

## Renesas RA ファミリ

### e<sup>2</sup> studio から IAR または Keil へのアプリケーションコンバート

#### 要旨

ルネサス RA マイコンを使用して製品を開発するお客様は、さまざまな統合開発環境を選択することができます。本アプリケーションノートでは、Renesas e<sup>2</sup> studio IDE のプロジェクトを参考にしながら、IAR Embedded Workbench for ARM や Keil Microcontroller Development Kit - ARM などの IDE でプロジェクトを作成するためのガイドラインを提供します。以下、このプロセスを、「IDE 移行」と呼びます。このガイドラインは、Renesas ホームページで公開されているほとんどのサンプルプロジェクトやアプリケーションプロジェクトに適用されます。このドキュメントで推奨されている手順を完了したら、移行後のプロジェクトと元の e<sup>2</sup> studio プロジェクトでベースとなるコンパイラとオプション設定が異なるため、RA MCU で実行した際の十分な動作比較のテストを行うことを推奨します。

#### 必要なリソース

##### 開発ツールとソフトウェア

- e<sup>2</sup> studio IDE v2021-10 以降([link](#))
- RA スマート・コンフィグレータ (RASC) v2021-10 またはそれ以降([link](#))
- IAR Embedded Workbench for ARM v8.50.9 以降([link](#))
- Keil MDK v5.34 以降([link](#))
- Flexible Software Package v3.1.0 以降([link](#))

#### 前提条件と対象ユーザ

本アプリケーションノートでは、Renesas e<sup>2</sup> studio IDE および RA Flexible Software Package (FSP) の使用経験があることを前提に説明しています。本アプリケーションノートの手順を実行する前に、FSP ユーザーズマニュアルの手順に従って、Blinky プロジェクトをビルドおよび実行してください。そうすることで、e<sup>2</sup> studio と FSP を理解し、ボードへのデバッグ接続が正しく機能することも確認できるようになります。

本アプリケーションノートの対象ユーザは、e<sup>2</sup> studio と GCC 用に一般公開されているサンプルプロジェクトとアプリケーションプロジェクトを、他の IDE やコンパイラ用に拡張したいユーザです。

注： e<sup>2</sup> studio および FSP を初めて使用する場合、システムに e<sup>2</sup> studio および FSP をインストールして Blinky プロジェクトを実行し、e<sup>2</sup> studio および FSP 開発環境に慣れてから次のセクションに進むことを強く推奨します。

#### 前提条件

- 公開済みの Renesas Flexible Software Package の最新ドキュメントへのアクセス
- e<sup>2</sup> studio と組み込み(またはスタンドアロンの)RA コンフィグレータの操作に関する事前知識
- IAR Embedded Workbench と IAR コンパイラの操作に関する事前知識
- Keil MDK と ARM コンパイラの操作に関する事前知識

## 目次

|                                                        |    |
|--------------------------------------------------------|----|
| 1. 概要 .....                                            | 3  |
| 1.1 RASC のインストール.....                                  | 3  |
| 1.2 FSP のインストール.....                                   | 5  |
| 2. e <sup>2</sup> studio プロジェクトの IAREW/IARCC への移行..... | 6  |
| 2.1 IAR IDE の構成とコンポーネントのインストール.....                    | 6  |
| 2.1.1 IAR のパッチインストール.....                              | 7  |
| 2.2 IAR 用の空のテンプレートプロジェクトの作成 .....                      | 8  |
| 2.2.1 IAR で空のテンプレートプロジェクトを作成する手順 .....                 | 8  |
| 2.3 ソースコードの追加 .....                                    | 10 |
| 2.3.1 src ファイルの追加.....                                 | 10 |
| 2.3.2 src ファイル追加後の設定とプロジェクト生成 .....                    | 10 |
| 2.4 プロジェクトの設定とコンパイル .....                              | 12 |
| 2.4.1 コンパイラオプションの設定.....                               | 12 |
| 2.4.2 プロジェクトのビルド .....                                 | 14 |
| 2.4.3 ビルド操作での警告とエラーへの対処 .....                          | 16 |
| 2.5 アプリケーションのダウンロードとデバッグ .....                         | 16 |
| 2.5.1 DFP のインストール .....                                | 17 |
| 2.5.2 Keil 用の空のテンプレートプロジェクトの作成 .....                   | 19 |
| 2.5.3 ソースコードの追加 .....                                  | 21 |
| 2.5.4 プロジェクトのコンパイル .....                               | 25 |
| 2.5.5 プロジェクトのダウンロードとデバッグ .....                         | 29 |

## 1. 概要

IDE 移行のプロセスは、IAR EW と Keil MDK の両方に共通に必要なステップを踏むことで行われます。これらのステップは、次のセクションで詳細に説明されます。

1. IDE の構成とコンポーネントのインストール
  1. RASC のインストールと登録
  2. FSP のインストール
  3. パッチ/DFP のインストール
2. 空のテンプレートプロジェクトの作成
  1. 空のプロジェクトの作成
3. ソースコードを追加
  1. ソースファイルを追加
  2. FSP 自動生成ファイルの生成
4. バイナリのコンパイル
  1. コンパイラオプションを調整します(インクルードパス、コンパイラフラグ、リンカーフラグなど)
  2. プロジェクトのビルド
  3. 警告およびエラーへの対処
5. バイナリのダウンロードとデバッグ
  1. デバッガのセットアップ
  2. ダウンロードの確認

### 1.1 RASC のインストール

入手可能な RASC 実行ファイルは、直接実行できます。以下の手順に従ってインストールしてください。

1. RASC をインストールしてセットアップを開始したら、別の場所にインストールする場合は、[All Users]を選択する必要があります。[Current user]を選択すると、別の場所にインストールできない場合があります。

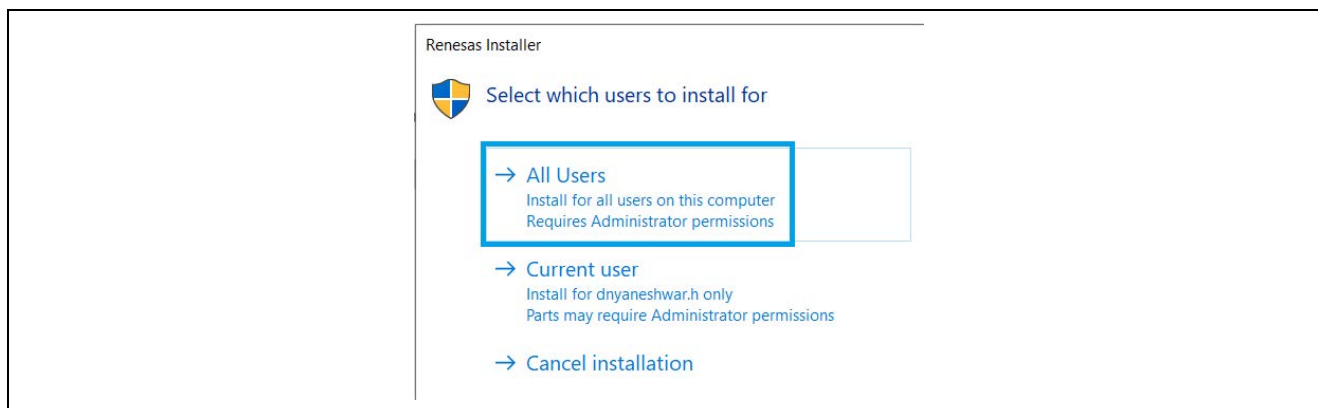


図 1 すべてのユーザを選択

2. [インストール(別の場所にインストールします)]を選択して、新しいRASC用に分割したインスタンスを作成します。次に、[Next]をクリックしてインストールを開始します。

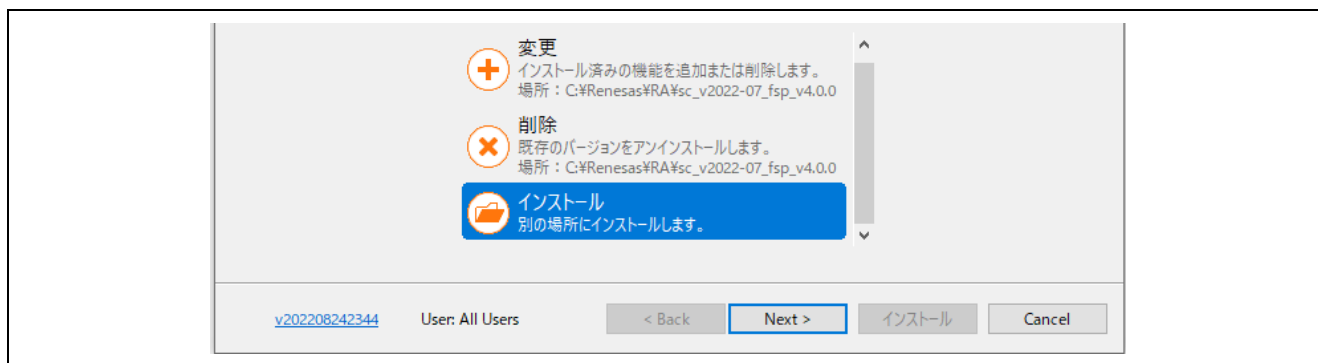


図2 インストールを選択

このプロセスでは、インストールを完了するために必要な前提条件が確認されます。デフォルトでは、C:\Renesas\RA\に分割したインストールフォルダが作成されますが、これは任意の場所に変更することができます。

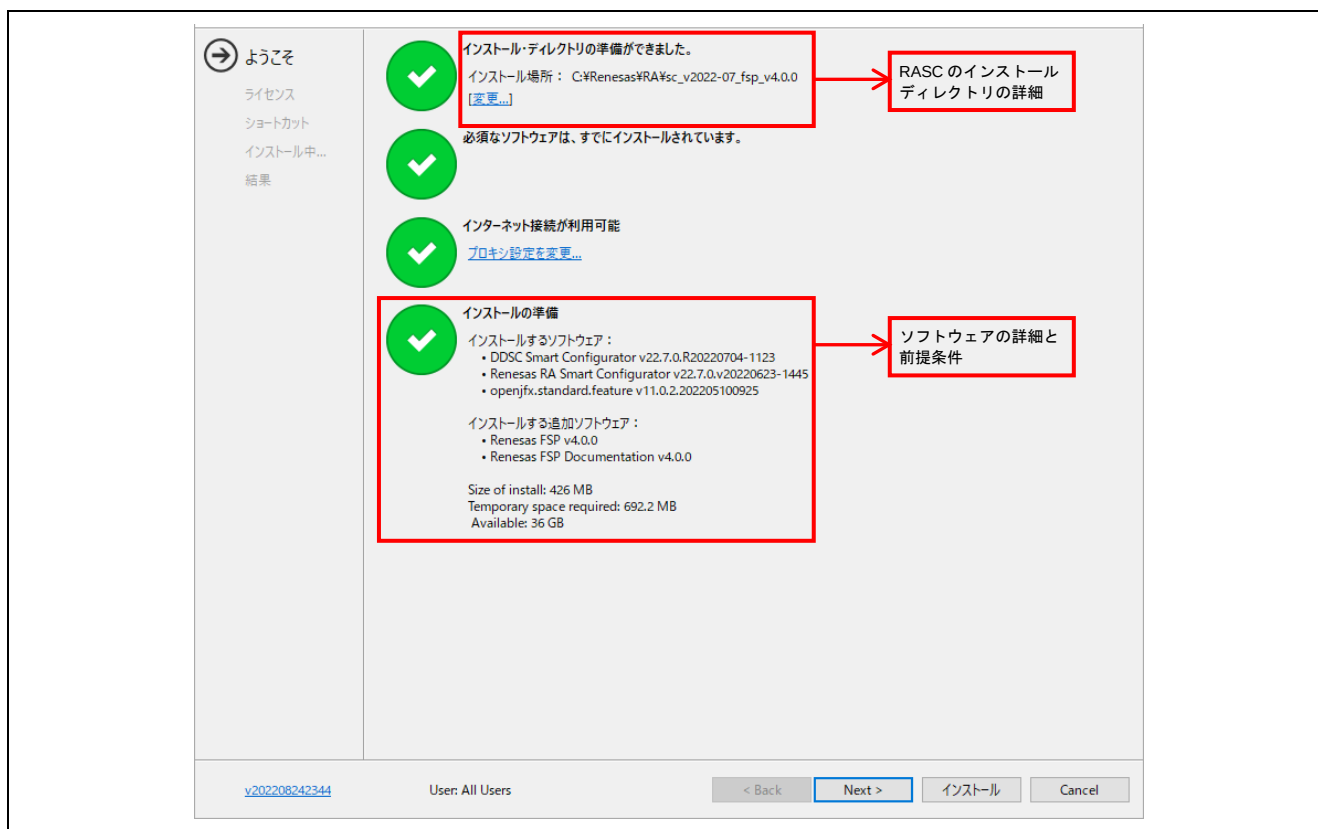


図3 インストールディレクトリの選択

3. [Next]をクリックして、次の手順に進みます。

4. チェックボックスをクリックして利用規約に同意し、[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。以下は、最後に[インストール]をクリックした後の様子です。

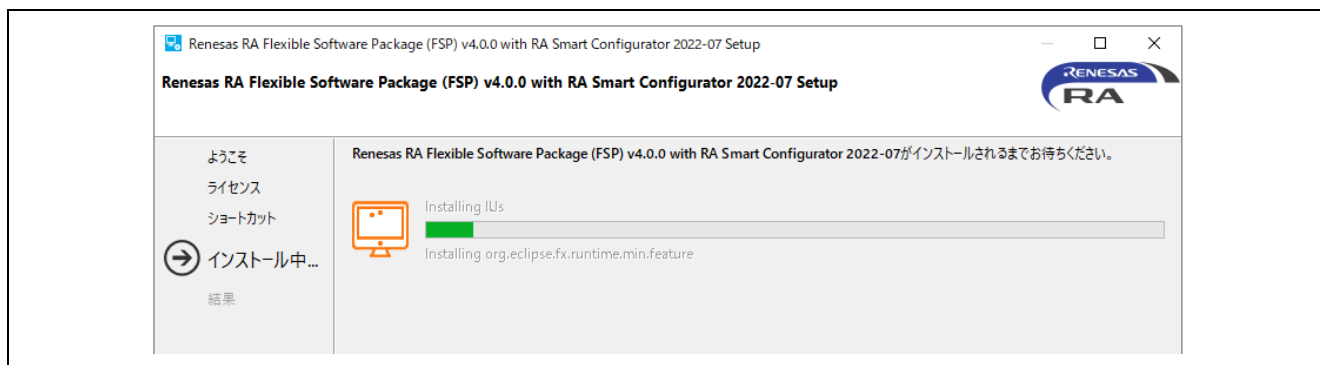


図 4 インストール

5. インストールしたフォルダを開き、目的の FSP バージョンの新しくインストールされた RASC フォルダを確認します。図 5 に示すようなフォルダが作成されます(既定のディレクトリに新しいインスタンスとしてインストールされます)。

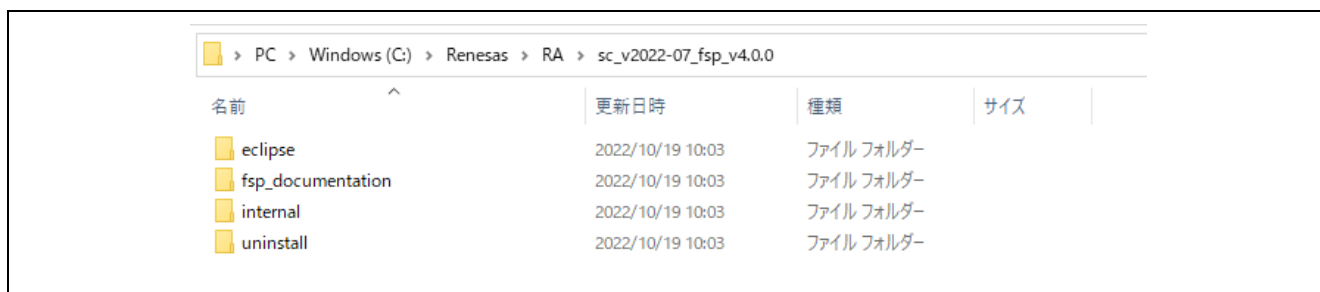


図 5 インストールフォルダ

注： 新規にインストールした pack ファイルは、図 6 のように RASC フォルダの内部ディレクトリ内に存在する必要があります。

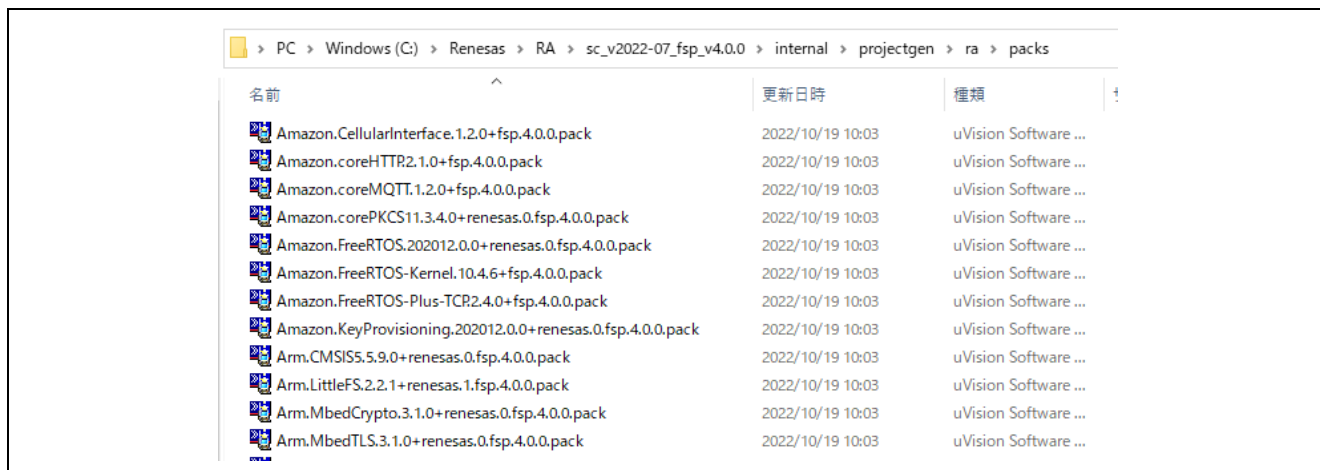


図 6 内部ディレクトリ

## 1.2 FSP のインストール

この手順は、Keil と IAR 両方の IDE のプロジェクト移行に共通しています。

これは、次のような方法で実現できます。

- 新しい RASC は、特定のバージョンのプリインストールの pack ファイルと一緒に直接ダウンロードされます。
- pack ファイルは次のディレクトリにあります。<rasc installation directory>\sc\_v2021-04\_fsp\_v3.2.0\internal\projectgen\ra\packs  
FSP パック(pack ファイル)は、現在インストール済みの RASC パックフォルダ内に直接配置できます。  
<installation directory>\sc\_v2021-04\_fsp\_v3.2.0\internal\projectgen\ra\packs

## 2. e<sup>2</sup> studio プロジェクトの IAREW/IARCC への移行

### 2.1 IAR IDE の構成とコンポーネントのインストール

参考ドキュメントとして、最新の FSP ユーザーズマニュアルの以下のセクションが利用可能です。

FSP User Manual→Starting Development→Using RA Smart Configurator with IAR EWARM

[https://renesas.github.io/fsp/starting\\_de\\_v.html#using-ra-smart-configurator-with-ewarm](https://renesas.github.io/fsp/starting_de_v.html#using-ra-smart-configurator-with-ewarm)

#### IAR IDE と RASC の設定

要件：

- IAR EW for ARM (使用時のバージョン：8.50.9)
- Flex Software Package (FSP) (使用時のバージョン：3.1.0)
- RA スマート・コンフィグレータ(RASC) (使用時のバージョン：3.1.0)

#### インストールされた IAR EW ソフトウェア用に RASC を構成する手順

インストールした IAR Embedded Workbench を開きます。[ツール]→[ツールの設定]に移動します。

1. [ツールの設定]→[新規作成]オプションをクリックすると、以下のウィンドウが表示されます。図 7 に示すように、関連するフィールドを設定します。

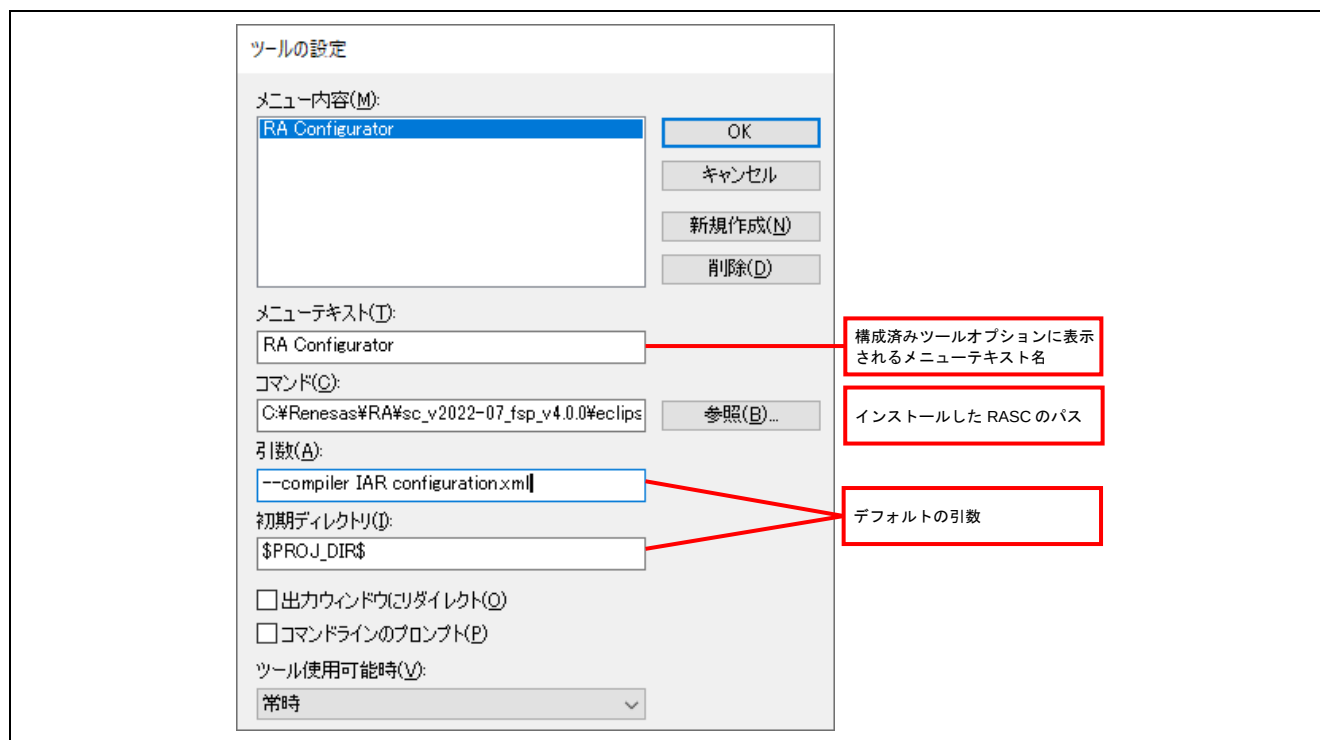


図 7 ツールの設定

2. 以降で説明する内容を、図 7 に示すように[ツールの設定]オプションで設定します。

メニューテキスト：設定済みツールに表示される RA スマート・コンフィグレータツールの名称です。

ツールの名称は、識別のために正しく指定する必要があります。

コマンド：移行に使用するインストール済みの RASC のパスです。

新しい RASC をインストールした場所を参照し、インストールした rasc.exe を選択します。

(例：C:\Renesas\RA\sc\_v2021-04\_fsp\_v3.1.0\eclipse)

引数：--compiler IAR configuration.xml を設定します。このフラグは、RASC が IAR コンパイラと互換性のあるコードを生成するように設定します。

初期ディレクトリ：\$PROJ\_DIR\$を使用します。

注： FSP のバージョンを変更する場合は、バージョンに合わせて[コマンド]フィールドの rasc.exe へのパスを調整します。

例：ここでは FSP v3.1.0 が使用されているため、設定されているパスは

C:\Renesas\RA\sc\_v2021-04\_fsp\_v3.1.0\eclipse\rasc.exe となります。

3. [OK]をクリックし、IDE による RASC の設定を完了します。

これで、設定した RASC を IDE から[ツールの設定][メニューテキスト]フィールドに設定された名称で実行できるようになりました。IAR IDE を起動し、[ツール]→[RA Configurator]をクリックします。RA コンフィグレータは、以下のようにここから実行することができます。

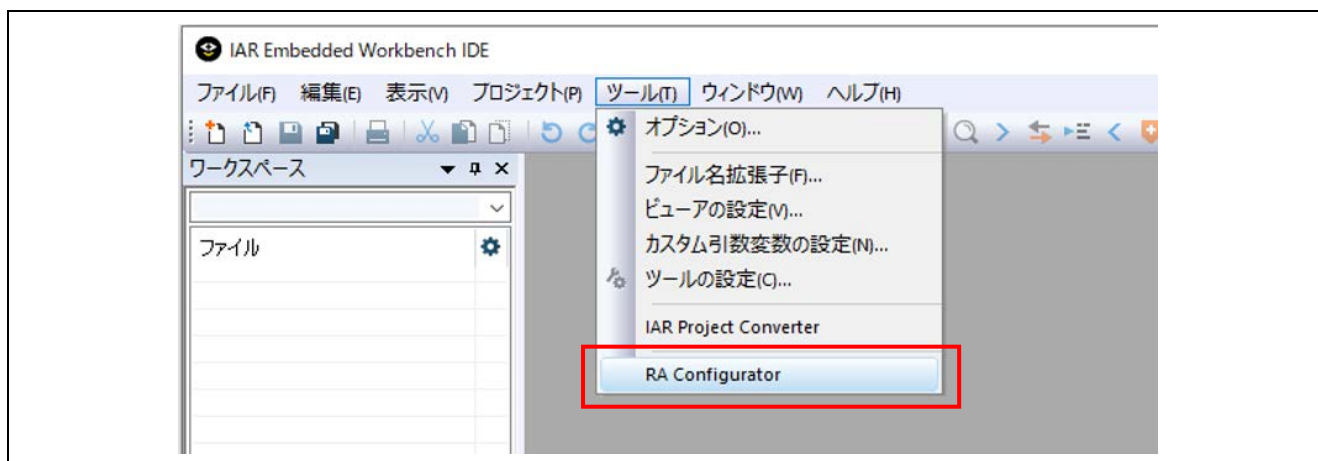


図 8 RA コンフィグレータの実行

### 2.1.1 IAR のパッチインストール

IAR Embedded Workbench に目的の RA MCU のサポートが含まれていない場合、IAR のパッチ適用が必要となる可能性があります。このパッチがないと、選択したマイクロコントローラのプロジェクト生成や操作ができない可能性があります。

パッチへのアクセスには、IAR との通信が推奨されます。アクセスできたら、サポートされているボードのパッチは、インストールされている IAR バージョンの[arm]ディレクトリ内にコピーする必要があります。

デフォルトの場所：C:\Program Files (x86)\IAR Systems\Embedded Workbench 8.4\arm

1. インストールされている IAR ディレクトリから arm ディレクトリを開きます。

2. パッチファイルを解凍し、次の新しいディレクトリ、つまり config, inc, src を取得します。

| 名前     | 更新日時             | 種類        | サイズ |
|--------|------------------|-----------|-----|
| config | 2022/10/19 15:50 | ファイルフォルダー |     |
| inc    | 2022/10/19 15:50 | ファイルフォルダー |     |
| src    | 2022/10/19 15:50 | ファイルフォルダー |     |

図 9 新規ディレクトリ

- 上記のディレクトリをコピーして、既定の場所または IAR がインストールされている arm ディレクトリの場所に貼り付けます。

## 2.2 IAR 用の空のテンプレートプロジェクトの作成

以下は、IAR で新しいプロジェクトを作成するための要件です。

- IAR EW for ARM (使用時のバージョン : 8.50.9)
  - 注 : FSP リリースに関連する最新の利用可能な IDE バージョンは、github で公開されている FSP リリースノートで確認することができます。
- Flex Software Package (FSP) (使用時のバージョン: 3.1.0)
- RA スマート・コンフィグレータ (RASC) (使用時のバージョン: 3.1.0)

### 2.2.1 IAR で空のテンプレートプロジェクトを作成する手順

- 前の章で説明したように pack ファイルをインストールした後、RASC (rasc.exe)を実行します。
- プロジェクトの[Project name]とワークスペースの[Location]を指定します。

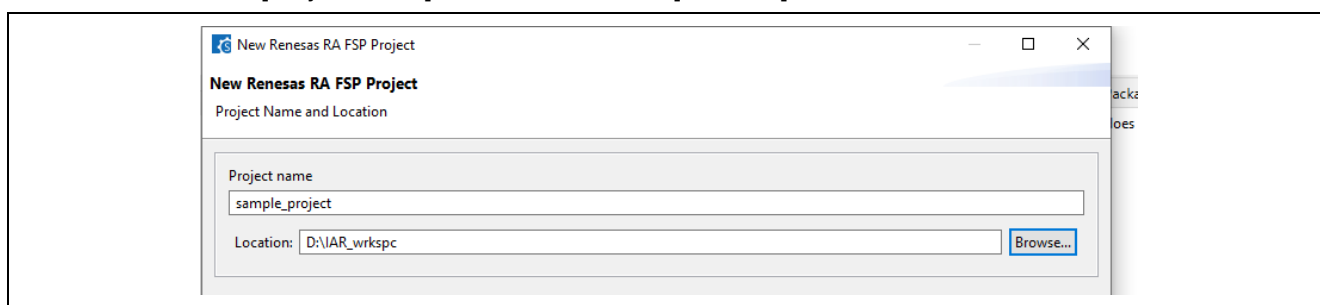


図 10 Project name と Location の設定

- [Next]をクリックします。以下の画面が表示されます。プロジェクトを作成する[FSP version]、[Board]、[Device]、および[IDE]を選択します。

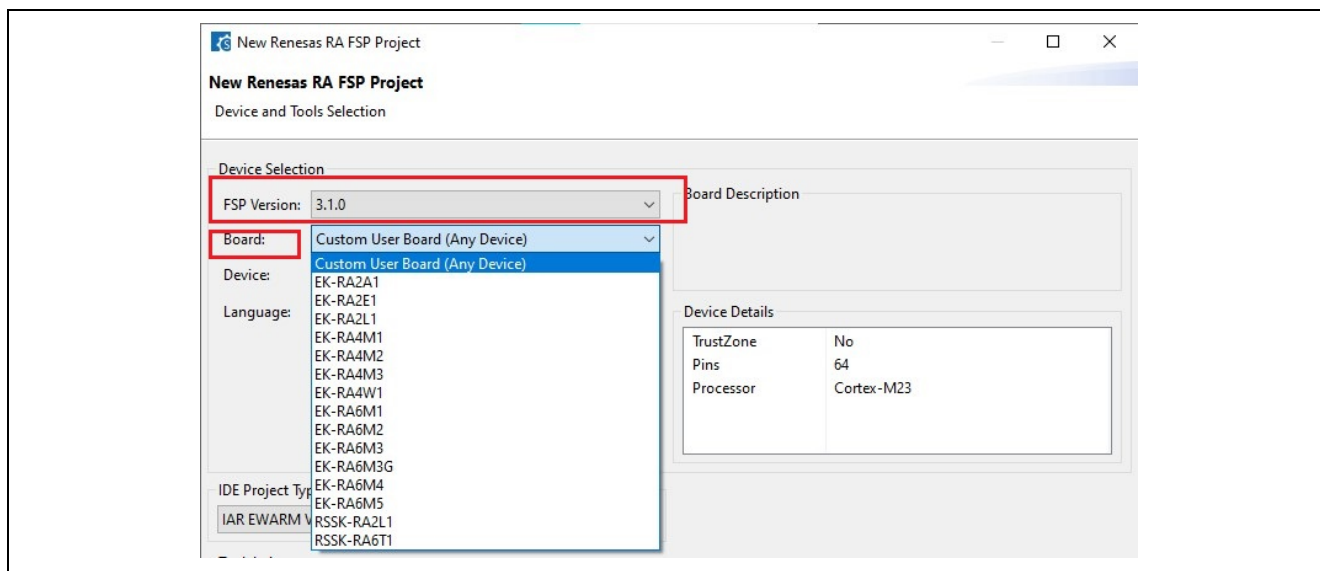


図 11 FSP Version、Board、Device、IDE の設定



4. FSP バージョンのドロップダウンに、利用可能なパッケージが表示されます。パッケージは、RASC のインストールディレクトリの内部フォルダ内に存在する必要があります。
5. デバイスと言語が自動的に「デフォルト」に選択されます。
6. IAR でサンプルプロジェクトを作成するために、[IDE Project type]を[IAR EWARM Version 8]に選択してください。

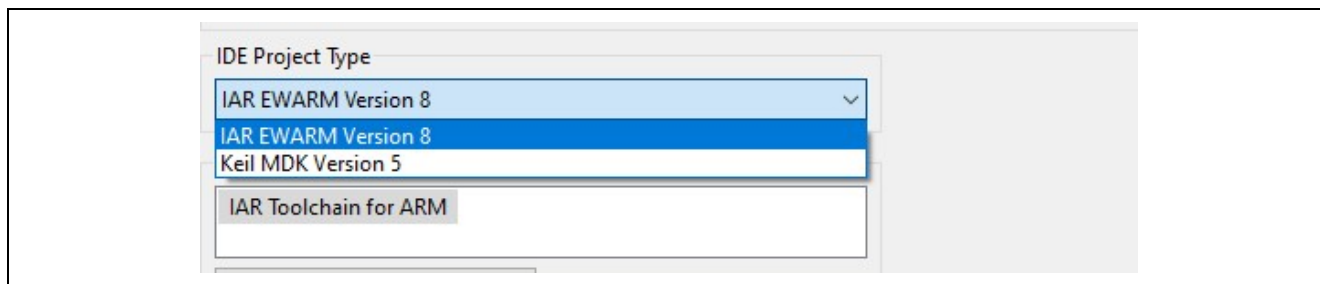


図 12 IDE プロジェクトタイプの設定

7. [Next]をクリックし、[Flat]プロジェクトの作成手順に従います。TrustZone をサポートする MCU (RA4M2、RA4M3、RA6M4、RA6M5)を選択すると、[Flat]、[Secure]、[Non-secure]のいずれかを選択するよう求められます。それ以外の場合は、Flat プロジェクトのみ選択可能です。

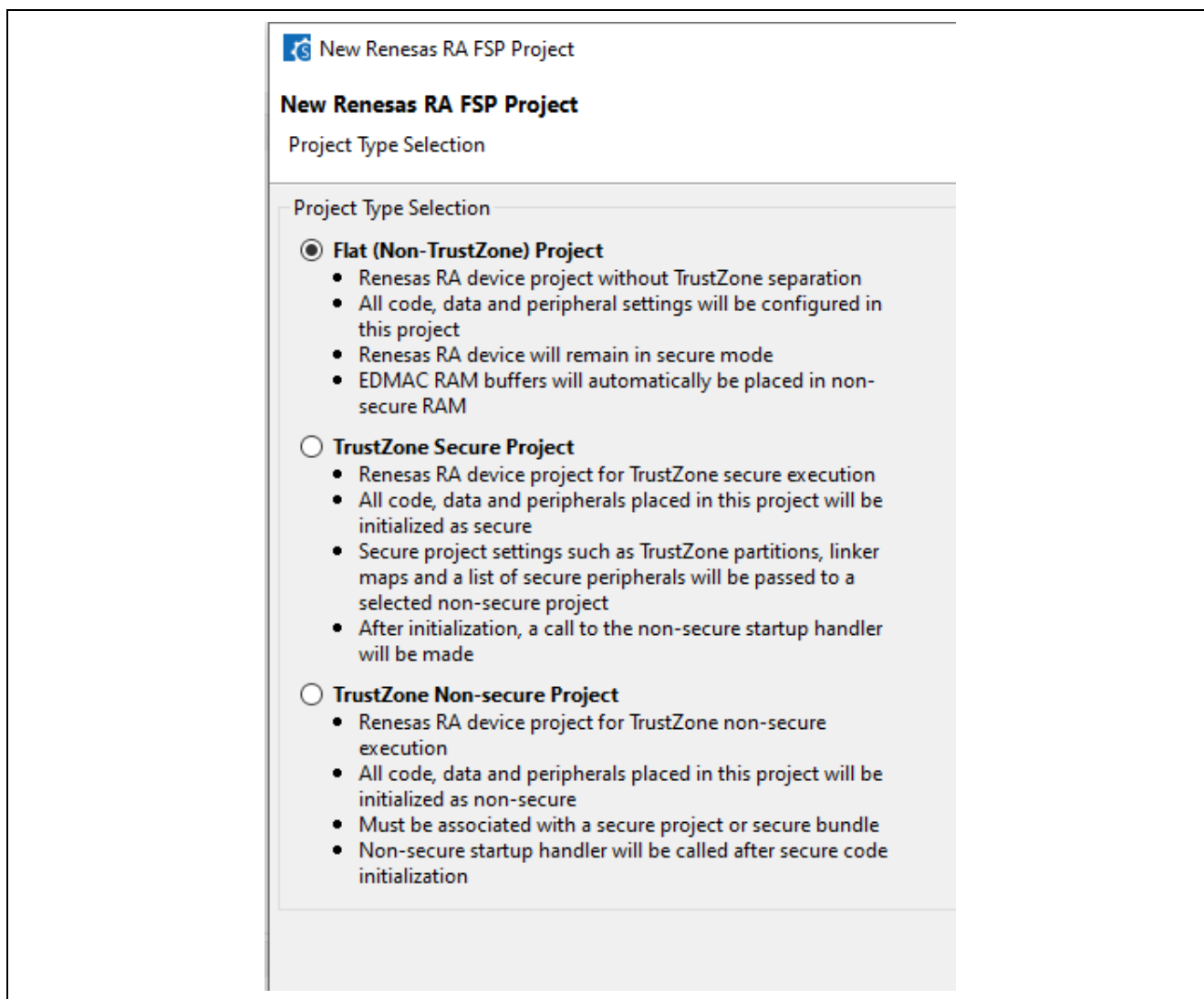


図 13 フラット、セキュア、ノンセキュアオプション

8. (手順 6 で)プロジェクトの種類を選択した後、プロジェクトの要件に応じて RTOS を選択します。デフォルトでは、[No RTOS]に設定されています。[Next]をクリックして次に進みます。

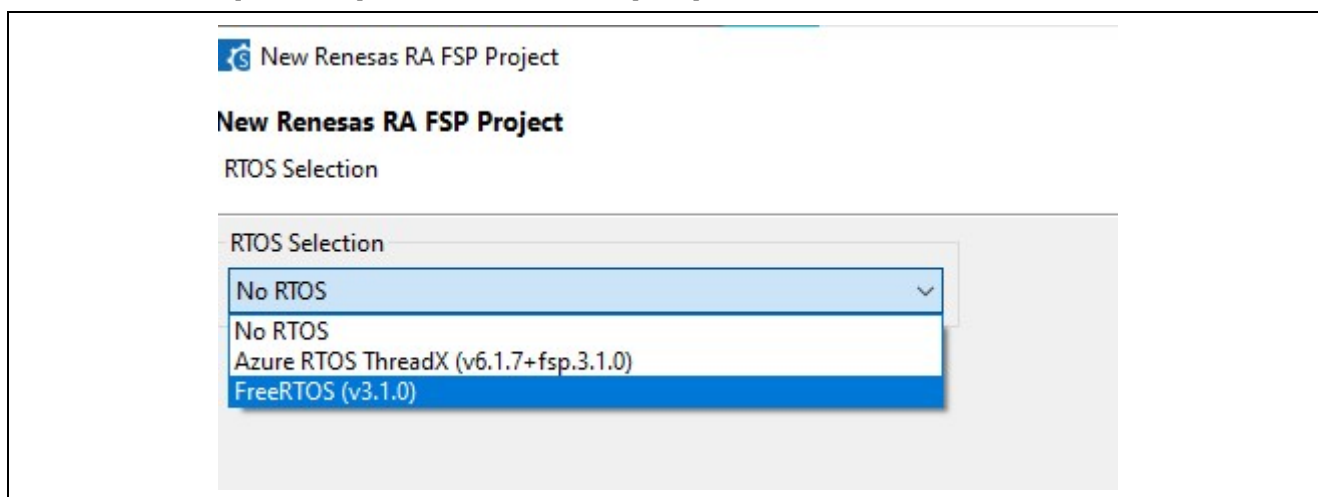


図 14 RTOS 選択

9. [Bare Metal - Minimal]を選択し、プロジェクトサンプルテンプレートを作成します。[Finish]をクリックして、プロジェクトテンプレートの作成を完了します。

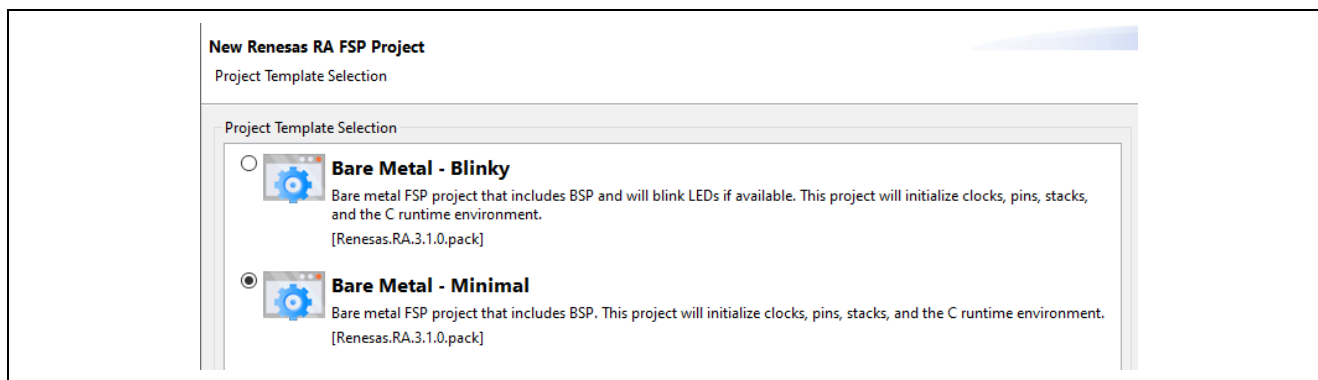


図 15 Bare Metal - Minimal 選択

## 2.3 ソースコードの追加

### 2.3.1 src ファイルの追加

- 上記の手順が正常に完了すると、IAR プロジェクトテンプレートが作成された場所に、既に src フォルダが存在することになります。

このプロジェクトテンプレートの src フォルダは、e<sup>2</sup> studio のプロジェクトフォルダから src フォルダに置き換えることができます。

### 2.3.2 src ファイル追加後の設定とプロジェクト生成

プロジェクトテンプレート作成後、選択したプロジェクトの場所に、必要な全ての初期化ファイルやその他の設定ファイルを含む .eww ファイルとしてプロジェクトが作成されます。

| 名前                   | 更新日時             | 種類                | サイズ    |
|----------------------|------------------|-------------------|--------|
| .settings            | 2022/10/19 17:05 | ファイル フォルダ         |        |
| ra                   | 2022/10/19 17:05 | ファイル フォルダ         |        |
| ra_cfg               | 2022/10/19 17:05 | ファイル フォルダ         |        |
| ra_gen               | 2022/10/19 17:05 | ファイル フォルダ         |        |
| script               | 2022/10/19 17:05 | ファイル フォルダ         |        |
| src                  | 2022/10/19 17:05 | ファイル フォルダ         |        |
| .api_xml             | 2022/10/19 17:05 | API_XML ファイル      | 1 KB   |
| .secure_azone        | 2022/10/19 17:05 | SECURE_AZONE フ... | 2 KB   |
| .secure_xml          | 2022/10/19 17:05 | SECURE_XML ファイル   | 3 KB   |
| buildinfo.ipcf       | 2022/10/19 17:05 | IPCF ファイル         | 10 KB  |
| configuration.xml    | 2022/10/19 17:05 | XML ファイル          | 20 KB  |
| memory_regions.icf   | 2022/10/19 17:05 | ICF ファイル          | 2 KB   |
| R7FA4M1AB3CFRpinconf | 2022/05/02 19:38 | PINCFG ファイル       | 2 KB   |
| sample_project.ewd   | 2022/10/19 17:05 | EWD ファイル          | 103 KB |
| sample_project.ewp   | 2022/10/19 17:05 | EWP ファイル          | 72 KB  |
| sample_project.eww   | 2022/10/19 17:05 | IAR IDE Workspace | 1 KB   |

図 16 プロジェクトファイル

### プロジェクト生成の手順：

1. 上記のように、プロジェクトテンプレートの作成で、<プロジェクト名>.eww ファイルが作成されます。.eww ファイルを実行すると、IAR EW でプロジェクトが開かれます。
2. RASC は IAR IDE で設定されているので、[ツール]オプションをクリックして[RA Configurator]を表示します(前述の IAR EW ソフトウェアをインストールした場合の RASC の設定手順を参照してください)
3. RASC が開いた後、図 17 のように複数のタブとオプションがある画面が表示されます。

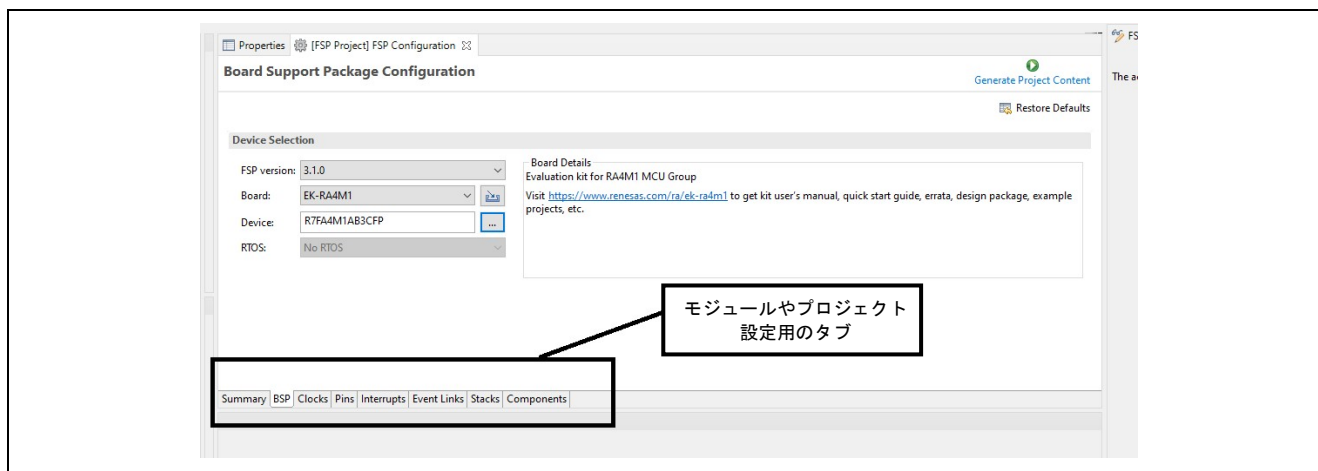


図 17 ボードサポートパッケージの構成

4. RASC のタブバーで必要なタブを選択し、プロジェクトの要件に応じて個々のコンポーネントのプロパティを設定することができます。
5. 通常、プロジェクトを生成する前に、以下のタブでプロジェクトの設定を行います。
  - a. [BSP]タブを選択し、[Properties]ウィンドウで BSP のパラメータを設定します。
  - b. 要件に応じて RASC 構成にスタックを追加します。それぞれのモジュールスタックは、[New Stack] オプションから選択できます。

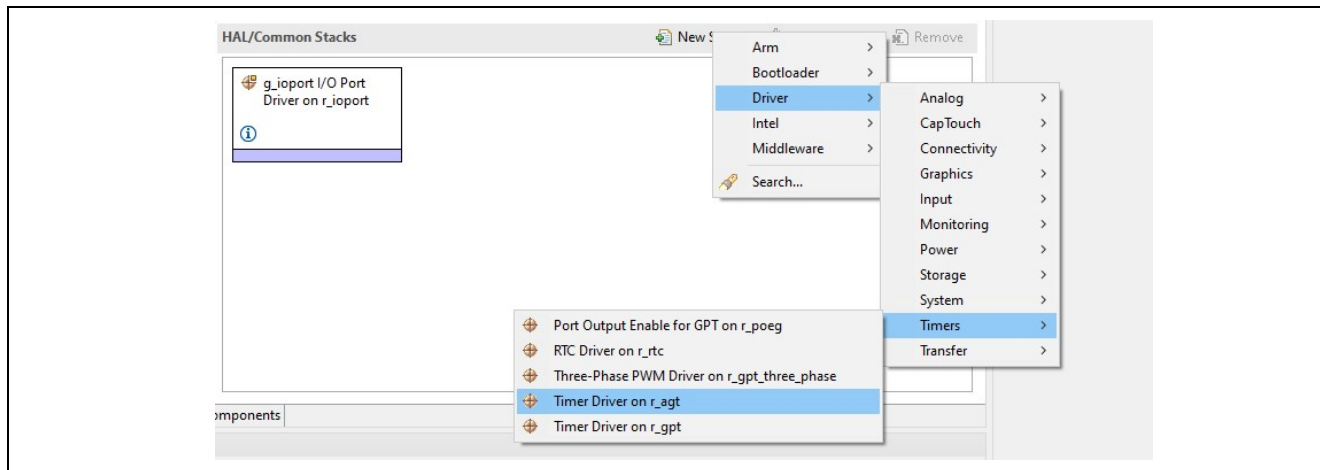


図 18 新しいスタックの追加

- c. 設定するモジュールに合わせて、[Pins]タブを設定します。

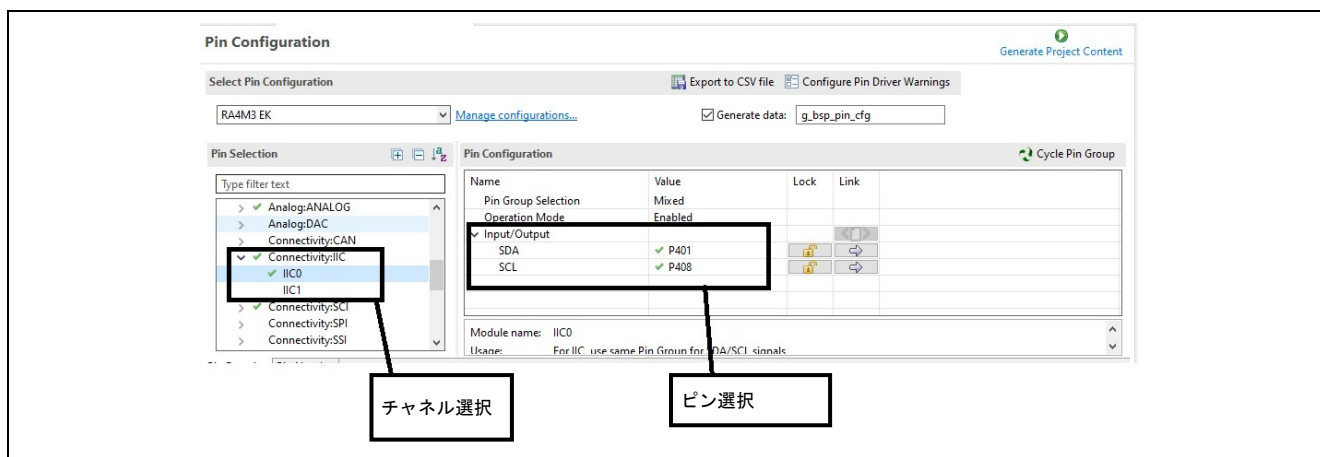


図 19 Pins タブ設定

6. 目的のモジュールに必要なタブ([BSP]、[Stacks]、[Clocks]など)をすべて設定したら、図のように保存 (Ctrl+S)して、[Generate Project Content]をクリックします。



図 20 プロジェクトコンテンツの生成

## 2.4 プロジェクトの設定とコンパイル

### 2.4.1 コンパイラオプションの設定

プロジェクトのコンパイラオプションは、次の手順で変更できます。

1. <プロジェクト名>.eww ファイルを開きます。ワークスペースで[プロジェクト]→[オプション]をクリックするか、Alt+F7 ショートカットを使用します。
2. 図 21 に示すように、[オプション]ウィンドウで、[カテゴリ]セクションの[C/C++コンパイラ]を選択し、[最適化] (右側)を選択します。

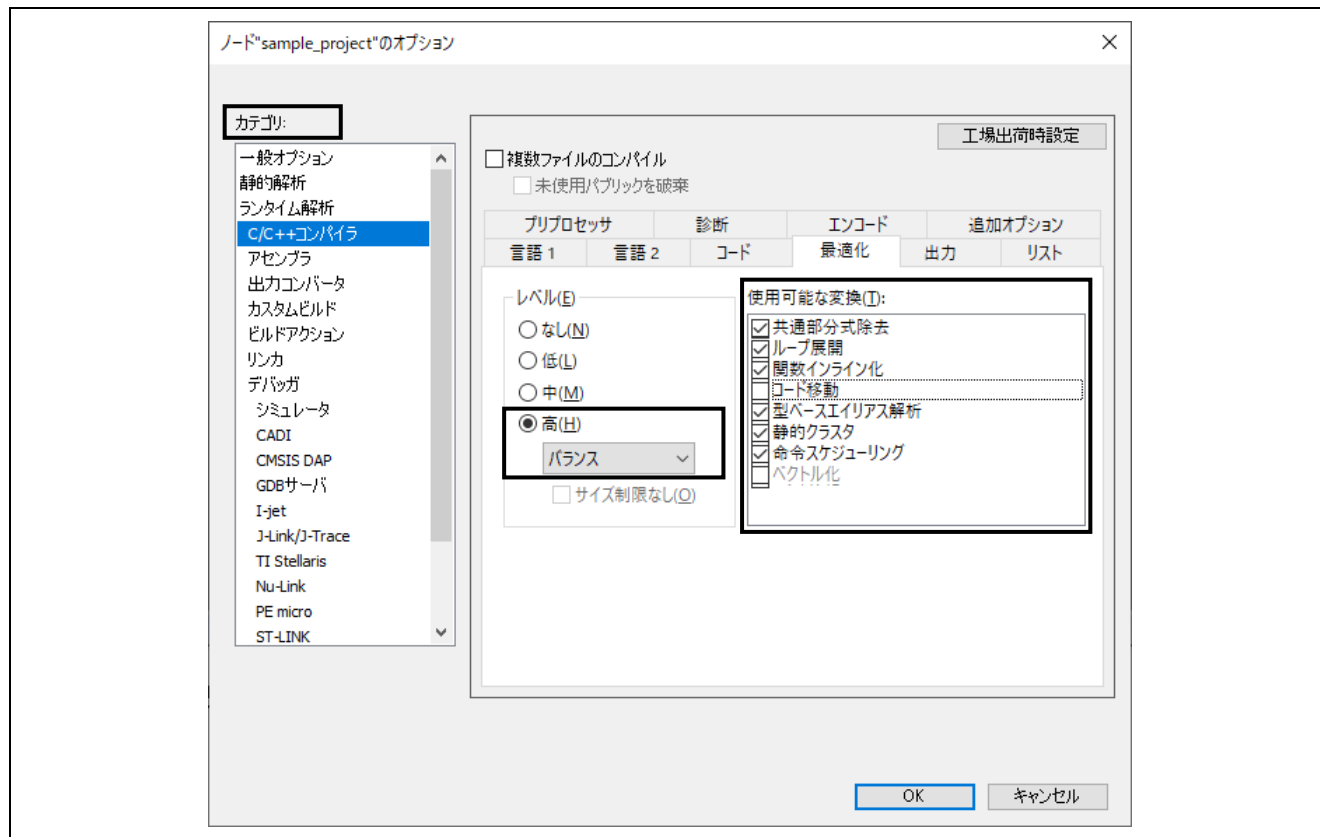


図 21 最適化

3. 図 21 に示すように、最適化の[レベル]を[高]に選択し、下のメニューで[バランス]を選択して、[コード移動]をオフにします。

4. 同じく[オプション]メニューの[カテゴリ]から[出力コンバータ]を選択します。図 22 に示すように、[追加出力ファイルを生成]をチェックして[出力フォーマット]オプションを[Intel Extended Hex]に設定します。

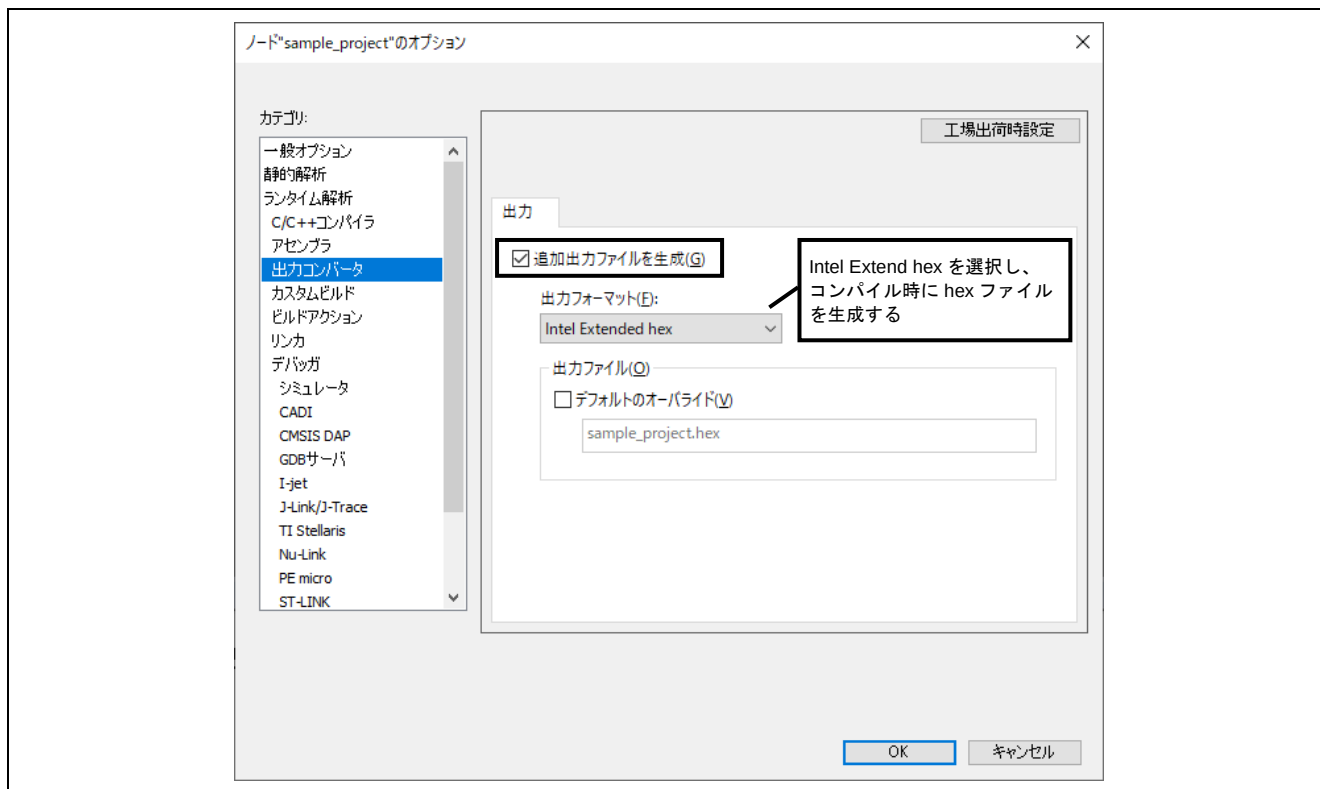


図 22 出力コンバータ

5. [オプション]ウィンドウの下部にある[OK]をクリックして、設定を適用します。

## 2.4.2 プロジェクトのビルド

プロジェクトをビルドする前に、以下のビルド前作業が完了していることを確認します。

- プロジェクトフォルダの中に、モジュール毎の src フォルダが存在すること
- 必要な構成([BSP]、[Clocks]、[Pins]など)を含むプロジェクトファイルが生成されていること
- コンパイラの設定が完了していること
- IAR EW ライセンスが有効になっていること

1. プロジェクト生成後、プロジェクト(.eww ファイル)を開きます。
2. プロジェクトを右クリックし、[すべてを再ビルド]オプションをクリックします。

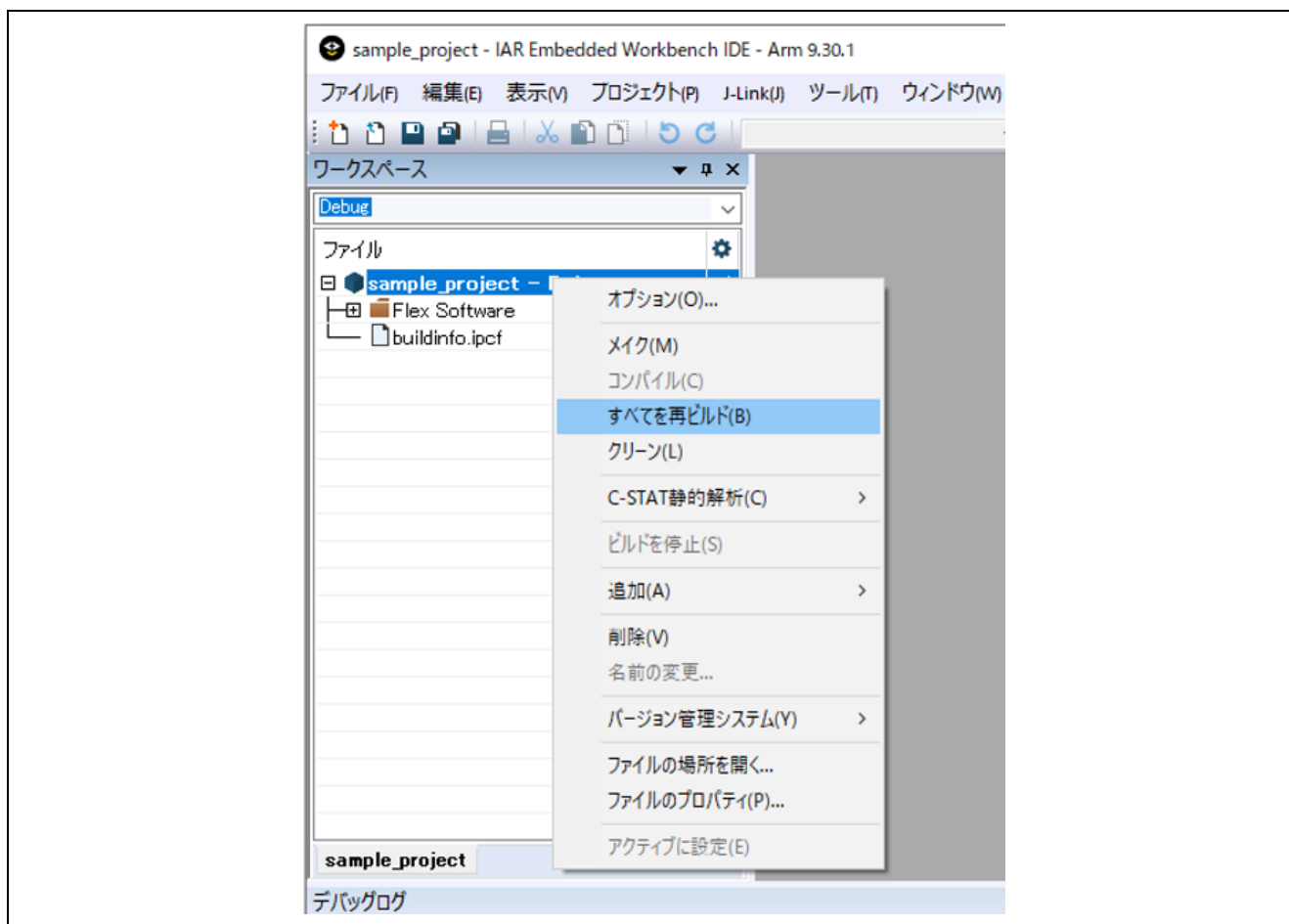


図 23 すべてを再ビルド

3. ライブログオプションを有効にすると、ログを取得することができます。
  - ワークスペースの下[ビルド]ウィンドウを右クリックします。
  - [ファイルへのライブログ]を選択し、作成するログファイルの場所を選択します。

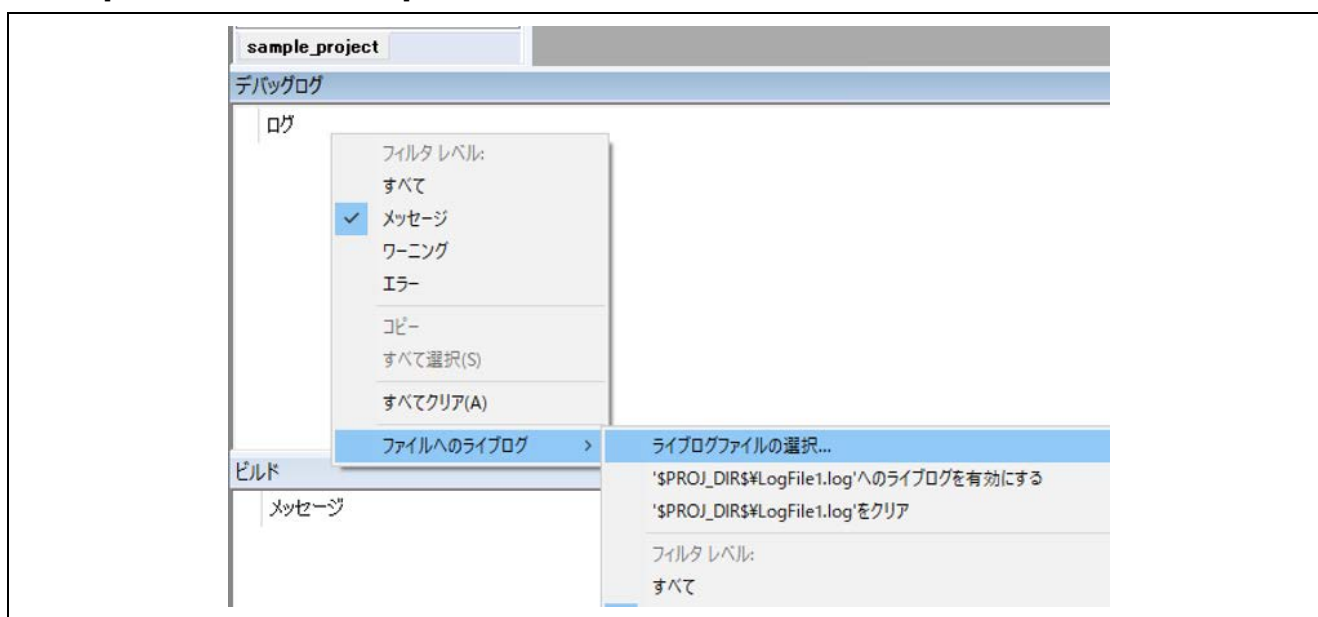


図 24 ライブログをファイルに保存

### 2.4.3 ビルド操作での警告とエラーへの対処

プロジェクトの生成とビルドが完了すると、[ビルド]ウィンドウのビルドログにその成否が表示されます。

[ビルド]ウィンドウが表示されていない場合は、[表示]→[メッセージ]→[ビルド]を選択します。

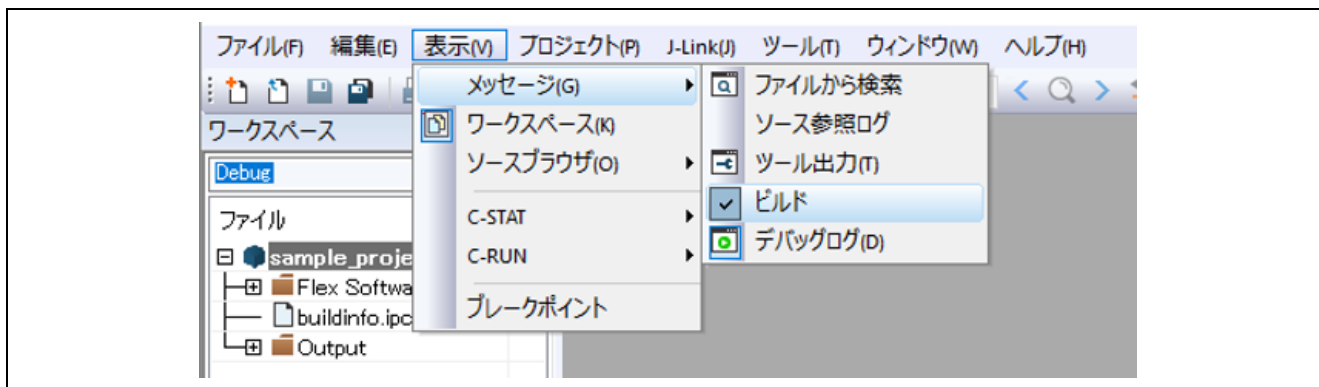


図 25 ビルドウィンドウの表示

[ビルド]ウィンドウでは、検出されたエラーと警告の総数とともに、エラーと警告のメッセージを詳細に確認できます。図 26 を参照してください。

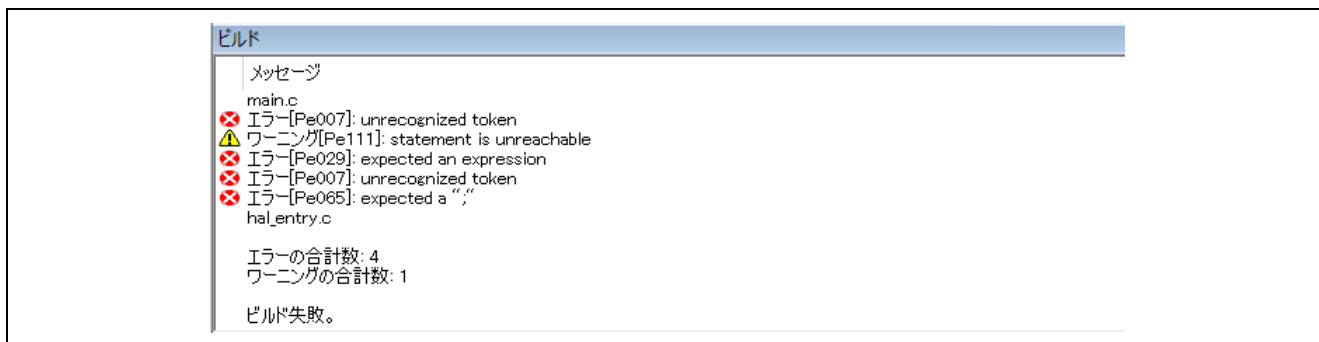


図 26 エラーと警告

## 2.5 アプリケーションのダウンロードとデバッグ

デバッグする前に、エラーがないことを確認し、コンパイラが生成した警告を確認してください。ターゲットボードがホストシステムに接続されていることを確認します。

- バイナリをデバッグするには、次のようにオプションパネルの[ダウンロードしてデバッグ]ボタンをクリックします。

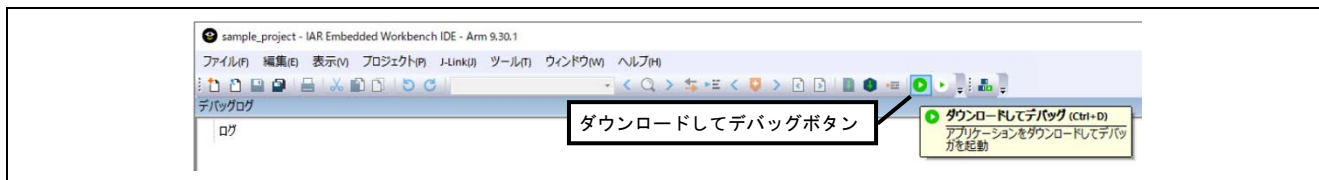


図 27 ダウンロードしてデバッグボタン



- ダウンロードしてデバッグが正常に完了したら、[実行]ボタンをクリックしてコードを実行します。



図 28 コード実行

- RTT は、バイナリにコンパイルされ、アプリケーションによって呼び出されると表示されます。

### e<sup>2</sup> studio プロジェクトの Keil MDK および ARM C コンパイラへの移行

RA ボードでの Keil IDE の使用方法については、以下のドキュメントを参照してください。

[https://www.keil.com/appnotes/files/apnt\\_330\\_v0.35.pdf](https://www.keil.com/appnotes/files/apnt_330_v0.35.pdf)

RA スマート・コンフィグレータを Keil MDK で使用方法については、以下のドキュメントを参照してください。

<https://renesas.github.io/fsp/ start dev.html#using-ra-smart-configurator-with-keil-mdk>

RASC/FSP のインストール手順については、FSP インストールのセクションを参照してください。

注： FSP バージョンを変更する場合は、それに応じて[ツールの設定]の[コマンド]フィールドで rasc.exe のパスを調整してください。

#### 2.5.1 DFP のインストール

- 最新の DFP pack をダウンロードし、解凍してください。

最新のデバイスファミリの pack は、次のリンクから入手できます。 - <https://www.keil.com/dd2/pack/>

- Keil IDE を開きます。(注：パックをインストールするには、管理者権限が必要です)

- Pack Installer を開きます。

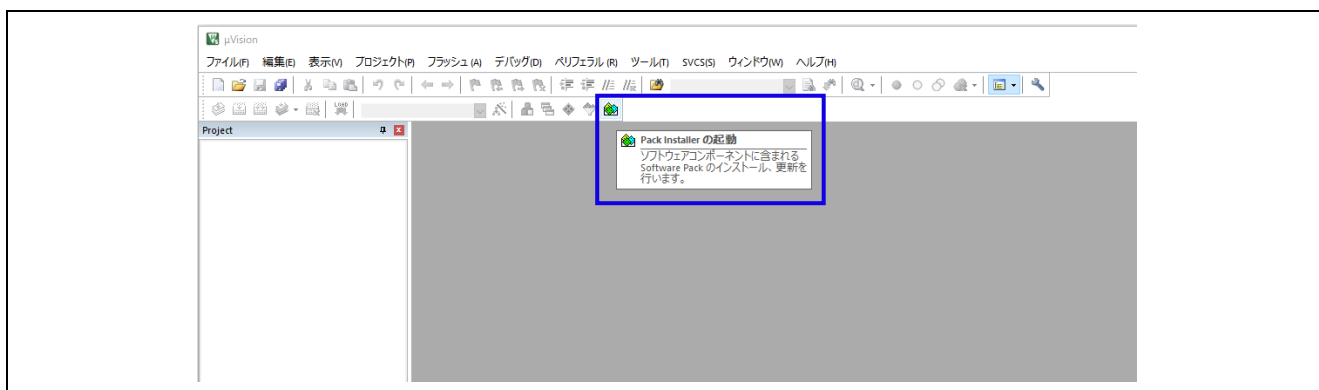


図 29 パックインストーラ

4. 図 30 に示すように、[File]→[Import from Folder...]と進んでください。

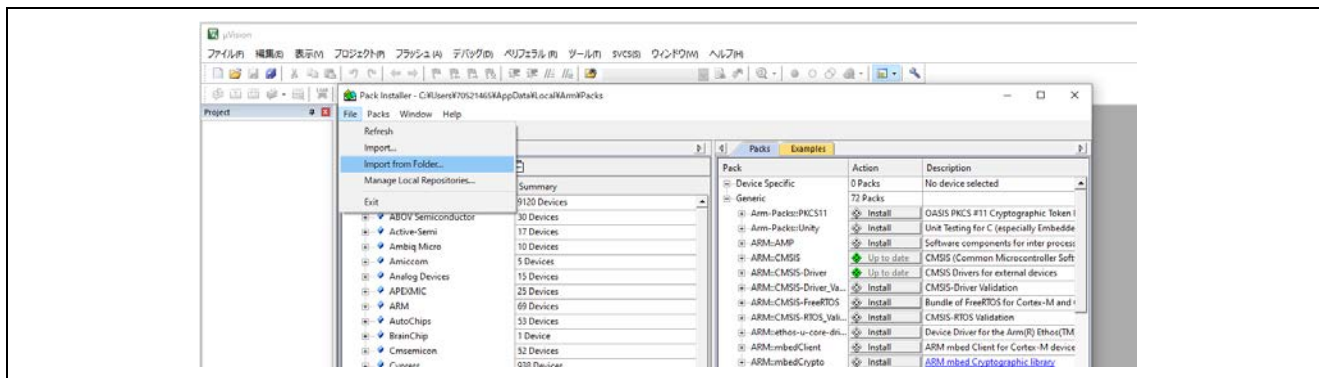


図 30 フォルダからインポートする

5. 開いたウィンドウから、(手順 1 で)解凍したフォルダを選択します。
6. インストールされている pack は、以下のように pack タブで選択された特定のデバイスに対して確認することができます。

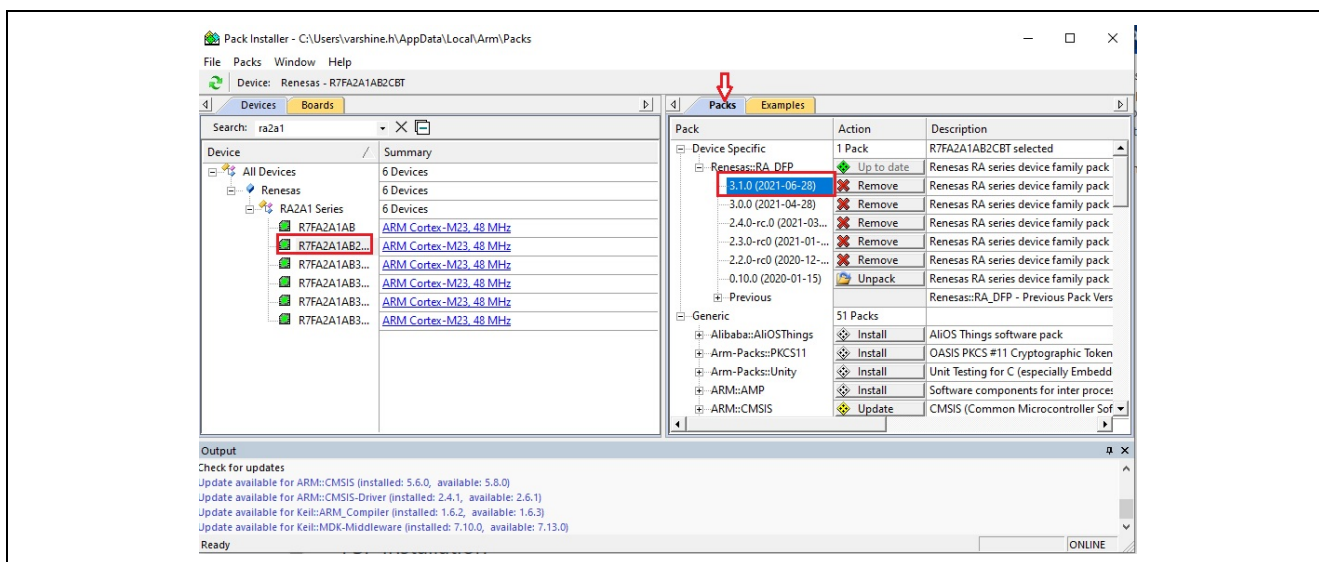


図 31 インストールされたパック

## 2.5.2 Keil 用の空のテンプレートプロジェクトの作成

1. フォルダ<rasc installation directory>\sc\_v2021-04\_fsp\_v3.1.0\eclipse に移動し、rasc.exe を起動します。
2. 開いているスマート・コンフィグレータで[Project name]とプロジェクトの[Location]を入力し、[Next]をクリックします。

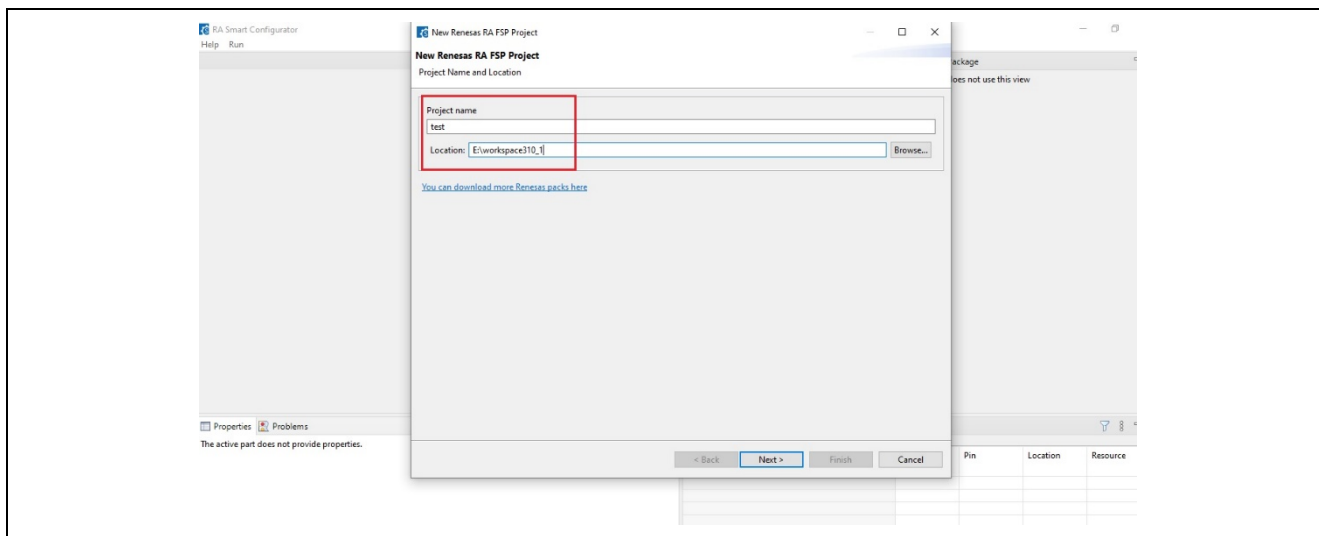


図 32 スマート・コンフィグレータ

3. 次のステップは、必要な[FSP Version]と[Board]を選択します。図 33 に示すように、このステップでは、特定の IDE プロジェクトの種類を選択できます。[Keil MDK Version 5]を選択します。

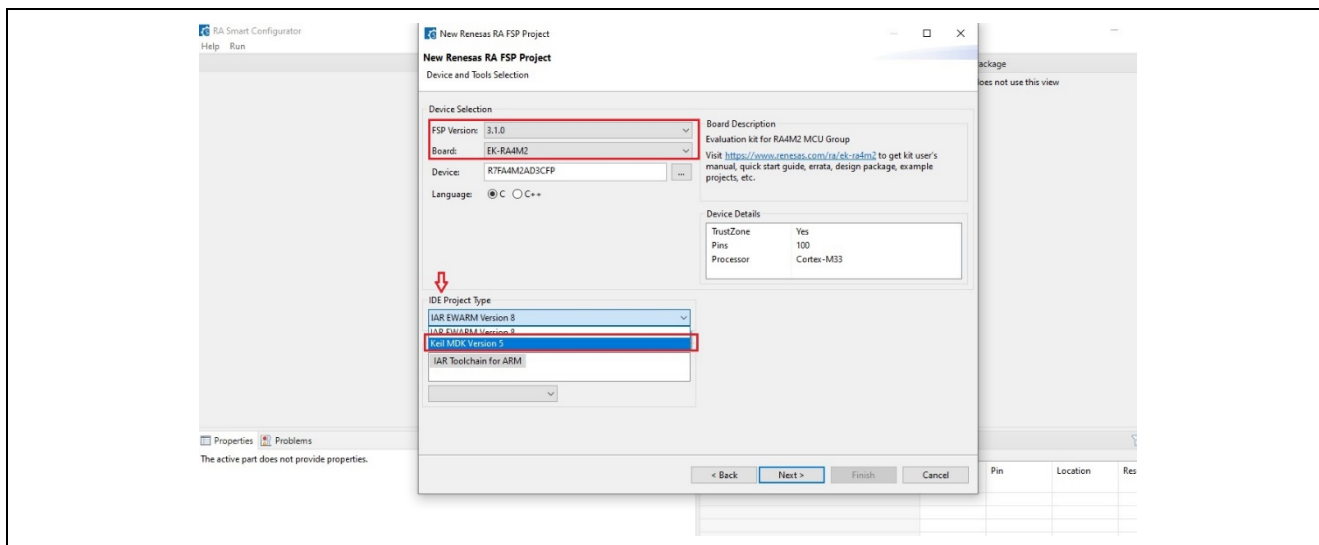


図 33 プロジェクトの種類を選択

4. 次のステップでは、プロジェクトの種類を、[Flat]、[TrustZone Secured]、[TrustZone Non-secured]から、必要に応じて選択します。この選択オプションは、TrustZone をサポートしている MCU でのみ利用可能です。

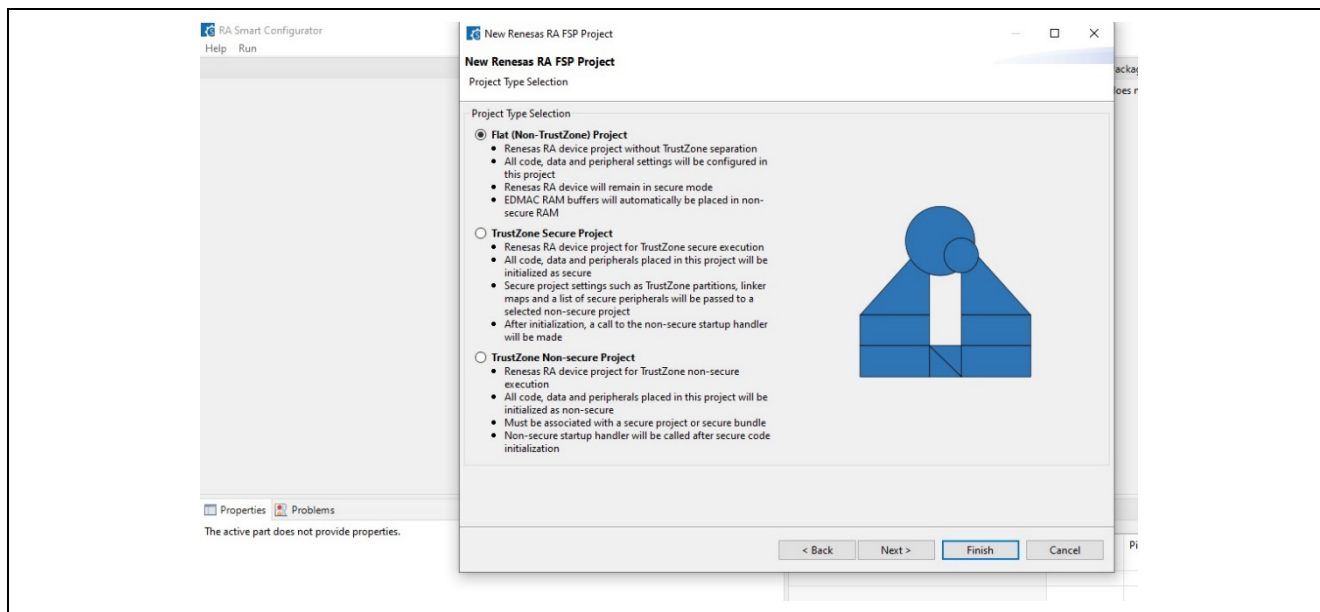


図 34 Flat、TrustZone Secured、TrustZone Non-secured 選択

5. 次に示すように、RTOS の選択はオプションで利用できます。プロジェクトに必要なオプションを選択します。

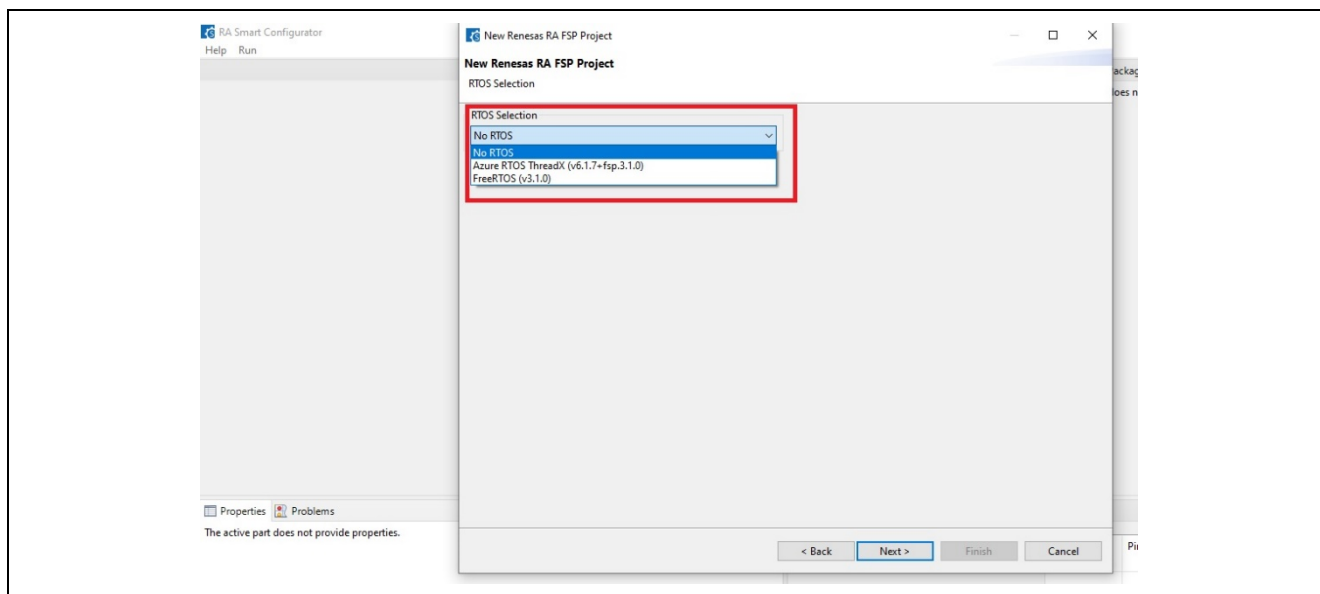


図 35 RTOS の選択

6. プロジェクトのサンプルテンプレートを作成するために、[Bare Metal - Minimal]を選択します。[Finish]をクリックして、プロジェクトテンプレートの作成を完了します。

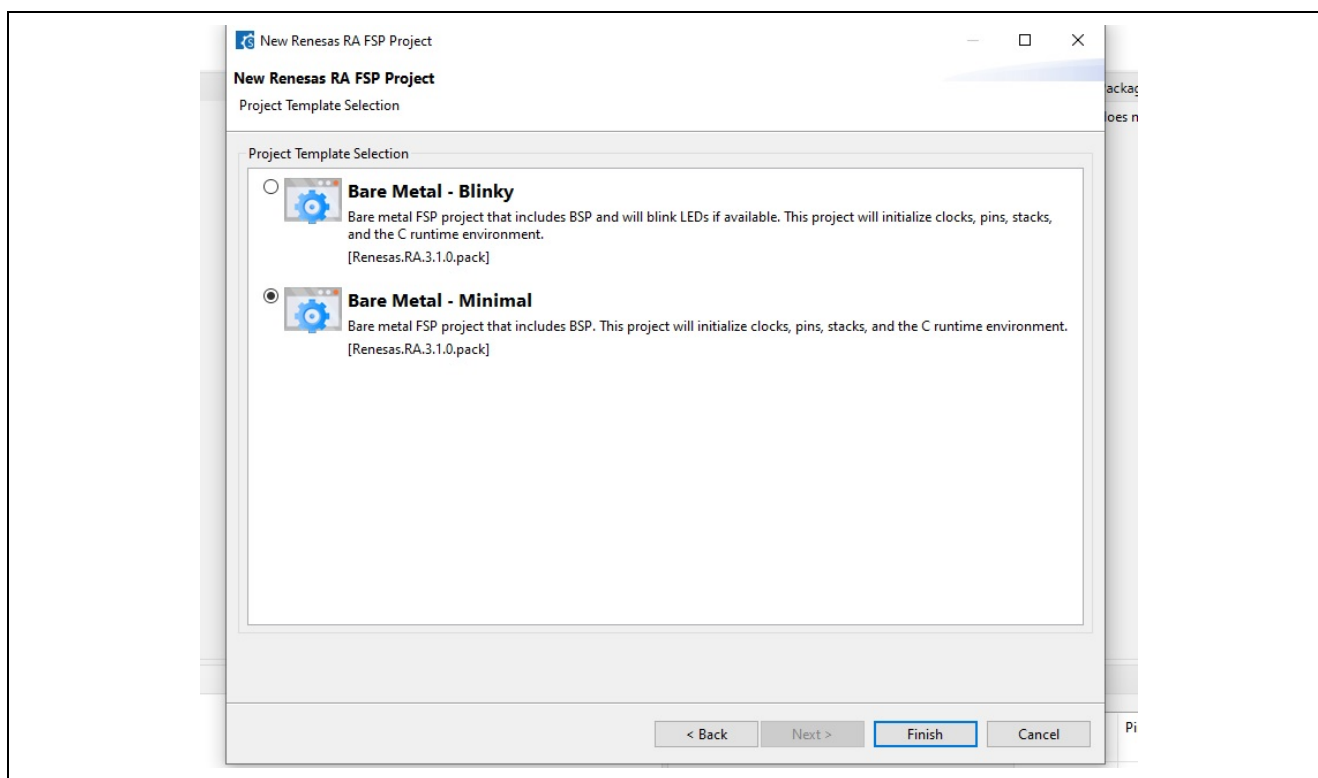


図 36 プロジェクトテンプレート

### 2.5.3 ソースコードの追加

1. プロジェクトテンプレートの作成プロセスが完了したら、作成した Keil プロジェクトを含むフォルダに移動します。

プロジェクトフォルダ内では、生成された src フォルダを e<sup>2</sup> studio プロジェクトの src フォルダに置き換える必要があります。

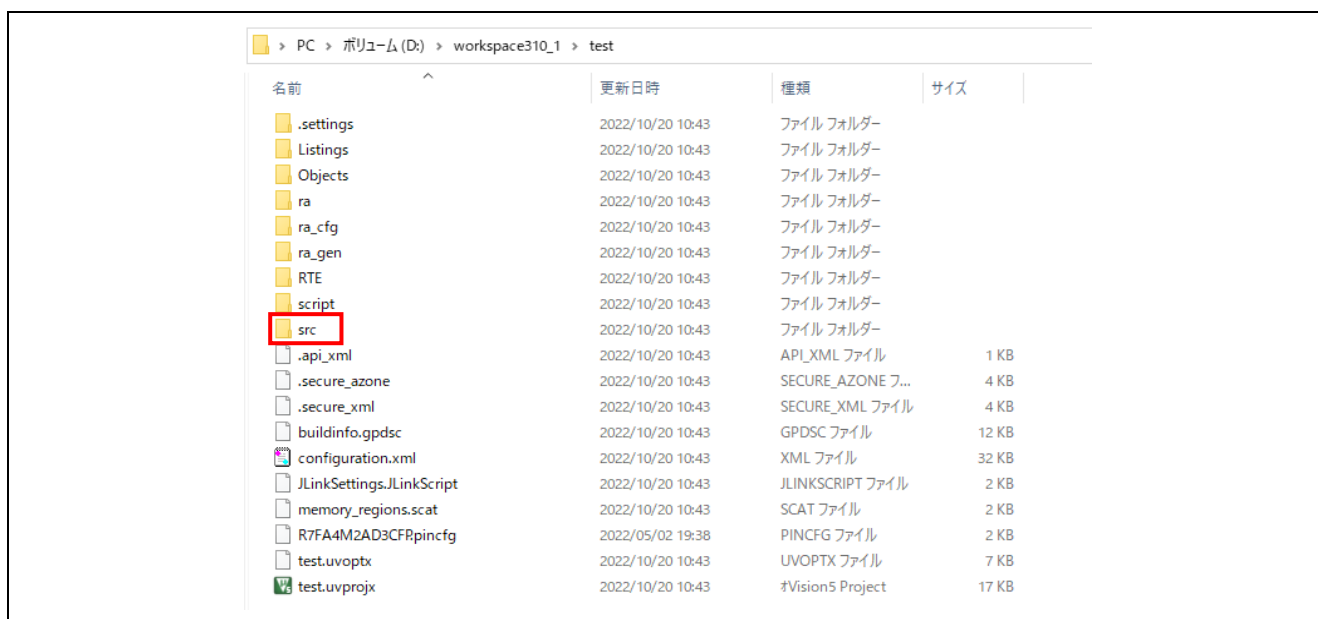


図 37 e<sup>2</sup> studio src フォルダ

2. フォルダ内の uVision5 プロジェクトファイルを開くと、Keil IDE でプロジェクトが開かれます。

| 名前                        | 更新日時             | 種類                | サイズ   |
|---------------------------|------------------|-------------------|-------|
| .settings                 | 2022/10/20 10:43 | ファイル フォルダ         |       |
| Listings                  | 2022/10/20 10:43 | ファイル フォルダ         |       |
| Objects                   | 2022/10/20 10:43 | ファイル フォルダ         |       |
| ra                        | 2022/10/20 10:43 | ファイル フォルダ         |       |
| ra_cfg                    | 2022/10/20 10:43 | ファイル フォルダ         |       |
| ra_gen                    | 2022/10/20 10:43 | ファイル フォルダ         |       |
| RTE                       | 2022/10/20 10:43 | ファイル フォルダ         |       |
| script                    | 2022/10/20 10:43 | ファイル フォルダ         |       |
| src                       | 2022/10/20 10:43 | ファイル フォルダ         |       |
| .api_xml                  | 2022/10/20 10:43 | API_XML ファイル      | 1 KB  |
| .secure_azone             | 2022/10/20 10:43 | SECURE_AZONE フ... | 4 KB  |
| .secure_xml               | 2022/10/20 10:43 | SECURE_XML ファイル   | 4 KB  |
| buildinfo.gpdsc           | 2022/10/20 10:43 | GPDSK ファイル        | 12 KB |
| configuration.xml         | 2022/10/20 10:43 | XML ファイル          | 32 KB |
| JLinkSettings.JLinkScript | 2022/10/20 10:43 | JLINKSCRIPT ファイル  | 2 KB  |
| memory_regions.scats      | 2022/10/20 10:43 | SCAT ファイル         | 2 KB  |
| R7FA4M2AD3CFR.pincfg      | 2022/05/02 19:38 | PINCFG ファイル       | 2 KB  |
| test.uvoptx               | 2022/10/20 10:43 | UVOPTX ファイル       | 7 KB  |
| test.uvprojx              | 2022/10/20 10:43 | uVision5 Project  | 17 KB |

図 38 uVision5 プロジェクトファイル

3. 図 39 に示すように、Keil プロジェクトが開かれます。

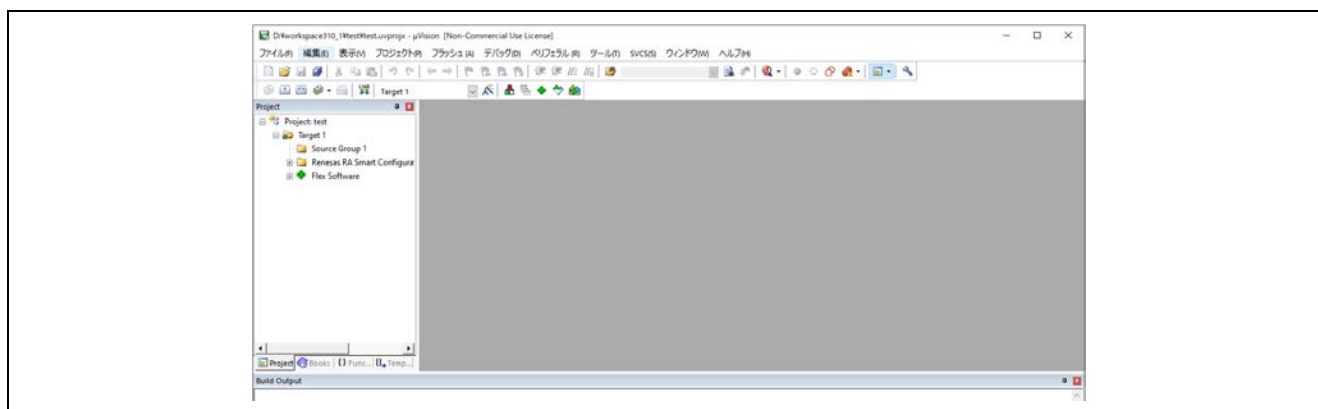


図 39 Keil プロジェクト

4. RASC を開くには、[ランタイム環境]アイコンを選択します。

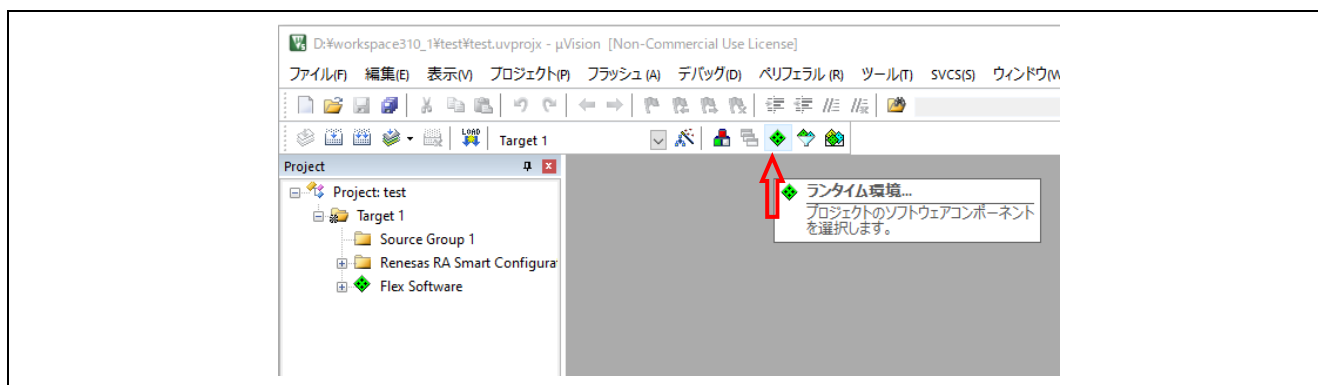


図 40 ランタイム環境

5. 次のようなダイアログボックスが表示されます。:

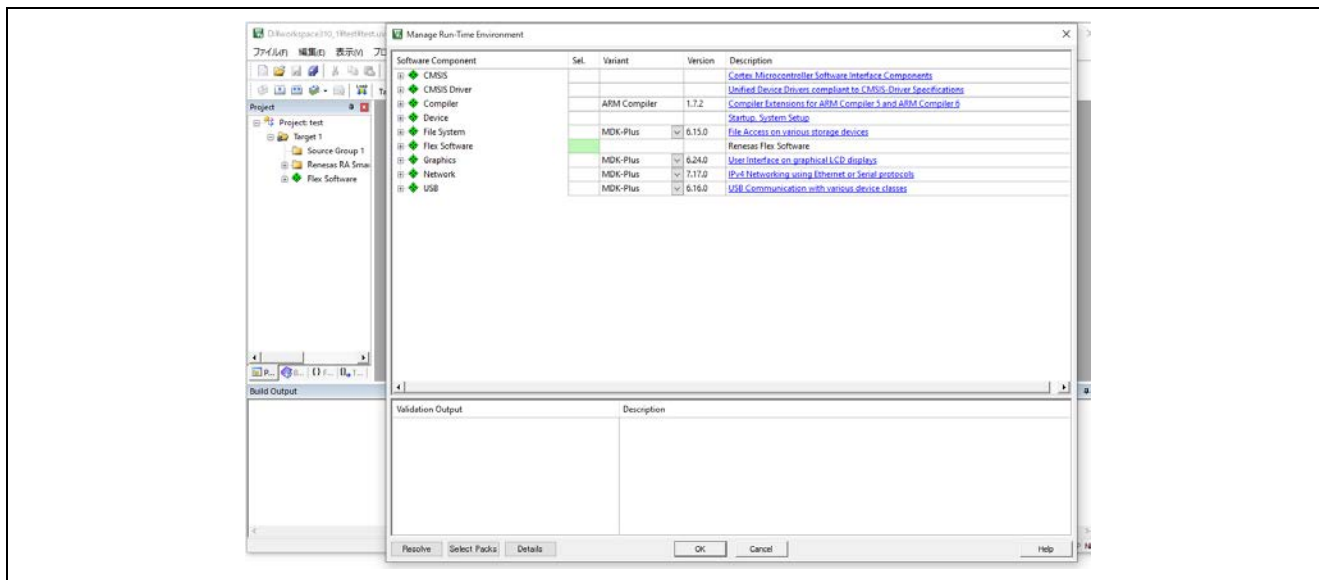


図 41 ランタイム環境の管理ダイアログ

6. RASC を実行するには、次のように[Flex Software]を展開し、[RA Configuration]を実行します。

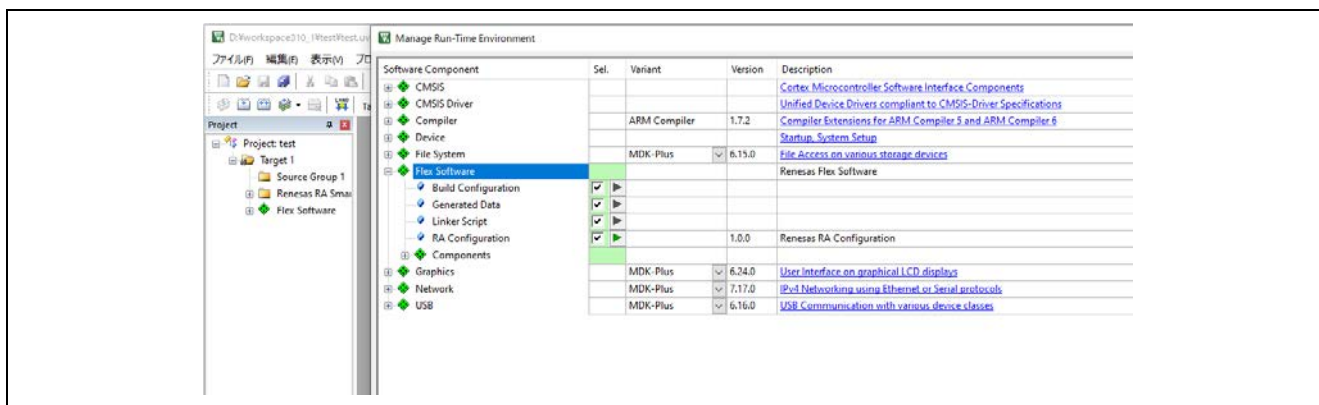


図 42 RA Configuration

7. インストールされている RASC バージョンのウィンドウが表示されます。該当する番号を入力して、必要なバージョンを選択します。

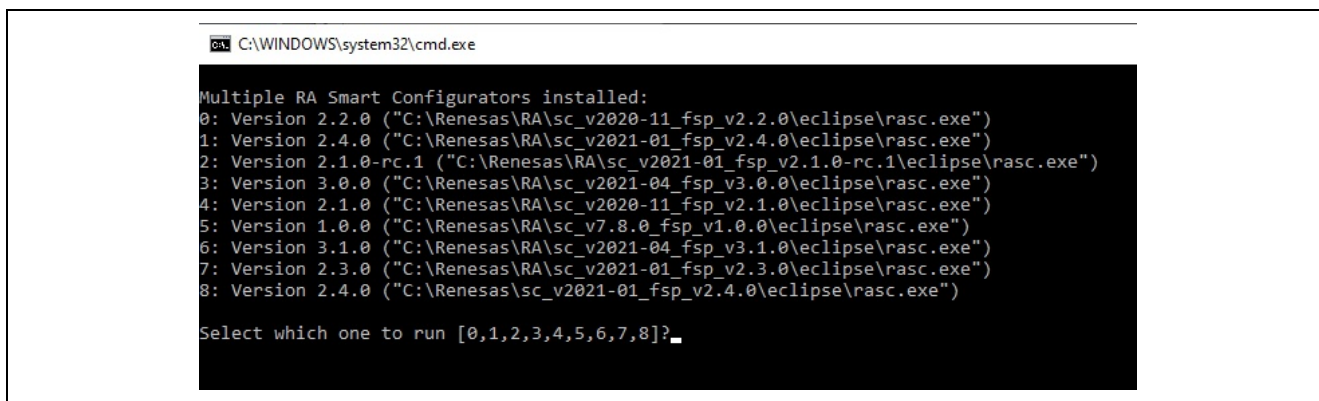


図 43 RASC バージョンの選択



8. スマート・コンフィグレータが開き、強調表示されている使用可能なタブから、[BSP]、[Pins]、[Stacks]などのさまざまなプロパティを設定できます。

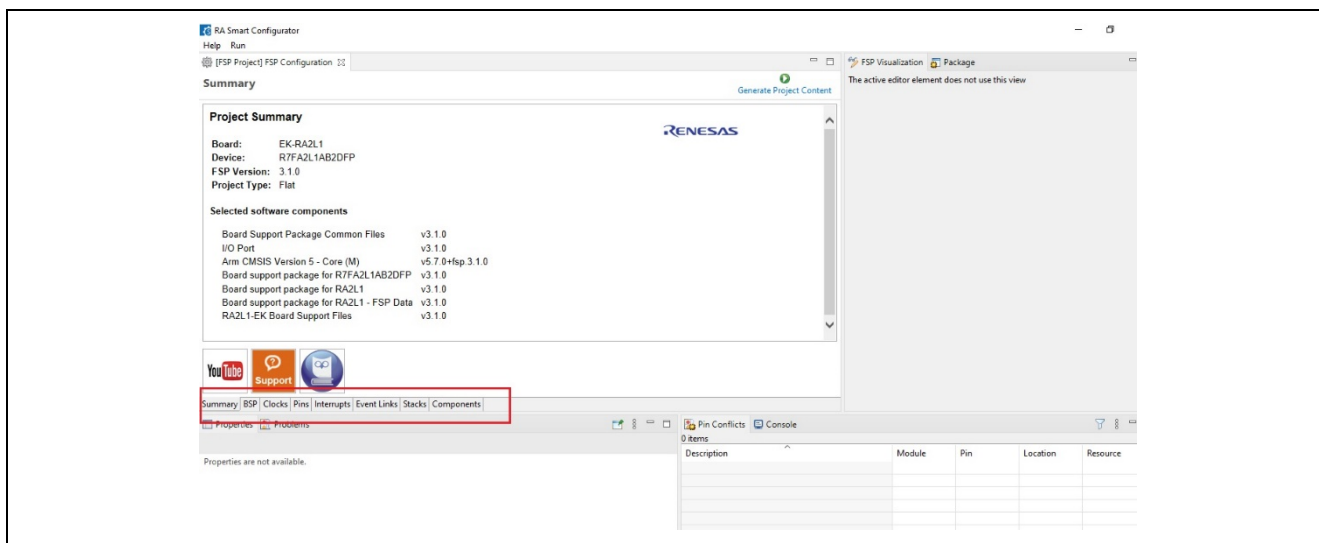


図 44 スマート・コンフィグレータ

[BSP]、[Pins]、[Stacks]、[Clocks]タブで設定するパラメータは、e<sup>2</sup> studio プロジェクトの設定に合わせて調整する必要があります。

9. 必要な設定が完了したら、[Ctrl+S] (保存)を実行し、[Generate Project Content]を選択します。

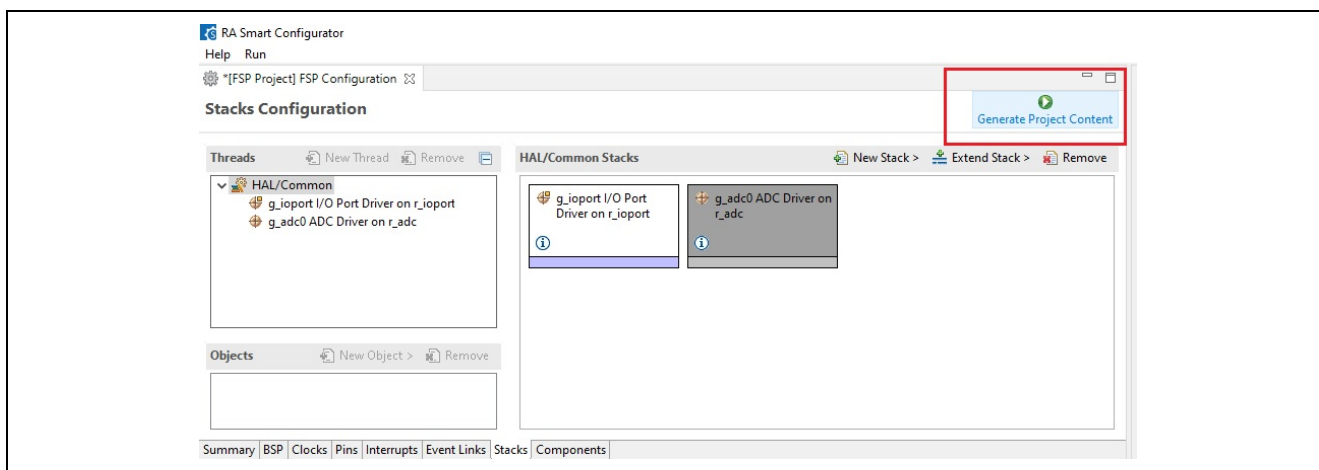


図 45 Generate Project Content



10. RASC を終了し、[実行環境の管理]ウィンドウで[OK]を選択します。
11. 完了すると、以下のダイアログボックスが表示されます。[はい]を選択すると、コンフィグレータで行った変更がインポートされます。

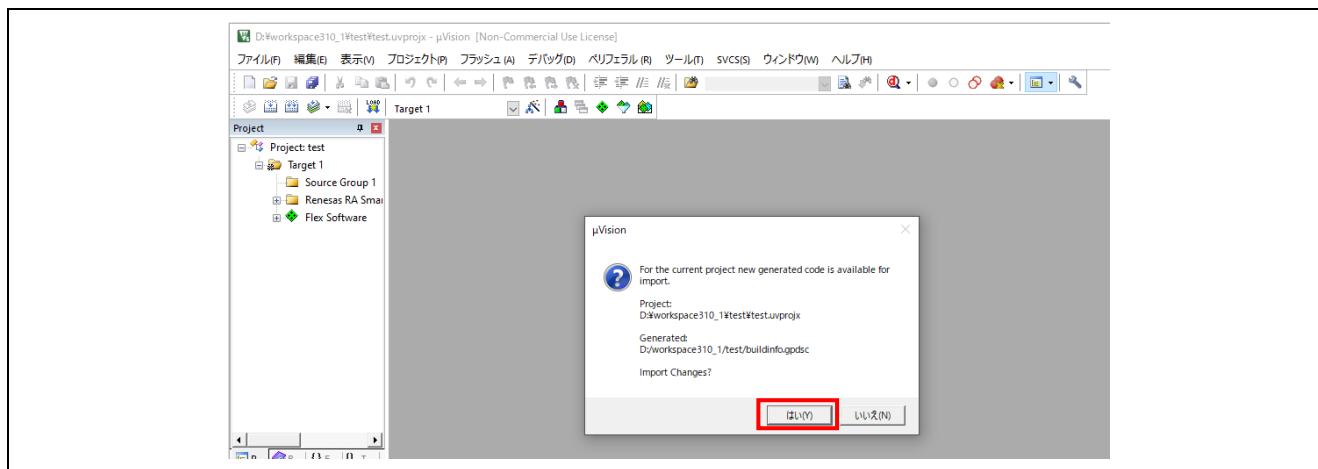


図 46 変更の保存

## 2.5.4 プロジェクトのコンパイル

1. 表示されるウィンドウで、以下のターゲットオプションを設定します。このウィンドウは、ハイライトされている[ターゲット]アイコンから選択することができます。

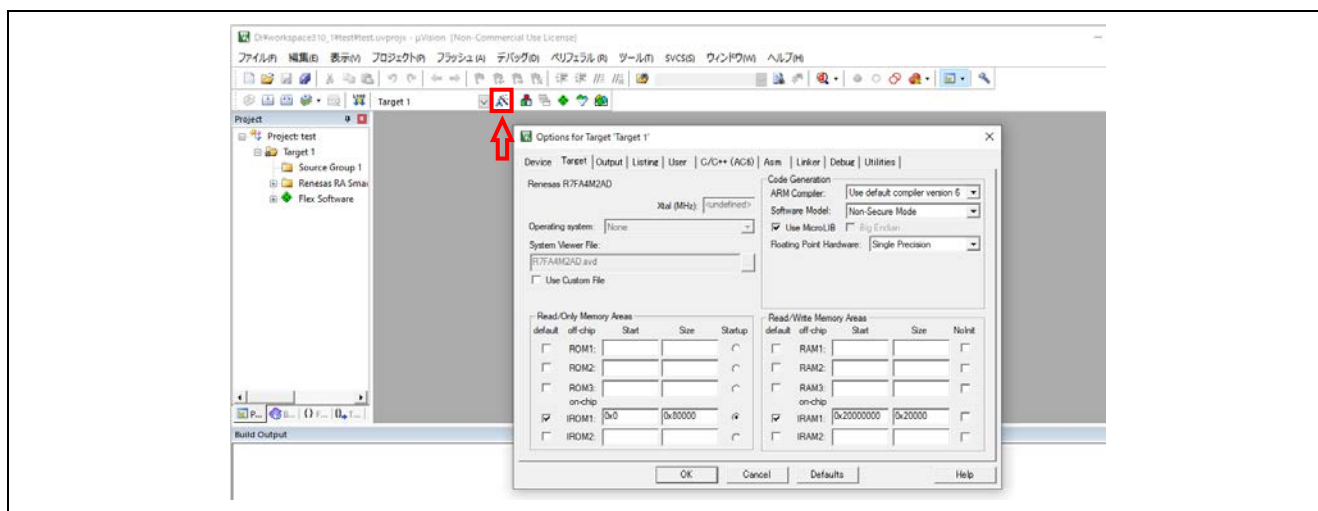


図 47 ターゲットオプション

2. [ARM Compiler]のバージョンを図 48 のように設定します。

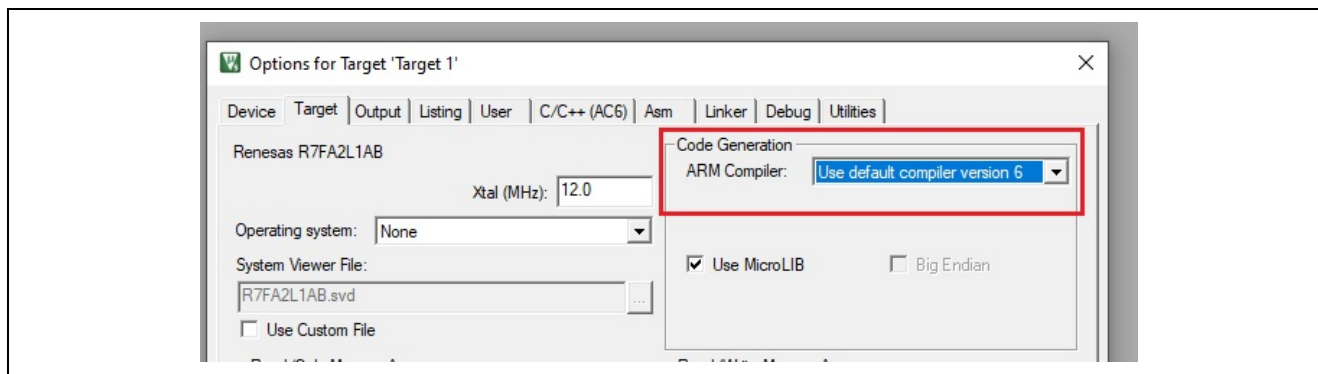


図 48 Arm コンパイラバージョン

3. [Output]タブで、[Create HEX File]オプションが選択されていることを確認し、ビルド中に HEX ファイルを生成します。

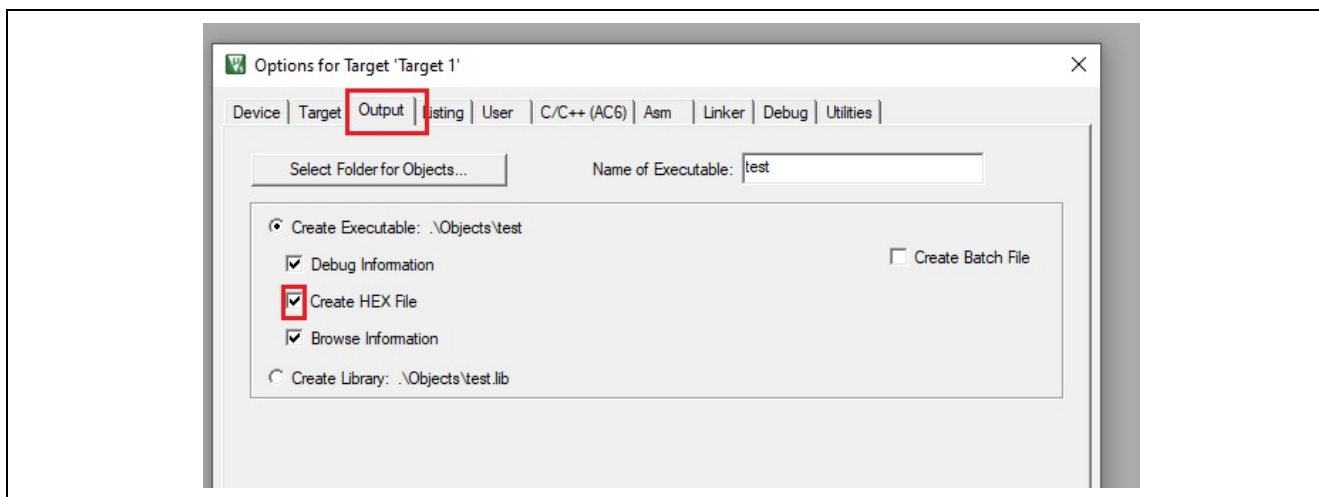


図 49 HEX ファイルの作成

4. [C/C++(AC6)]タブでターゲットの最適化を設定します。最適化を[-O2]として選択します。[Misc Controls]で、以下のようにパラメータ[-Os]を追加します。

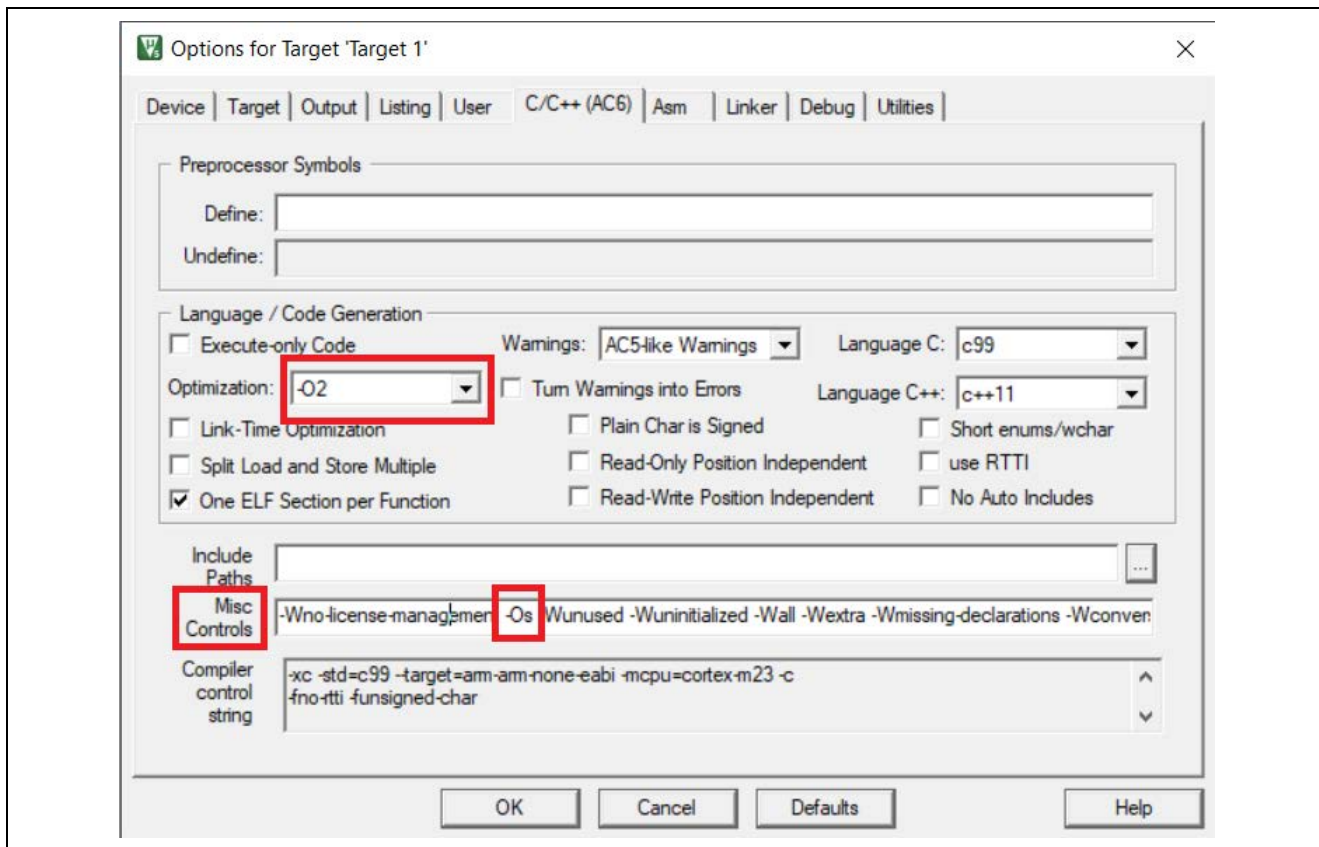


図 50 最適化とその他コントロール

5. [Debug]オプションで、[J-Link/J-TRACE Cortex]オプションを選択します。また、[Settings]オプションを選択し、J リンク設定が設定され、生成されていることを確認します。

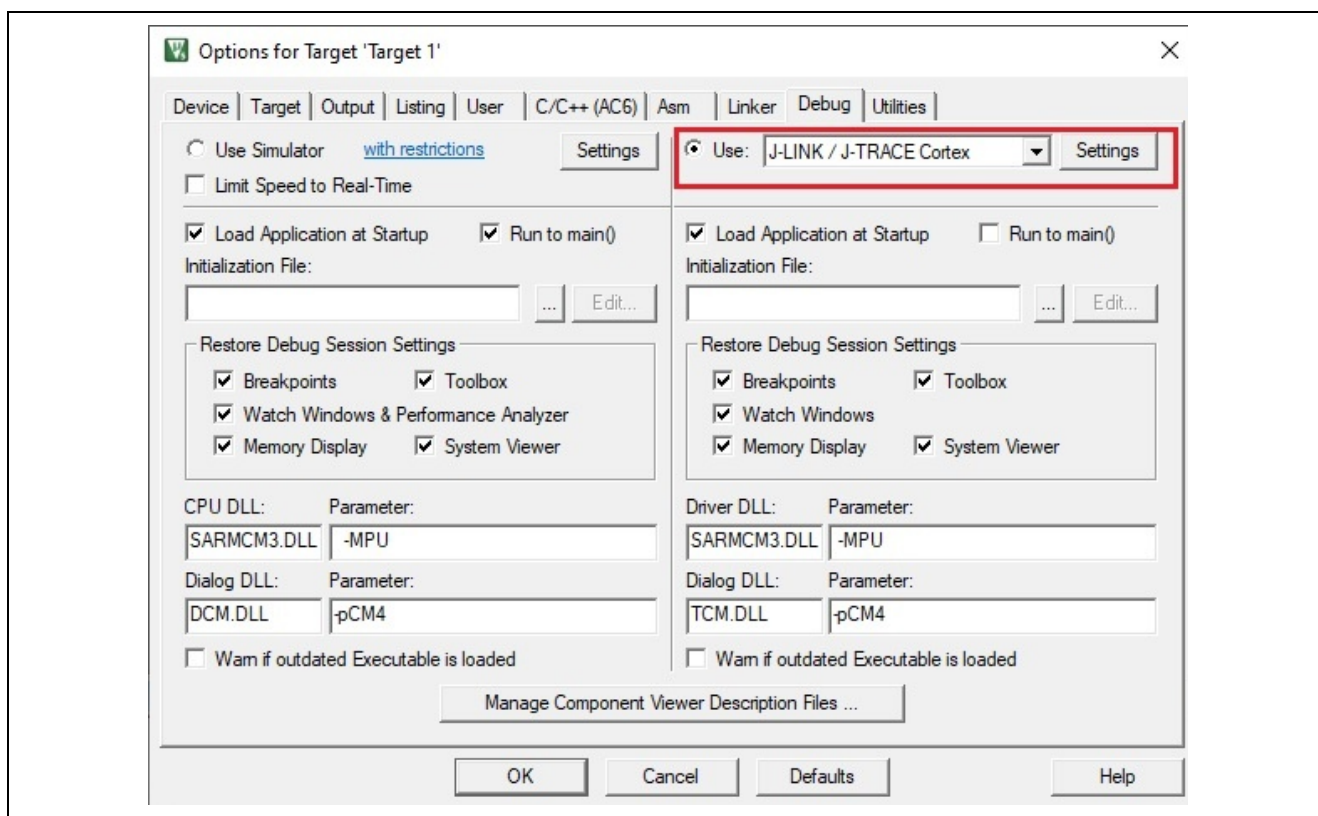


図 51 デバッグオプション

6. 前述の[debug]ウィンドウで[Settings]が選択された後、次のウィンドウが表示されます。適切な J-Link ID が選択されていることを確認してください。

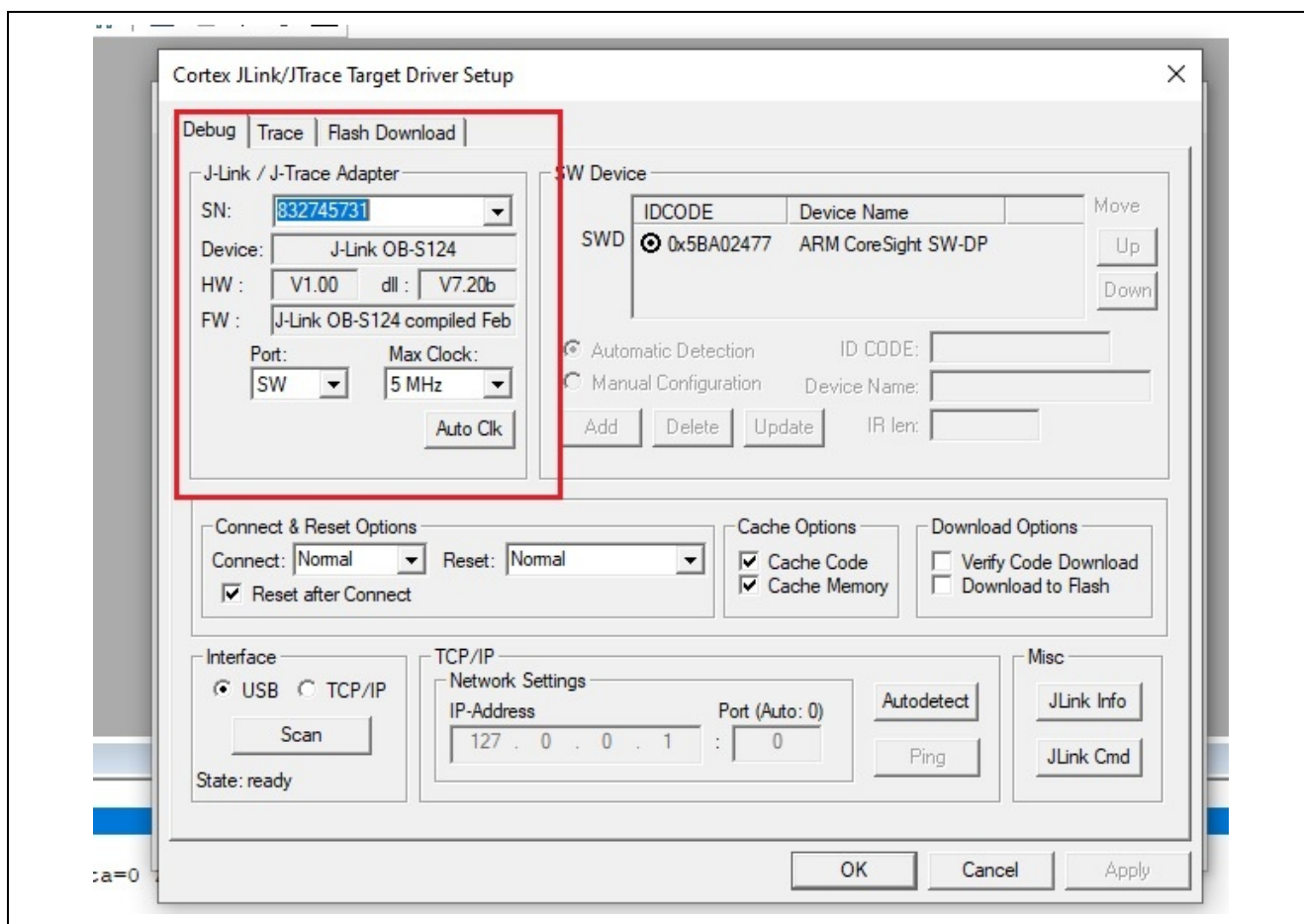


図 52 J リンク ID の選択

7. [OK]を選択します。

8. プロジェクトを選択します。[プロジェクト]→[ビルドターゲット]を選択します。プロジェクトがビルドされたら、警告やエラーがないか確認します。

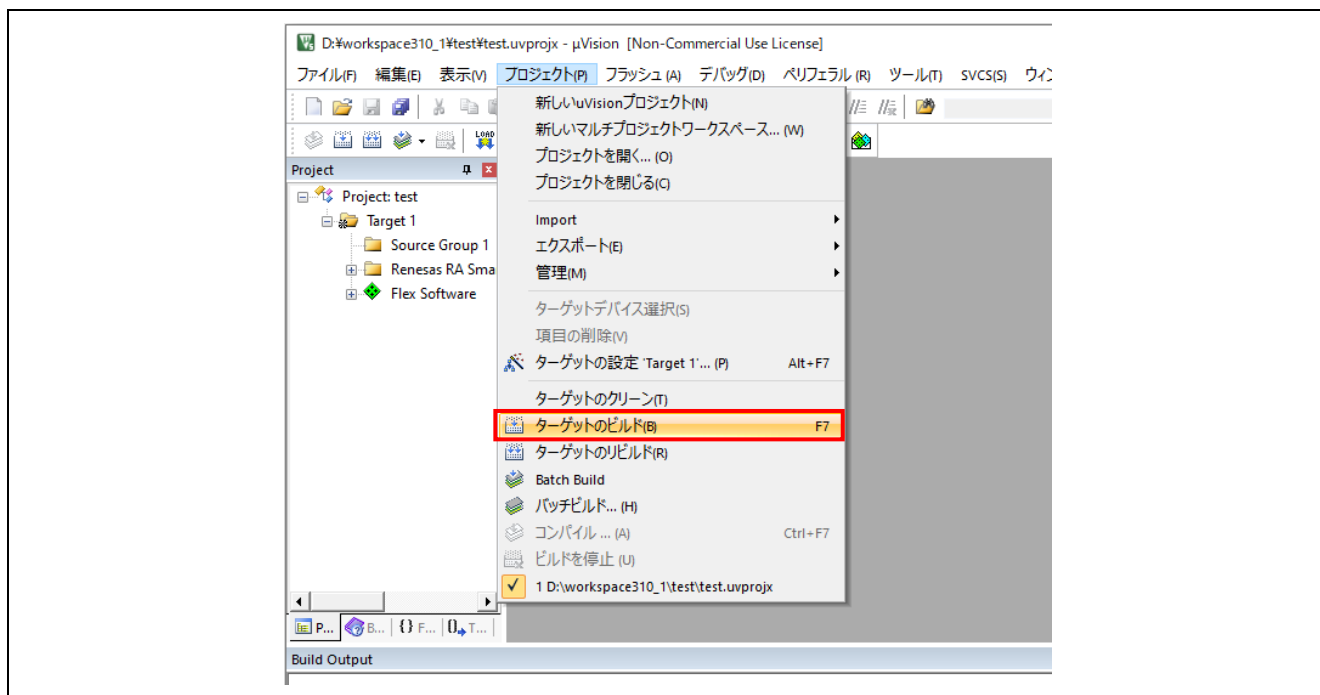


図 53 ビルドターゲット

## 2.5.5 プロジェクトのダウンロードとデバッグ

1. プロジェクトのビルドに成功し、エラーや警告が出なくなったら、[デバッグセッションの開始/停止]アイコンを選択します。[ターゲット]オプションを設定する際に、J-Link の設定がされていることを確認してください。

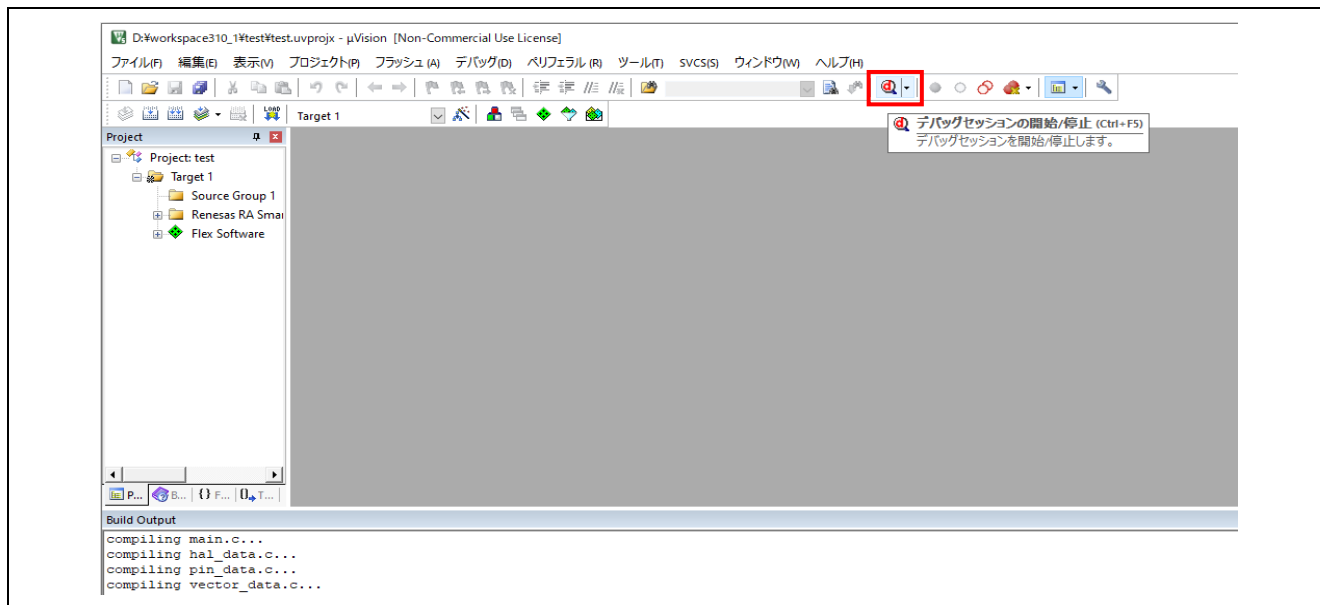


図 54 デバッグセッションの開始

2. [デバッグ]ウィンドウが表示されます。[リセット]および[実行]アイコンを選択して、プロジェクトを実行します。

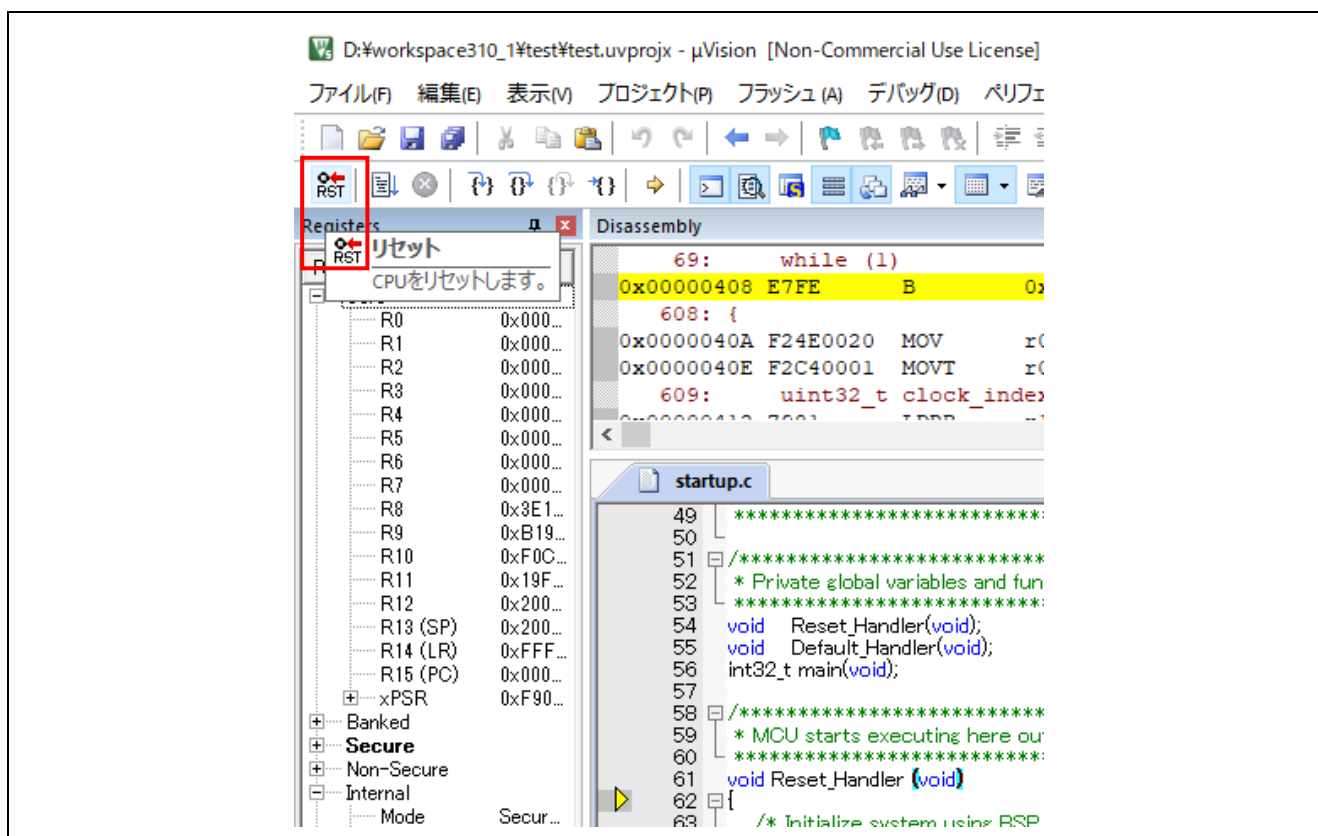


図 55 リセットアイコン

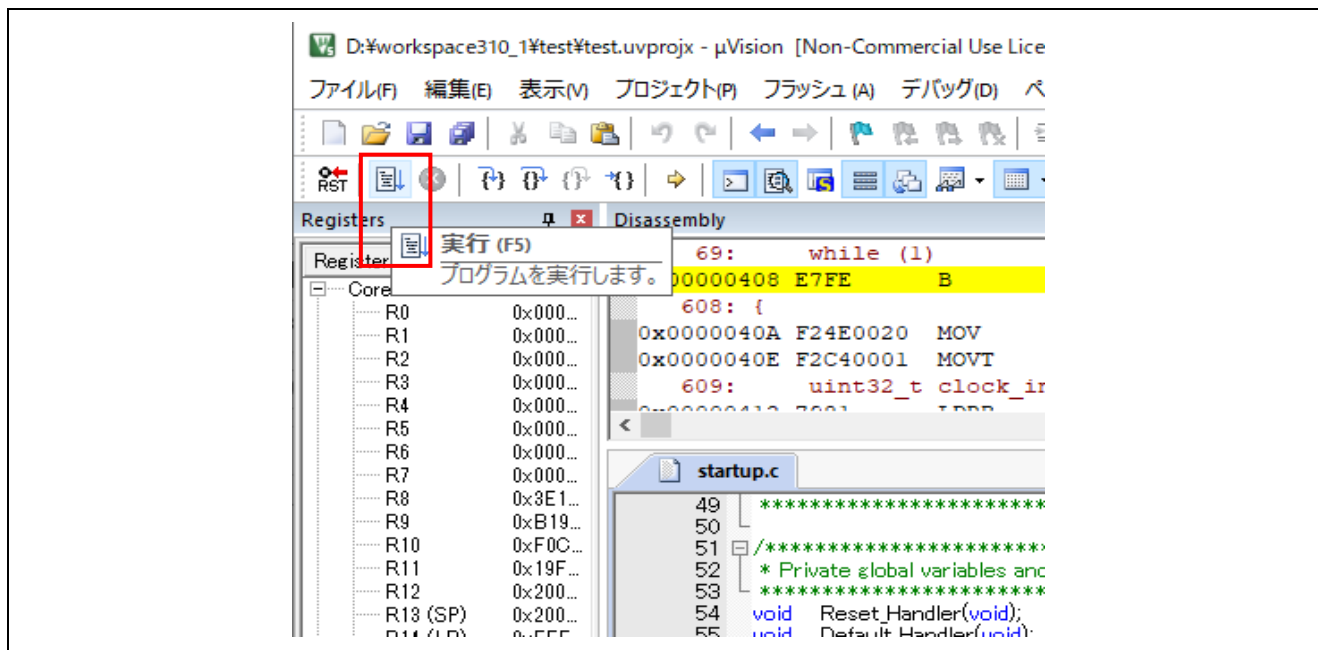


図 56 実行アイコン

3. [停止]アイコンを選択することで、デバッグセッションを停止することができます。

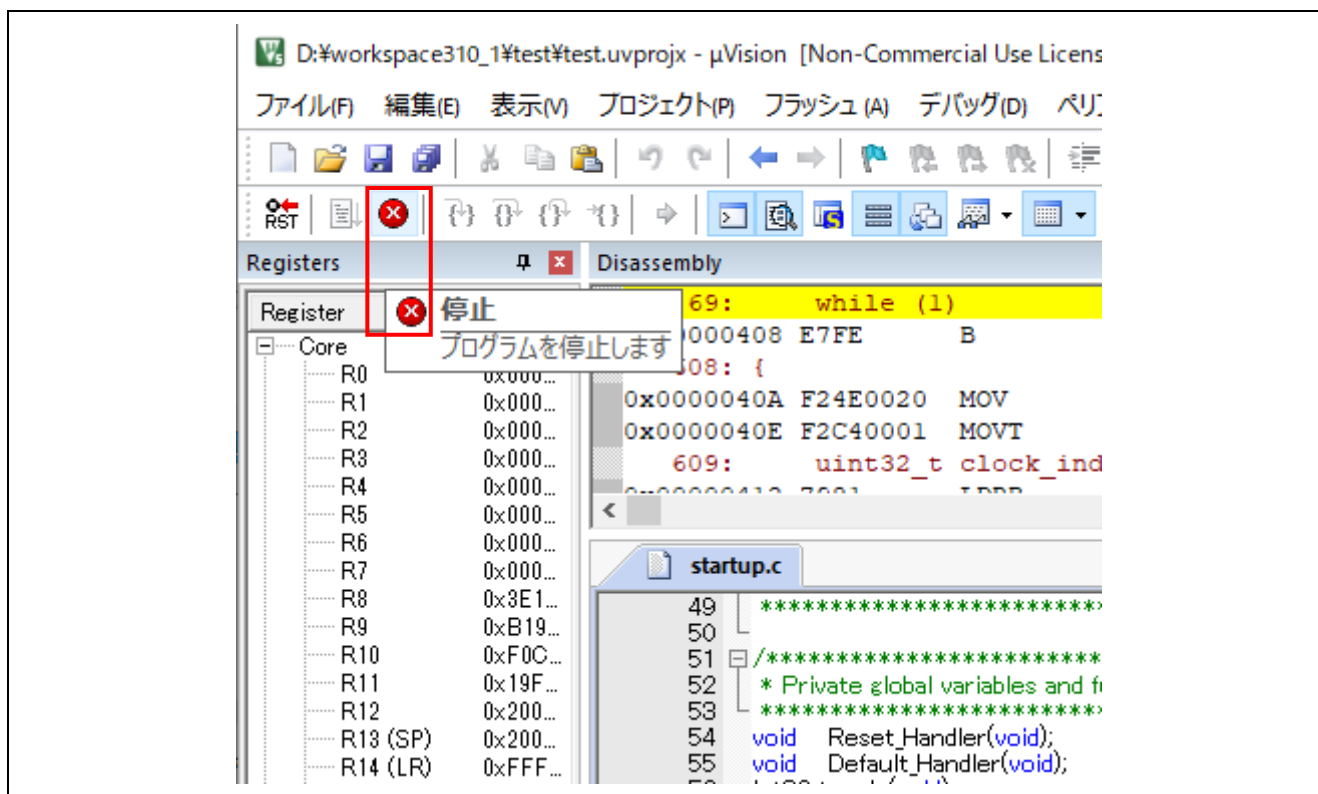


図 57 停止アイコン

実行後、SEGGER RTT Viewer (アプリケーションバイナリと共にコンパイルされ、使用されている場合のみ利用可能)で出力を監視することができます。



## ウェブサイトとサポート

以下の Vanity URL から、RA ファミリの主要な要素についての説明、コンポーネントや関連ドキュメントのダウンロード、サポートを受けることができます。

|                              |                                                                        |
|------------------------------|------------------------------------------------------------------------|
| RA 製品情報                      | <a href="http://www.renesas.com/ra">www.renesas.com/ra</a>             |
| RA 製品サポートフォーラム               | <a href="http://www.renesas.com/ra/forum">www.renesas.com/ra/forum</a> |
| RA Flexible Software Package | <a href="http://www.renesas.com/FSP">www.renesas.com/FSP</a>           |
| Renesas サポート                 | <a href="http://www.renesas.com/support">www.renesas.com/support</a>   |

改訂記録

| Rev. | 発行日       | 改訂内容 |      |
|------|-----------|------|------|
|      |           | ページ  | ポイント |
| 1.00 | Mar.27.23 | －    | 初版発行 |

## 製品ご使用上の注意事項

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

### 1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

### 2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

### 3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

### 4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

### 5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

### 6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 $V_{IL}$  (Max.) から  $V_{IH}$  (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 $V_{IL}$  (Max.) から  $V_{IH}$  (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

### 7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違くと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

## 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、変更、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
  6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。  
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等  
当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
  7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な変更、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、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
  12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものいたします。
  13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
  14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。
- 注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1 2020.10)

## 本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

[www.renesas.com](http://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/](http://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。