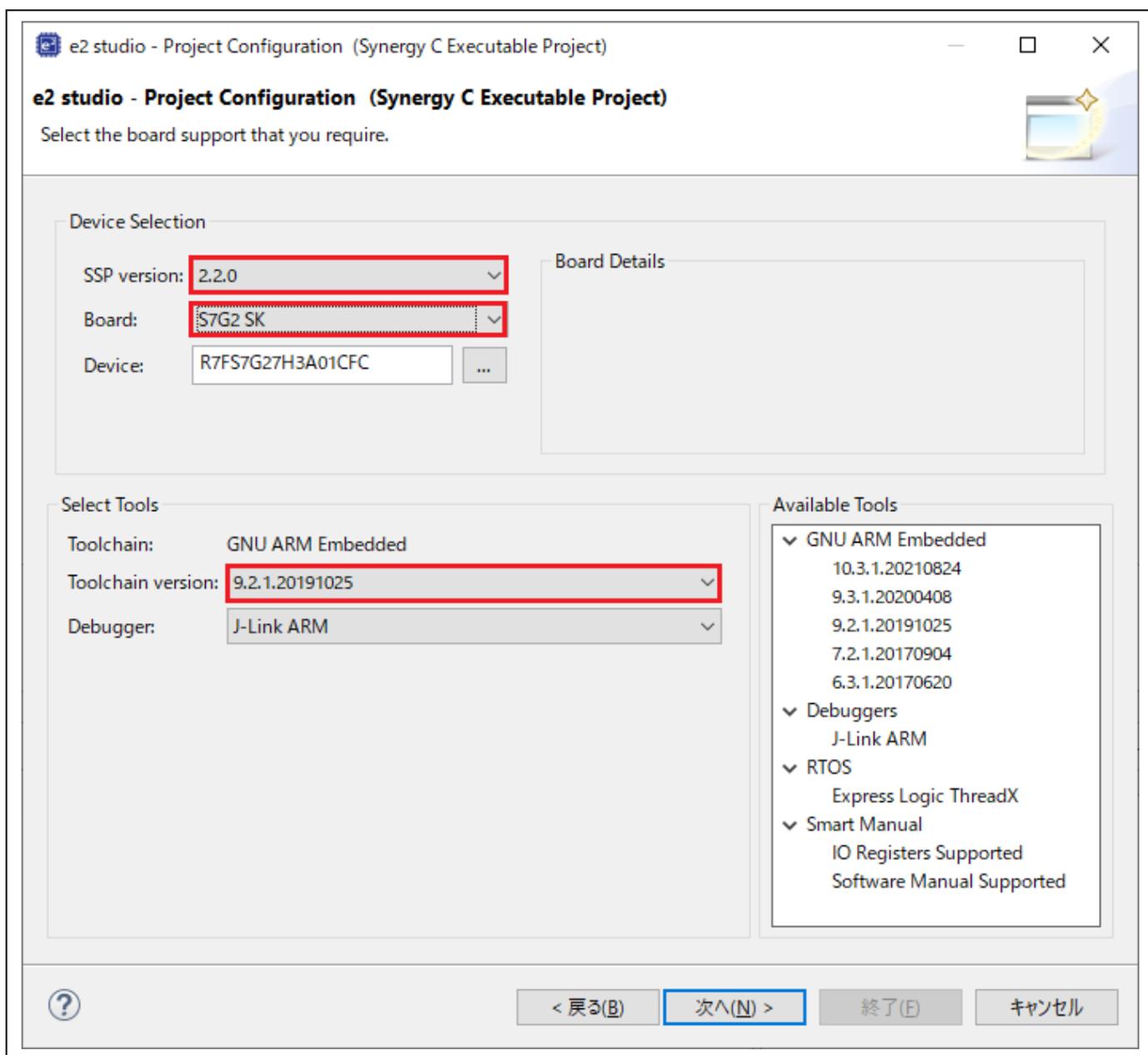


図 3-3 プロジェクトの作成 - デバイスの選択

(4) プロジェクトテンプレート選択の画面でテンプレートを選択します（例：Blinky）。

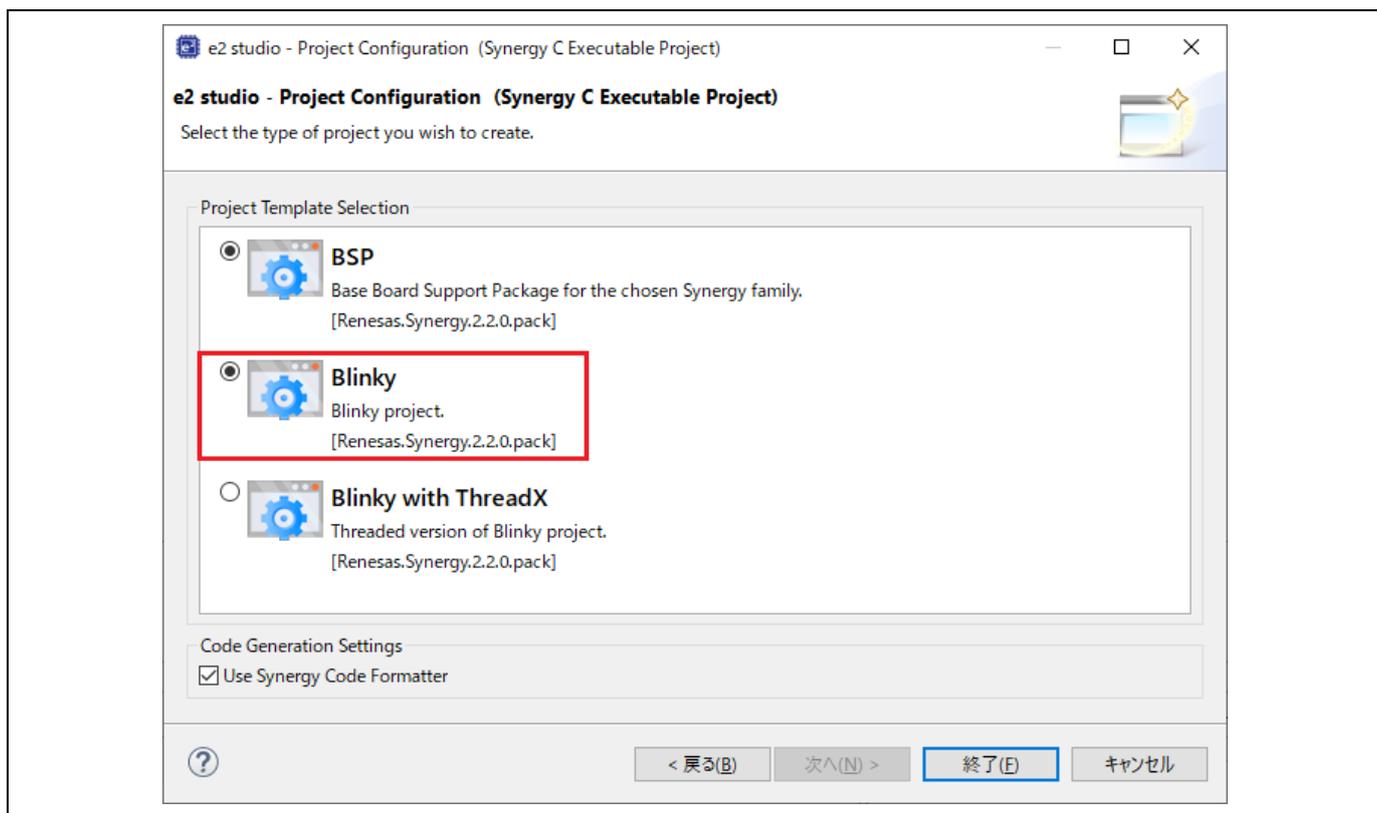


図 3-4 プロジェクトの作成 – プロジェクトテンプレート

(5) [終了] ボタンを押すと新規プロジェクトが作成されます。

[Synergy 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る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、[Synergy プロジェクト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。



図 3-5 プロジェクトの作成 – 新規プロジェクトのビュー

## 3.2 既存 Synergy プロジェクトのインポート

既存の Synergy プロジェクトは以下の手順でインポートできます。

(1) [ファイル] → [インポート] の順に選択します。

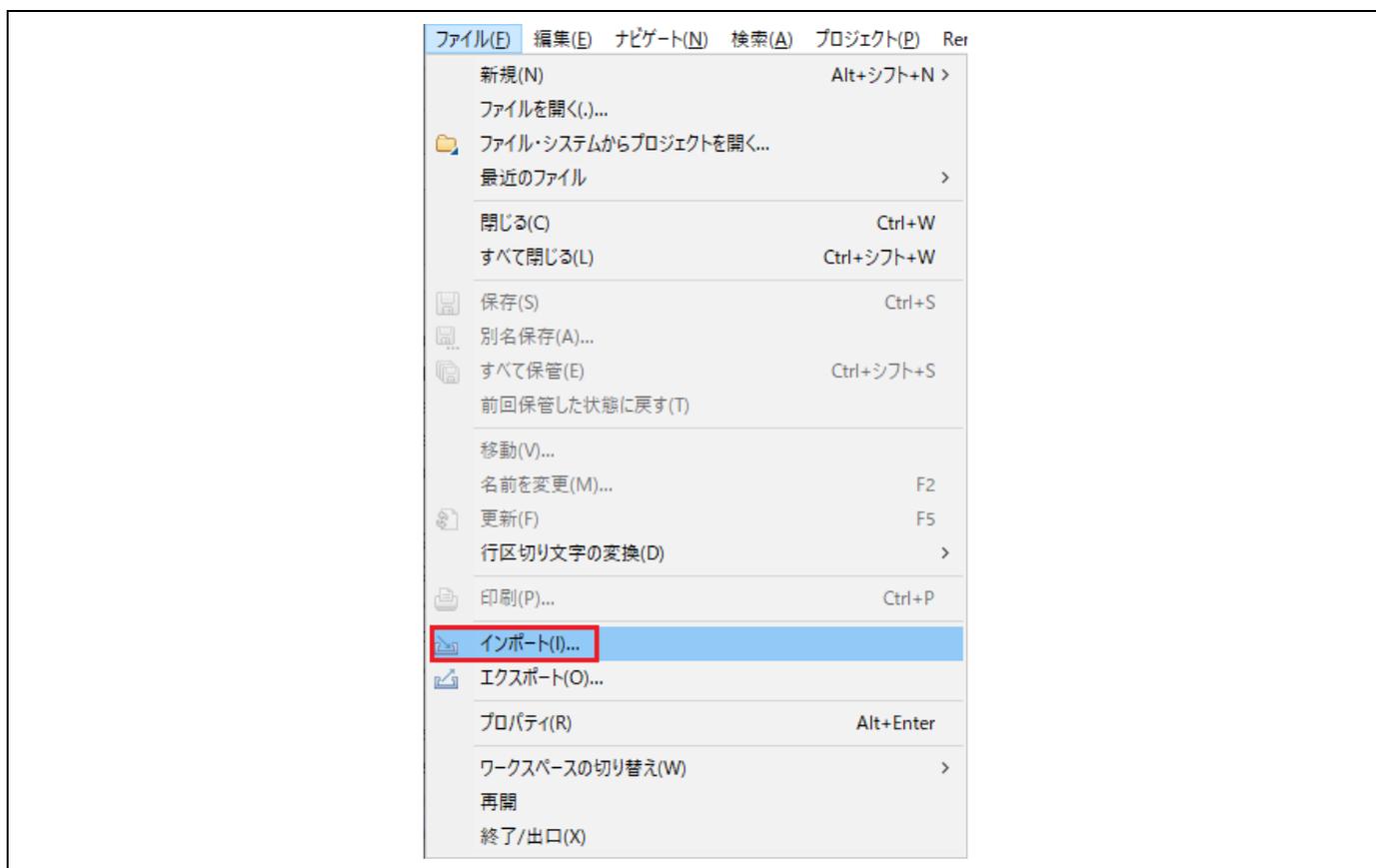


図 3-6 プロジェクトのインポート

(2) [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] を選択してください。[次へ] で次に進みます。

注：インポートするプロジェクトの名称を変更するには、[一般] → [Rename & Import Existing C/C++ Project into Workspace] を選択してください。

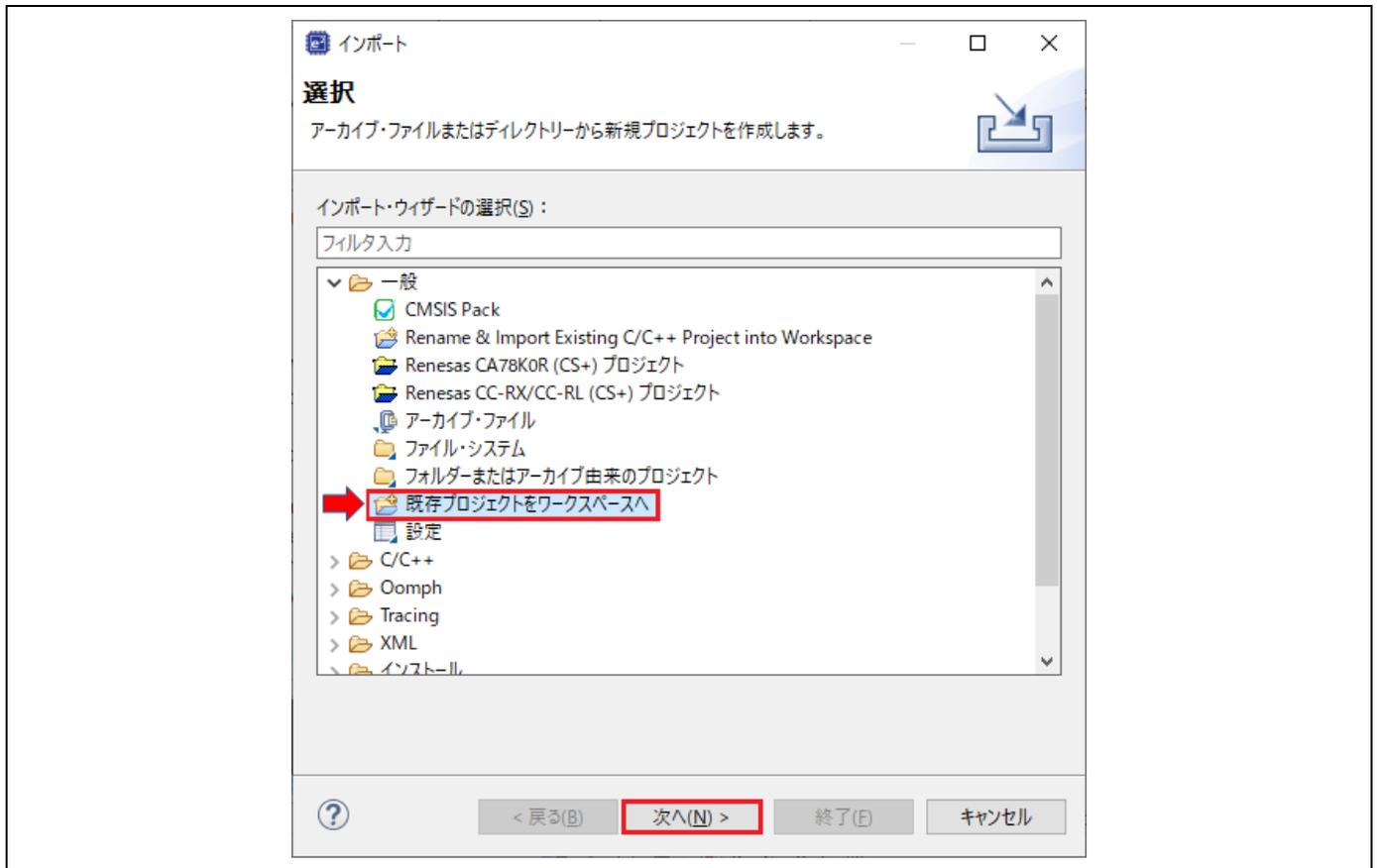


図 3-7 インポート方法の選択

- (3) [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、右の [参照...] ボタンでプロジェクトが入っている圧縮ファイル (.zip) を選択します。  
 既存プロジェクトがフォルダに保存されている場合は、“ルート・ディレクトリーの選択” を選んでください。
- (4) インポートす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。

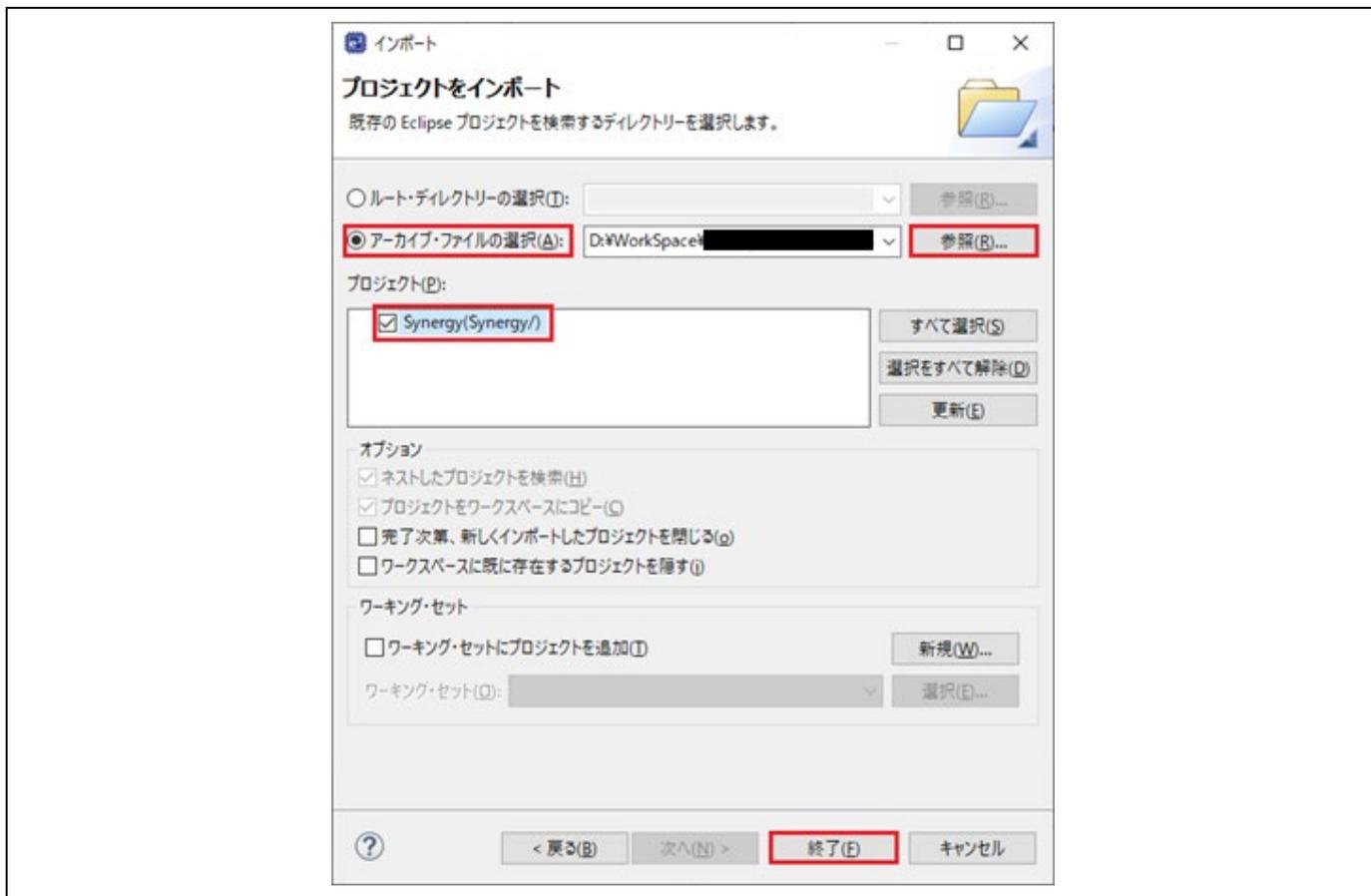


図 3-8 圧縮ファイルに保存されているプロジェクトの選択

- (5) 選択したプロジェクトが e<sup>2</sup> studio にインポートされます。

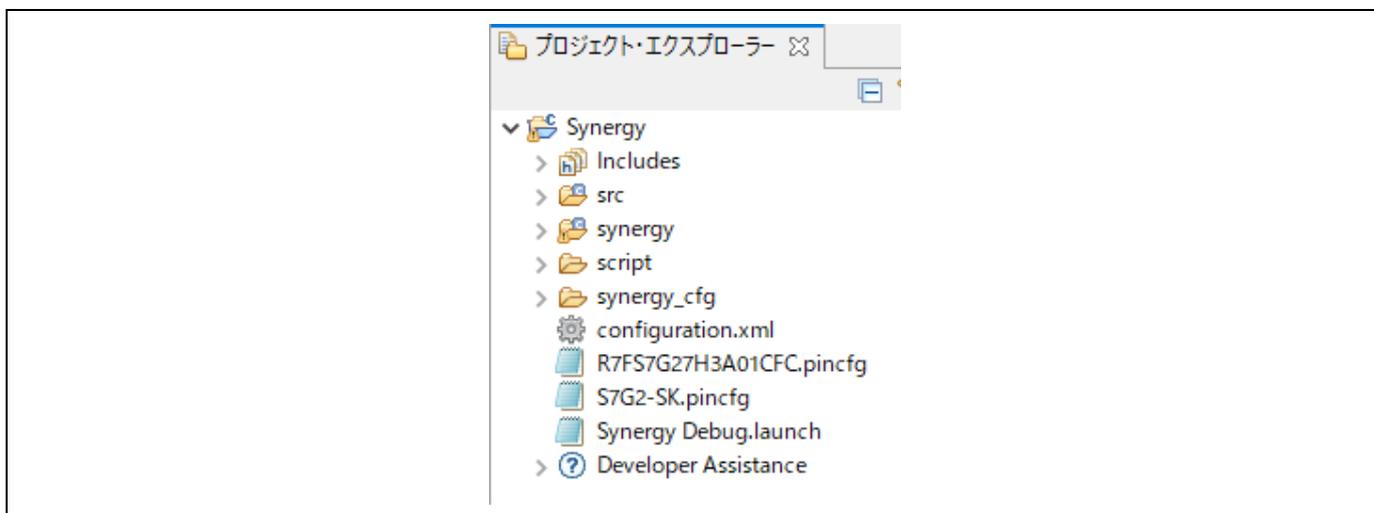


図 3-9 インポートされたプロジェクト

### 3.3 Synergy スタティックライブラリの作成と使用

この章では Synergy 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

Synergy 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) [ファイル] → [新規] → [Synergy C/C++ Project] の順に選択します。
- (2) “Renesas Synergy C Library 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。

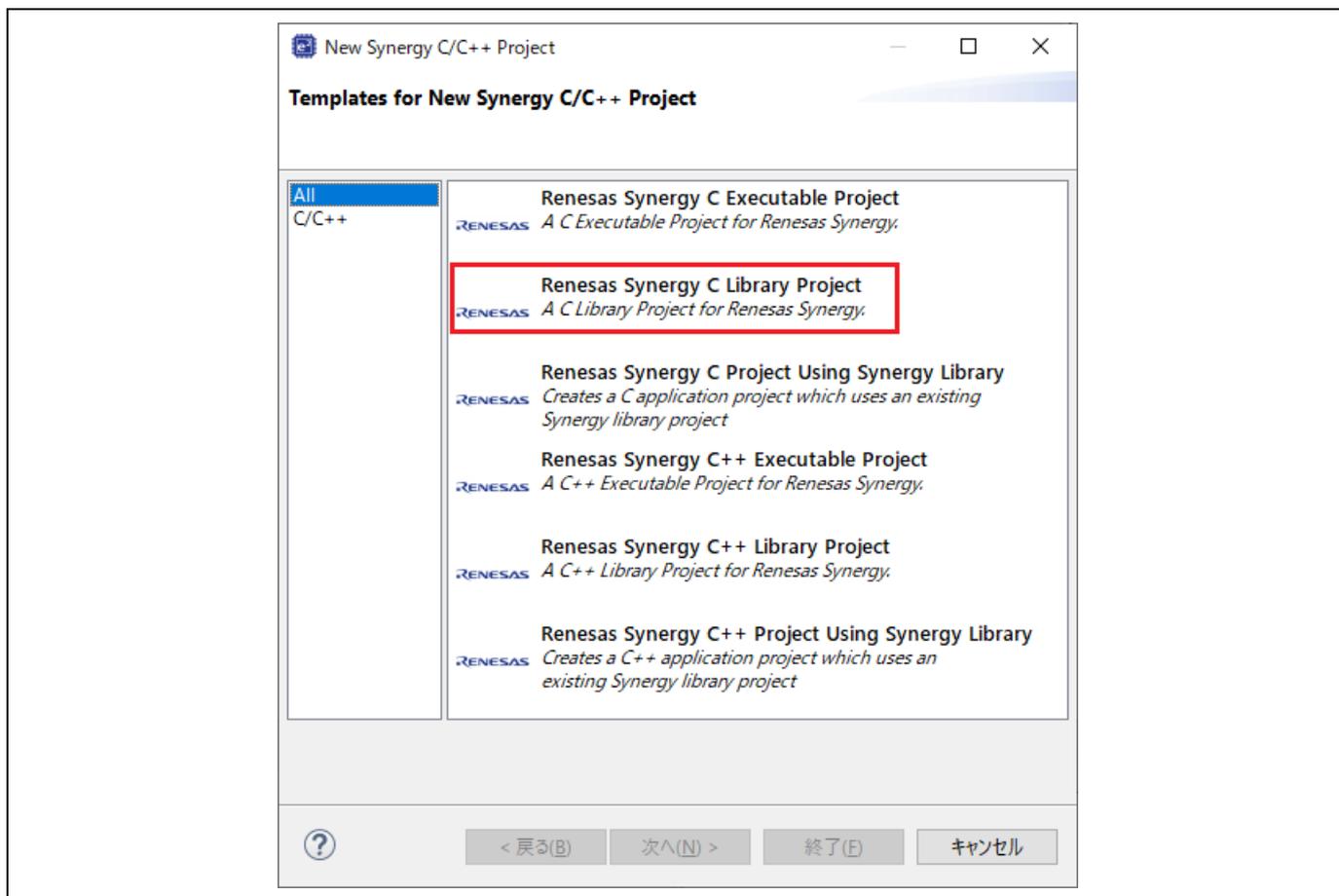


図 3-10 プロジェクトの作成 – ライブラリプロジェクトテンプレートの選択

- (3) デバイスとツール選択の画面で、実行プロジェクトで指定したものと同一デバイスおよびツールチェーンを選択し、[次へ] をクリックします。

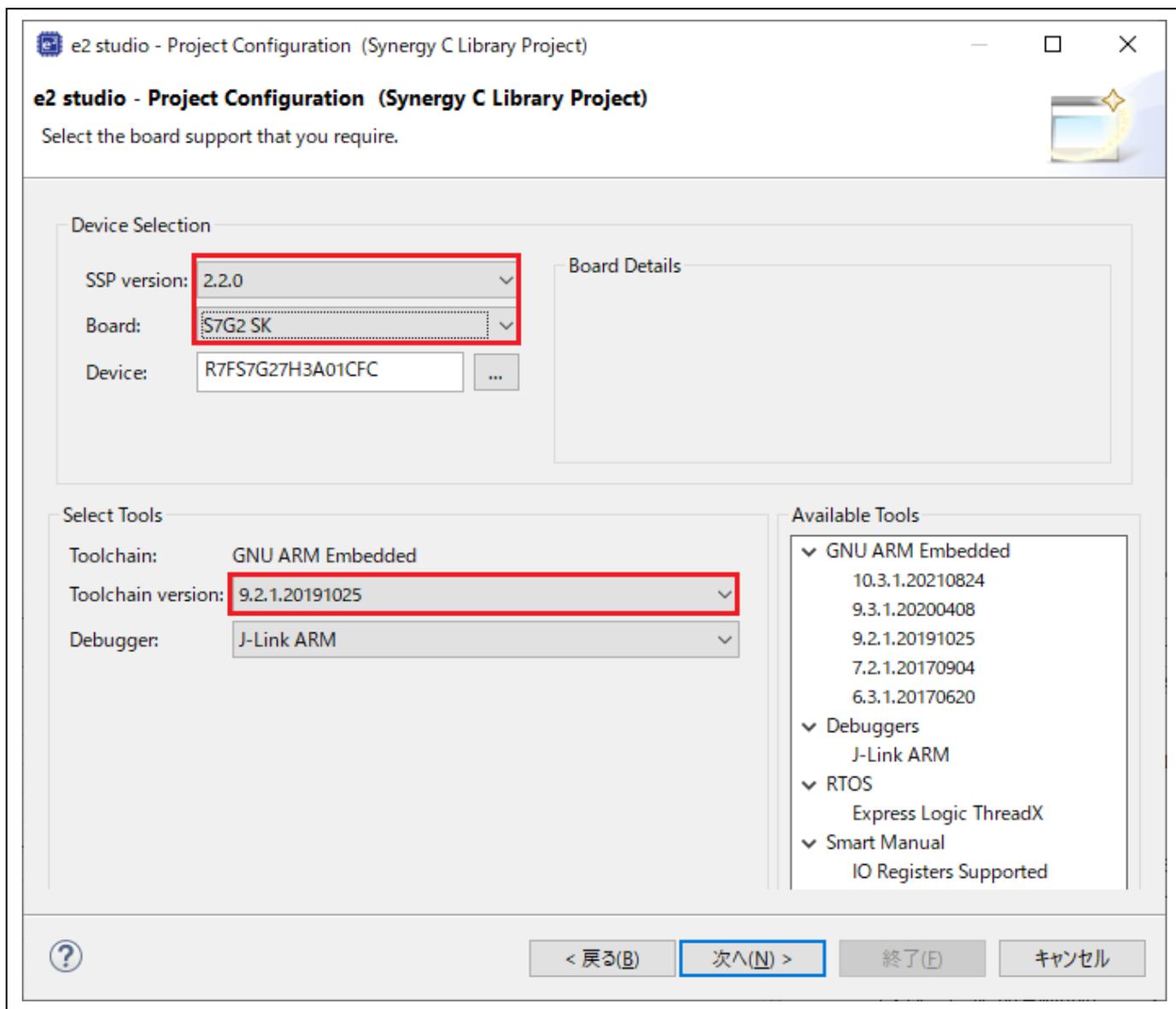


図 3-11 デバイスとツールチェーンの選択

- (4) プロジェクトテンプレート選択の画面で、“Blinky” テンプレートを選択してください。  
[終了] ボタンを押すとプロジェクトが作成されます。

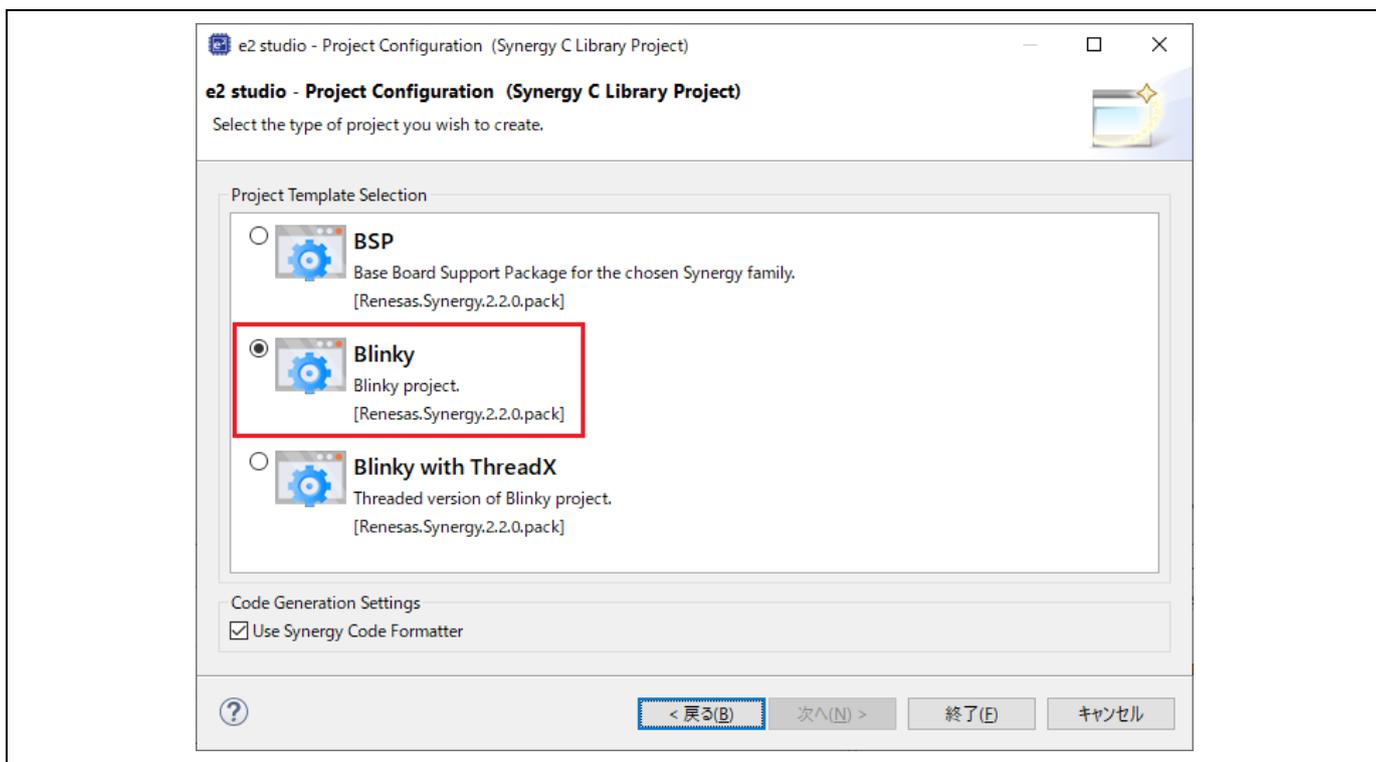


図 3-12 ライブラリ用プロジェクトテンプレートの選択

- (5) [Synergy] パースペクティブを開くよう e<sup>2</sup> studio の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。  
(6) [Generate Project Content] をクリックしてください。

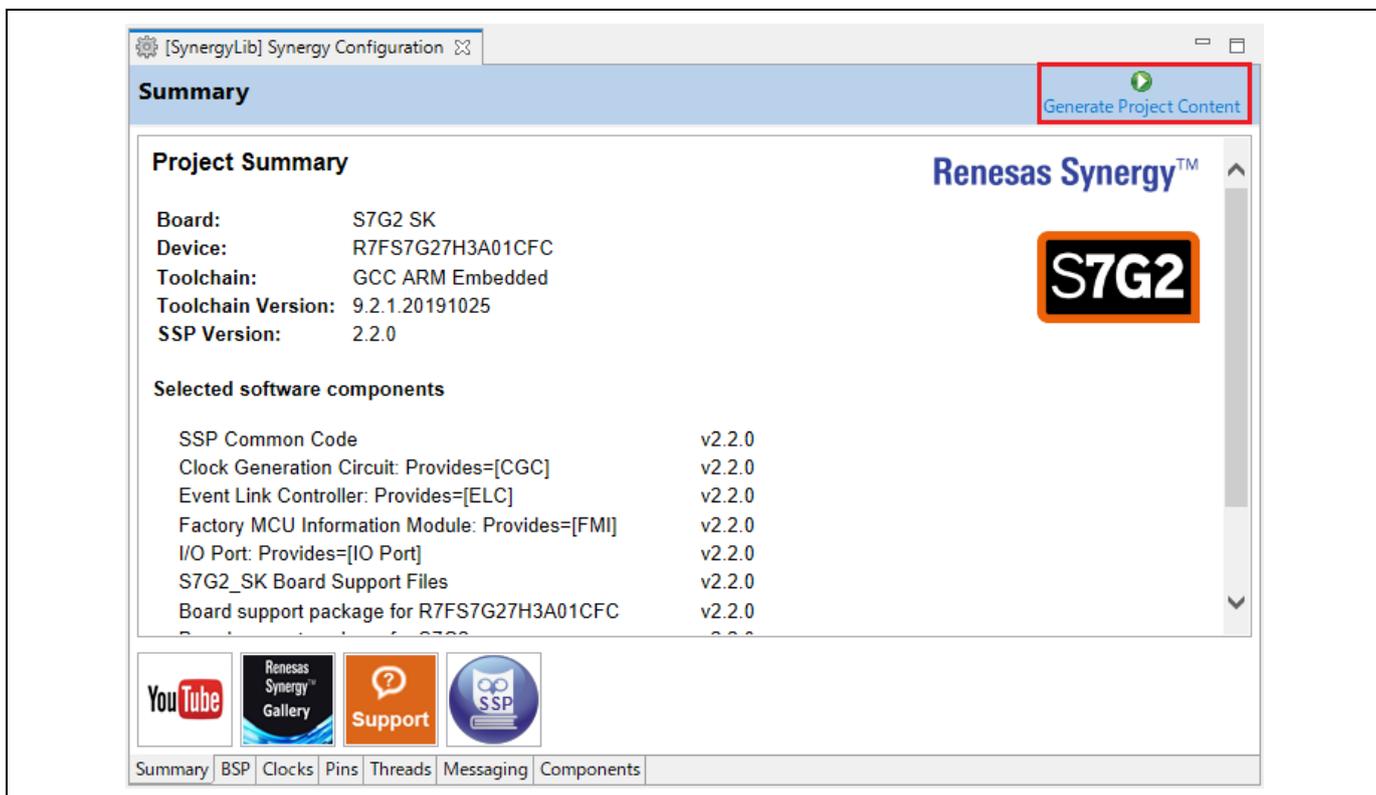


図 3-13 ライブラリプロジェクトの生成

(7) [プロジェクト・エクスプローラー] ビューで、synergylib¥src¥ 下の “hal\_entry.c” を開いてください。

```

2      * Copyright [2015-2021] Renesas Electronics Corporation and/or its
20     * File Name      : hal_entry.c
23
24     #include "hal_data.h"
25
27     * @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_MILLIS
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 */
    
```

図 3-14 既存の “hal\_entry.c”

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。

```

2      * Copyright [2015-2021] Renesas Electronics Corporation ar
20     * File Name      : hal_entry.c
23
24     #include "hal_data.h"
25
26     void hal_entry_lib();
28     * @brief Blinky example application
34     void hal_entry_lib(void) {
35
36         /* Define the units to be used with the software delay
37         const bsp_delay_units_t bsp_delay_units = BSP_DELAY_UNI
38         /* Set the blink frequency (must be <= bsp_delay_units
39         const uint32_t freq_in_hz = 2;
40         /* Calculate the delay in terms of bsp_delay_units */
41         const uint32_t delay = bsp_delay_units/freq_in_hz;
42         /* LED type structure */
43         bsp_leds_t leds;
    
```

図 3-15 新しい “hal\_entry.c”

- (8) 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 Synergylib¥Debug¥libSynergylib.a” が出力されます。

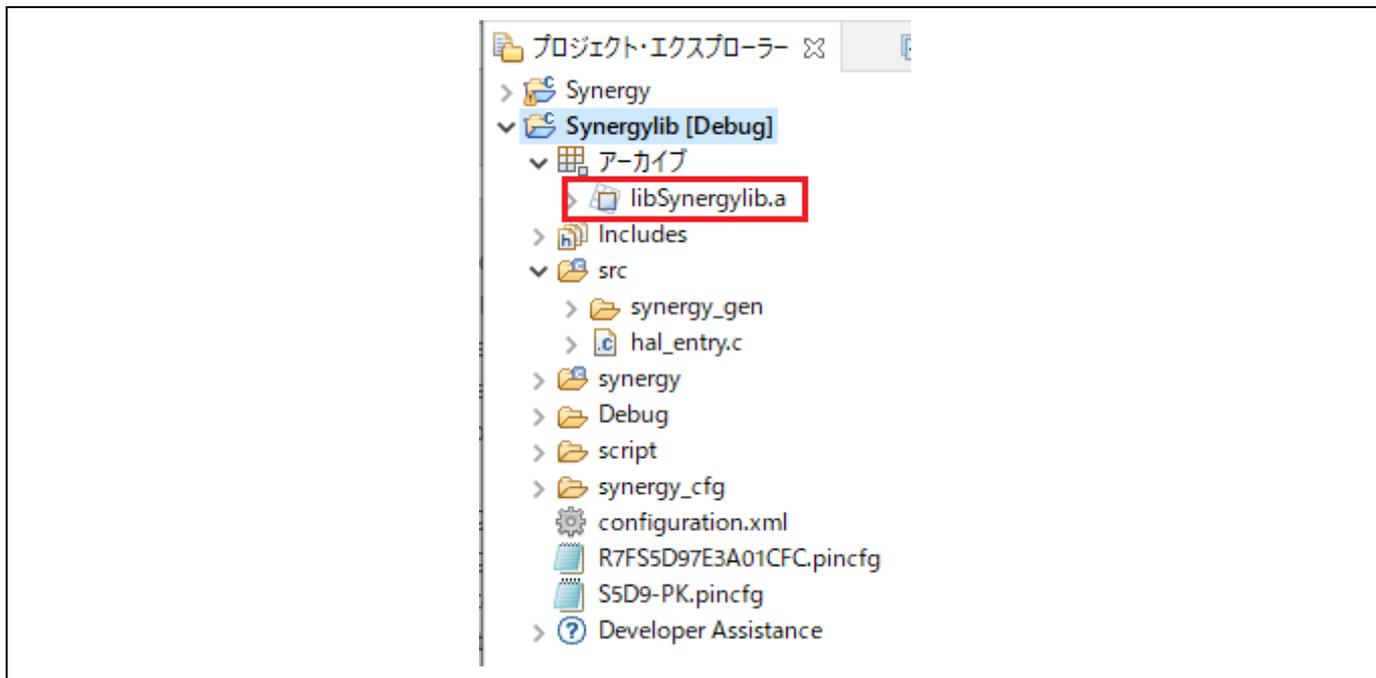


図 3-16 ビルドされたスタティックライブラリ

### 3.3.2 スタティックライブラリを実行プロジェクトで使用する

前章（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 Synergy 実行プロジェクトで使用方法を、以下の手順で説明します。

- Synergy 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
- Synergy 実行プロジェクトをビルドして実行する

以下の手順を実行します。

- (1) “Renesas Synergy C Project Using Synergy Library” テンプレートで実行プロジェクトを作成します。

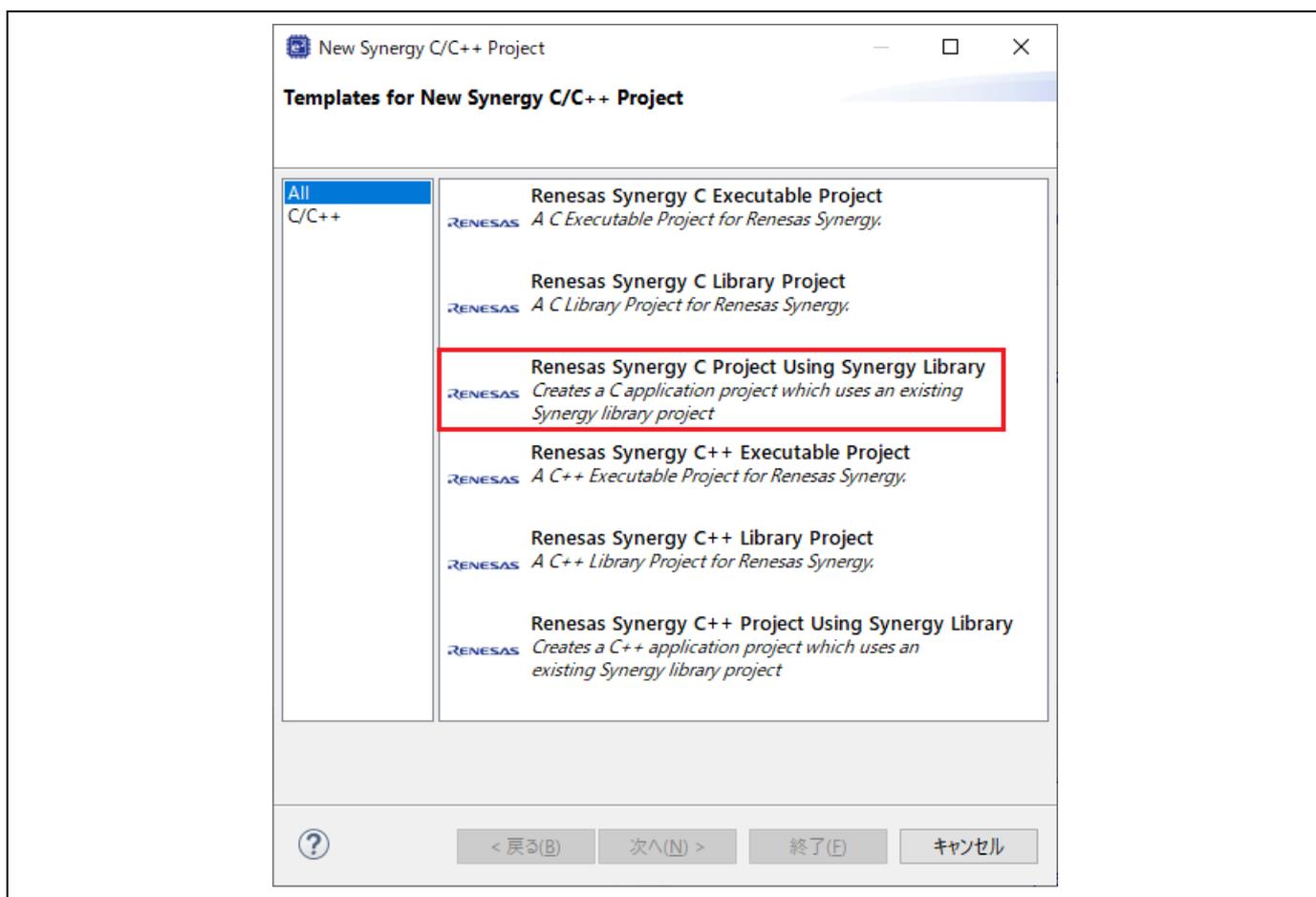


図 3-17 Synergy ライブラリを用いる実行プロジェクト用のテンプレート選択

(2) プロジェクト名を“Synergyapp”と指定します。

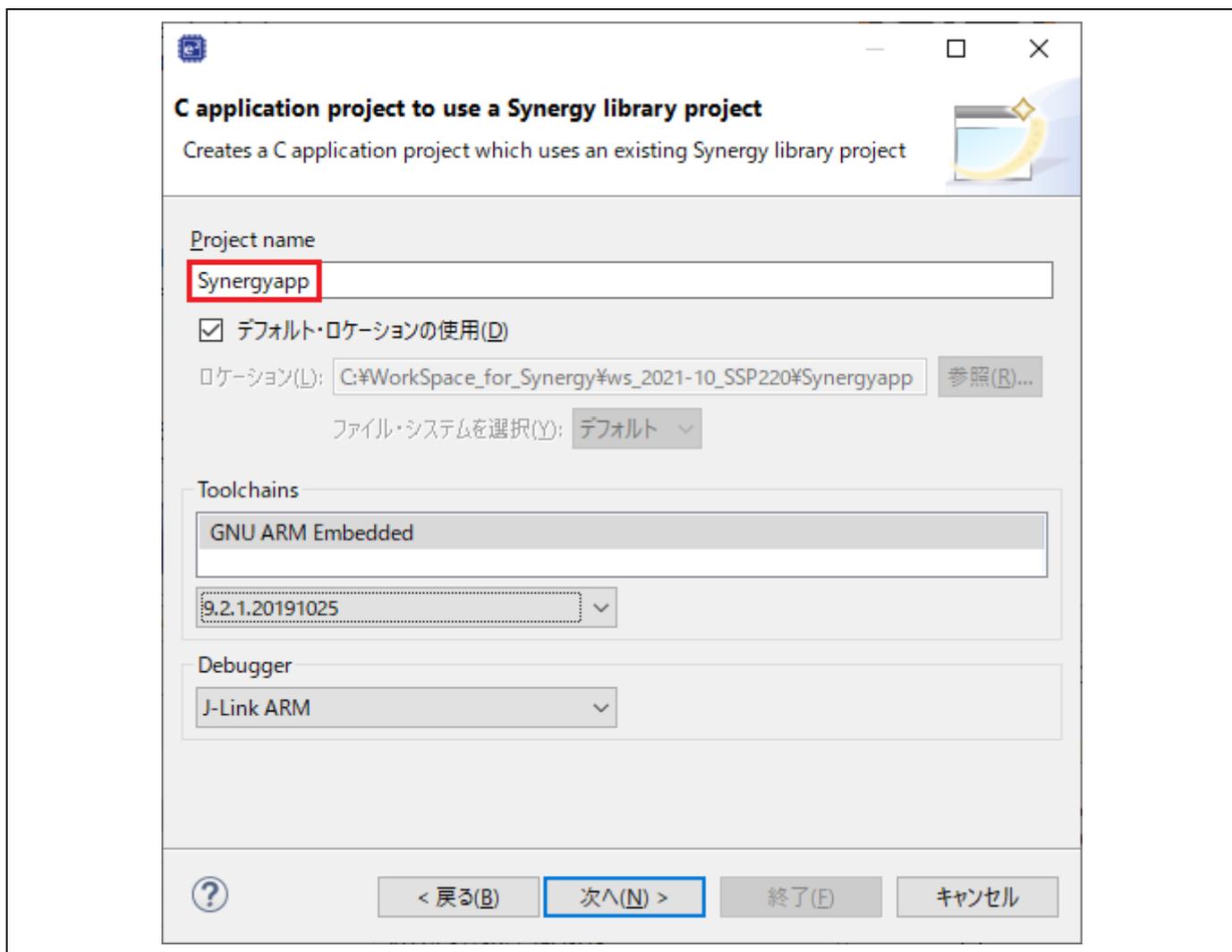


図 3-18 プロジェクト名の指定

(3) ライブラリプロジェクトを選択してください。[終了] ボタンを押すとプロジェクトが作成されます。

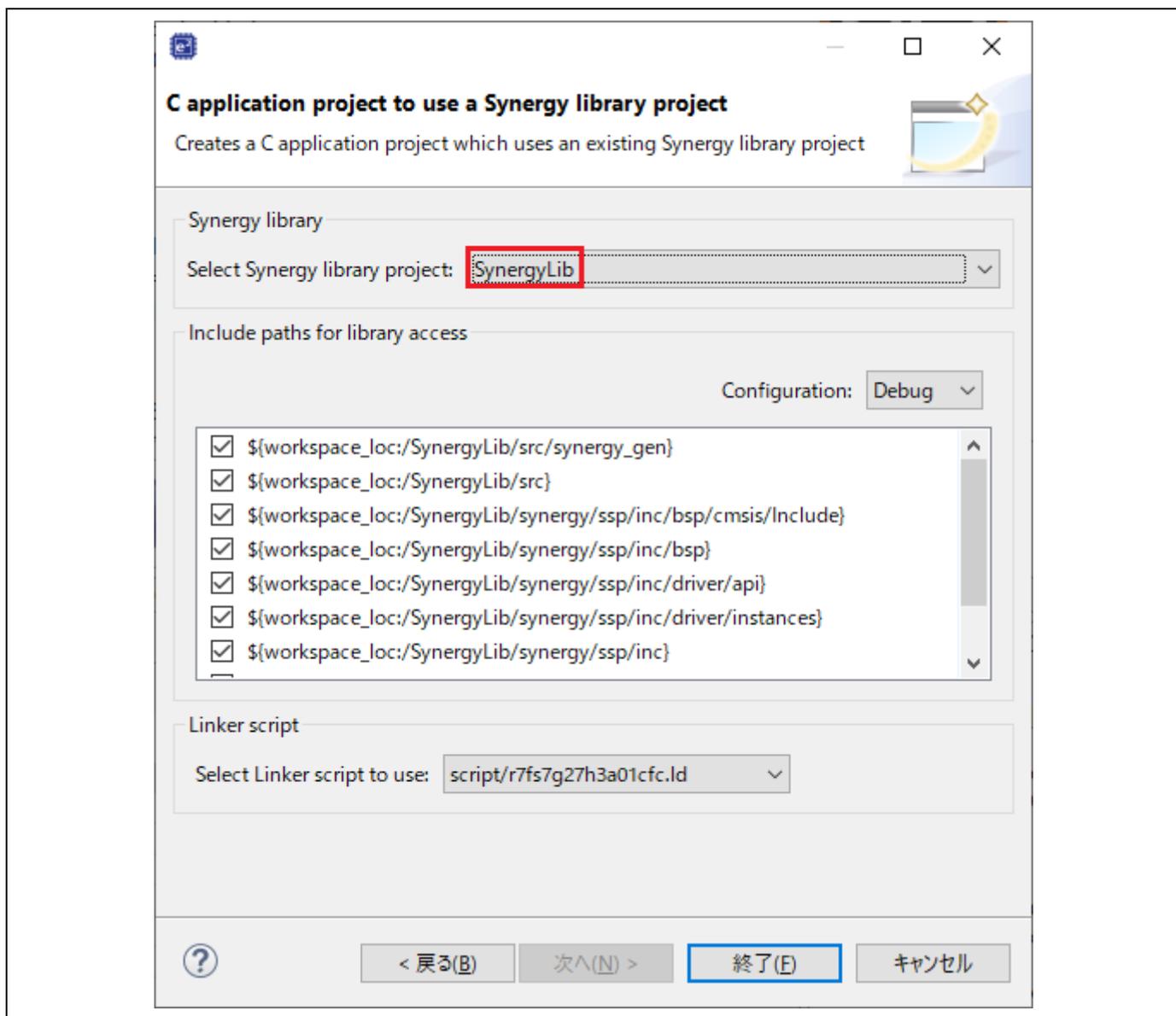


図 3-19 ライブラリプロジェクトの選択

- (4) 実行プロジェクトの作成後、[プロジェクト・エクスプローラー] ビューで、Synergyapp¥src¥ 下の “hal\_entry.c” を開いてください。

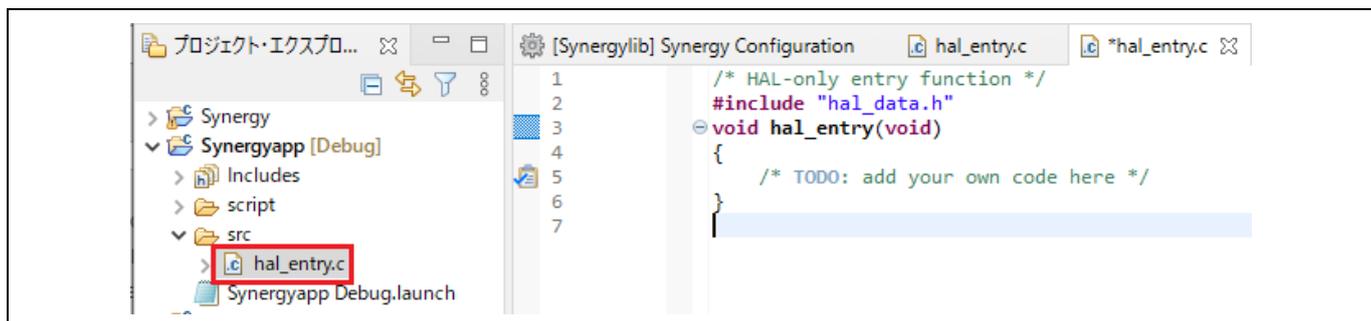


図 3-20 既存の “hal\_entry.c”

hal\_entry() 関数内に LED 点滅用ライブラリ関数 “hal\_entry\_lib()” を呼び出すコードを追加し、ライブラリ関数の宣言を追加します。

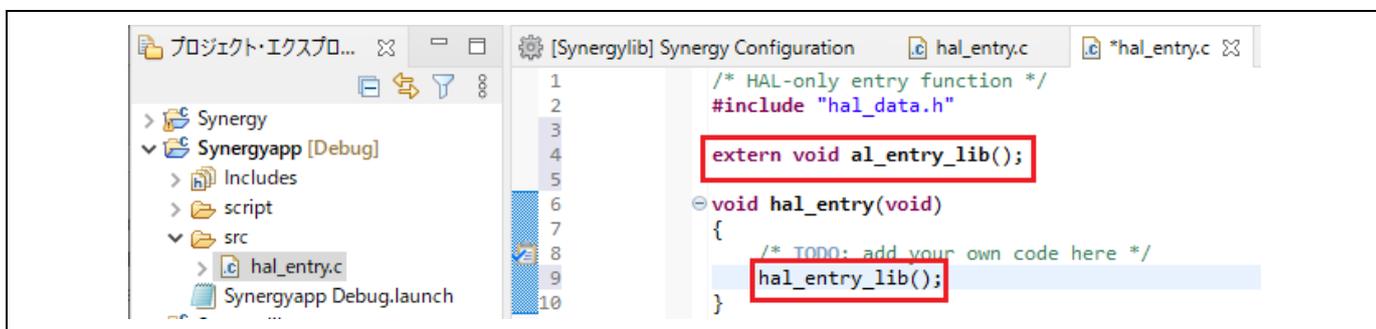


図 3-21 新しい “hal\_entry.c”

- (5) アプリケーションプロジェクトをビルドします。
- (6) 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 Synergyapp プロジェクトを実行します。ブレークポイントの設定方法については、「5.3.1 ブレークポイントビュー」を参照してください。
- (7) 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。

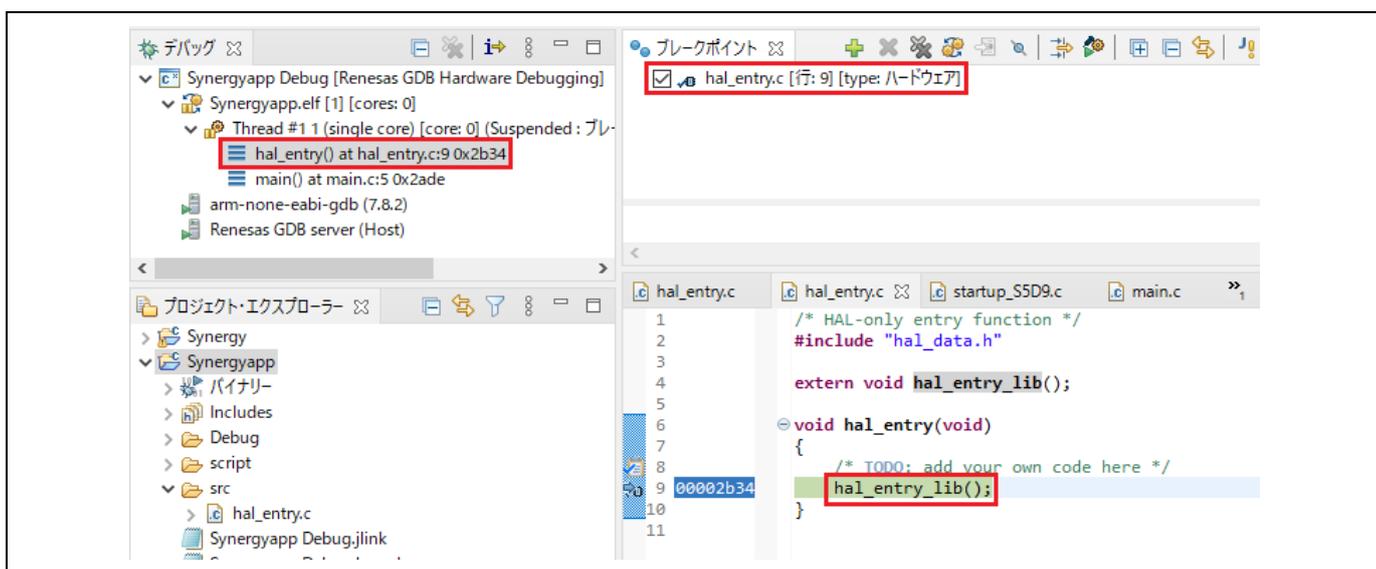


図 3-22 アプリケーションプロジェクトでのライブラリ関数の実行

### 3.4 Synergy プロジェクトコンフィグレーションエディタ

[Synergy プロジェクト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)。ドライバには、ハードウェアレベルのシンプルなものから、RTOS 対応のアプリケーションまでを含みます。マルチスレッド専用のコンポーネント (ミューテックス、セマフォ、イベントなど) も設定できます。

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。

- e<sup>2</sup> studio ウィンドウの右上の表示で [Synergy Configuration] パースペクティブが選択されていることを確認してください。選択されていない場合は、[ウィンドウ] 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] → [Synergy Configuration] の順に選択してください。
- “configuration.xml” ファイルが開いていることを確認してください。

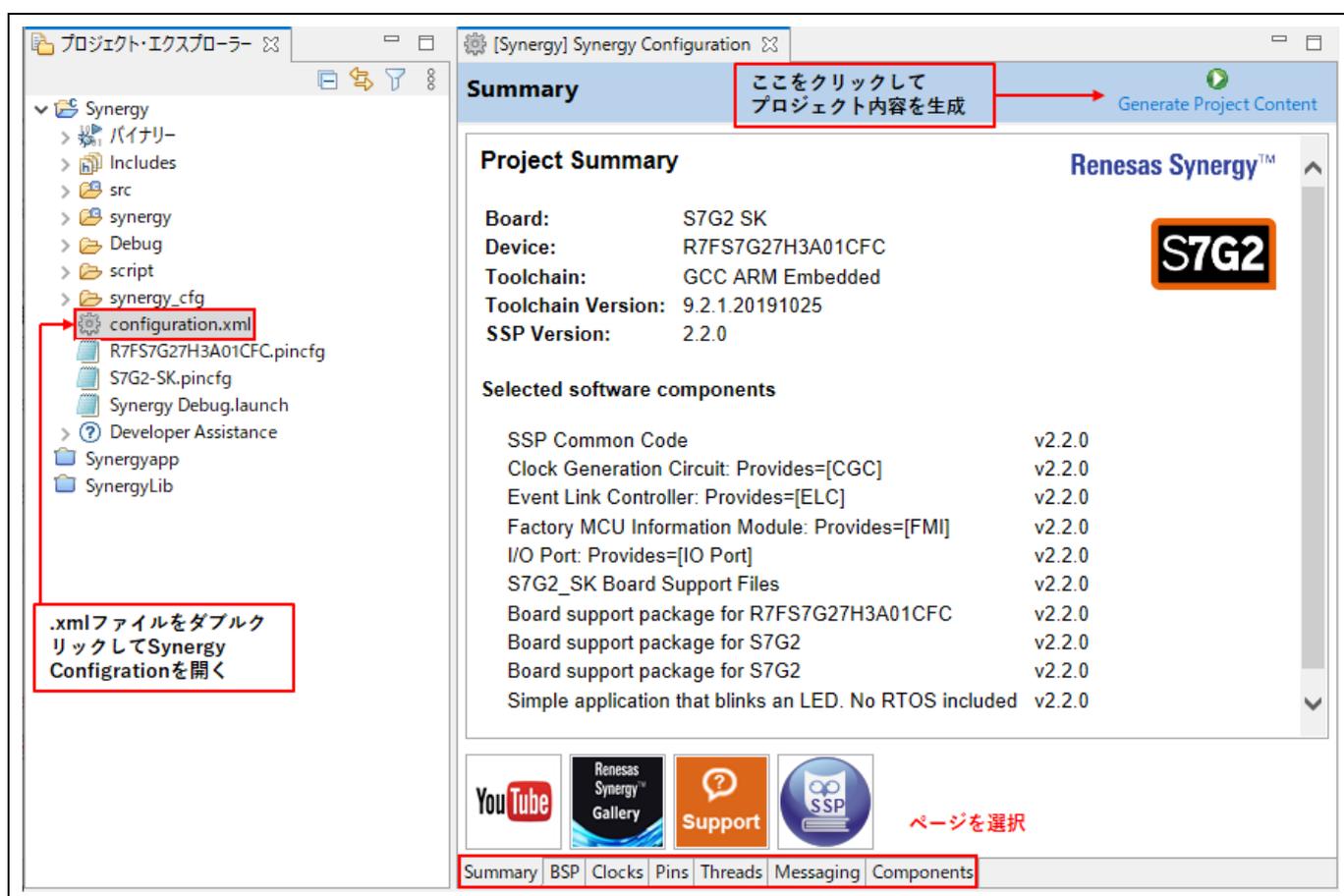


図 3-23 Synergy プロジェクト構成 – [Synergy プロジェクトコンフィグレーションエディタ] ビュー

[Synergy プロジェクトコンフィグレーションエディタ] ビューには 7 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、SSP のバージョン、Synergy 向けのボードの種類、デバイスを選択できます。

クロック (Clocks)、端子 (Pins)、スレッド (Threads)、メッセージング (Messaging)、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。

### 3.4.1 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、Synergy 向けの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、Synergy Platform Web サイト、YouTube の 'Renesas Presents' チャンネル、SSP 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。

新しいスレッドを追加した場合や、スレッドにモジュール/オブジェクトを追加した場合、その情報は [Summary] ページにも追加されます。

**Project Summary**

Board:	S7G2 SK
Device:	R7FS7G27H3A01CFC
Toolchain:	GCC ARM Embedded
Toolchain Version:	7.2.1.20170904
SSP Version:	2.2.0

ボード、デバイス、ツールチェーン、SSPに関する情報

**Threads Summary**

Property	Value
Priority	GUI Thread
	10

スレッドに関する情報

**Interrupts Summary**

Property	Value
ELC Software Event Interrupt Priority	g_transfer0 Transfer Driver on r_dtc SPI0 TXI
ELC Software Event Interrupt Priority	g_transfer1 Transfer Driver on r_dtc SPI0 RXI
Underflow 2 Interrupt Priority	g_display0 Display Driver on r_glcd
Transmit End Interrupt Priority	g_rspi_lcdc SPI Driver on r_rspi
Error Interrupt Priority	g_rspi_lcdc SPI Driver on r_rspi
Receive Interrupt Priority	g_rspi_lcdc SPI Driver on r_rspi
Transmit Interrupt Priority	g_rspi_lcdc SPI Driver on r_rspi
Data Transfer Interrupt Priority	g_jpeg_decode0 JPEG Decode Driver on r_jpeg_decode
Decompression Interrupt Priority	g_jpeg_decode0 JPEG Decode Driver on r_jpeg_decode
Line Detect Interrupt Priority	g_display0 Display Driver on r_glcd
Underflow 1 Interrupt Priority	g_display0 Display Driver on r_glcd

割込みにに関する情報

**Selected software components**

SSP Common Code	v2.2.0
SF_EL_GX GUIX Adaptation Framework: Provides=[SSP GUIX Adaptation Framework] , Requires=[ThreadX , GUIX]	v2.2.0
Framework JPEG Decode: Provides=[SF JPEG Decode] , Requires=[ThreadX ,JPEG Decode]	v2.2.0
TES D/AVE 2D Port: Provides=[D/AVE 2D Port] , Requires=[ThreadX ,D/AVE 2D]	v2.2.0
TES D/AVE 2D: Provides=[D/AVE 2D]	v2.2.0
Clock Generation Circuit: Provides=[CGC]	v2.2.0
Data Transfer Controller: Provides=[Transfer]	v2.2.0
Event Link Controller: Provides=[ELC]	v2.2.0
Factory MCU Information Module: Provides=[FMI]	v2.2.0
Graphics LCD: Provides=[Display]	v2.2.0
I/O Port: Provides=[IO Port]	v2.2.0
IPEG Common	v2.2.0

プロジェクトに含まれているソフトウェアコンポーネント

便利なサイトへのリンク

Summary | BSP | Clocks | Pins | Threads | Messaging | Components

図 3-24 [Summary] ページ

### 3.4.2 BSP ページ

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

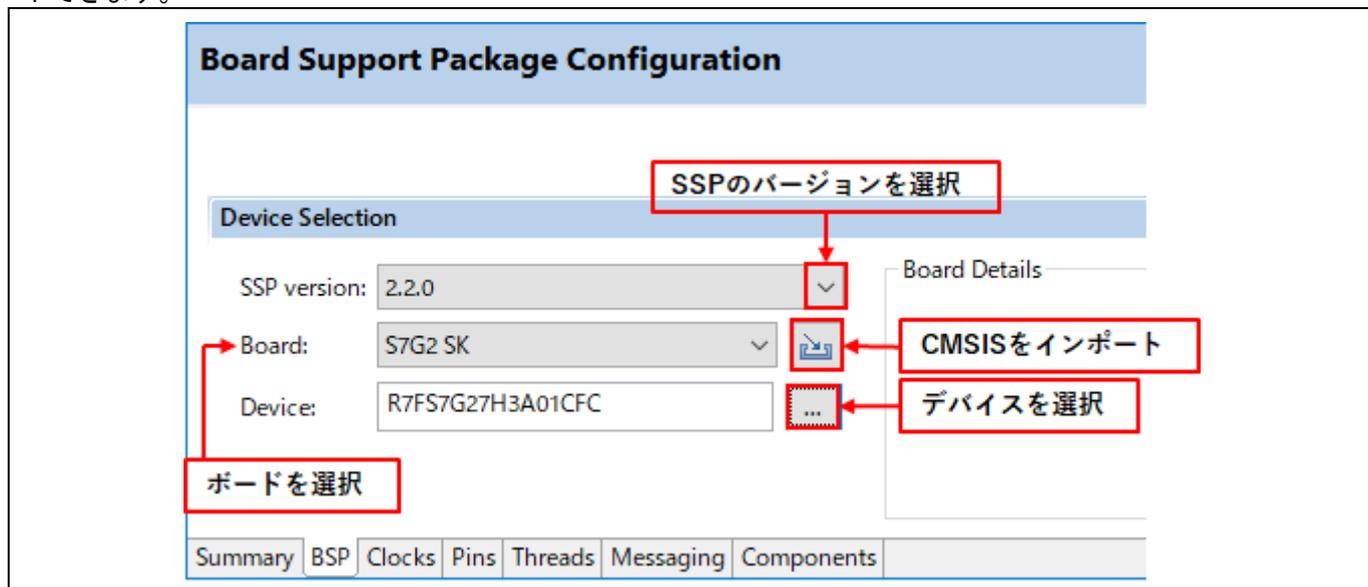


図 3-25 Synergy プロジェクト構成 - [BSP] ページ

### 3.4.3 クロック構成 (Clocks Configuration) ページ

[Clocks Configuration] 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、Synergy 製品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。

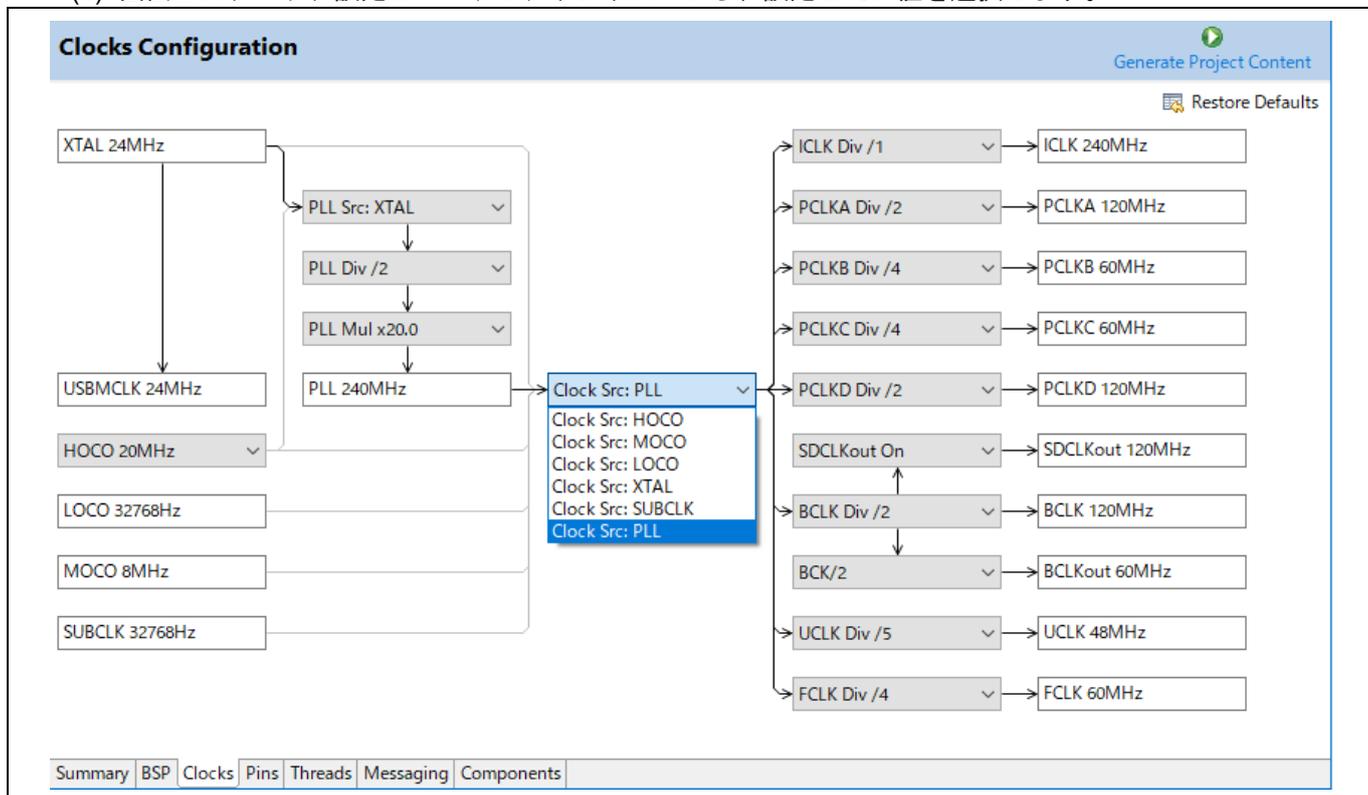


図 3-26 Synergy プロジェクト構成 – [Clocks Configuration] ページ

- (2) [Ctrl+S] キーなどで、プロジェクト構成の設定を保存します。

- (3) [Generate Project Content] Generate Project Content ボタンを押します。

- (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。

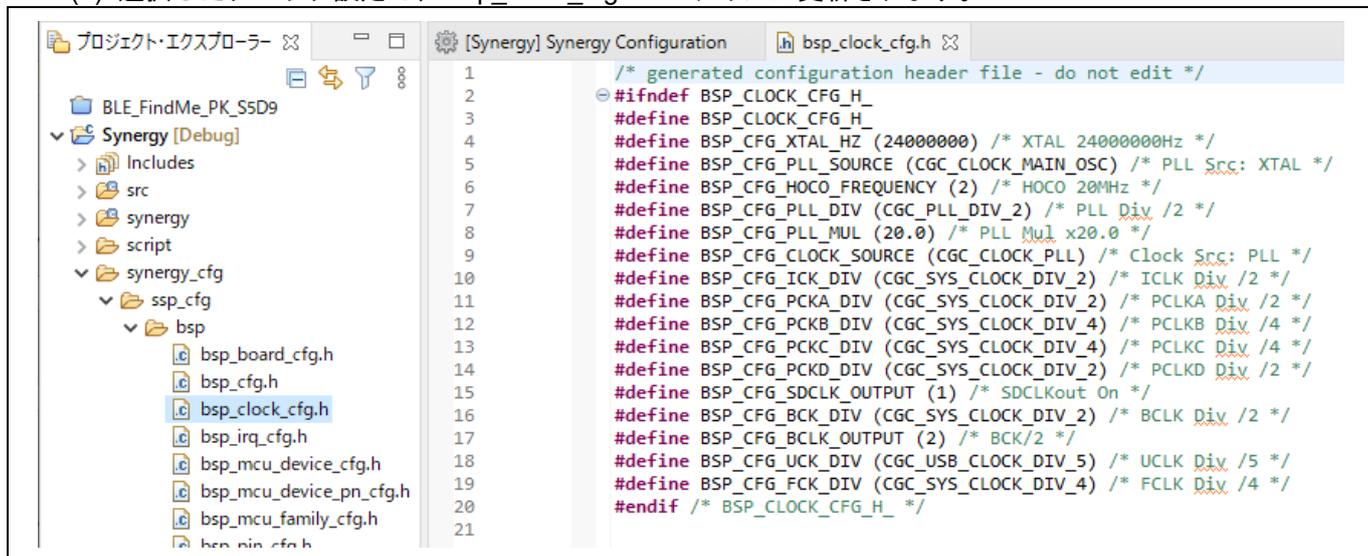


図 3-27 “bsp\_clock\_cfg.h” の更新

### 3.4.4 端子構成 (Pins Configuration) ページ

[Pins Configuration] ページでは、プロジェクトの端子構成を生成します。

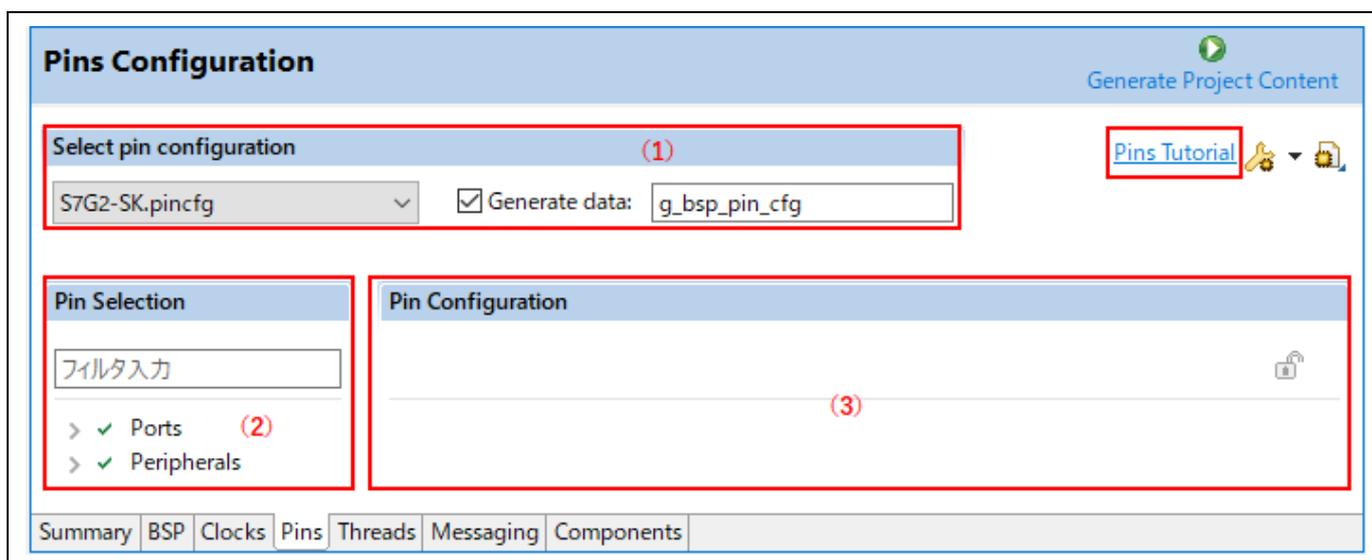


図 3-28 Synergy プロジェクト構成 – [Pins Configuration] ページ

[Pins Configuration] 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。
  - a. 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。
  - b. 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。
  - c. [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。
  - d. 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。

[Pins Tutorial] をクリックして、端子構成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。

端子構成を設定するには、プロジェクトで使用する周辺機能ごとに設定するのが最もわかりやすい方法です。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] → [SCI2]）。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。

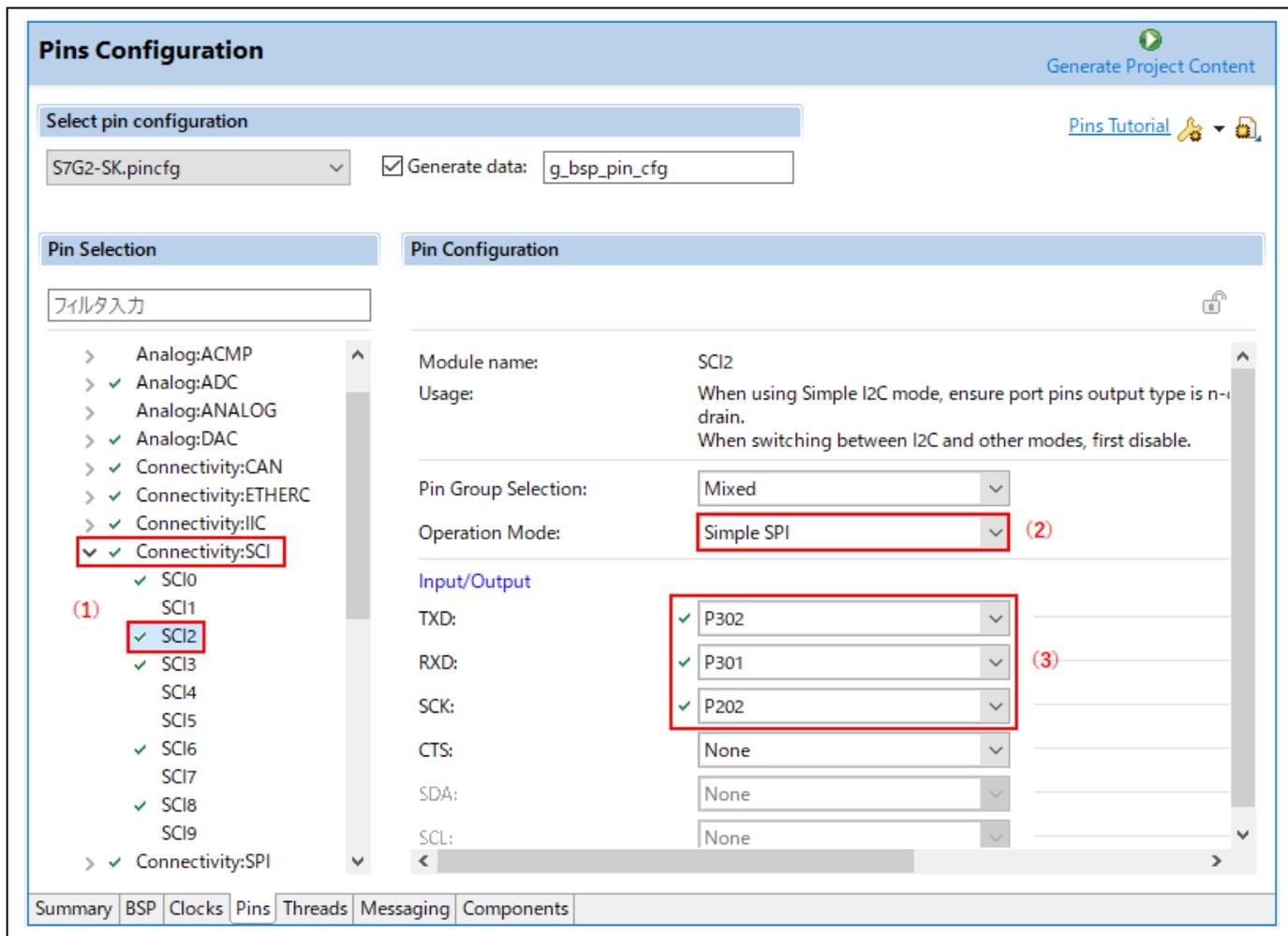


図 3-29 Synergy プロジェクト構成 — 端子構成の設定（周辺機能ごとに設定）

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。

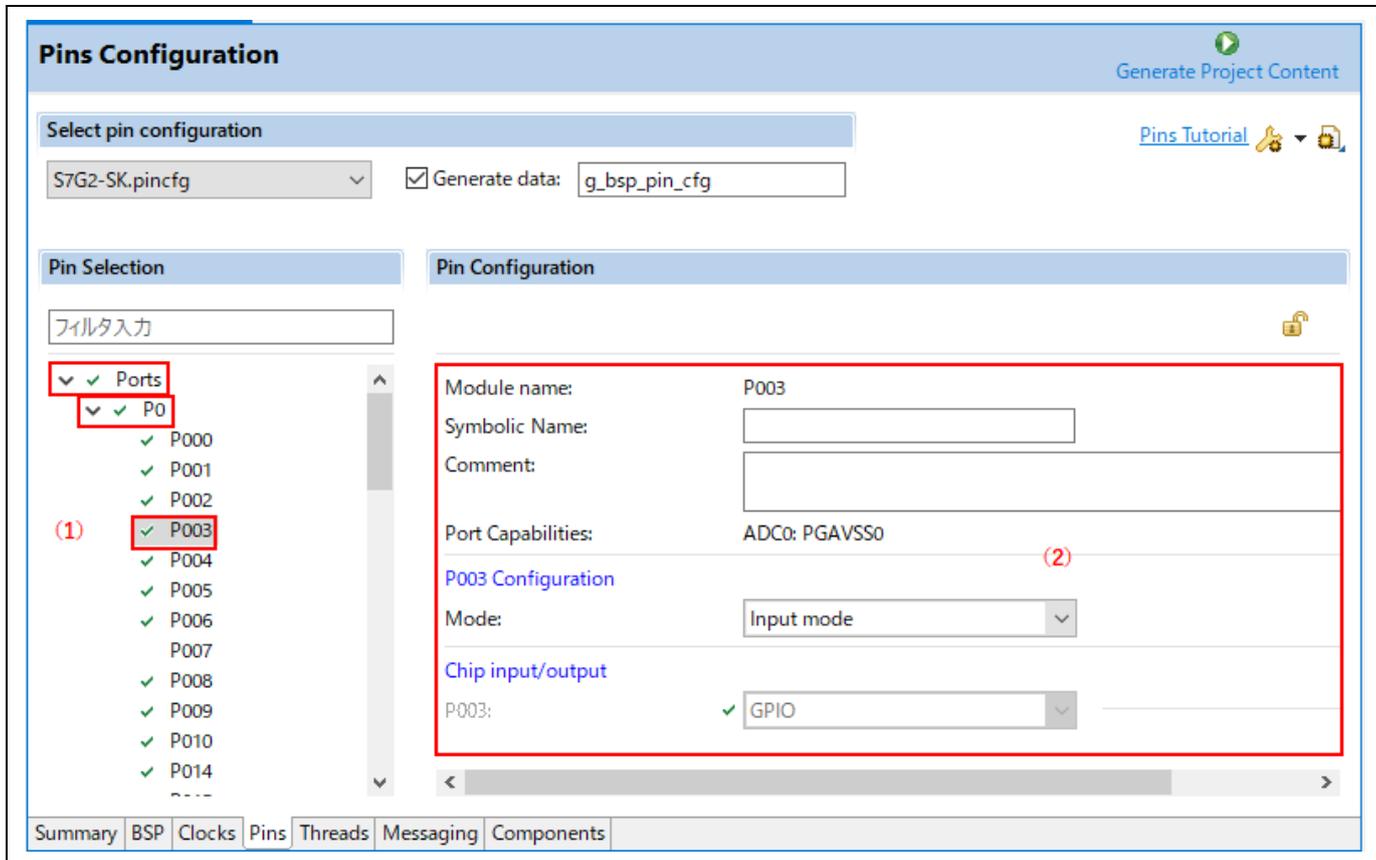


図 3-30 Synergy プロジェクト構成 - 端子構成の設定（端子ごとに設定）

- (3) 変更した端子設定は [Package] ビューに表示されます。

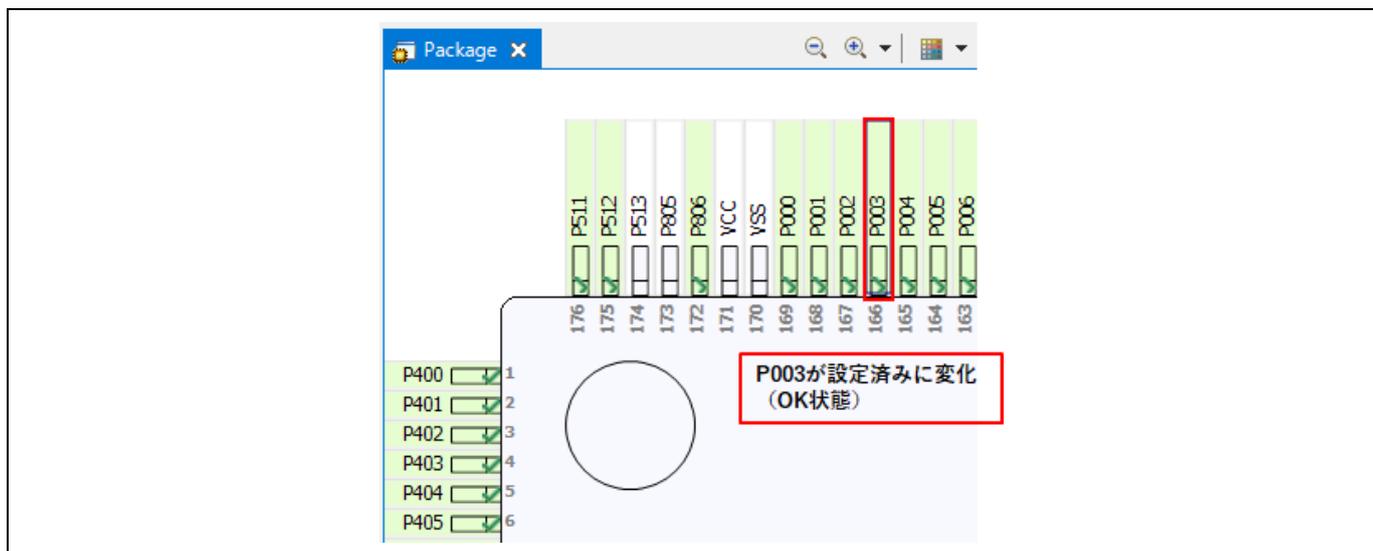


図 3-31 Synergy プロジェクト構成 - [Package] ビュー（接続状態）

[Pins Configuration] ページでは、端子構成を別のデバイスから移行することもできます。移行にはツールバー上の [Import a pin configuration] ボタンを使います。この機能は、ユーザが設定した既存の端子構成を新たなデバイスに移行できます。

既存の端子構成を現在のプロジェクトにインポートするには、[Import a pin configuration]  ボタンを押し、インポートする端子構成ファイルを選択します。

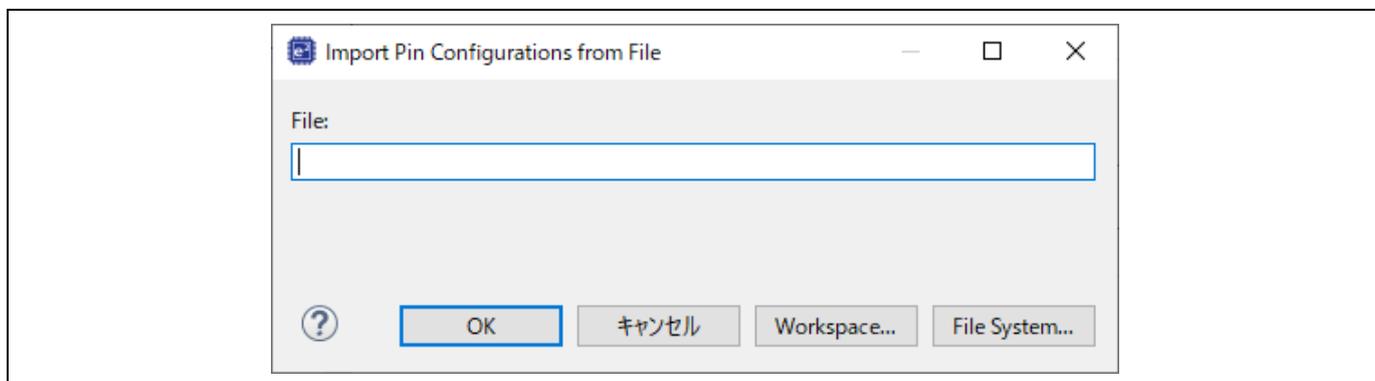


図 3-32 既存の端子構成を現在のプロジェクトにインポート

競合が発生する場合はメッセージが表示されますので、次のいずれかを選択します。

1. インポート動作をキャンセルする
2. 競合する設定をそのままインポートする
3. 競合する設定をインポートせずに、インポート操作を継続する

**注：**端子情報として、プロジェクトフォルダ “¥synergy\_cfg¥ssp\_cfg¥bsp..” に “.csv” ファイルが生成されます。  
これ以外の情報は出力されません。

### 3.4.5 スレッド構成 (Threads Configuration) ページ

[Threads Configuration] ページでは次の操作が可能です。

- Synergy プロジェクト内でスレッドを設定する
- スレッドに Synergy 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

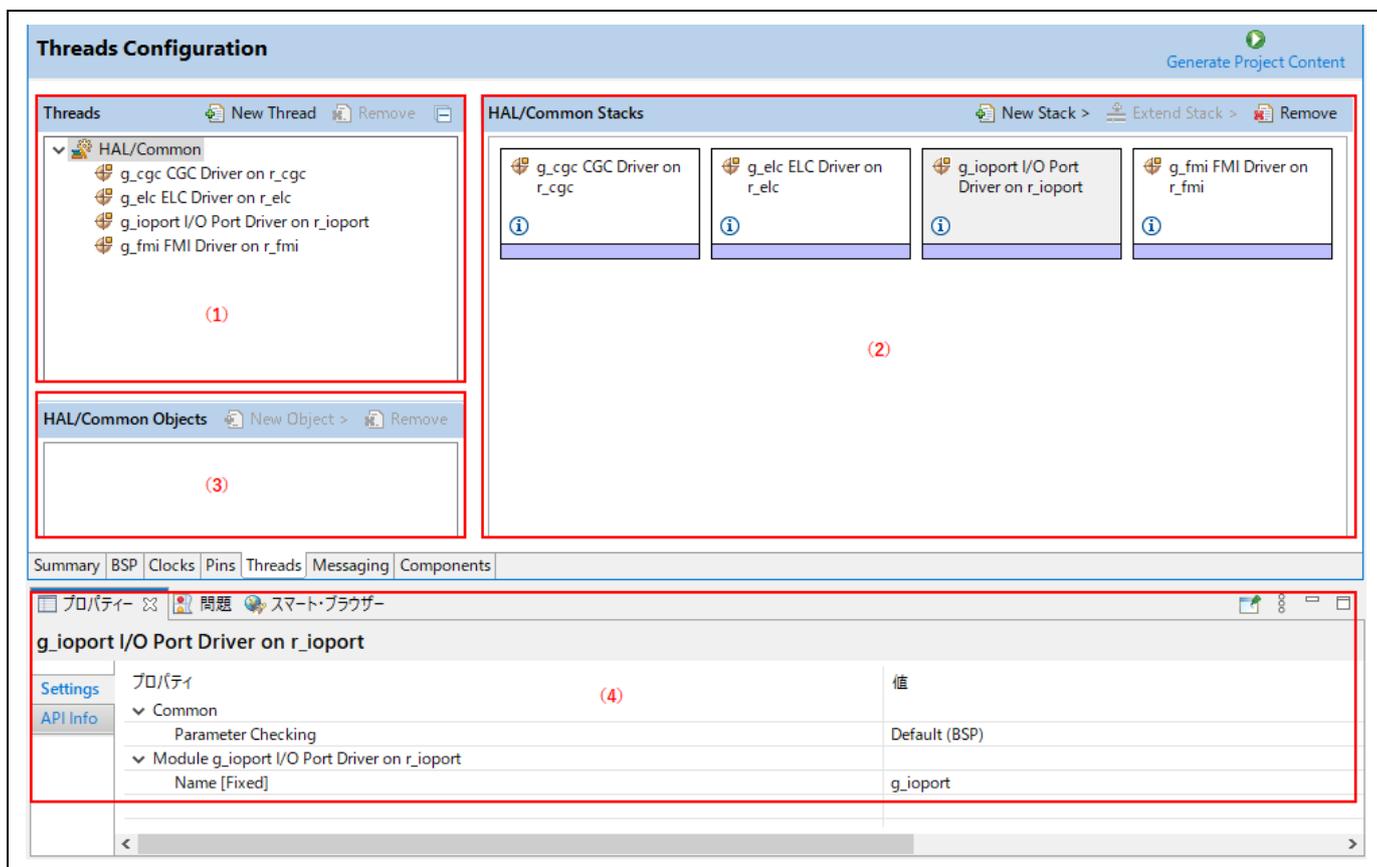


図 3-33 Synergy プロジェクト構成 – [Threads Configuration] ページ

[Threads Configuration] 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。詳細は「6 章 ThreadX アプリケーションの設定」で説明します。
2. [Stacks] ペイン：SSP 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。詳細は「6 章 ThreadX アプリケーションの設定」で説明します。

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] → [Monitoring] → [Clock Accuracy Circuit Driver on r\_cac] の順に選択)。
- (3) [Generate Project Content] ボタンを押してソースコードを生成します。
- (4) 選択したモジュールのプロパティが [プロパティ] ビューに表示されます。必要に応じてプロパティを変更できます。

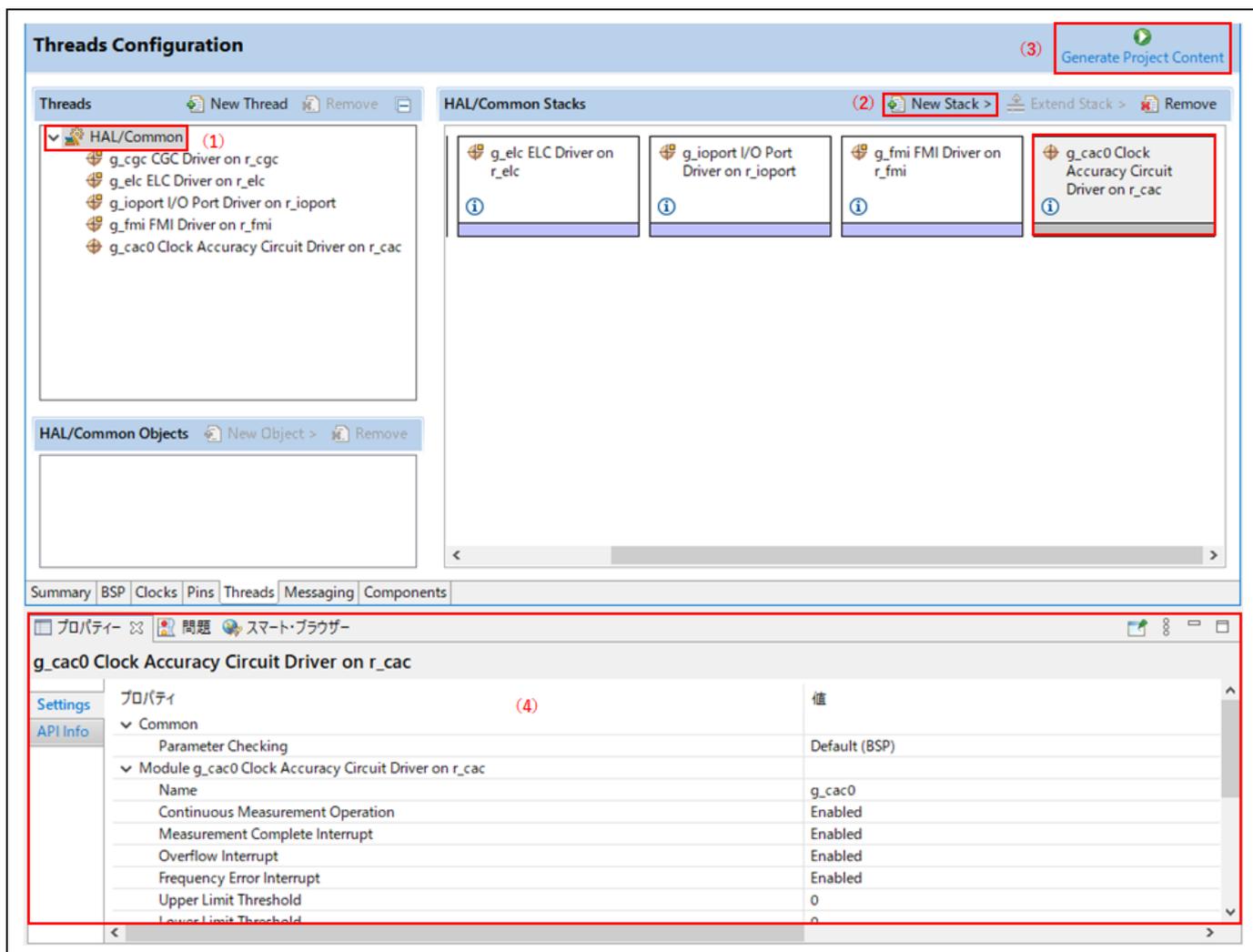


図 3-34 Synergy プロジェクト構成 – スレッドにモジュールを追加

注: HAL/Common は実際はスレッドではありませんが、既存プロジェクトへのモジュール追加手順を説明するために、この例ではスレッドとして用いています。別の操作例として、「6.1 ThreadX での汎用 PWM タイマ使用例」では汎用 PWM タイマ (GPT) モジュールを “Blinky スレッド” に追加する手順を説明しています。

モジュールを追加すると（例：UART Driver on r\_sci\_uart）、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。任意で追加したいものはユーザーが手動で追加します。追加可能なモジュールは別に表示されますので、追加したいものをクリックして選択し、プロパティを設定してください。

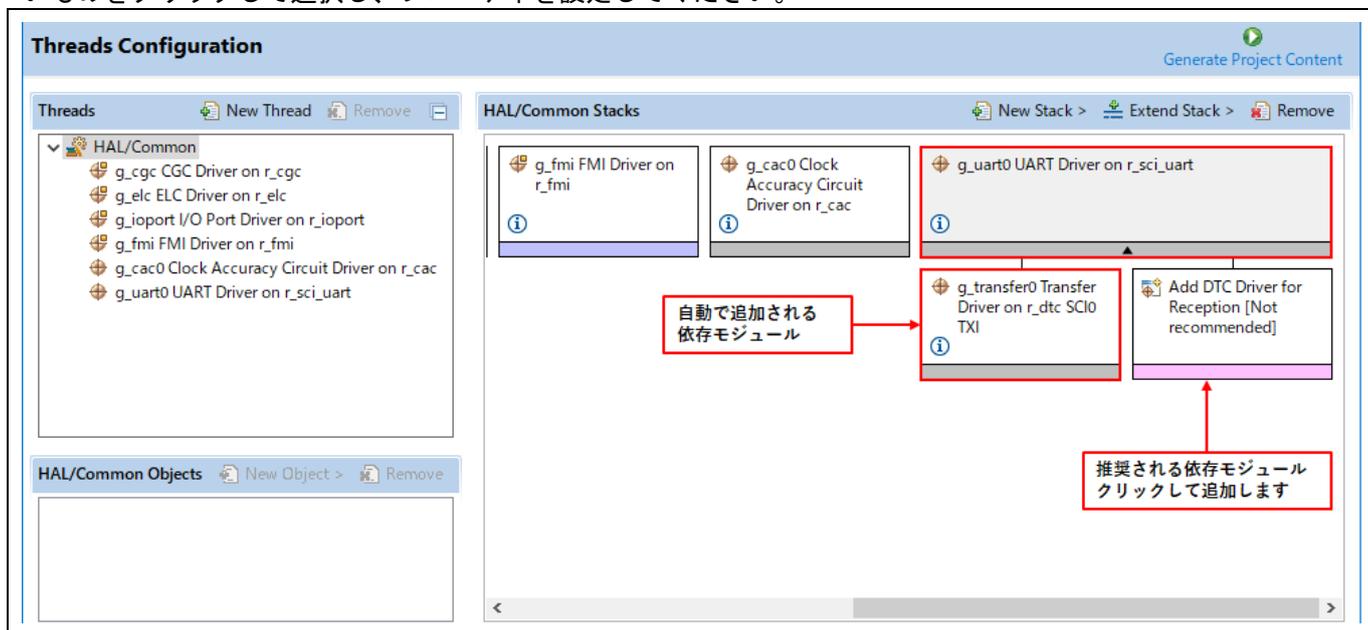


図 3-35 Synergy プロジェクト構成 – 依存関係のあるモジュール

[Threads Configuration] 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。

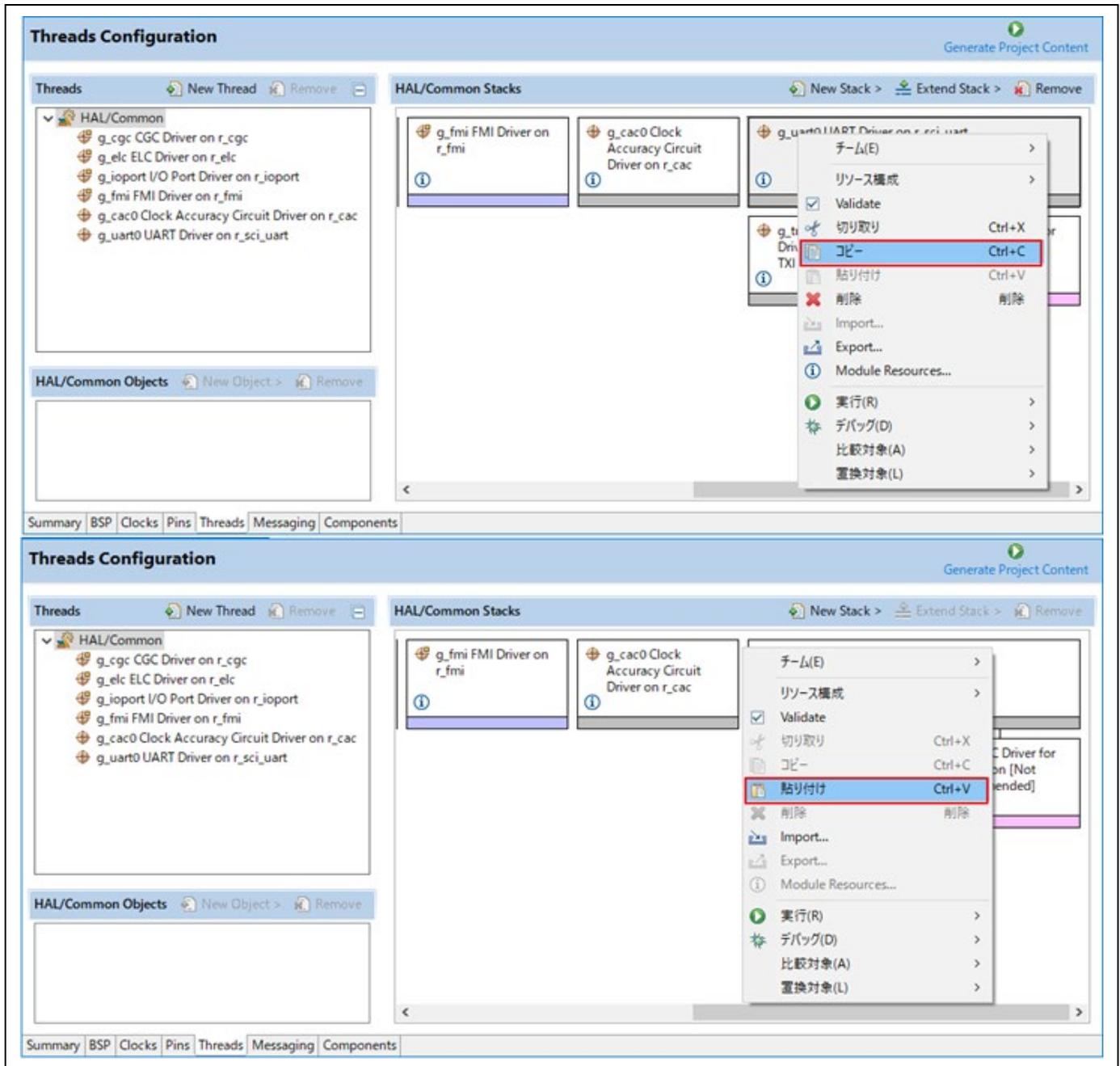


図 3-36 コピー&ペーストの操作

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。

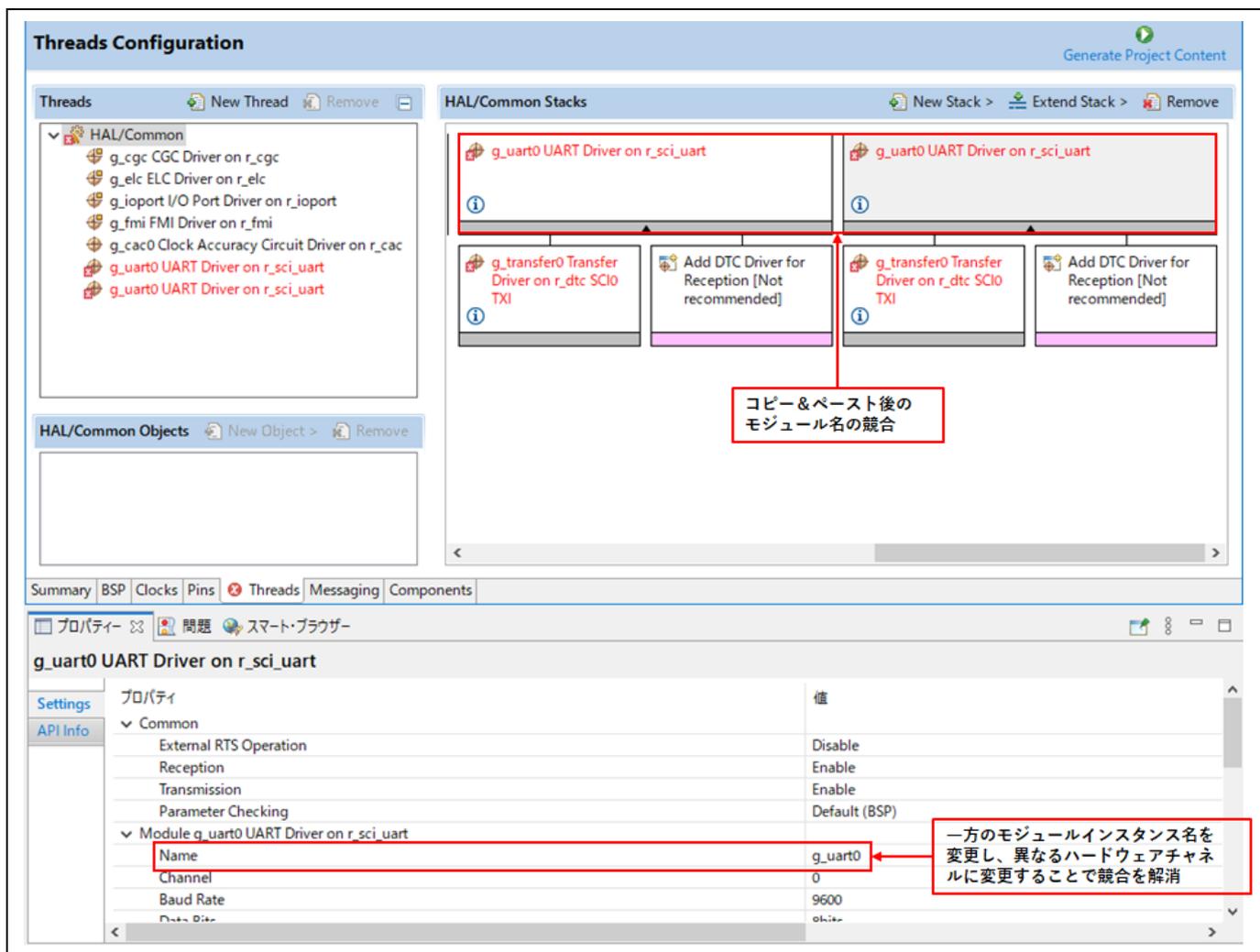


図 3-37 モジュールインスタンス名の競合

[Threads Configuration] 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。モジュール名の競合は、一方のモジュールインスタンス名を変更して解消します。

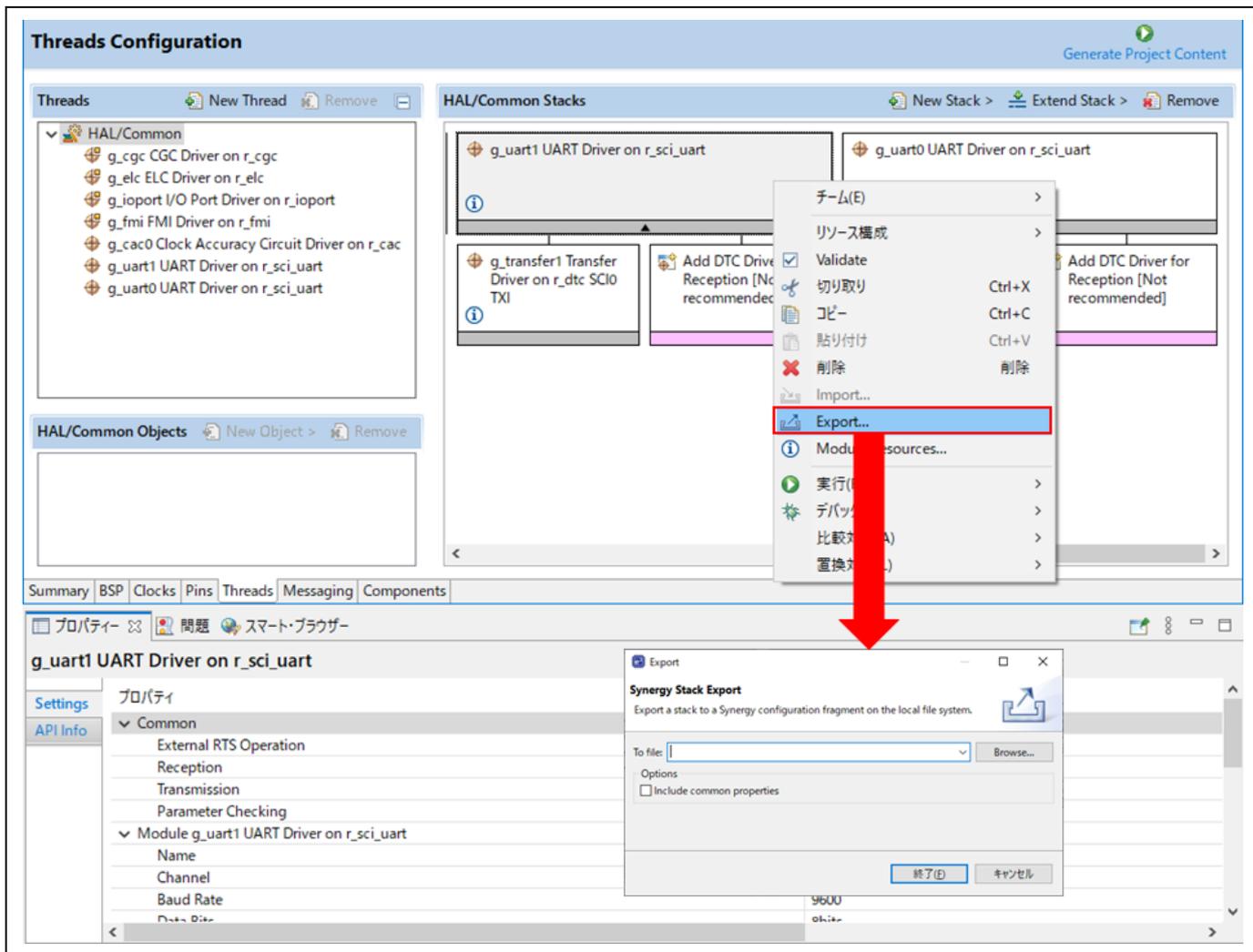


図 3-38 Synergy スタックのエクスポート

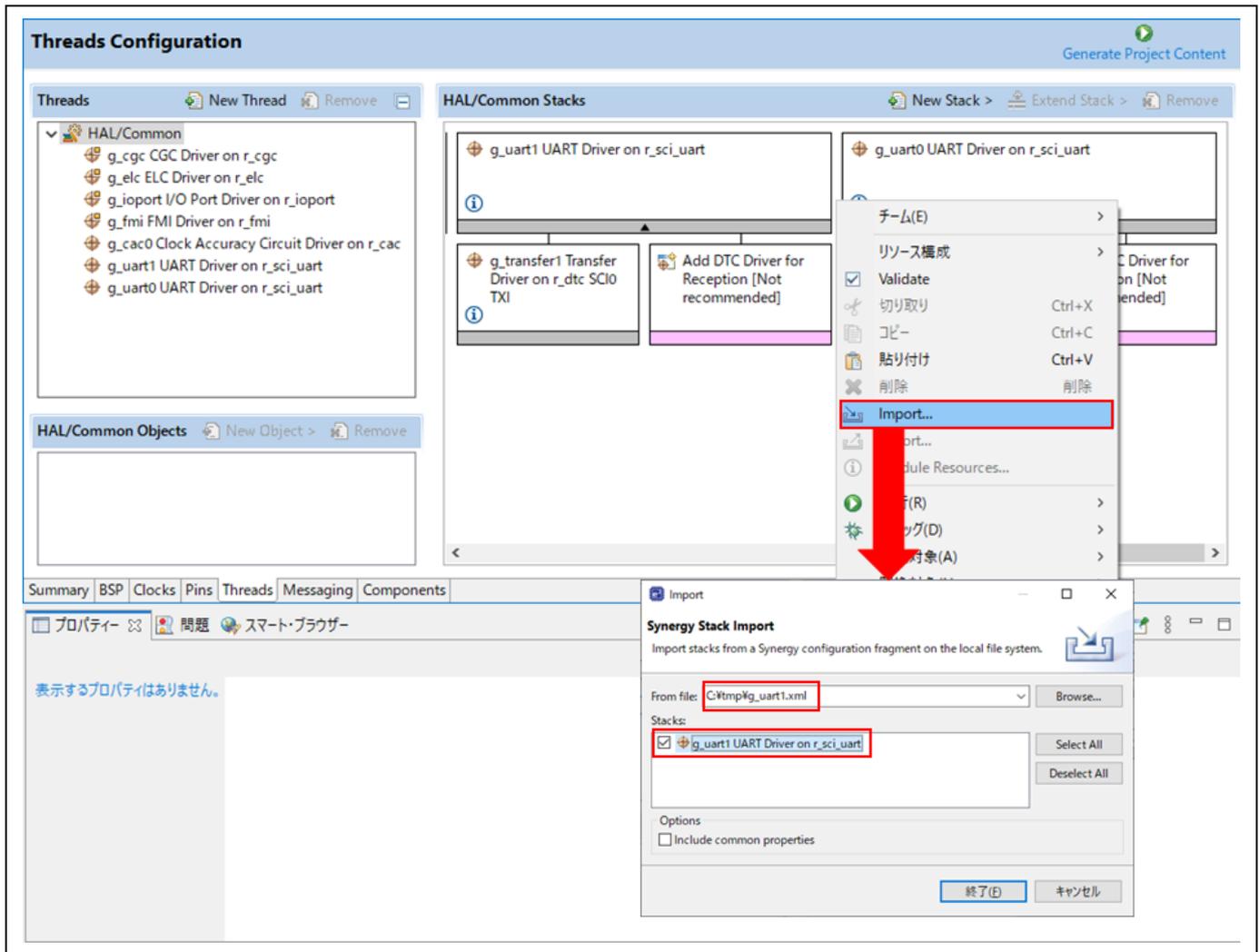


図 3-39 Synergy スタックのインポート

### 3.4.6 メッセージング (Messaging) ページ

[Messaging] ページでは、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使用することができます。

[Messaging] ページは 3 つのペインで構成されています。

1. [Event Classes] ペイン：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] ペイン：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] ペイン：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。

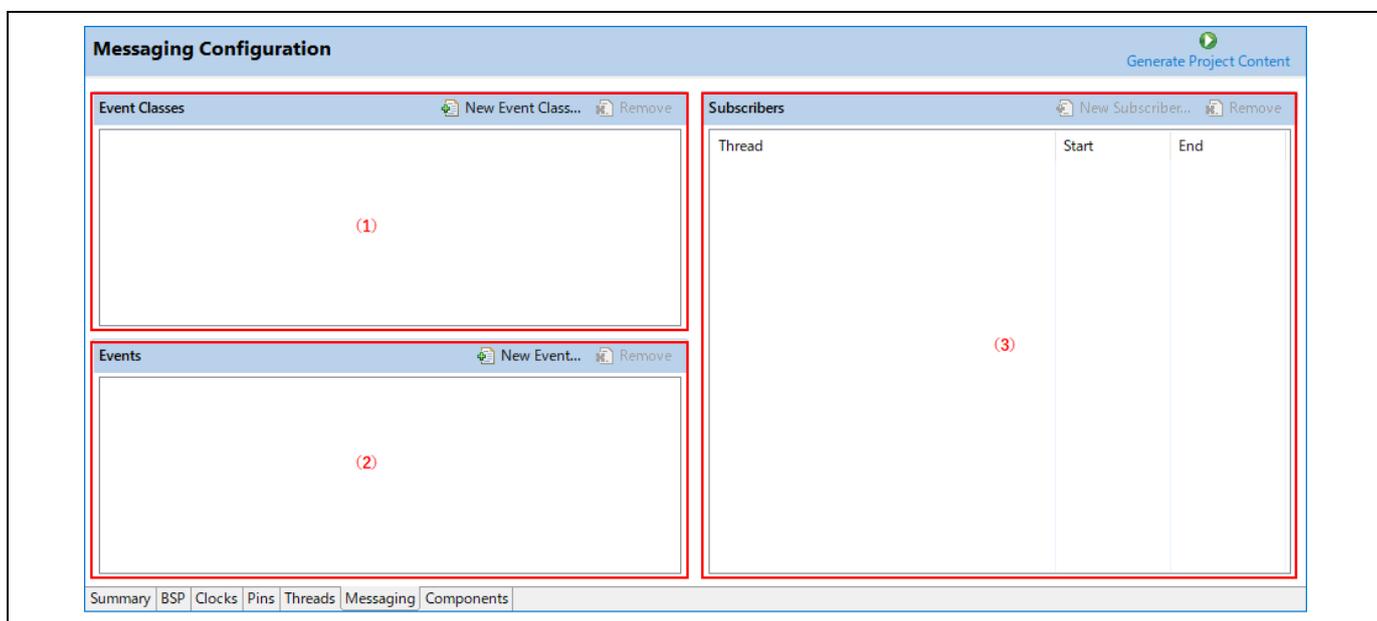


図 3-40 Synergy プロジェクト構成 – [Messaging] ページ

各ペインで [New Event Class...], [New Event...], または [New Subscriber...] 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。

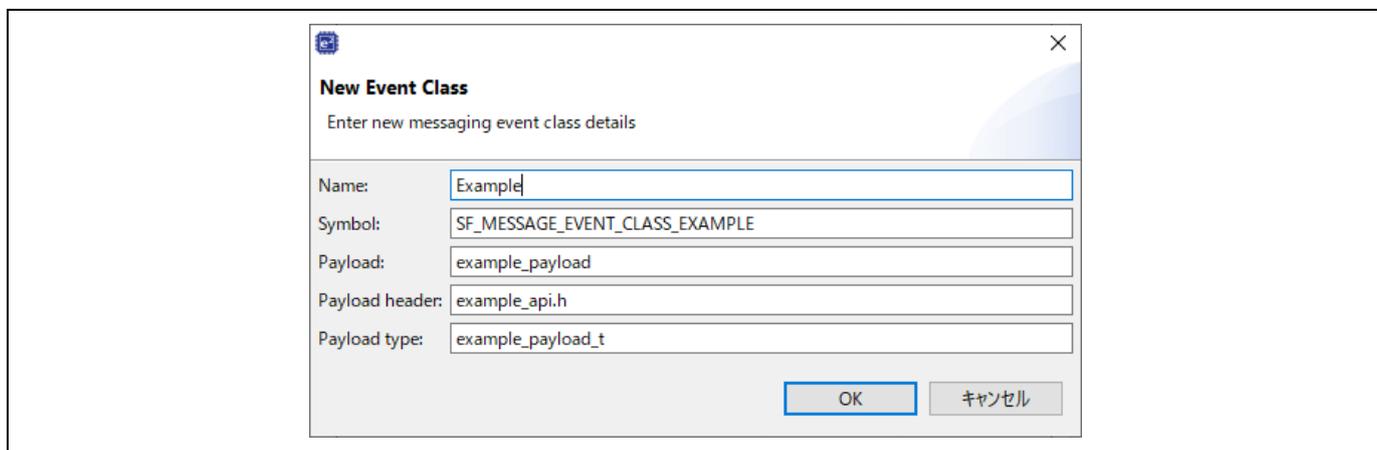


図 3-41 [Messaging] ページ – 新規イベントクラスの追加

手動で作成した項目を削除するには、該当するペインでその項目を選択し、 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）。

いずれかの項目を選択すると、e<sup>2</sup> studio の [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するプロパティが表示されます。

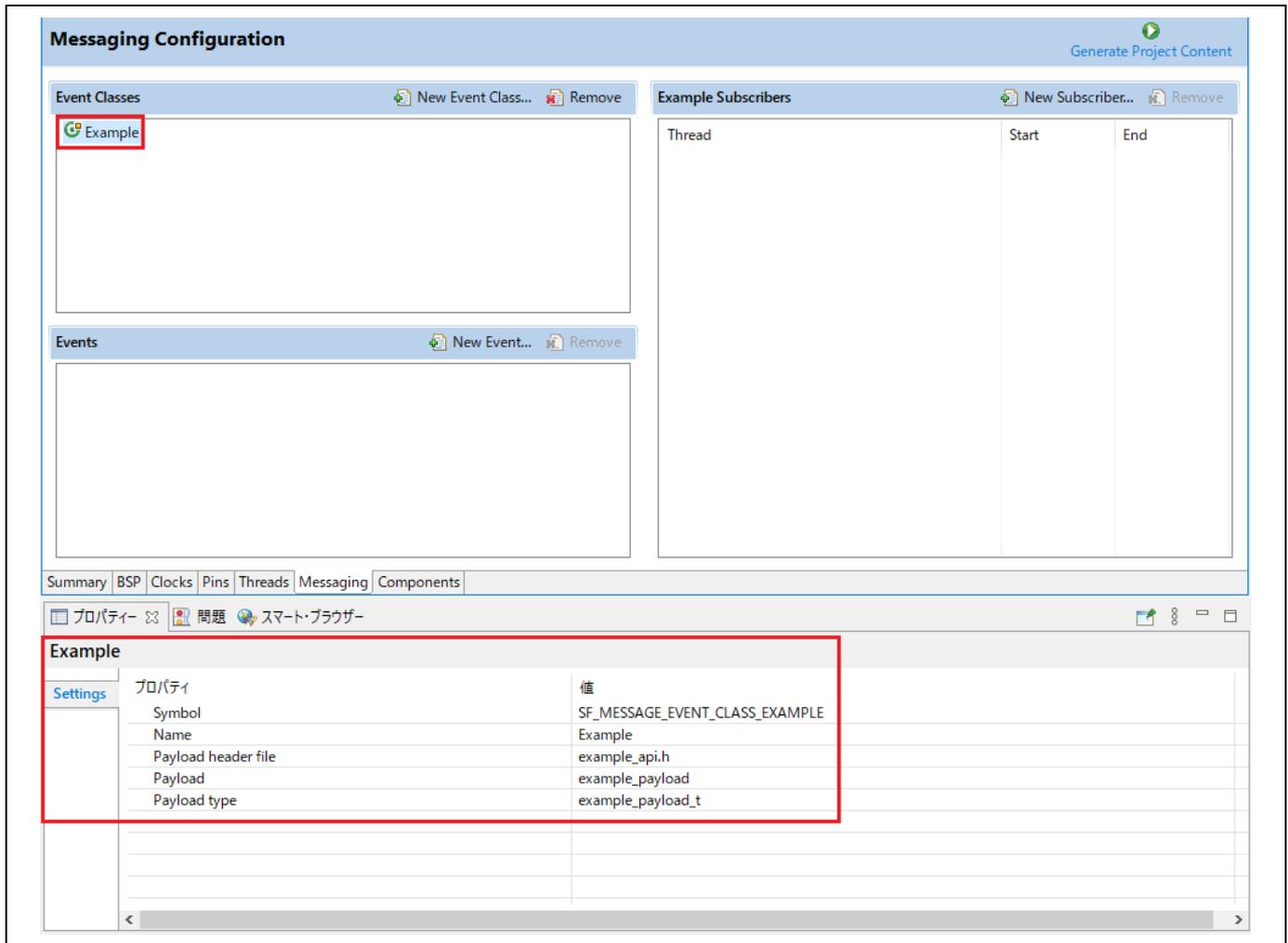


図 3-42 メッセージングのプロパティ

### 3.4.7 コンポーネント構成 (Components Configuration) ページ

[Components Configuration] ページでは、アプリケーションに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 Synergy 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 → all → r\_cgc)。

[Threads] 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 [Threads Configuration] ページを用います。[Components Configuration] ページは、主に、インストール済み SSP 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。

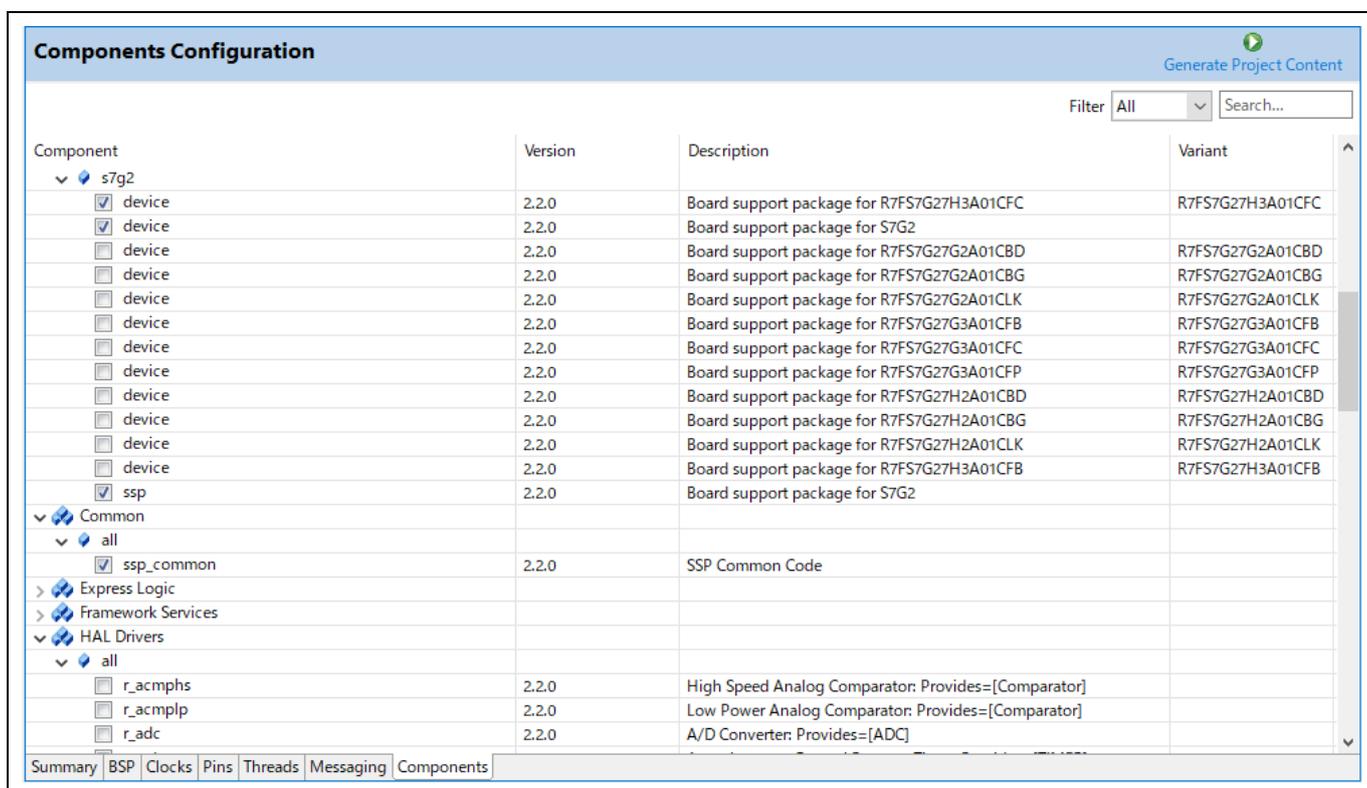


図 3-43 Synergy プロジェクト構成 - [Components Configuration] ページ

### 3.5 エディタ画面の吹き出し機能

e<sup>2</sup> studio はテキストエディタ画面での吹き出し機能をサポートしています。[ウィンドウ] メニュー → [設定] → [C/C++] → [エディター] → [吹き出し] の順に選択して、吹き出し機能を有効または無効にします。

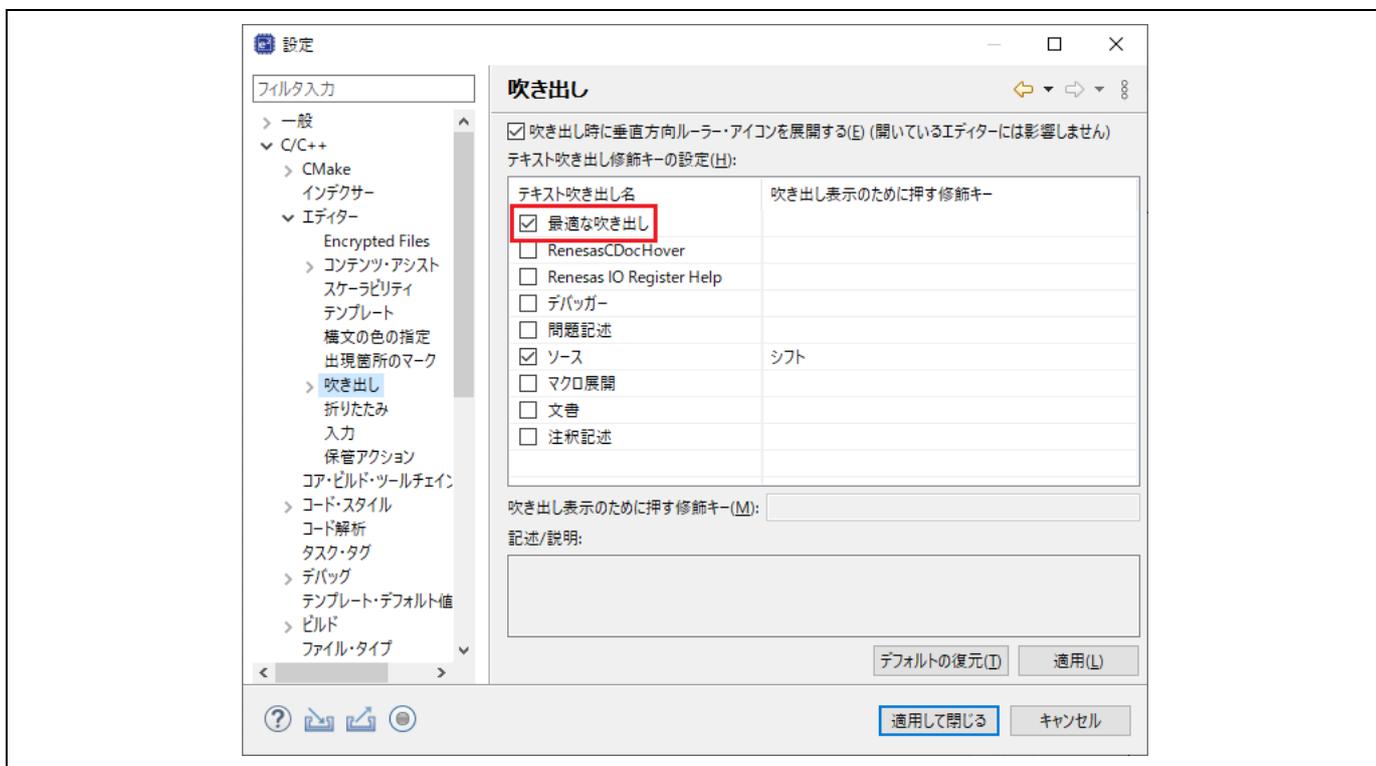


図 3-44 吹き出しの設定

吹き出しを有効にするには、“最適な吹き出し”を選択します。無効にするには、チェックをはずしてください。

吹き出しはデフォルトで有効に設定されています。

吹き出し機能により、ソースコード中の各識別子の詳細情報を見ることができます。識別子の上にマウスポインタを置くとポップアップ表示が出ます。

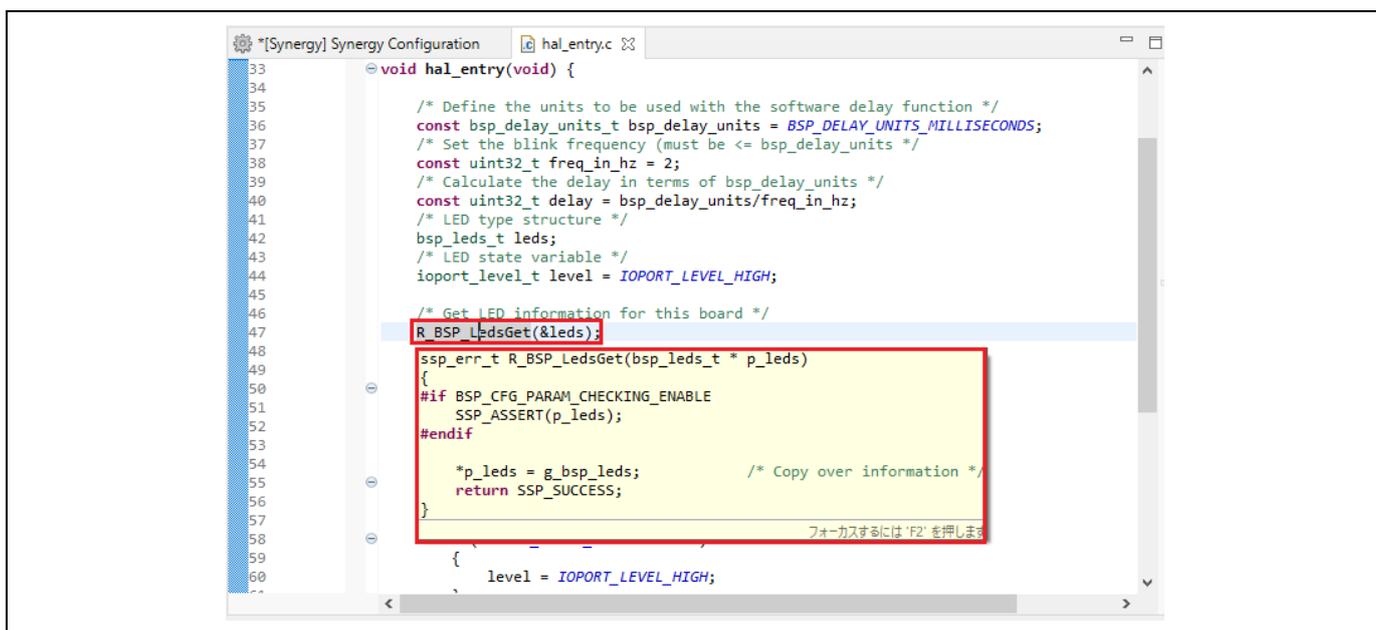


図 3-45 吹き出し機能による情報の表示

### 3.6 開発者支援機能 (Developer Assistance)

Synergy Developer Assistance により、e<sup>2</sup> 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 コンフィグレーションエディタで Synergy プロジェクトのスレッドとソフトウェアスタックを設定した後、すぐに設定済みのスタックモジュールを使用してプロジェクトの C/C++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 が支援を行います。

(1) [プロジェクト・エクスプローラー] ビューを展開し、[Developer Assistance] を表示します。

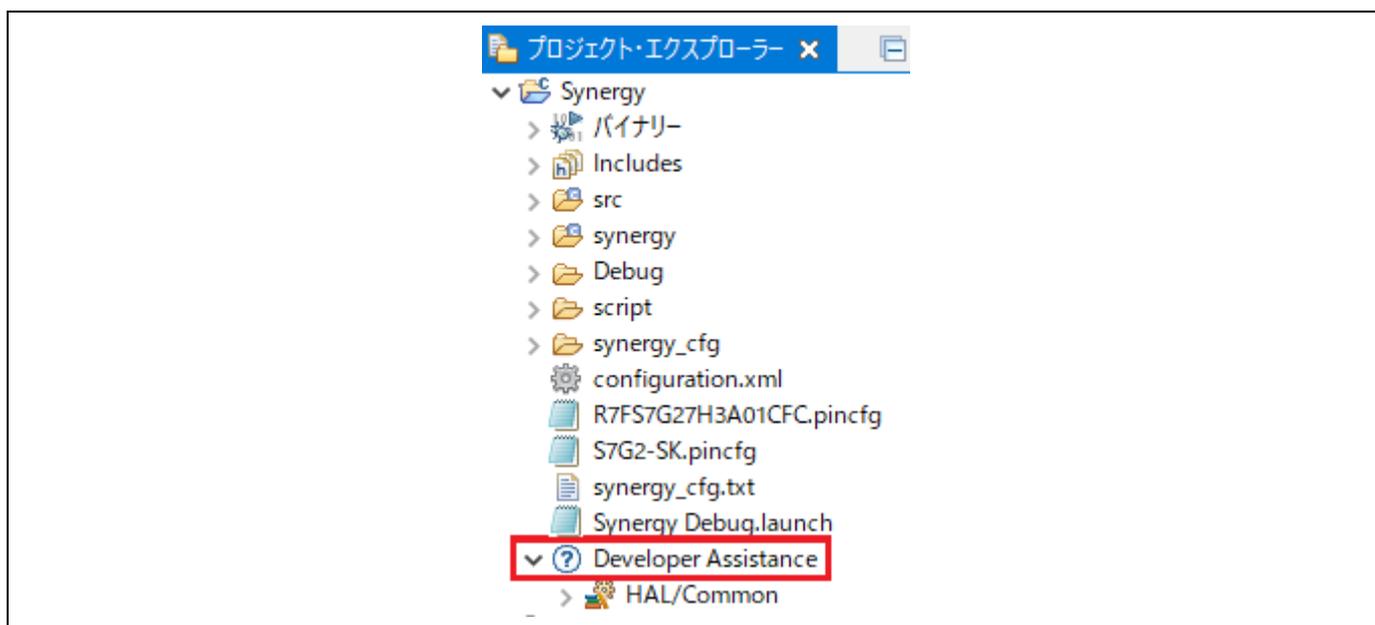


図 3-46 Synergy Developer Assistance

(2) スタックモジュールを展開し、その API を表示します。

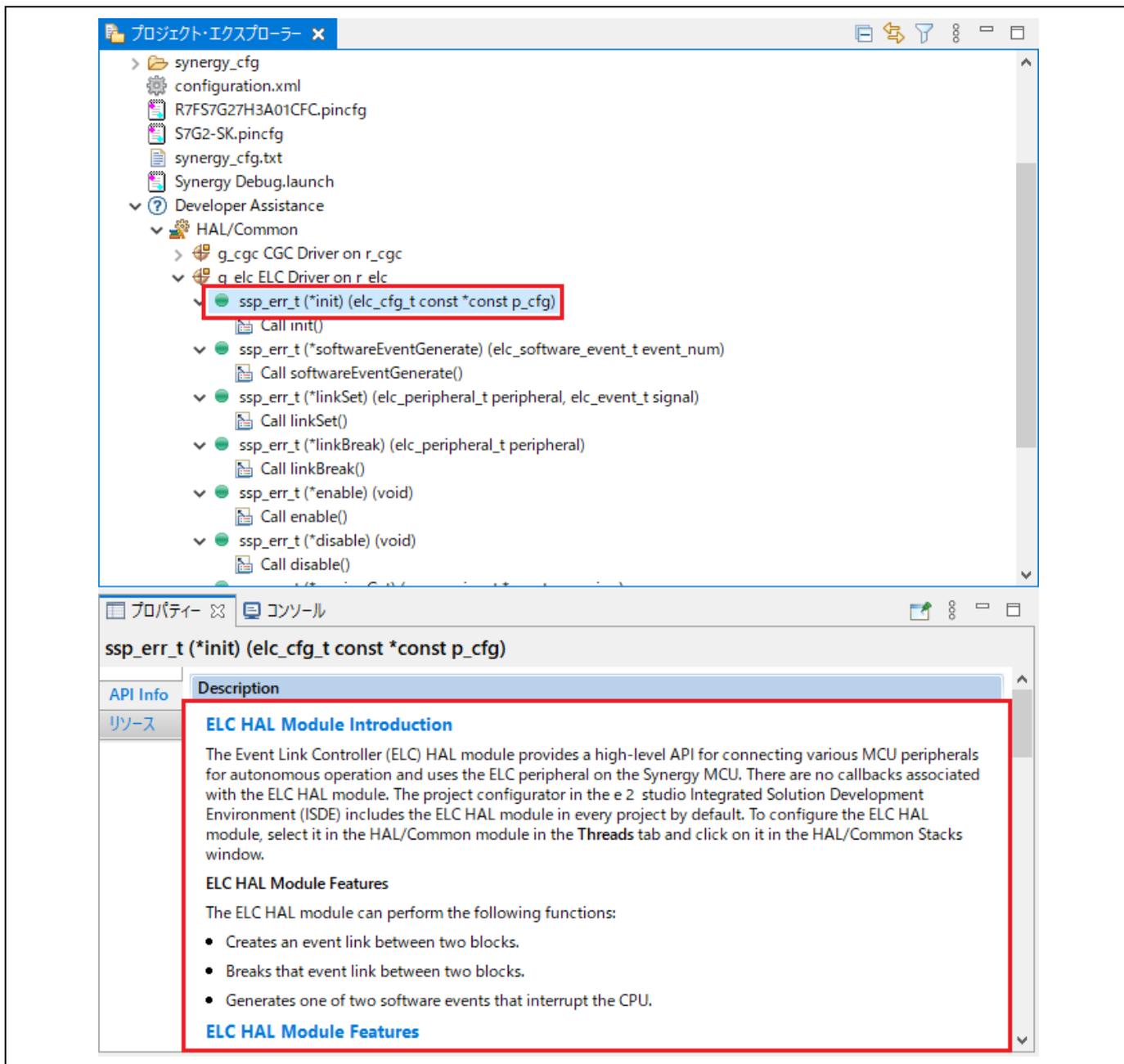


図 3-47 API 情報

(3) Developer Assistance から API をソースファイルにドラッグ&ドロップすると、ソースコードをすぐに  
入力できます。

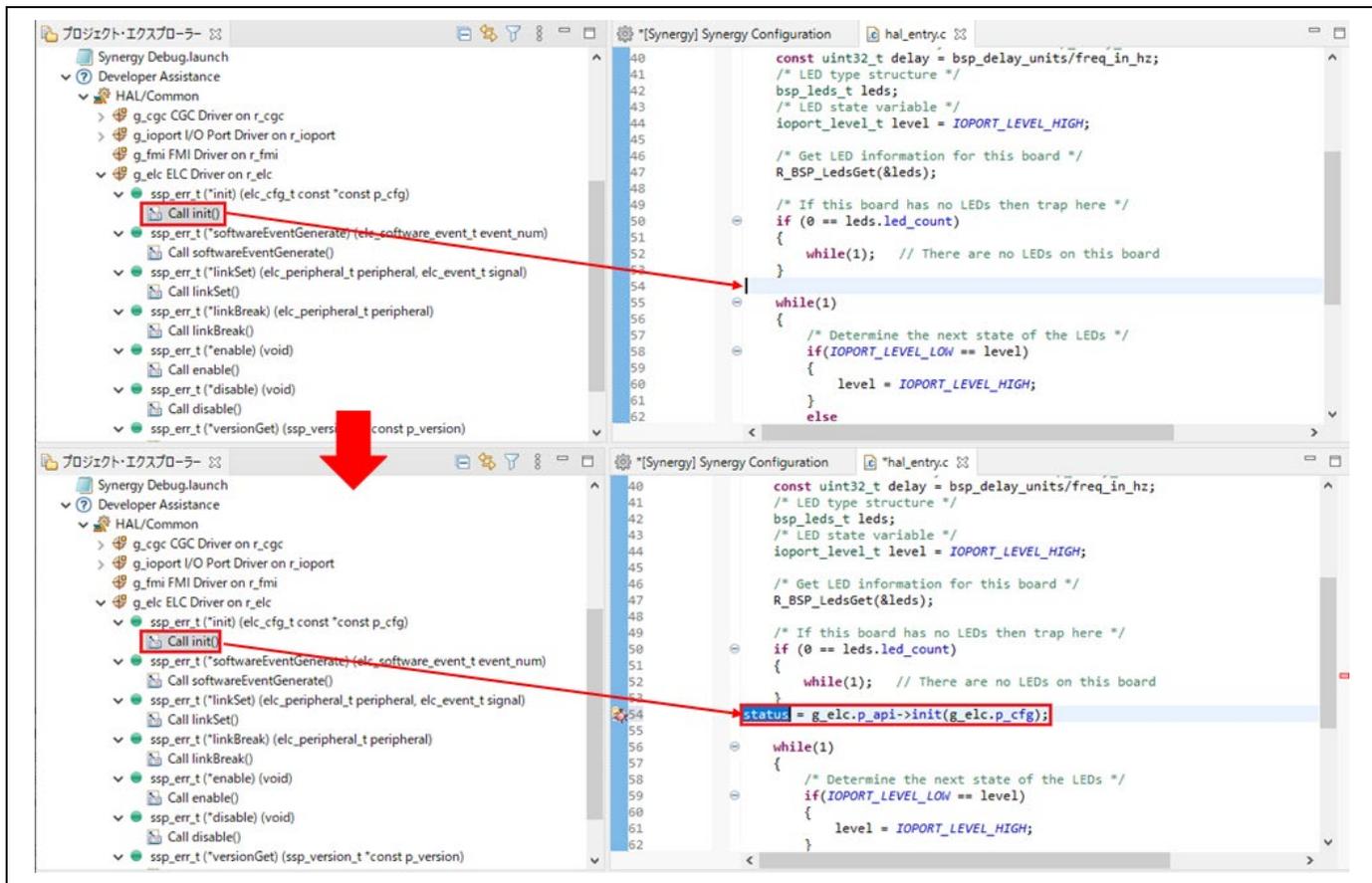


図 3-48 Developer Assistance – API のドラッグ&ドロップ

## 4. ビルド

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

### 4.1 ビルドオプションの設定

プロジェクト作成時はデフォルトのビルドオプションが生成され、通常はデフォルト設定でプロジェクトをビルドできます。

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、ビルドする前に以下の手順を実行してください。

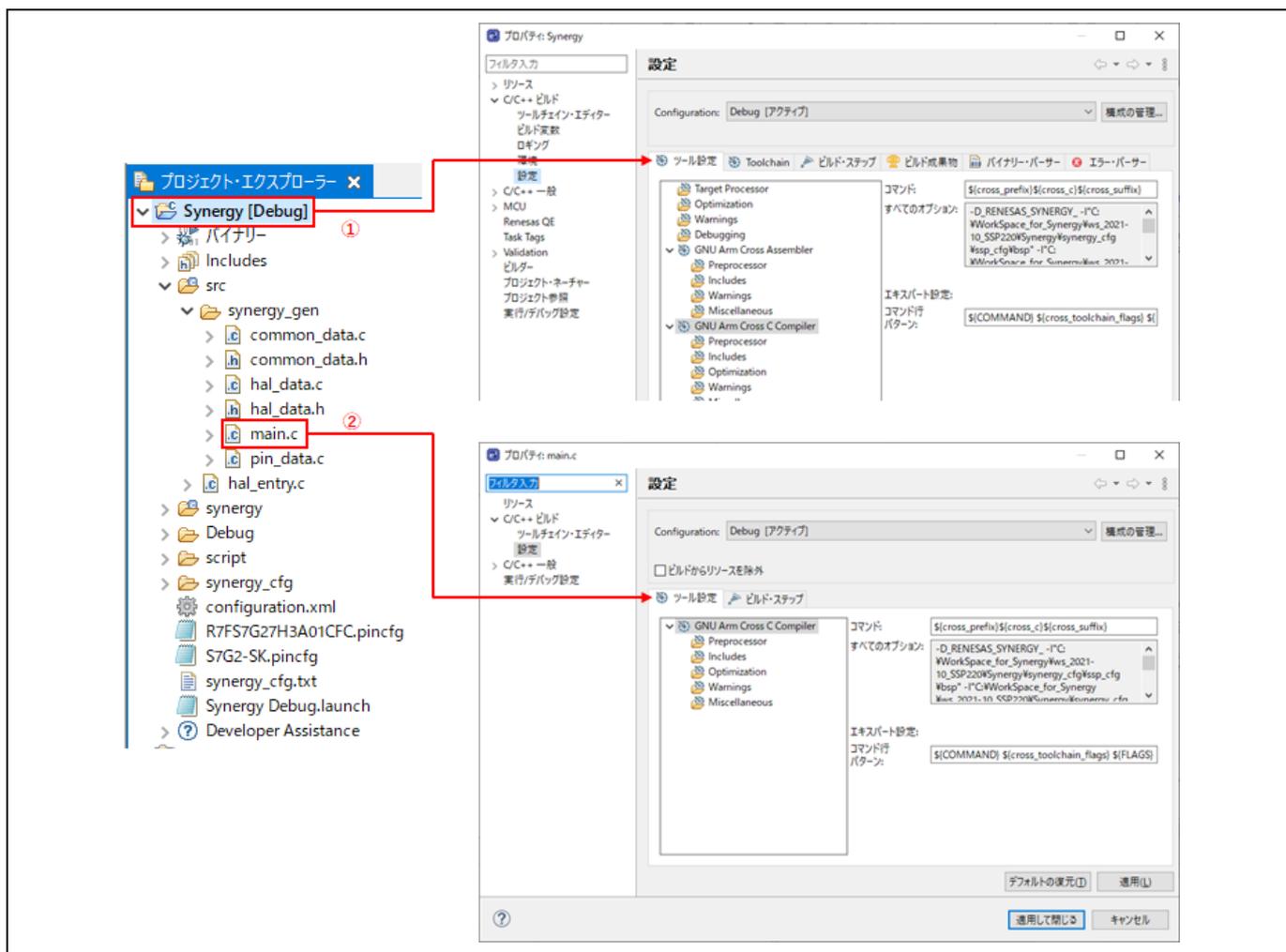


図 4-1 ビルド – Synergy プロジェクトのプロパティと “main.c” ソースファイル

ビルドオプションは、プロジェクトまたはソースファイルの [プロパティ] 画面で設定できます。

- (1) ① プロジェクト名または ② ソースファイル名を選択します。
- (2) 右クリックして [プロパティ] を選択するか、ショートカットキー [Alt] + [Enter] で [プロパティ] 画面を開きます。
- (3) “C/C++ビルド” → “設定” の順に選択して、オプション設定を表示し編集します。

[プロパティ] 画面は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] 画面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内の全てのファイルに共通に適用されます。

## 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・エクスプローラー] ビューで Synergy プロジェクトをクリックして選択します。
- (2) [プロジェクト] → [プロジェクトのビルド] の順に選択するか、 アイコンを押してプロジェクトをビルドします。
- (3) ビルド終了後、エラーがないことを確認してください。

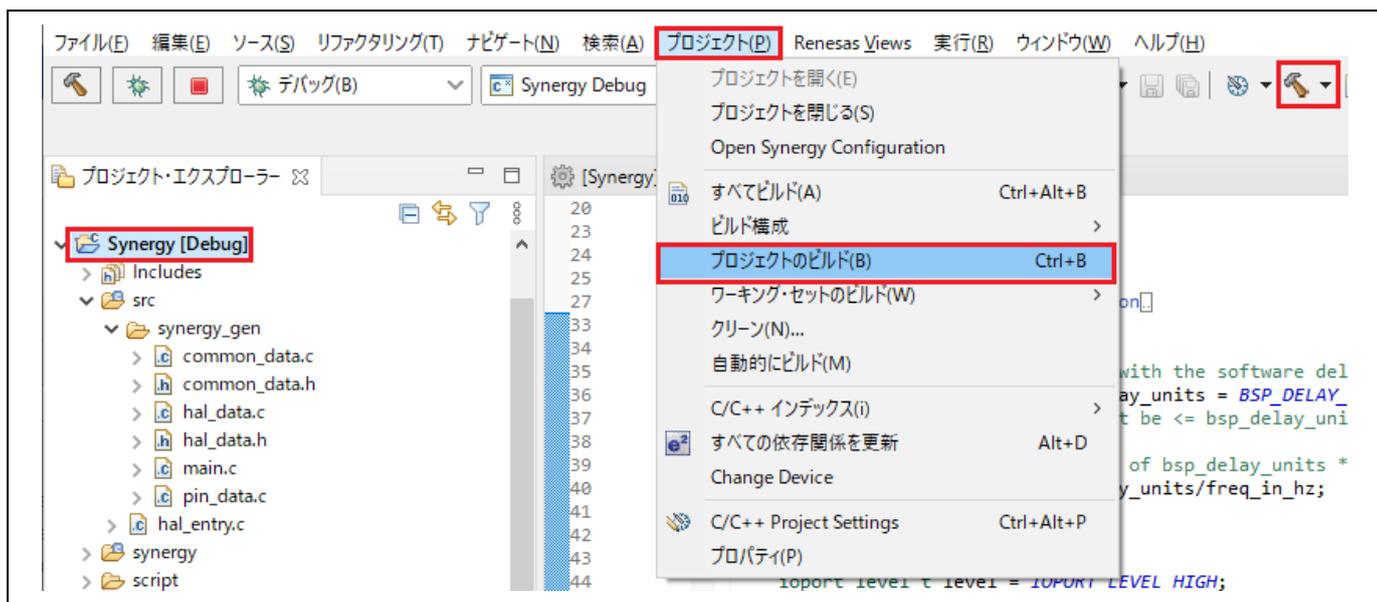


図 4-2 ビルド – サンプルプロジェクトのビルド

### 4.3 プロジェクトレポート機能（ビルドオプション一覧の出力）

プロジェクトレポート機能により、e2 studio のプロジェクト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) [Renesas C/C++ Project Settings] → [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。

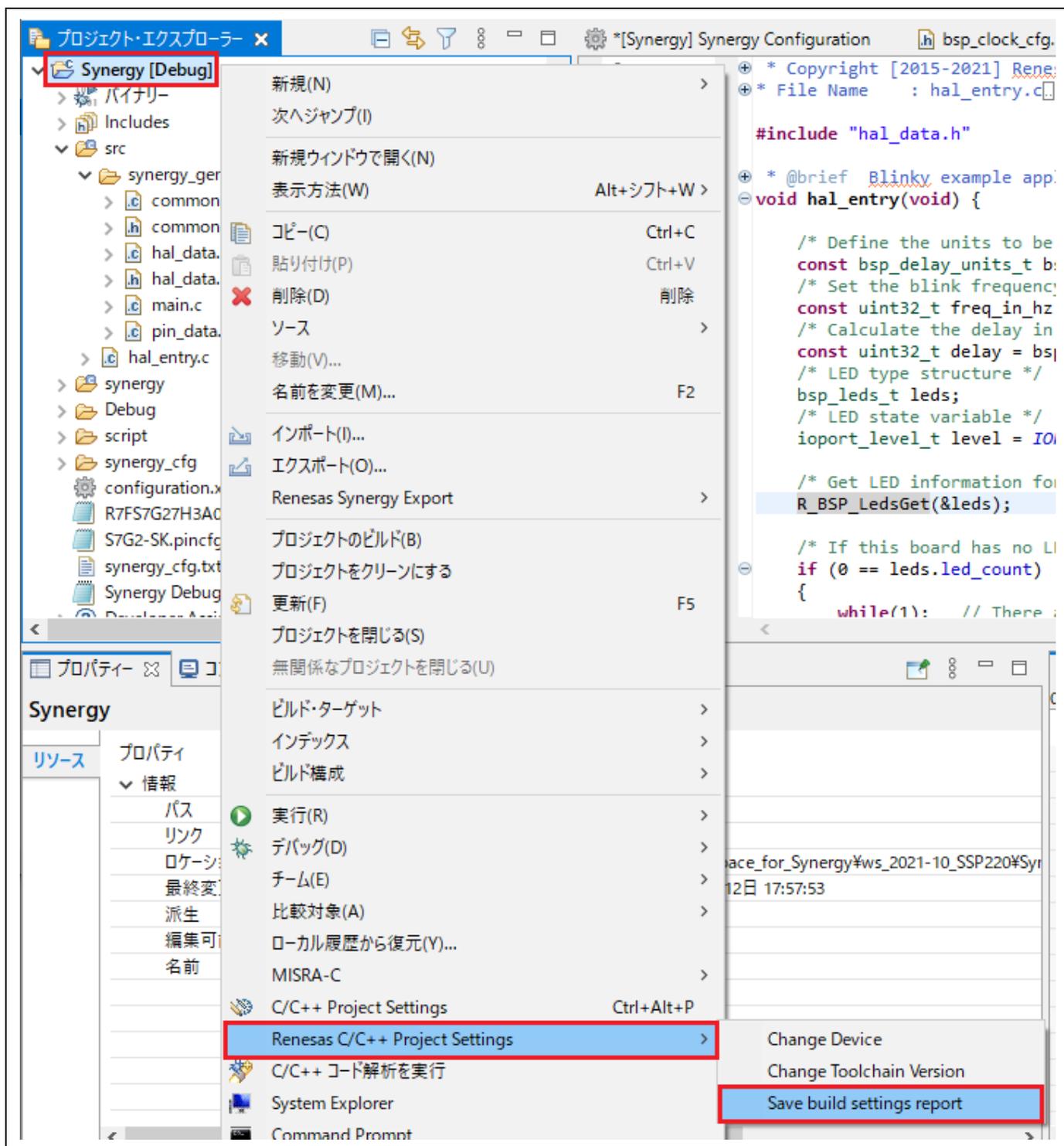


図 4-3 ビルド - プロジェクトレポート機能（ビルドオプション一覧の出力）

## 5. デバッグ

この章では、e<sup>2</sup> 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。J-Link ARM エミュレータと Synergy S7G2 SK ボードを動作環境とする Synergy プロジェクト（「4.2 サンプルプロジェクトのビルド」参照）を例に説明します。

TraceX<sup>®</sup>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、マイクロソフトの Web サイト（<https://docs.microsoft.com/ja-jp/azure/rtos/tracex/about-this-guide>）で入手できる『TraceX<sup>®</sup>ユーザーズマニュアル』を参照してください。ThreadX プロジェクトのデバッグは、この章では説明しません。

いずれかのパースペクティブアイコン上で右クリックし、[テキストの表示] を選択してアイコン名を表示します。

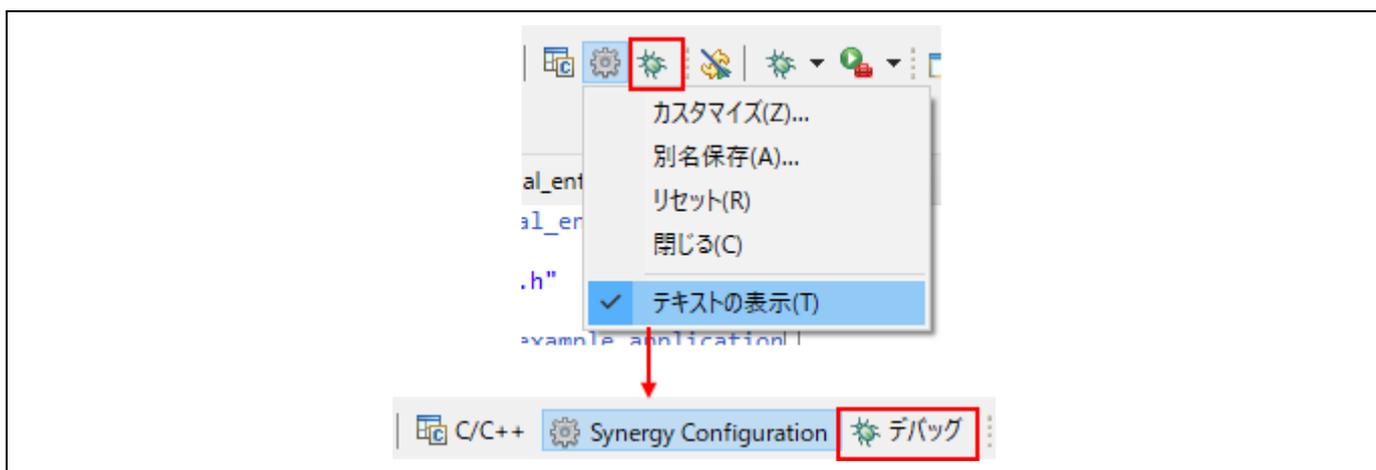


図 5-1 デバッグ – [デバッグ] パースペクティブへの切り替え

e<sup>2</sup> studio で Synergy プロジェクトを開き、[デバッグ] をクリックして [デバッグ] パースペクティブに切り替えます。

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で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。
- [Synergy Configuration] パースペクティブ：エディタ画面の“configuration.xml”とあわせて Synergy configuration を開き、プロジェクト構成を設定する [Package] 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時、e<sup>2</sup> 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中に、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ブ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

Synergy プロジェクトを作成すると、プロジェクトごとに一度だけデフォルトのデバッグ構成が自動で作成されます。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] ビューでプロジェクト名をクリックします。
- (2) [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。

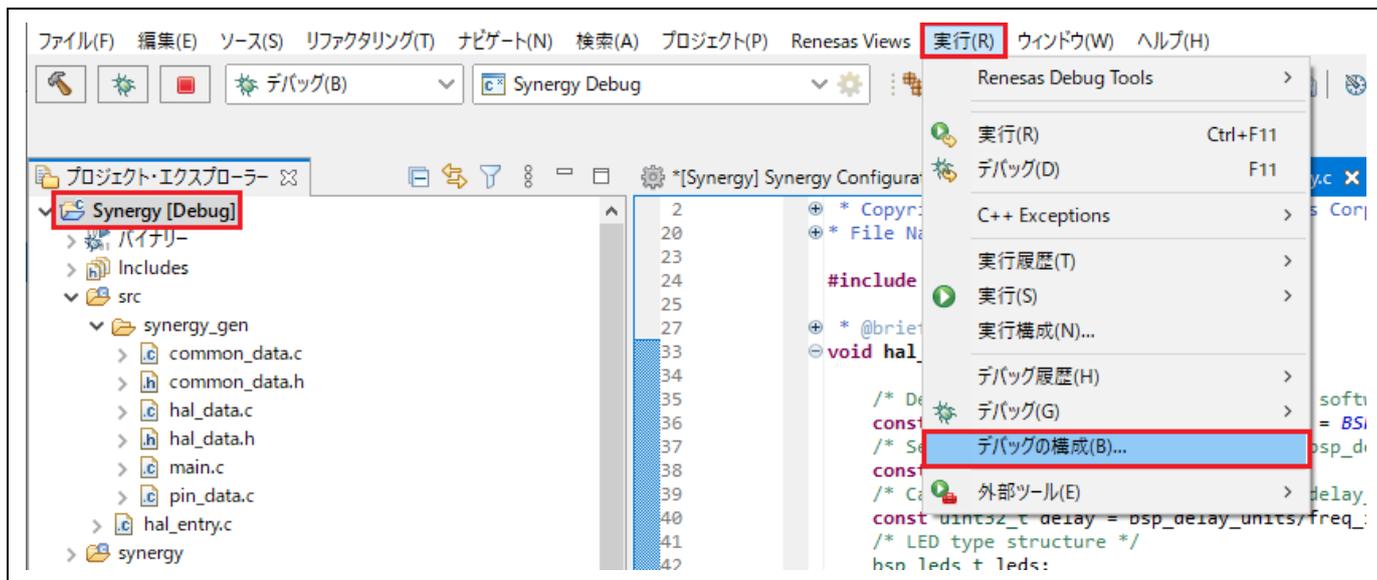


図 5-2 デバッグ - [デバッグ構成] ウィンドウを開く

- (3) [デバッグ構成] ウィンドウで、“Renesas GDB Hardware Debugging” デバッグ構成の表示を展開し、既存のデバッグ構成（例：“Synergy Debug”）をクリックしてください。
- (4) [メイン] タブを選択し、プロジェクトビルドフォルダにあるロードモジュール（例：“Synergy.elf”）を指定してください。

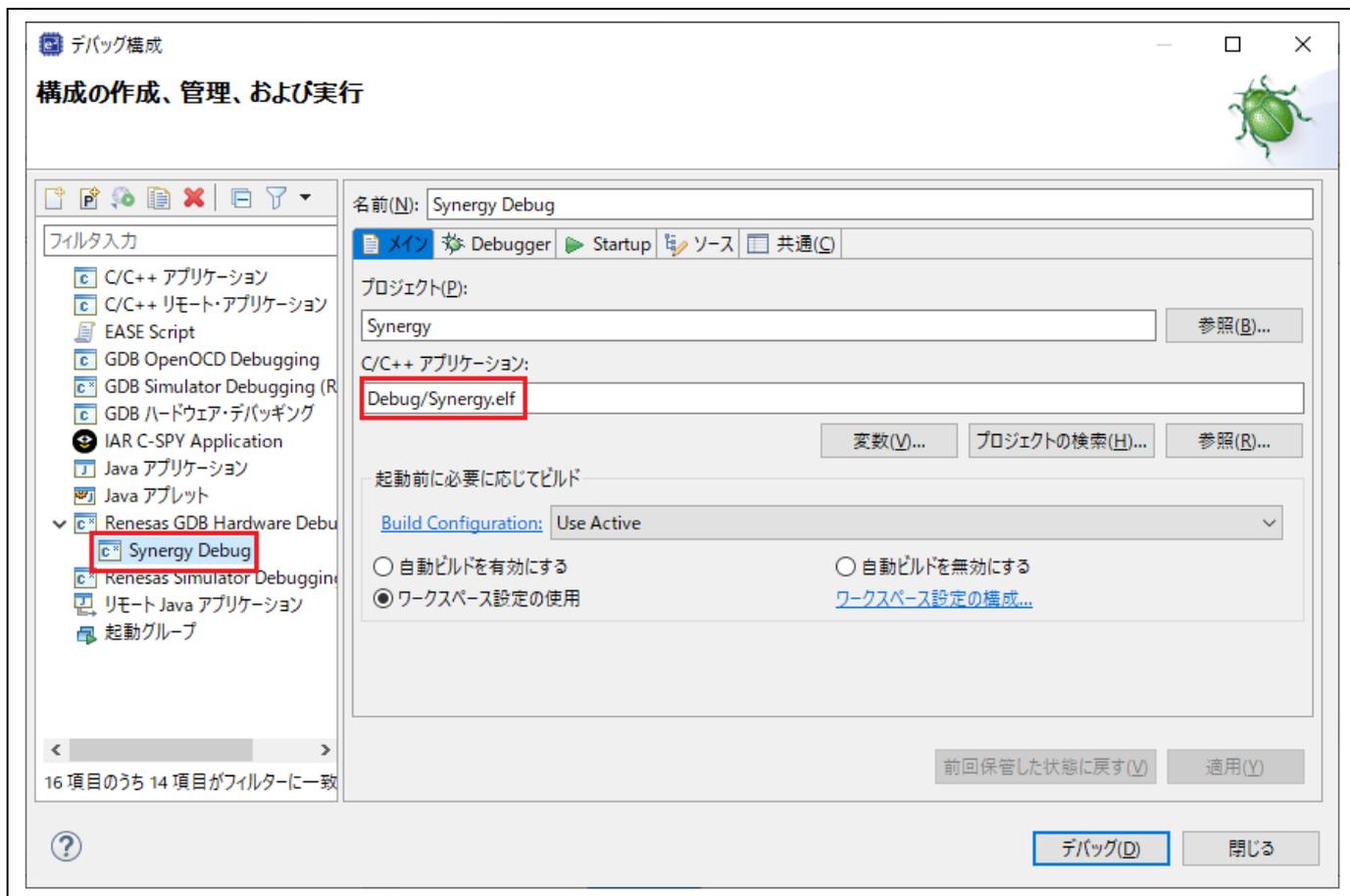


図 5-3 デバッグ – ロードモジュールの選択

- (5) [Debugger] タブに切り替え、エミュレータの種類とデバイス名を選択してください。  
 Debug hardware : “J-Link ARM”  
 Target Device : “R7FS7G27H”
- (6) [適用] ボタンを押すと設定が保存されます。
- (7) [デバッグ] ボタンを押すと、デバッグ起動構成が実行され、J-Link エミュレータと Synergy ボードへ接続されます。



図 5-4 デバッグ – 接続設定の変更

- (8) 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\_S7G2.c” の “Reset\_Handler()”）でプログラムの実行が一旦中断されます。

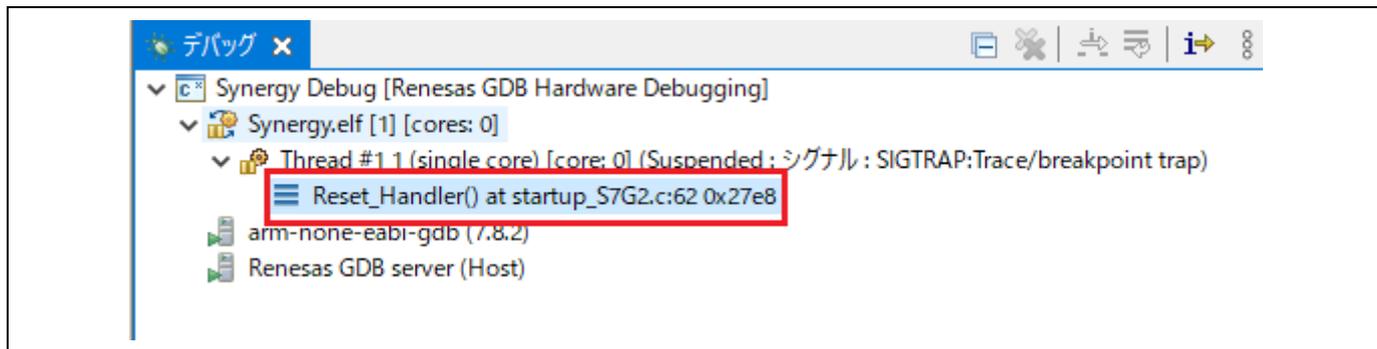


図 5-5 デバッグ - ターゲット接続後の [デバッグ] ビュー表示

## 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) [デバッグ構成] ウィンドウを開きます（「図 5-2」参照）。
- (2) [デバッグ構成] ウィンドウの左ペインに表示されたデバッグ構成（例：“Synergy Debug”）を選択し、 アイコンをクリック（現在選択している起動構成をコピー）すると、複製されたデバッグ構成（例：“Synergy Debug (1)”）が作成されます。
- (3) デバッグ構成は「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。

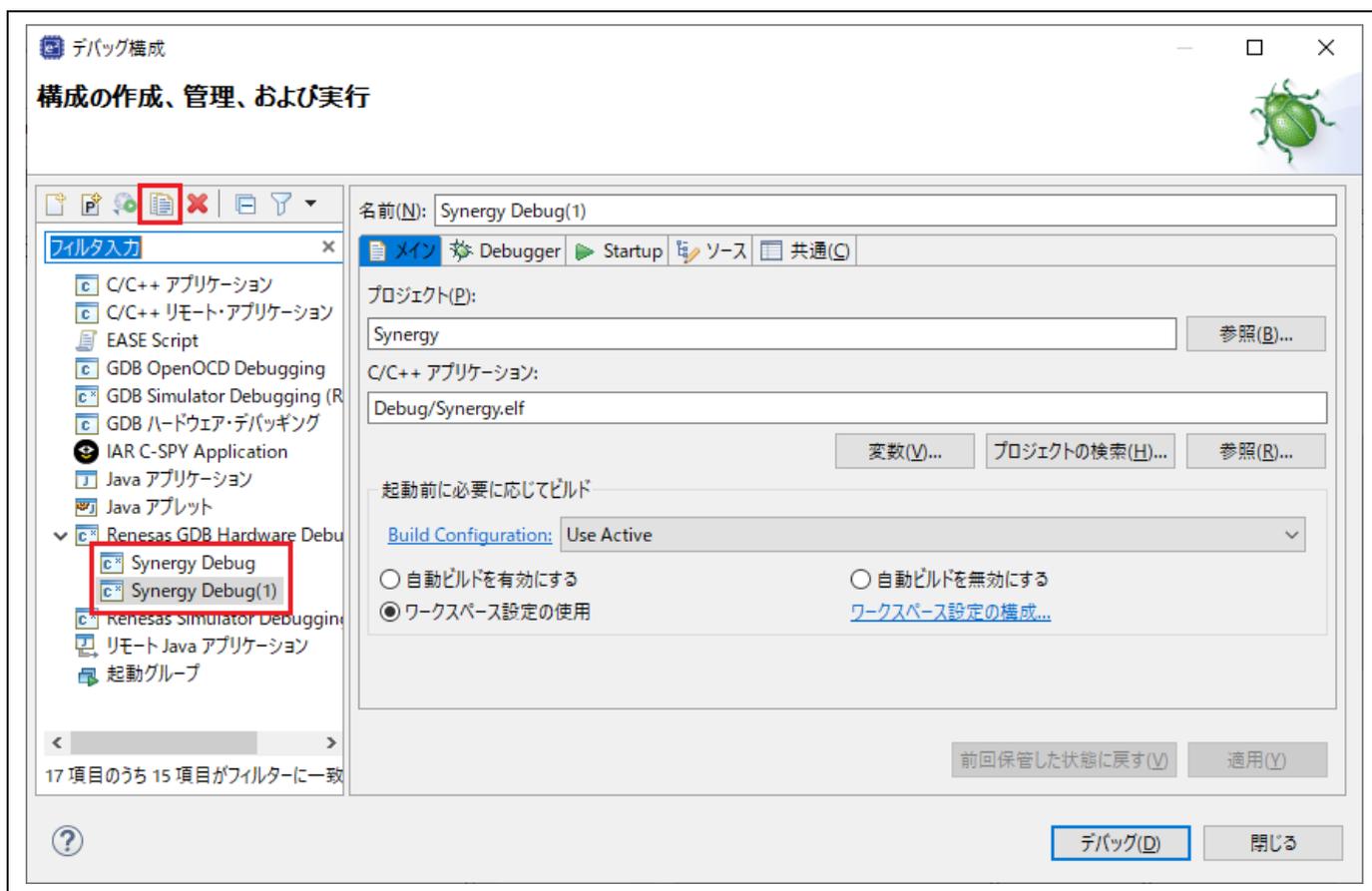


図 5-6 デバッグ — 選択したデバッグ構成の複製

### 5.3 基本的なデバッグ機能

この章では、e<sup>2</sup> 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（MMU ビューは Synergy 向けにはサポートされていません）
- GDB デバッガの Renesas による拡張機能のビュー：IO レジスタ、イベントポイント、トレース、Fault Status

デバッグツールバーを開くには、プルダウンメニューボタンをクリックし、[デバッグツールバーを表示] を選択します。[デバッグ] ビューには、下記のような便利なツールバーがあります。

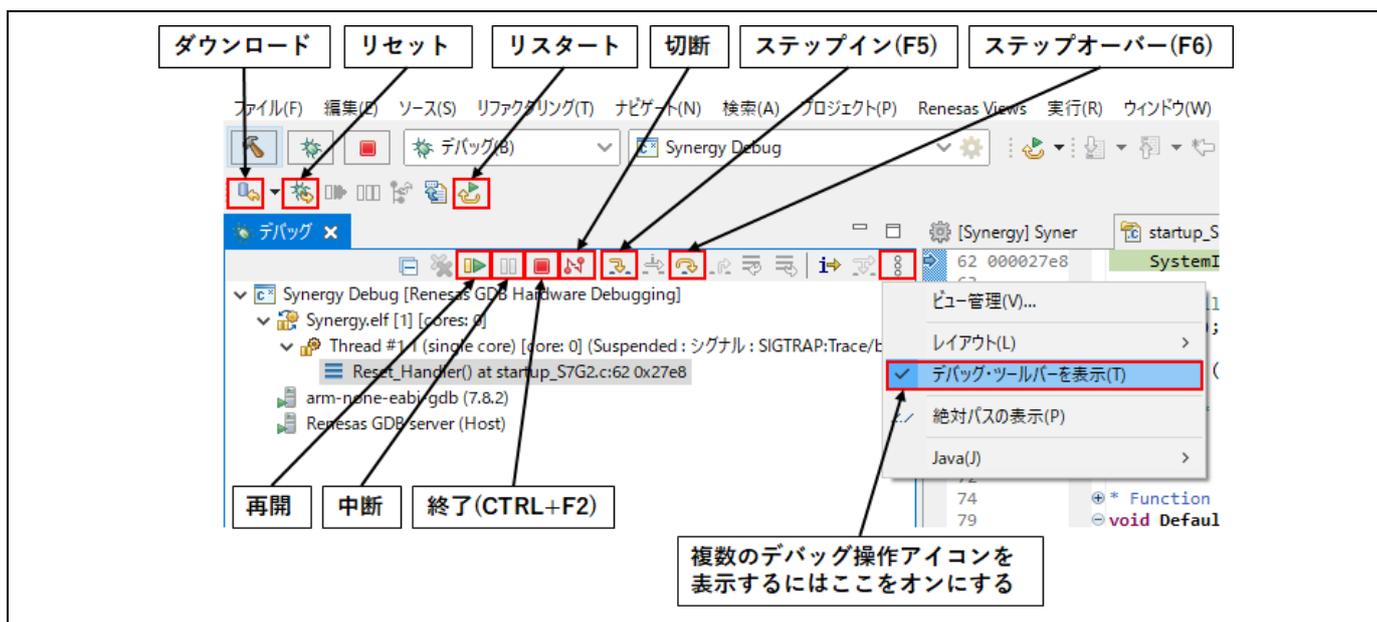


図 5-7 デバッグ - [デバッグ] ビューの便利なツールバー

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

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

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

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

他に以下のような操作が可能です。

- ボタンは、リセット後にプログラムを実行します。デバッグ構成でブレークポイントが設定されていると、main()関数で実行停止します。
- 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。

### 5.3.1 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。

e<sup>2</sup> studio ではソフトウェアブレークポイント (🔴 のマーカー) とハードウェアブレークポイント (🔵 のマーカー) を区別して設定できます。ダブルクリックによりブレークポイントを設定すると、デフォルトのブレークポイント型が適用されます。ハードウェアブレークポイントの場合、ブレークポイント用のハードウェアリソースが残っていないときは設定できません。その場合は、ソフトウェアブレークポイントに置き換えるとのメッセージが表示されます。

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

#### (1) エディタ画面上でブレークポイントを設定する

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

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

#### (2) 「ブレークポイント」ビューでブレークポイントプロパティを設定する

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

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

- エディタの左余白に表示されたブレークポイントマーカー (🔴または🔵) を右クリックし、コンテキストメニューの「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。

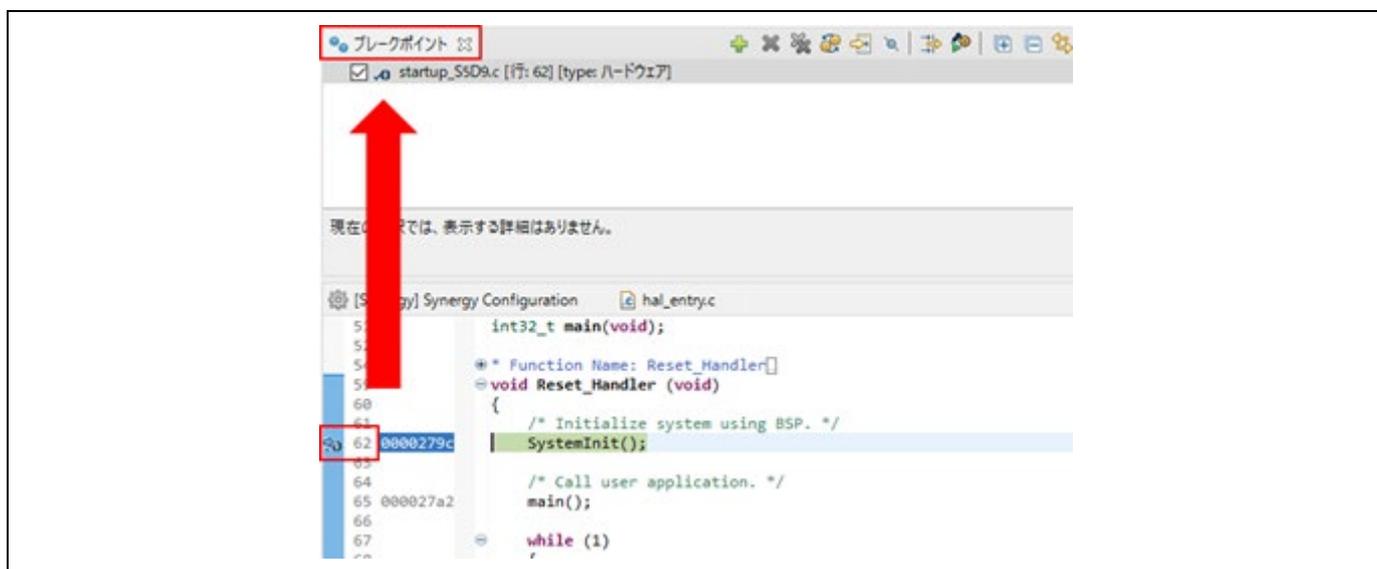


図 5-8 デバッグ - [ブレークポイント] ビュー

### 5.3.2 式ビュー

[式] ビューでは、デバッグ中のグローバル変数、静的変数、ローカル変数の値をモニタできます。変数名で登録する他に、計算式（例：“Aval+Bval\*2”）や型キャスト（例：“(struct mystr \*)&buf[1]”）を使って表示させることができます。

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) エディタ上で変数名（例：“r\_cg.c” 内の “g\_cg\_version”）を選択状態にしてから [式] ビューヘドラッグ&ドロップします。（または、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。）

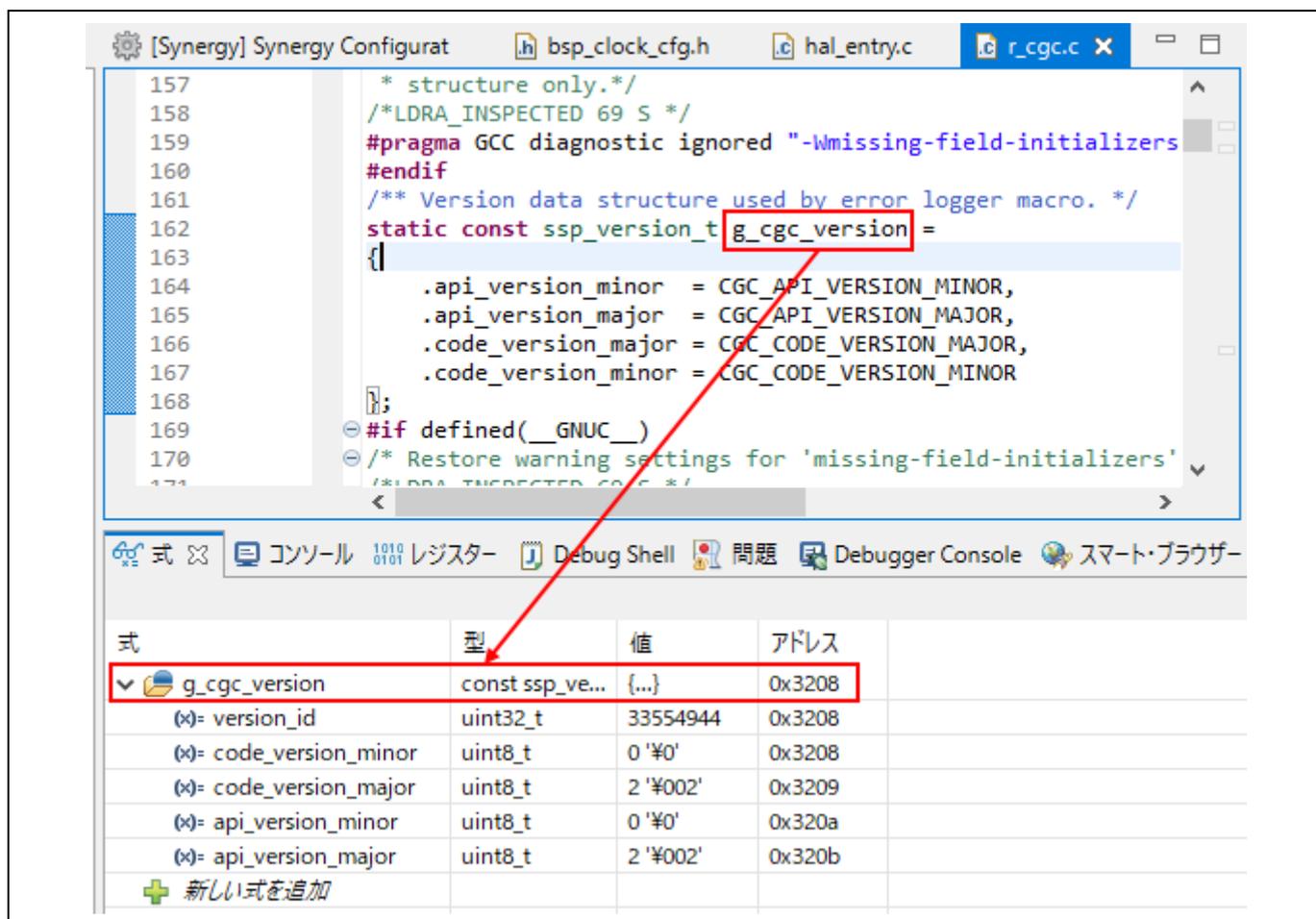


図 5-9 デバッグ - [式] ビュー

### 5.3.3 レジスタビュー

[レジスタ] ビューは、Synergy マイクロコントローラの汎用レジスタについての情報を表示します。

- (1)  のアイコンが付いた [レジスタ]ビューを選択するか、表示されていない場合は [ウィンドウ] メニューの [ビューの表示] → [レジスタ]を選択して表示させます。
- (2) レジスタ名を選択すると詳細欄には各基数のフォーマットで値が表示されます。

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

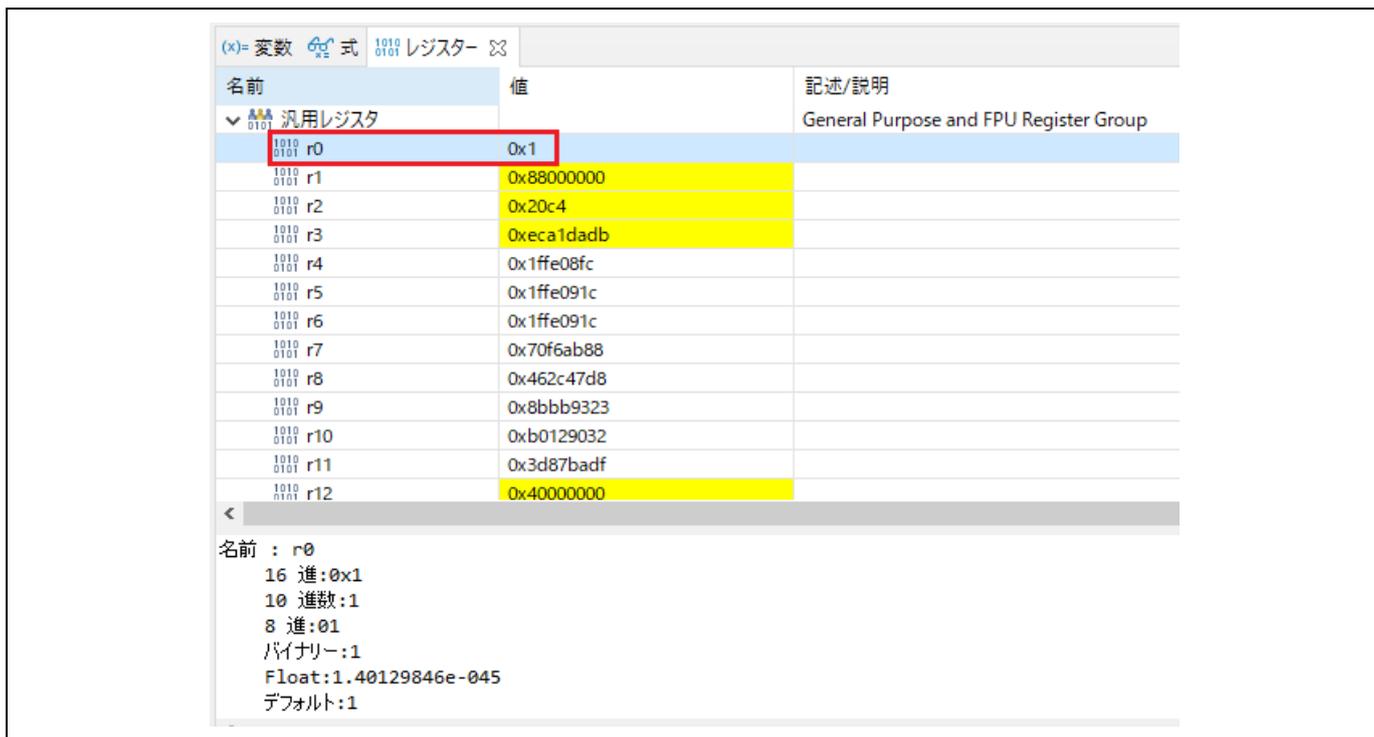


図 5-10 デバッグ - [レジスタ] ビュー

### 5.3.4 メモリービュー

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

変数（例：“g\_ssp\_version\_build\_string”）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、メモリーの表示開始アドレスを指定してください。下記の例では変数のアドレス “&g\_ssp\_version\_build\_string” を指定しています。

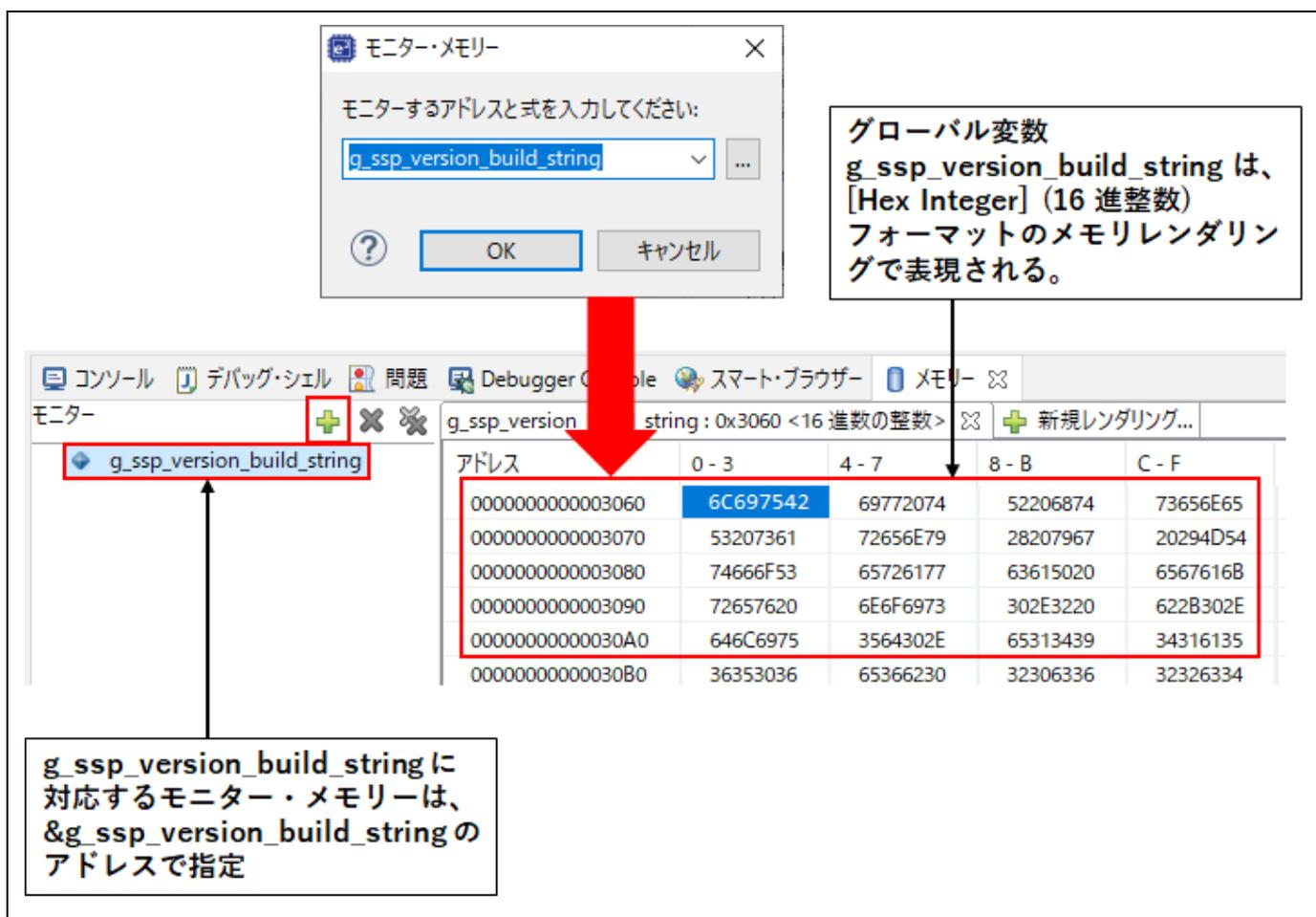


図 5-11 デバッグ - [メモリー]ビュー

指定したモニター・メモリーに対するレンダリング（表示形式）を追加するには、

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

下記の例では変数 “g\_ssp\_version\_build\_string” を「16 進数の整数」と「ASCII」の二種類で表示を切り替えられるようにしています。

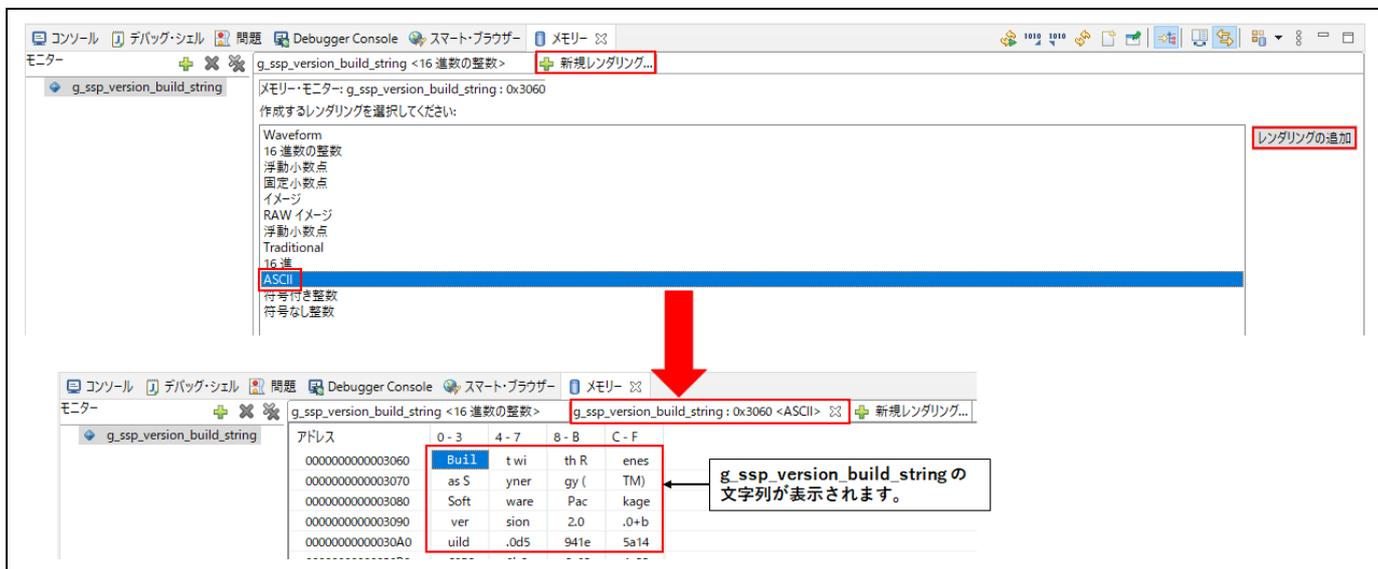


図 5-12 デバッグ - [メモリー]ビューの新規レンダリング（表示方式）

### 5.3.5 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) の情報を取得します。表示するのは対象プロジェクトの合計のメモリーサイズ、ROM/RAM の各使用率、セクション/オブジェクト/シンボル/モジュール単位の情報、ベクタテーブルやクロスリファレンスです。

e<sup>2</sup> studio のバージョン 7.3 以上では、ROM/RAM の使用量をグラフ表示できます。

[メモリー使用量] ビューを使うには、[ウィンドウ] メニュー → [ビューの表示] → [その他...] の [C/C++] カテゴリから [メモリー使用量] を選択し、[開く] ボタンでビューを開いてください。

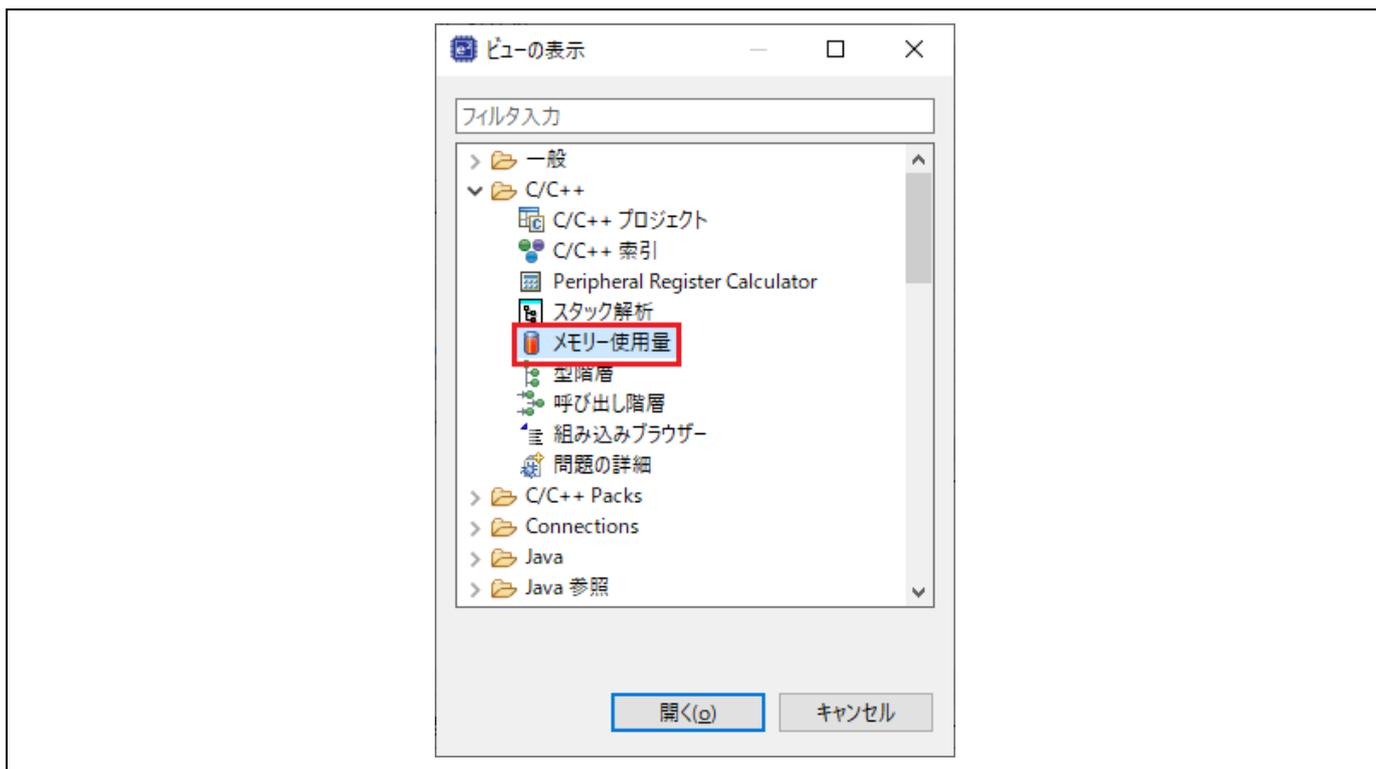


図 5-13 [メモリー使用量] ビューの表示

[メモリー使用量] ビューは次の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は未サポート)、(3) 詳細

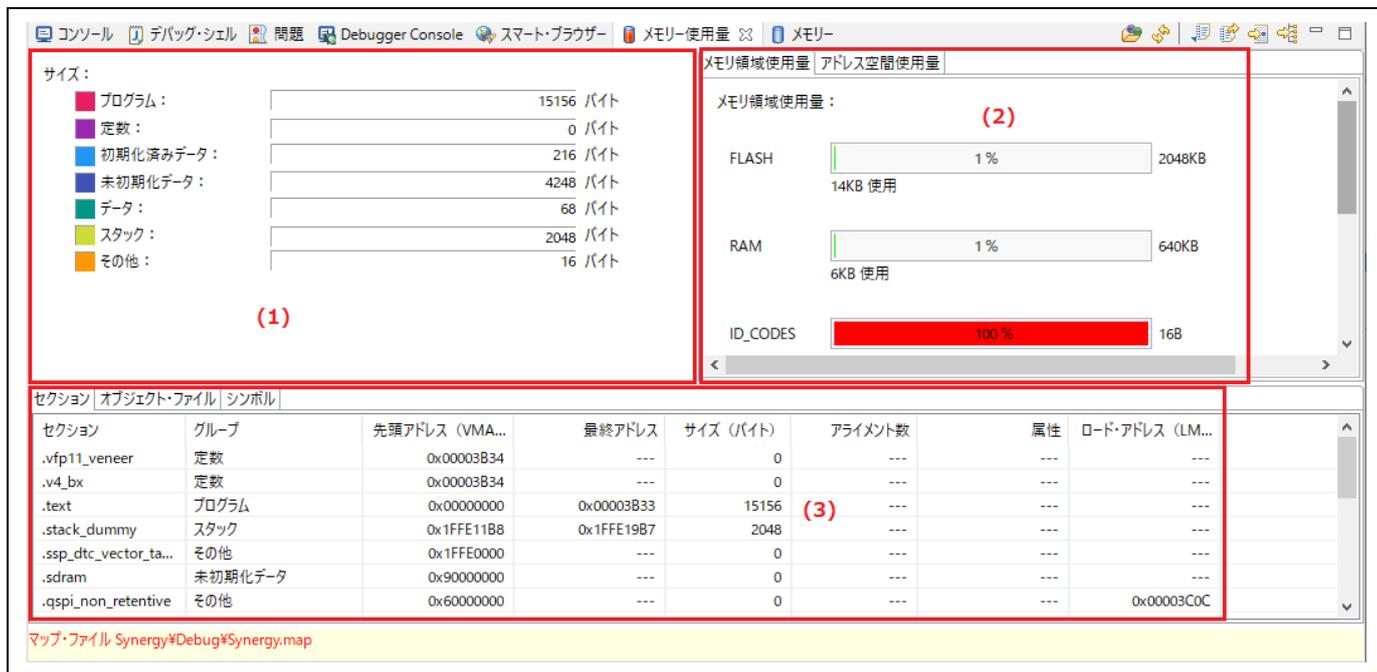


図 5-14 [メモリー使用量] ビューのエリア

[メモリー使用量] ビューでは以下の操作が可能です。

- : メモリー使用量を表示するためのマップファイル (\*.map) やライブラリリストファイル (\*.lbp) を選択します。
- : [メモリー使用量] ビューの情報をすべて更新します。
- : 詳細エリアのすべてのタブのデータをエクスポートします。
- : マップファイルやライブラリリストファイルをエディタで開きます (Synergy ライブラリプロジェクトにはライブラリリストファイルはありません)。
- : 選択したプロジェクトのマップファイルを出力するためのページを開きます。
- : 選択したプロジェクトの [セクション] ページを開きます。

### 5.3.6 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。

[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。すでに [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、アドレスカラムを右クリックし [オPCODEを表示]、[関数オフセットを表示] でオPCODE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。

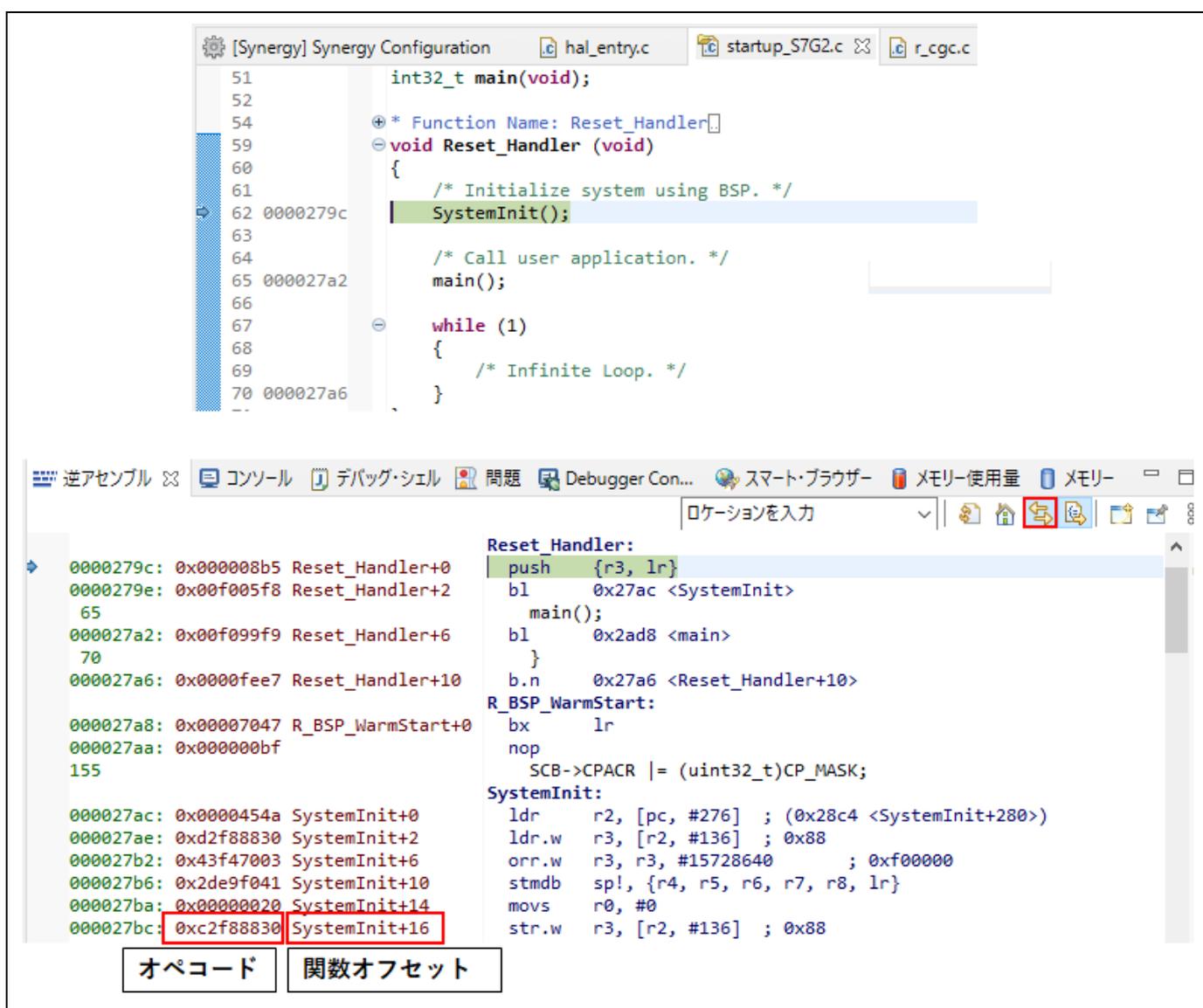


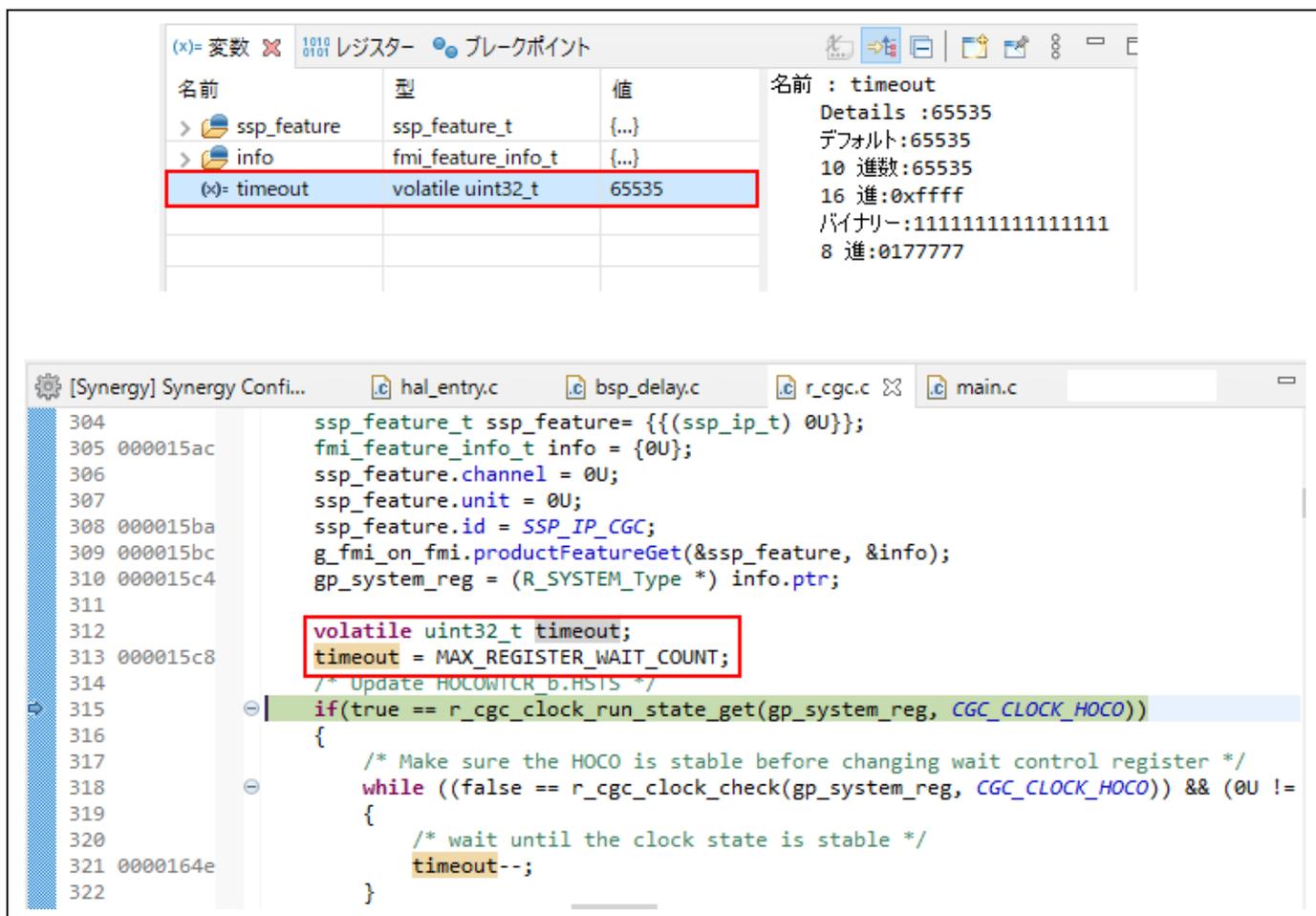
図 5-15 デバッグ - [逆アセンブル] ビュー

### 5.3.7 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。

ローカル変数を見るには、

- (1) [変数] ビュー (  ) のアイコン) を選択するか、[ウィンドウ] メニュー → [ビューの表示] → [変数] でビューを開きます。
- (2) ステップ実行で関数内に入ると、変数ビューにローカル変数とその値が表示されます。(下記の例では R\_CGC\_Init() 関数内の "timeout" 変数)



The screenshot displays the Synergy IDE interface. At the top, the 'Variable View' window is open, showing a table of local variables. The 'timeout' variable is highlighted with a red box. Below it, the source code for the `R_CGC_Init()` function is shown, with the declaration and assignment of the `timeout` variable also highlighted with a red box.

名前	型	値	名前 : timeout
> ssp_feature	ssp_feature_t	{...}	Details : 65535
> info	fmi_feature_info_t	{...}	デフォルト: 65535
(x) timeout	volatile uint32_t	65535	10 進数: 65535
			16 進: 0xffff
			バイナリ: 1111111111111111
			8 進: 0177777

```

304     ssp_feature_t ssp_feature= {{{(ssp_ip_t) 0U}}};
305 000015ac     fmi_feature_info_t info = {0U};
306     ssp_feature.channel = 0U;
307     ssp_feature.unit = 0U;
308 000015ba     ssp_feature.id = SSP_IP_CGC;
309 000015bc     g_fmi_on_fmi.productFeatureGet(&ssp_feature, &info);
310 000015c4     gp_system_reg = (R_SYSTEM_Type *) info.ptr;
311
312     volatile uint32_t timeout;
313 000015c8     timeout = MAX_REGISTER_WAIT_COUNT;
314     /* Update HOCLOCK_B.HSIS */
315     if(true == r_cgc_clock_run_state_get(gp_system_reg, CGC_CLOCK_HOCO))
316     {
317         /* Make sure the HOCO is stable before changing wait control register */
318         while ((false == r_cgc_clock_check(gp_system_reg, CGC_CLOCK_HOCO)) && (0U !=
319         {
320             /* wait until the clock state is stable */
321             timeout--;
322         }

```

図 5-16 デバッグ - [変数] ビュー

### 5.3.8 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表示します。ユーザは、[選択されたレジスタ] に必要な IO レジスタを選択して追加することによって、[IO Registers] ビューをカスタマイズすることができます。

選択した IO レジスタの値を見るには、

- (1) [IO Registers] ビュー (  のアイコン) を選択するか、[ウィンドウ] メニュー → [ビューの表示] → [IO Registers] を選択してビューを開いてください。
- (2) [すべてのレジスタ] タブで、対象のモジュール (例 : CAC) を探し、モジュールの IO レジスタ一覧を展開してください。
- (3) レジスタ (例 : CAICR と CASTR) を [選択されたレジスタ] 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] タブを選んで、選択したレジスタ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] 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] を使うことをお勧めします。

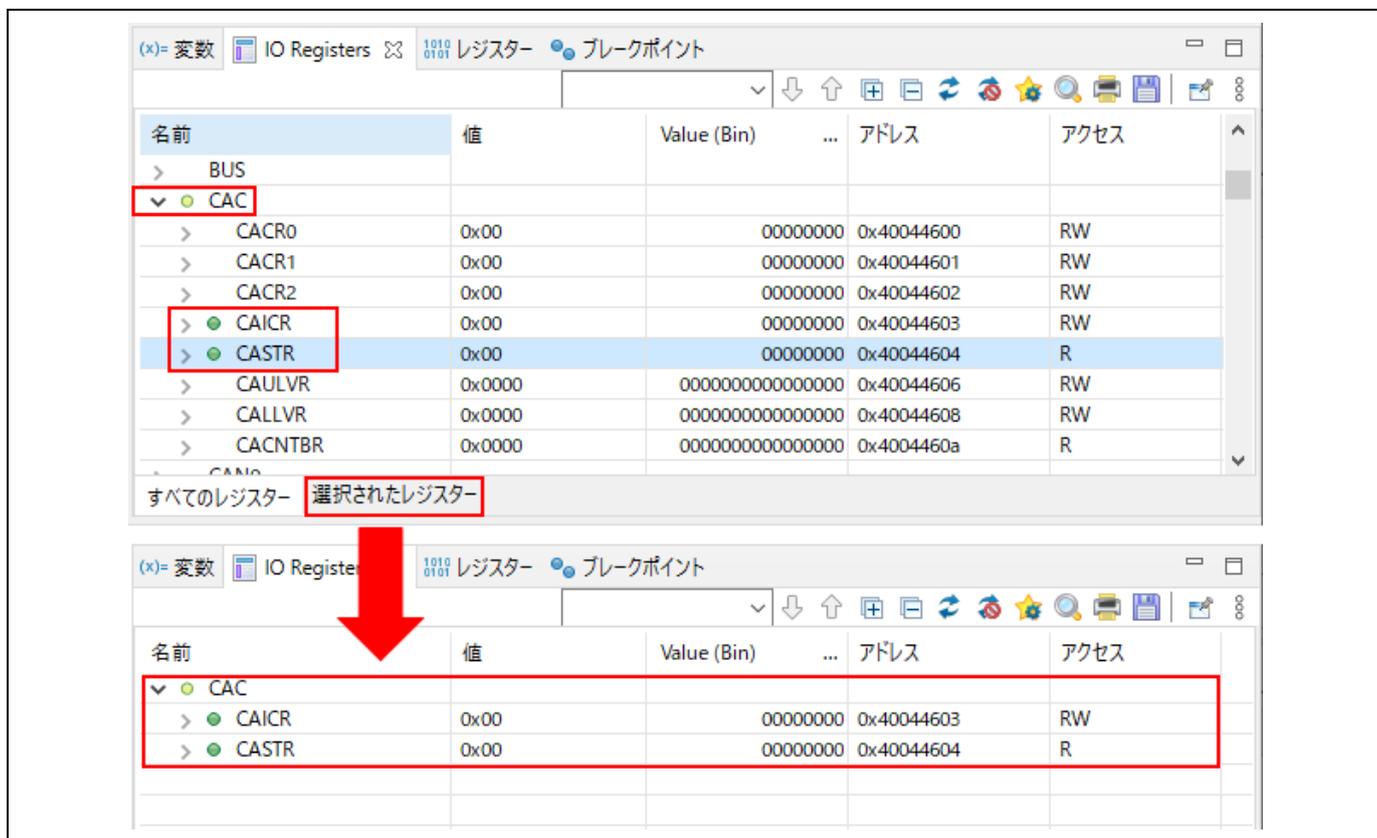


図 5-17 デバッグ - [IO Registers] ビュー

IOレジスタを検索するには、

- (1) 検索ボックスにIOレジスタ名を入力し、下向き矢印ボタン  をクリックします。
- (2) 検索したレジスタが [IO Registers] ビューに表示されます。

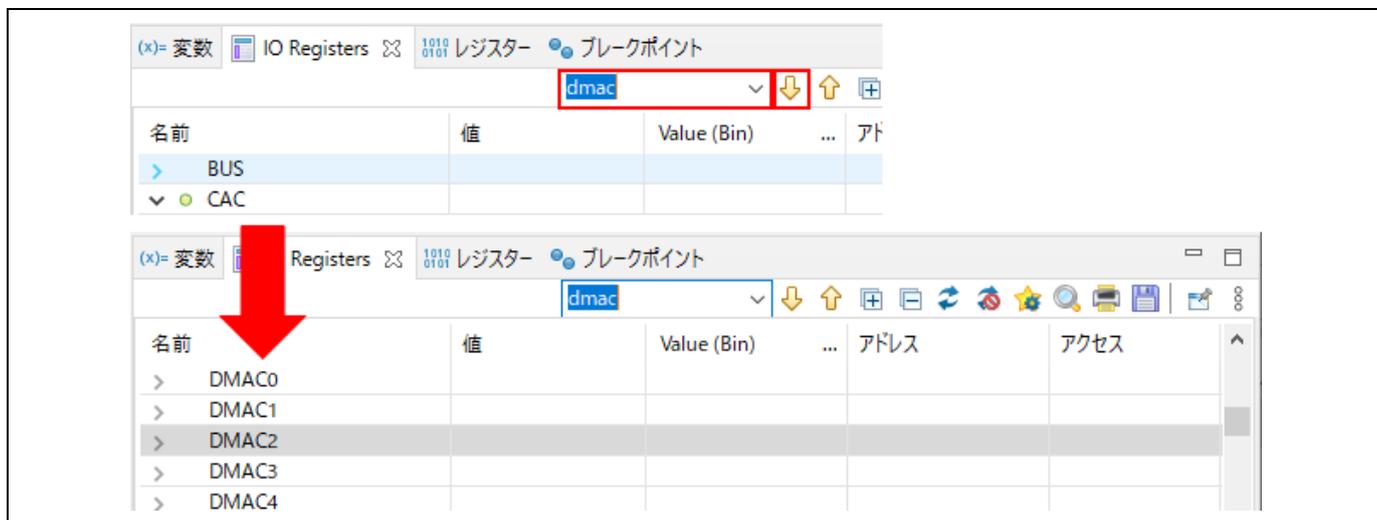


図 5-18 デバッグ - [IO Registers] ビュー (レジスタの検索)

### 5.3.9 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、イベントブレークなどを設定、表示することができます。

Synergy プロジェクトでは、データアクセスによるイベントブレークをサポートしています。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。

表 5-1 イベントの組み合わせ条件

組み合わせ条件	説明
OR	指定したイベントのうち1つが一致すると条件成立。
AND（組み合わせ）	指定したイベントが、順序は問わず、すべて一致すると条件成立。
シーケンシャル	指定したイベントが、指定した順序で一致すると条件成立。

アドレスまたはデータが一致する条件（例：g\_bsp\_leds へのアクセス時）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) [Renesas Views] → [デバッグ] → [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て [イベントポイント] ビューを開きます。
- (2) タイプカラムの“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。

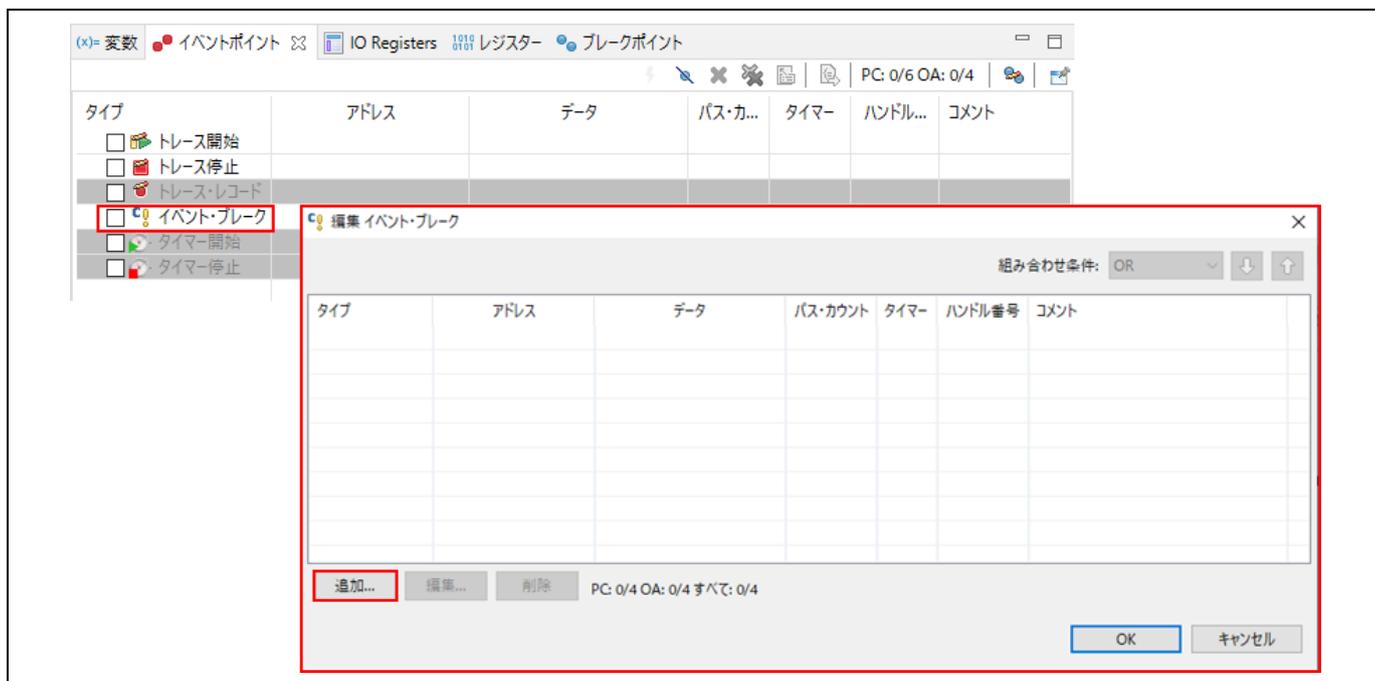


図 5-19 デバッグ - [イベントポイント] ビュー(1/2)

- (4) [イベントポイント・タイプ] はデフォルトで “Data Access” が選択されています。
- (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 “g\_bsp\_leds” を検索します。（グローバル変数のアドレスは “&g\_bsp\_leds” 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 “Read” を選択します。
- (7) [OK] ボタンで次に進みます。

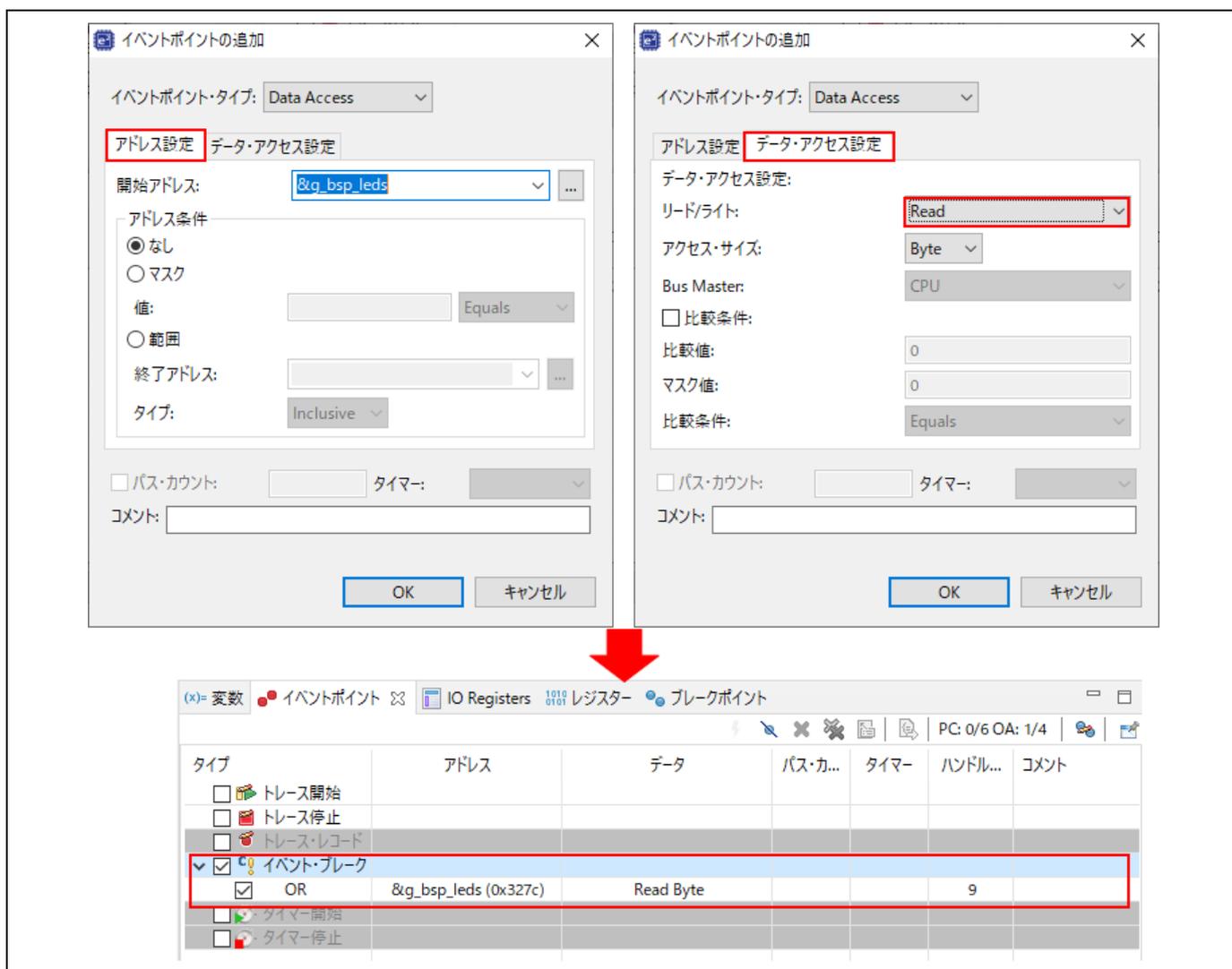


図 5-20 デバッグ - [イベントポイント] ビュー(2/2)

- (8) プログラムを最初から実行するためにリセットします。
- (9) 下図は、条件が成立してイベントブレークが掛かった様子を示します。“g\_bsp\_leds” のアクセス（リード）でブレークしています。

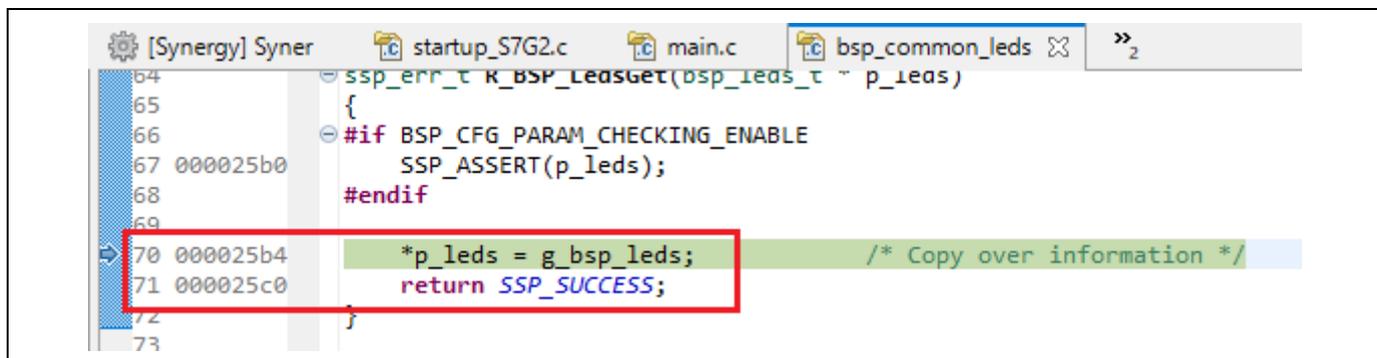


図 5-21 デバッグ — イベントブレークの実行

### 5.3.10 トレースビュー

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

トレースバッファは有限なため、バッファがいっぱいになると、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て [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取得を有効にしてください。

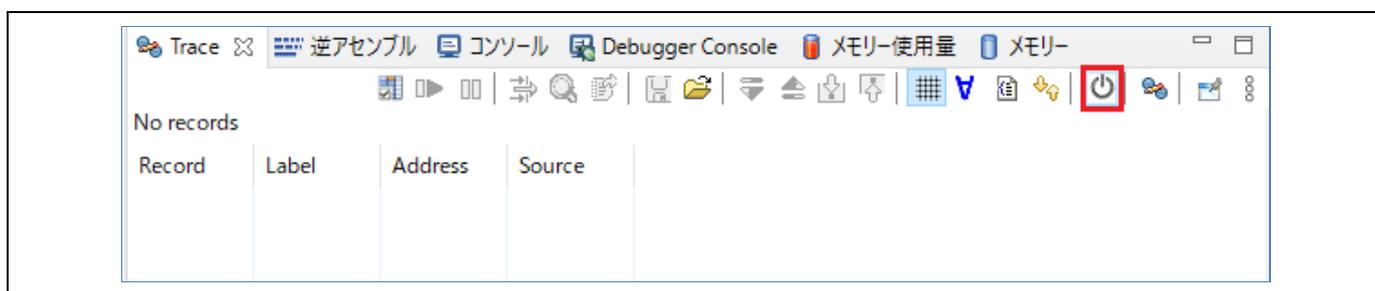


図 5-22 デバッグ – トレース有効

- (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。

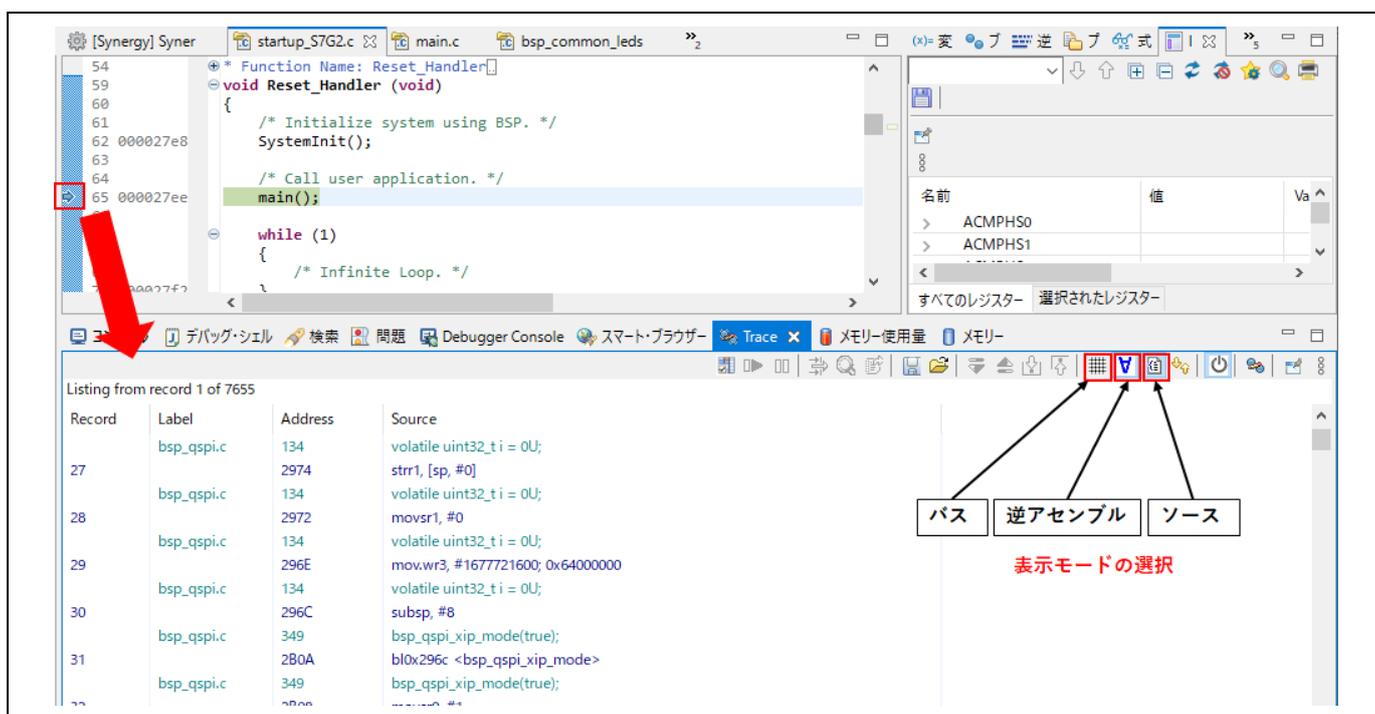


図 5-23 デバッグ – [トレース] ビューの表示モード選択

(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。

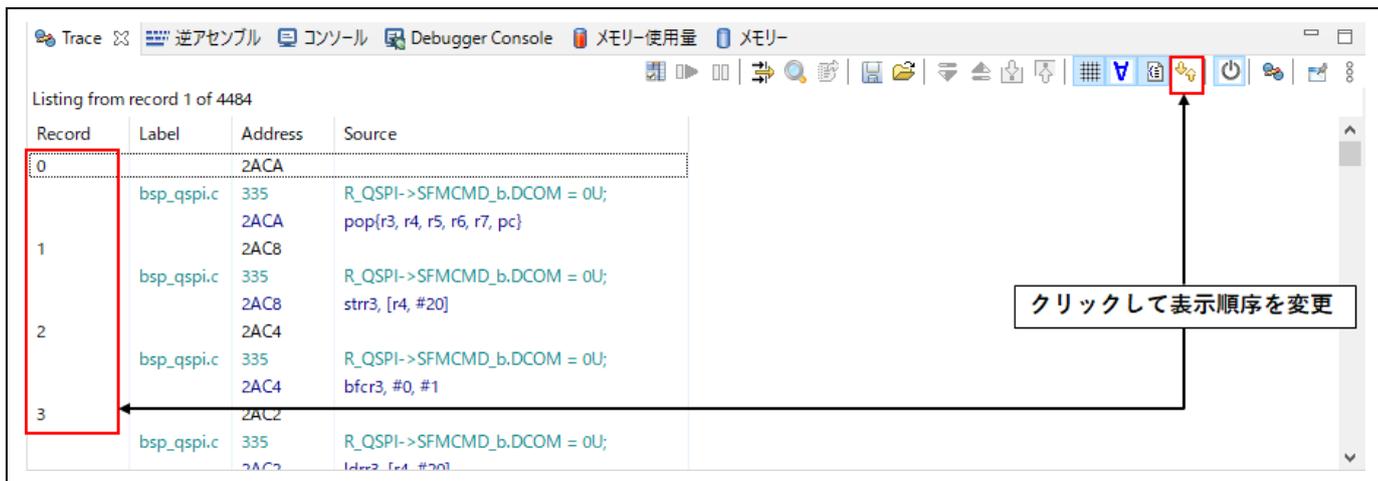


図 5-24 表示順の変更

(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。

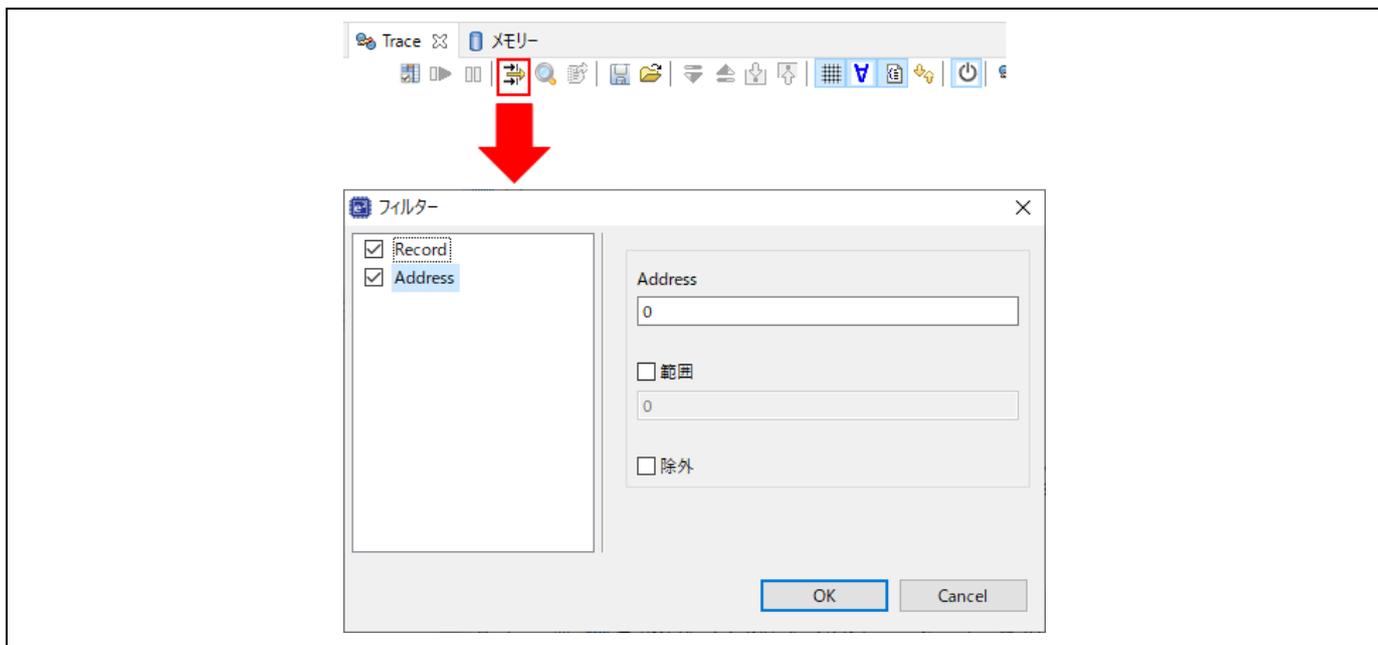


図 5-25 デバッグ - トレース結果のフィルタ

(7) トレース結果は .csv ファイルに保存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。

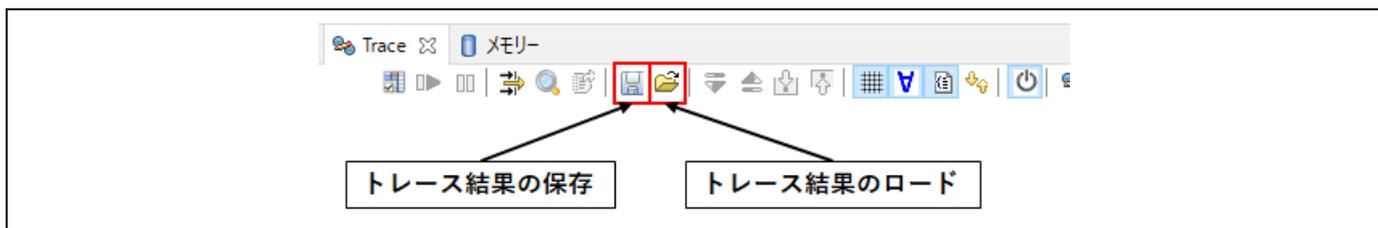


図 5-26 デバッグ - トレース結果の保存とロード

### 5.3.11 Fault Status ビュー

このビューは、ハードウェアフォルト発生時に、fault status レジスタのビット値とその他の主なレジスタの値を表示します。ハードウェアフォルトが発生すると、フォルト要因に関するレジスタのビット値をチェックして表示し、また r0、r1、r2、r3、r12、lr、pc、psr レジスタの値を表示します。図 5-28 にその画面を示します。この機能は e<sup>2</sup> studio v5.2 以上でサポートしています。

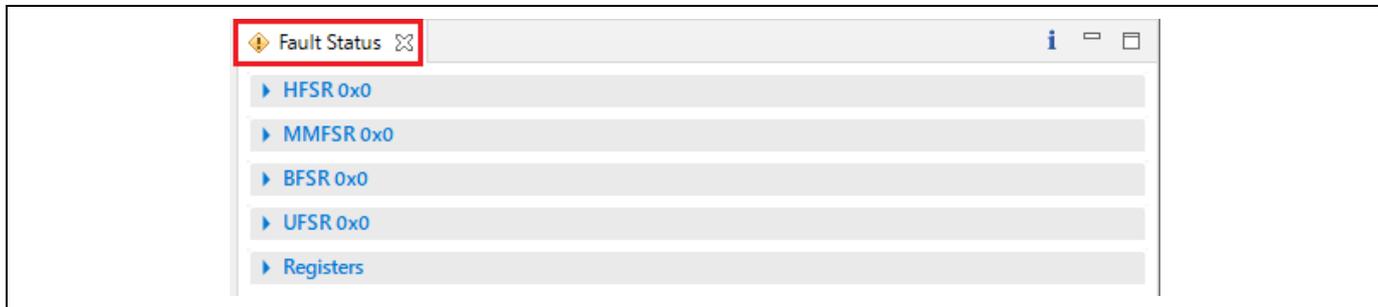


図 5-27 [Fault Status] ビュー – ハードウェアフォルト無し

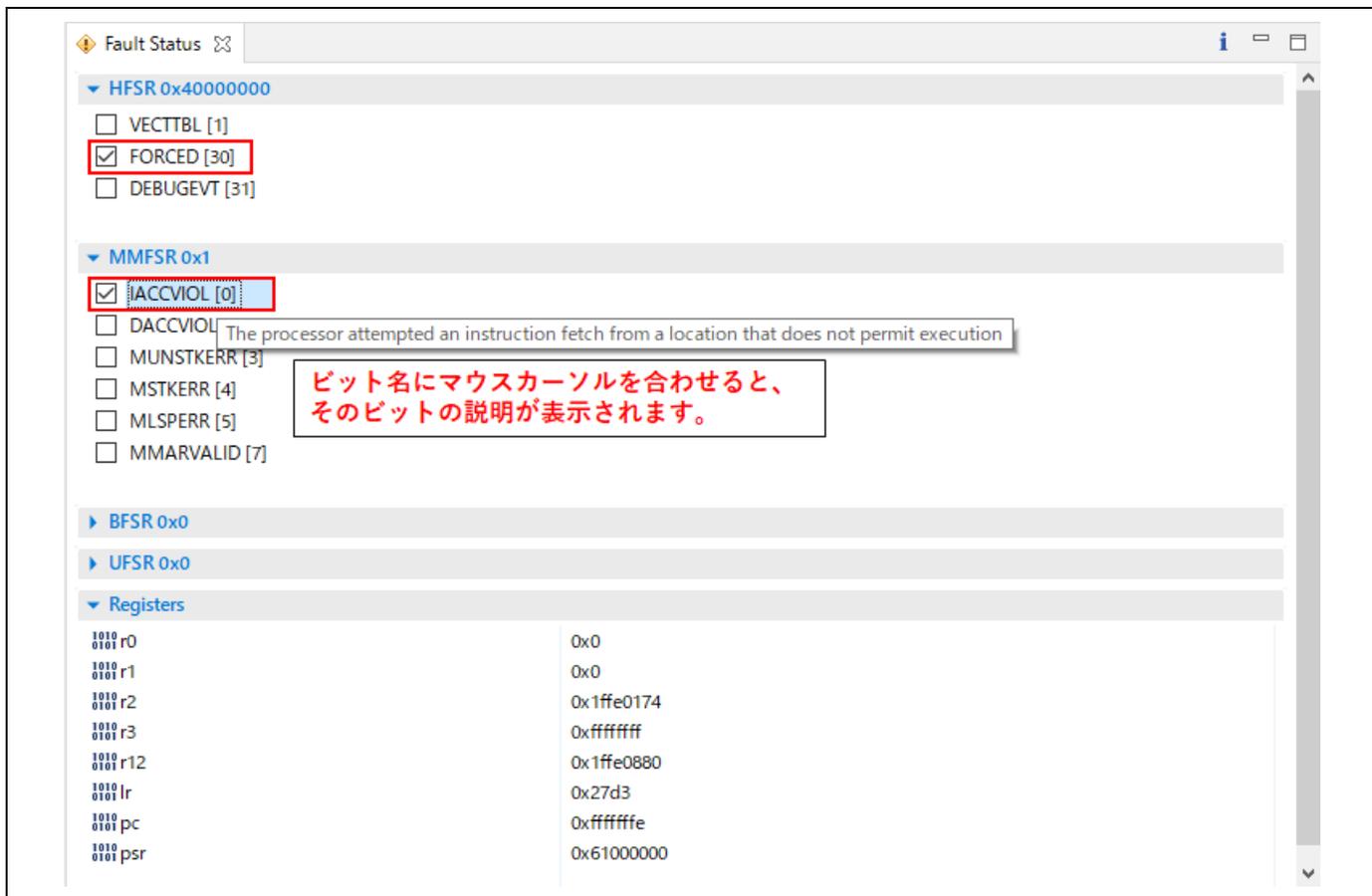


図 5-28 [Fault Status] ビュー – ハードウェアフォルト発生

### 5.3.12 Run Break Timer

Run Break Timer 機能は、直前のプログラム実行のパフォーマンスをステータスバーに表示します。プログラム実行の停止後、現在のプログラムカウンタ（PC）値、直前の実行時間と CPU サイクル、精度または測定の方法を確認できます。

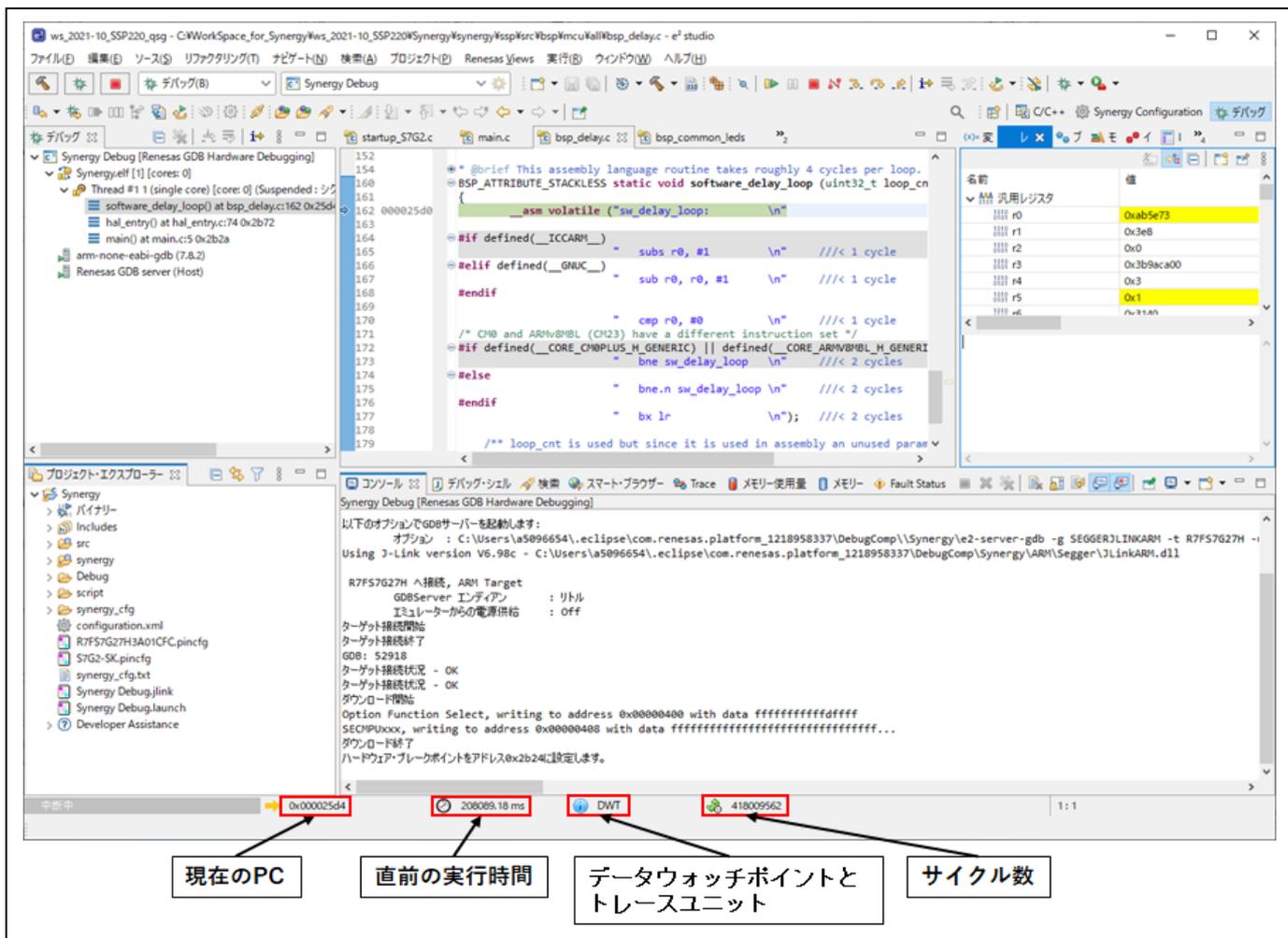


図 5-29 Run Break Timer による直前のプログラム実行のパフォーマンス表示

Synergy デバイスごとの Run Break Timer 機能のサポート状況を以下に示します。

表 5-2 Run Break Timer のサポート状況

デバイス	デバッガ	サポート状況
Synergy S1 シリーズ (Cortex-M0+, Cortex-M23)	J-Link	System Time
Synergy S3, Synergy S5、 Synergy S7 シリーズ	J-Link	Data Watchpoint and Trace Unit (DWT) — System Time で計算したサイクル数とオーバフロー回数

Run Break Timer は e<sup>2</sup> studio v7.3.0 以上でサポートしています。機能のアップデート情報については、e<sup>2</sup> studio のリリースノート（以下のリンク）を参照してください。

<https://www.renesas.com/jp/ja/software-tool/e-studio> の「e<sup>2</sup> studio xxxx-xx Release Note」（xxxx-xx はバージョンを表しています）

## 6. ThreadX アプリケーションの設定

この章では、ThreadX オブジェクトと汎用 PWM タイマ（GPT）モジュールを用いる Synergy プロジェクトを、プロジェクトテンプレート “Blinky with ThreadX” を使用して作成する例を説明します。

### 6.1 ThreadX での汎用 PWM タイマ使用例

Synergy プロジェクトの “Blinky with ThreadX” テンプレートでは、LED 状態を切り替える前にタスクをスリープ状態にすることで LED の点滅を行ないます。

ここで説明する例では、スリープによるディレイを入れるかわりに、Blinky スレッドをセマフォ待ち状態にし、GPT のタイマ割り込みで 1 秒間隔でセマフォを解放し、これによりスレッドの実行を再開します。

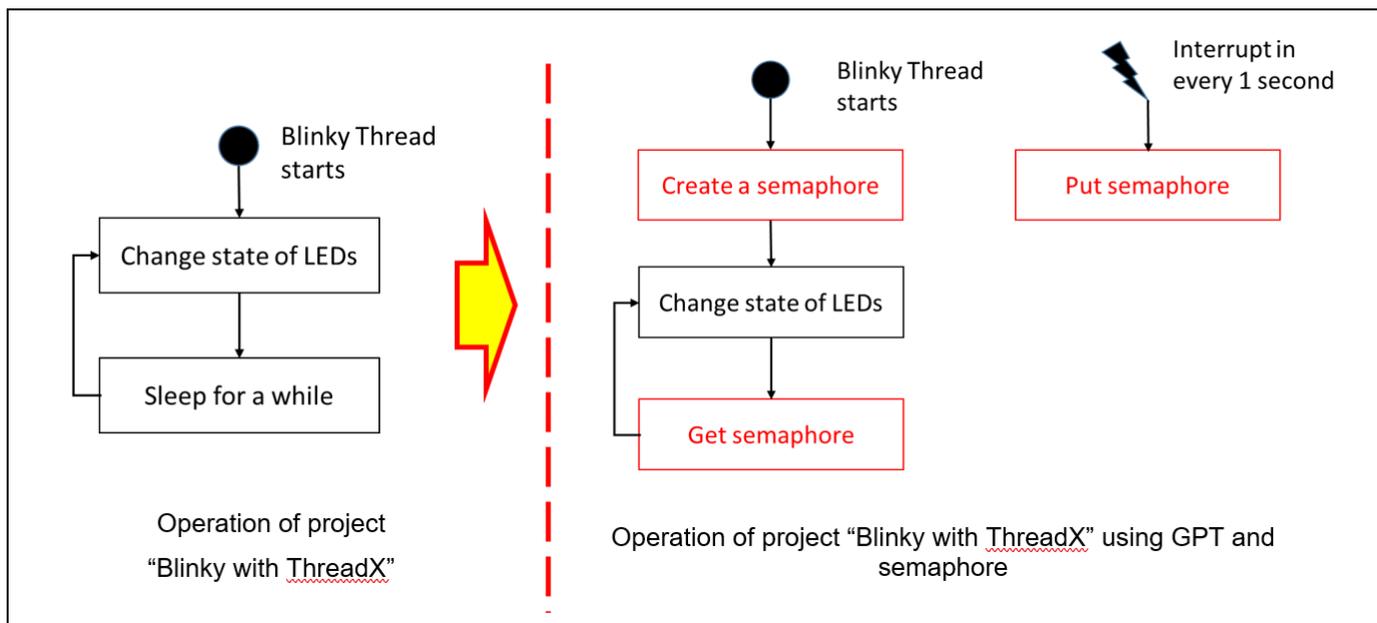


図 6-1 ThreadX アプリケーションの設定 - 概要

## 6.2 サンプルプロジェクトの作成

GPT とセマフォを使用する ThreadX サンプルプロジェクトを作成するには、以下の手順で Synergy プロジェクトを構成してください。

- (1) 新規プロジェクト作成用ウィザードを立ち上げ、「3.1 新規 Synergy プロジェクトの作成」の手順に従って新規プロジェクトを作成します。ただし、最後の [Project Template Selection] 画面では “Blinky with ThreadX” を選択してください。

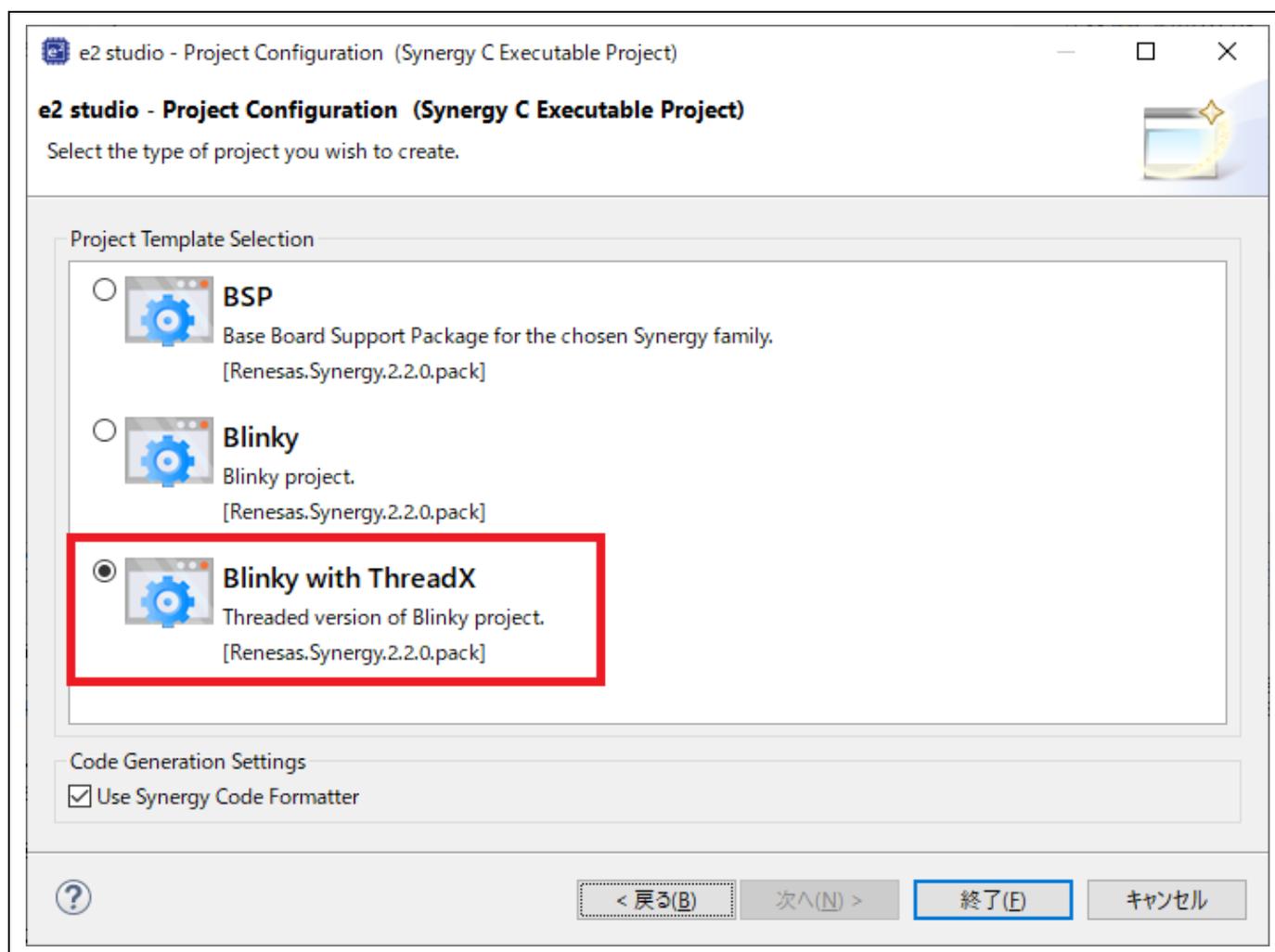


図 6-2 ThreadX アプリケーションの設定 - Blinky with ThreadX テンプレートの選択

- (2) [Synergy プロジェクトコンフィグレーションエディタ] ビューの [Threads Configuration] ページを開きます。「3.4.5 スレッド構成 (Threads Configuration) ページ」を参照してください。

- (3) [Threads] ペインで “Blinky Thread” を選択し、[Stacks] ペインで [New Stack] → [Driver] → [Timers] → [Timer Driver on r\_gpt] の順に選択して、Blinky スレッドに GPT モジュールを追加します。

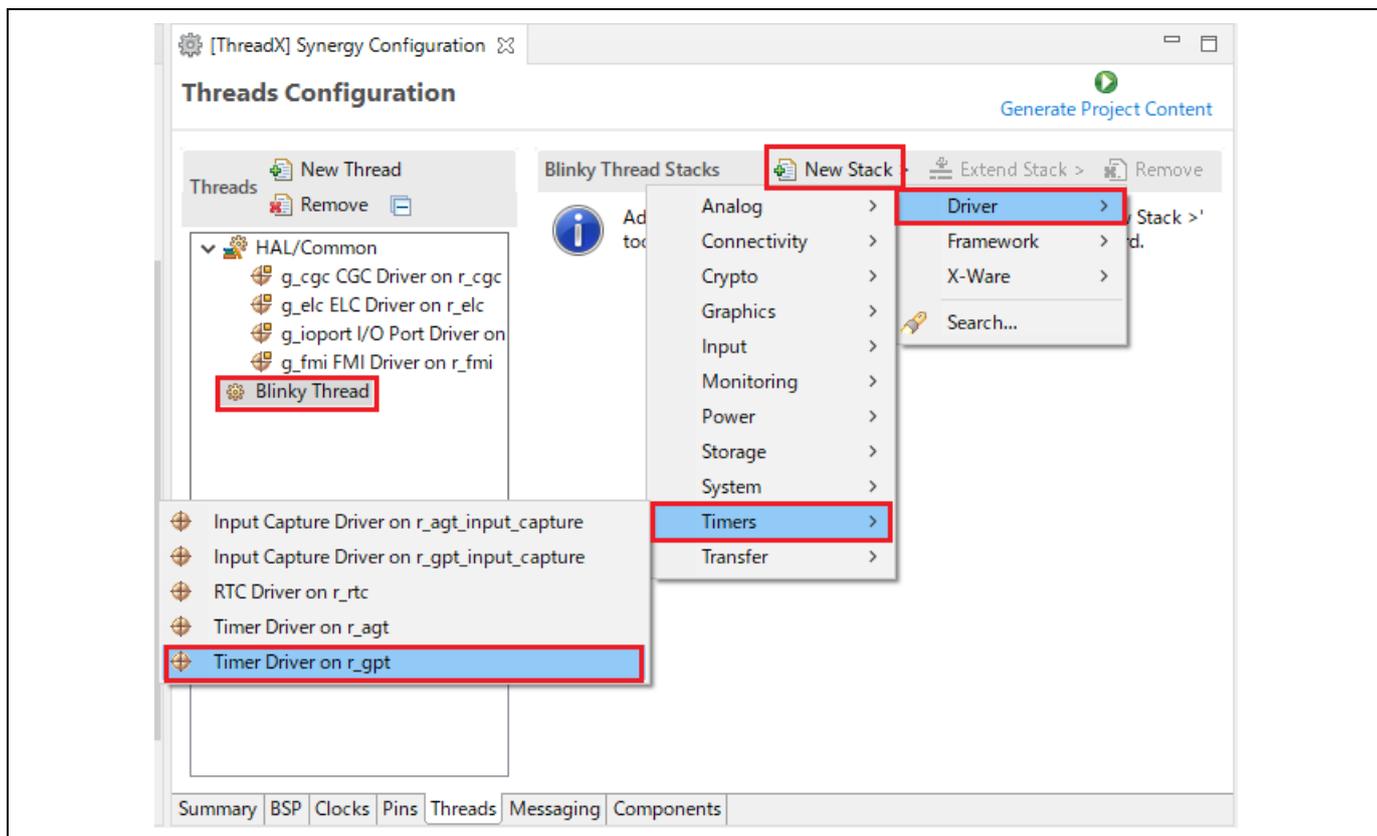


図 6-3 ThreadX アプリケーションの設定 - GPT モジュールの追加

(4) GPT モジュールを次のように設定します。

- Name : g\_timer
- Mode : Periodic
- Period Value : 1
- Period Unit : Seconds
- Callback : gpt\_callback
- Overflow Interrupt Priority : Priority 2

The screenshot shows the Synergy Configuration tool interface. The top window is titled "Threads Configuration" and displays a tree view of threads. Under "Blinky Thread", the "g\_timer Timer Driver on r\_gpt" is highlighted with a red box. Below this, the "g\_timer0 Timer Driver on r\_gpt Objects" section is visible. The bottom window shows the "g\_timer0 Timer Driver on r\_gpt" settings table, with several rows highlighted in red to match the configuration steps in the text above.

Parameter	Value
Name	g_timer
Channel	0
Mode	Periodic
Duty Cycle Range (only applicable)	Shortest: 2 PCLK, Longest: (Period - 1) PCLK
Period Value	1
Period Unit	Seconds
Duty Cycle Value	50
Duty Cycle Unit	Unit Raw Counts
Auto Start	True
GTIOCA Output Enabled	False
GTIOCA Stop Level	Pin Level Low
GTIOCB Output Enabled	False
GTIOCB Stop Level	Pin Level Low
Callback	gpt_callback
Overflow Interrupt Priority	Priority 2

図 6-4 ThreadX アプリケーションの設定 - GPT モジュールの設定

- (5) [Threads] ペインで “Blinky Thread” を選択し、[Objects] ペインで [New Object] → [Semaphore] の順に選択して、Blinky スレッドにセマフォオブジェクトを追加します。

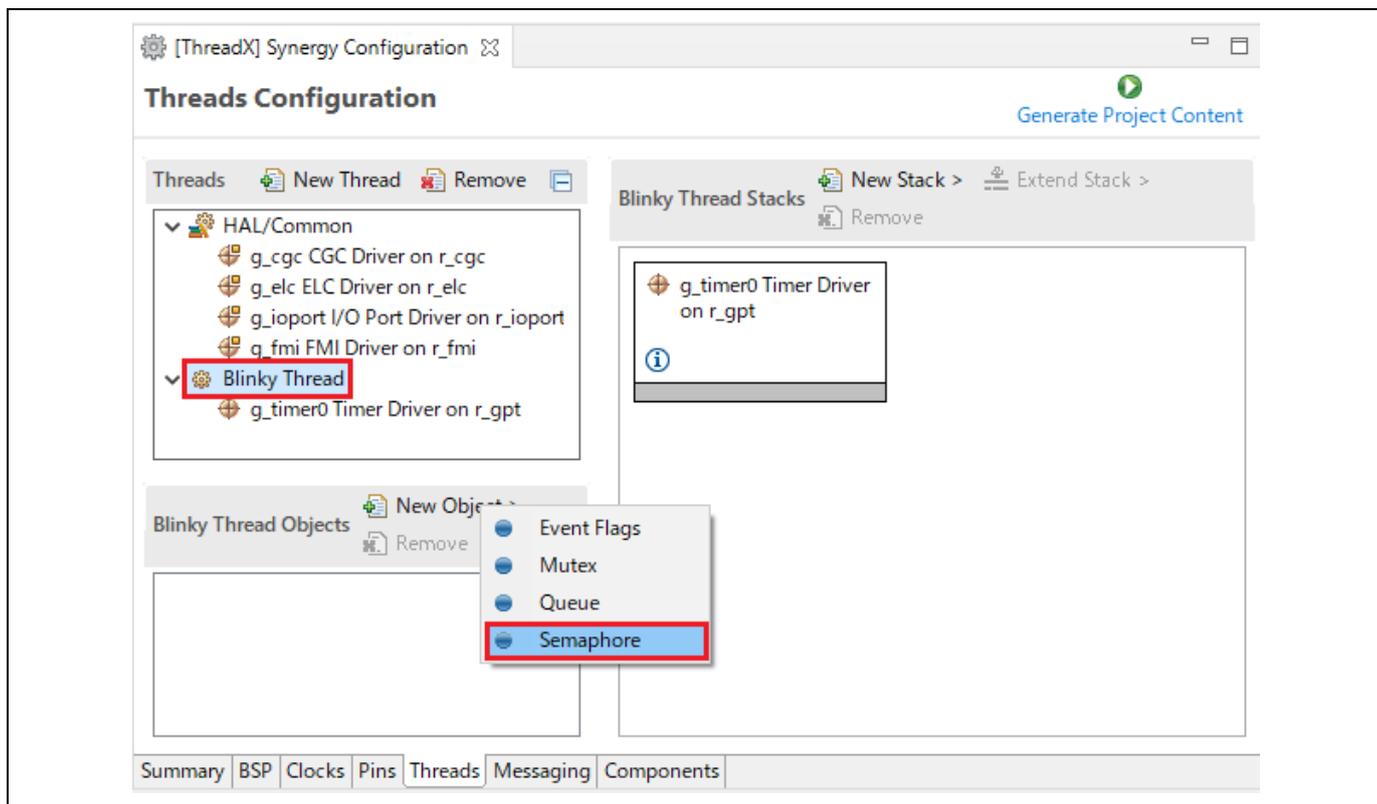


図 6-5 ThreadX アプリケーションの設定 – セマフォオブジェクトの追加

(6) 作成したセマフォを次のように設定します。

- Name : Blinky Semaphore
- Symbol : g\_blinky\_semaphore
- Initial count : 0

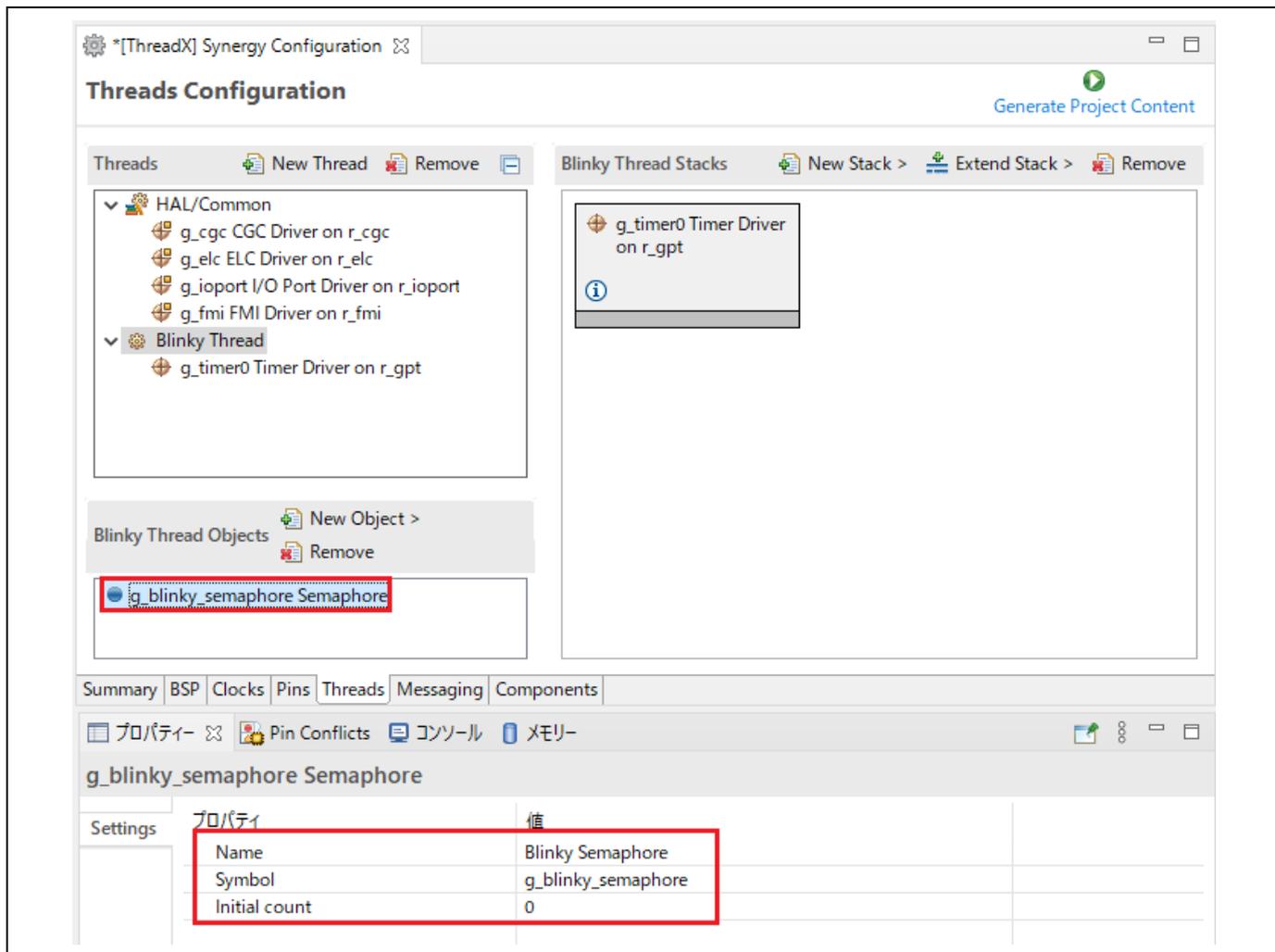


図 6-6 ThreadX アプリケーションの設定 – セマフォオブジェクトの設定

(7) [Ctrl + S] キーで設定を保存し、Generate Project Content ボタンでソースコードを生成します。

(8) “blinky\_thread\_entry.c” ファイルを開き、次のように変更します。

- blinky\_thread\_entry() 関数の “while(1)” ループの前に、GPT モジュール初期化のコードを追加します。  
`g_timer.p_api->open(g_timer.p_ctrl, g_timer.p_cfg);`
- blinky\_thread\_entry() 関数のスレッドスリープ命令を削除し、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);`
- Blinky スレッドにセマフォを通知する関数 gpt\_callback() を追加します。  

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

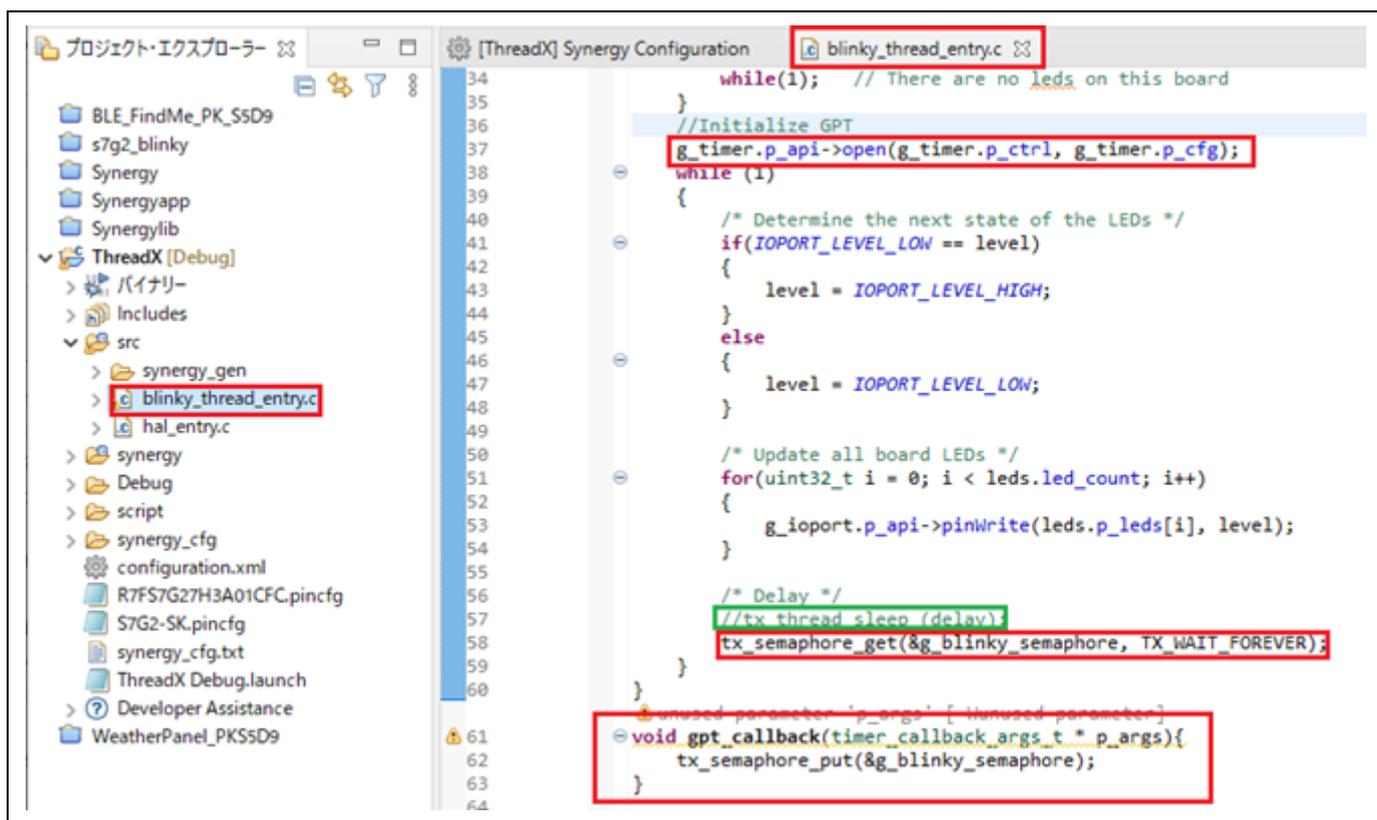


図 6-7 ThreadX アプリケーションの設定 - ユーザソースコードの追加

- (9) プロジェクトをビルドして、Synergy S7G2 ボードで実行します。1 秒間隔で LED が点滅することを確認してください。

## 7. 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 e<sup>2</sup> 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。

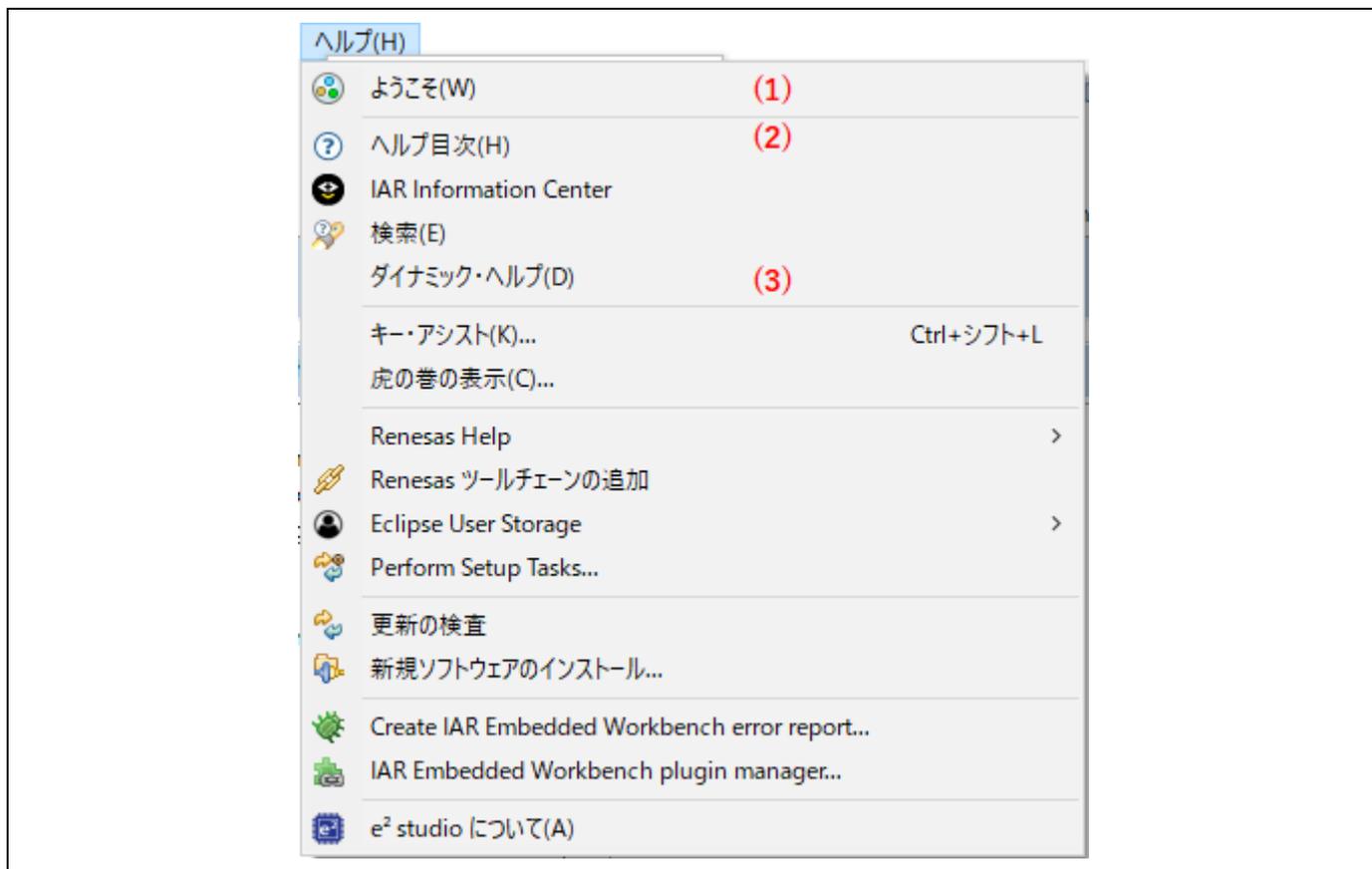


図 7-1 ヘルプメニュー

ヘルプメニューの機能：

(1) [ようこそ] をクリックすると、e<sup>2</sup> studio の概要とリリースノートを表示します。

(2) [ヘルプ目次] をクリックすると、新たにヘルプウィンドウが開きヘルプを検索できます。

(3) [ダイナミック・ヘルプ] をクリックすると、ワークベンチにヘルプ画面を開きます。

[ヘルプ目次] 内には多くの有用なコンテンツが入っています。

- “デバッグに関する機能” にはデバッガの設定、ブレークポイントの使用可能数、などが書かれています。

[ヘルプ] メニューから [ヘルプ目次] → [e<sup>2</sup> studio ユーザガイド] → [デバッグに関する機能] の順に選択してください。

- “Synergy トピックス” には Synergy プロジェクトの作成方法、Synergy コンフィグレーションエディタの使い方、FAQ、などが書かれています。

[ヘルプ] メニューから [ヘルプ目次] → [Synergy トピックス] の順に選択してください。

改訂記録	Renesas Synergy™ e <sup>2</sup> studio 2021-10 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	2021.11.12	—	初版発行
1.10	2022.5.31		e <sup>2</sup> studio と FSP のバージョンアップに伴う更新。 「2.4 Synergy ライセンスの確認と登録」の削除。（SSP2.2.0 以上はライセンス不要）
		7	Quick Install では日本語フォントがインストールされない注記を追加。
		18	「3.1 新規 Synergy プロジェクトの作成」の更新。
		95	「7. ヘルプ」メニューの更新。

---

Renesas Synergy™ e<sup>2</sup> studio 2021-10 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 2022 年 5 月 31 日 Rev.1.10

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)

---

統合開発環境 e<sup>2</sup> studio 2021-10 以上  
ユーザーズマニュアル：クイックスタートガイド

Renesas Synergy<sup>TM</sup>



ルネサスエレクトロニクス株式会社

R20UT5036JJ0110