

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

要旨 (Introduction)

このアプリケーションノートは、Renesas Synergy プロジェクトを IAR Embedded Workbench® for Renesas Synergy™ (IAR EW for Synergy) または e² studio 統合ソリューション開発環境 (integrated solutions development environment, ISDE) にインポートし、その後、そのプロジェクトアプリケーションのビルドと実行をするプロセスについて説明します。このアプリケーションノートの手順は、すべての Renesas Synergy デバイスと開発ボード、および「必須リソース」の章に記載したすべてのソフトウェアに適用されます。

注記： このアプリケーションノートでは、例として NetX™ DNS プロジェクトを使用します。必要に応じて、任意のプロジェクトを代わりに使用することもできます。

達成目標 (Goals and Objectives)

このアプリケーションノートの目標は、サンプルアプリケーションのような既存の Renesas Synergy Project を IAR EW for Synergy または e² studio にインポートできるようにすることです。

前提条件 (Prerequisites)

このアプリケーションノートは、読者が IAR EW for Synergy と Renesas e² studio ISDE のどちらか、および SSP を使用した経験があることを前提としています。例えば、このアプリケーションノートの手順を実行する前に、お持ちのボードのクイックスタートガイドの手順に従い「Blinky」プロジェクトをビルドして実行する必要があります。それにより、IAR EW for Synergy または e² studio と SSP の使用方法を理解し、ボードへのデバッグ接続が適切に機能していることを確認することができます。

必須リソース (Required Resources)

プロジェクトインポートの手順を実行するには、Microsoft® Windows® 7 または 10 が動作しており、以下の Renesas ソフトウェアがインストールされている PC が必要です

- Renesas Synergy 開発ボード (例えば、DK-S7G2)
- Microsoft® Windows® 7 または 10 が動作しており、以下の Renesas ソフトウェアがインストールされている PC:
 - IAR EW for Synergy v8.21.1 またはそれ以降、あるいは e² studio ISDE v6.2.0 またはそれ以降
 - Synergy Software Package (SSP) v1.4.0 またはそれ以降
 - Renesas Synergy™ Standalone Configurator (SSC) v6.2.0 またはそれ以降 (IAR EW for Synergy の場合のみ)。

必要な Renesas ソフトウェアは Renesas Synergy WEBサイト

(<https://www.renesas.com/jp/ja/products/synergy.html>) からダウンロードできます。

目次

1. IAR EW for Synergy でのプロジェクトのインポートとビルド (Importing and Building Projects with IAR EW for Synergy)	3
1.1 IAR EW for Synergy への既存のプロジェクトのインポート (Importing an Existing Project into IAR EW for Synergy)	3
1.2 IAR EW for Synergy を使用したプロジェクトファイルの生成 (Generating the Project Files in the IAR EW for Synergy)	4
1.3 アプリケーションのビルド (Building the Application)	8
1.4 コマンドラインからのビルド (Building from the command line)	9
1.4.1 GUIX Studio のコマンドライン (GUIX Studio Command Line)	10
1.5 アプリケーションの実行 (Running the Application)	11
2. e ² studio ISDE でのプロジェクトのインポートとビルド (Importing and Building Projects with e ² studio ISDE)	12
2.1 e ² studio の ISDE への既存プロジェクトのインポート (Importing an Existing Project in e ² studio ISDE)	12
2.2 Synergy ライセンスのインストール (Installing the Synergy License)	15
2.3 ISDE を使用したプロジェクトファイルの生成 (Generating the Project Files in the ISDE)	17
2.4 アプリケーションのビルド (Building the Application)	19
2.5 コマンドラインからのビルド (Building from the command line)	19
2.5.1 プロジェクトのビルド方法 (How to build projects)	19
2.6 アプリケーションの実行 (Running the Application)	21
3. IAR コンパイラを使用してビルドを行うための e ² studio の構成 (Configuring e ² studio to build with the IAR compiler)	24
改訂履歴	28

1. IAR EW for Synergy でのプロジェクトのインポートとビルド (Importing and Building Projects with IAR EW for Synergy)

1.1 IAR EW for Synergy への既存のプロジェクトのインポート (Importing an Existing Project into IAR EW for Synergy)

1. IAR EW for Synergy を開いて起動します。
2. サンプルプロジェクト **NETX_DNS_DK-S7G2.zip** を何らかの解凍先フォルダに解凍 (unzip) します。
例: ...\Desktop\NETX_DNS_DK-S7G2。
3. **[File] (ファイル) > [Open Workspace] (ワークスペースを開く)** をクリックして、IAR EW ワークスペースファイル (.eww) を開きます。

NETX_DNS_DK-S7G2.zip または類似の zip プロジェクトの解凍先フォルダに移動します。

4. **NETX_DUO_DNS.eww** ワークスペースファイル (.eww) を選択し、**[Open] (開く)** をクリックします。

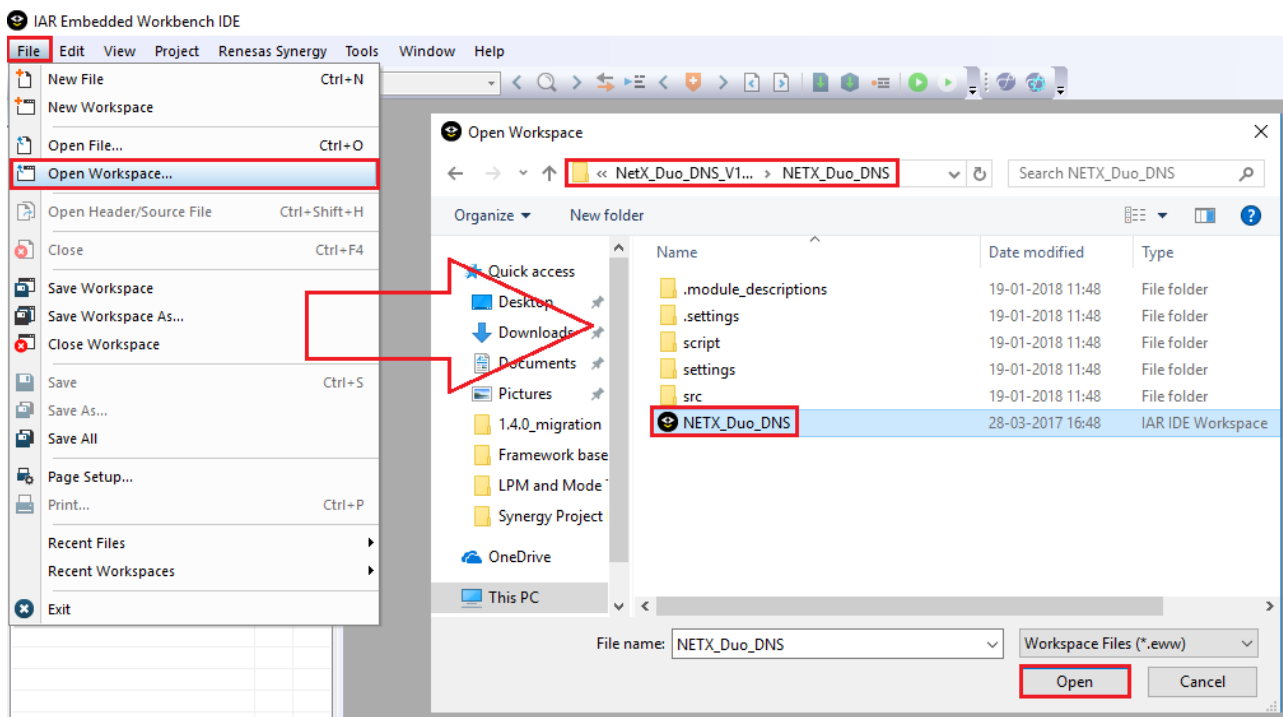


図 1.1 IAR EW for Synergy でワークスペースを開く

5. プロジェクトを開いた後、IAR EW for Synergy の ISDE で、プロジェクト構造 (図 1.2) が表示されます。

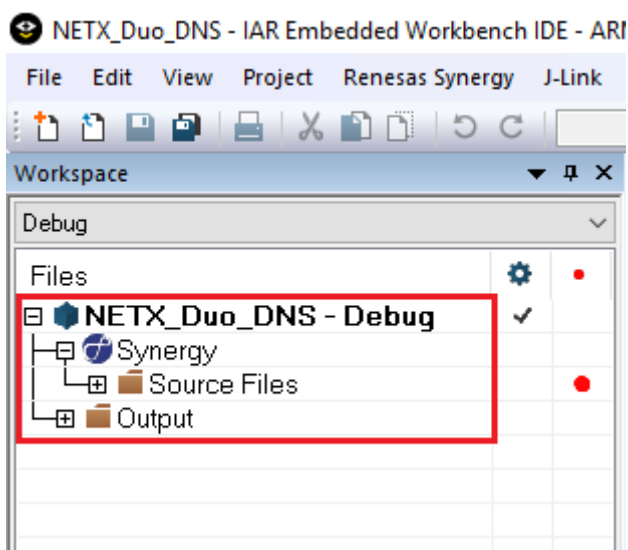


図 1.2 IAR EW for Synergy にロードされたプロジェクト

1.2 IAR EW for Synergy を使用したプロジェクトファイルの生成 (Generating the Project Files in the IAR EW for Synergy)

これで、プロジェクトを正常にロードできました。使用しているハードウェアに合わせてプロジェクトの構成を開始できます。

ファイルの生成を開始する前に、Renesas Synergy™ Standalone Configurator (SSC) と Synergy Software Package (SSP) へのパスを設定する必要があります。以下の手順を実行しない場合、デフォルトでパス設定作業を求められます。

プロジェクトファイルを生成するには、以下の手順に従います。

1. **[Renesas Synergy] > [Settings] (設定)** をクリックして、**[Renesas Synergy Settings] (Renesas Synergy の設定)** を開きます。ライセンスファイルと SSC/SSP のフォルダが既に構成されている場合、フォームのうち、**[License]** (ライセンス) 領域と SSC/SSP の場所が表示されます (図 1.3)。これらの設定が表示された場合、次のステップを省略し、ステップ 3 に進みます。

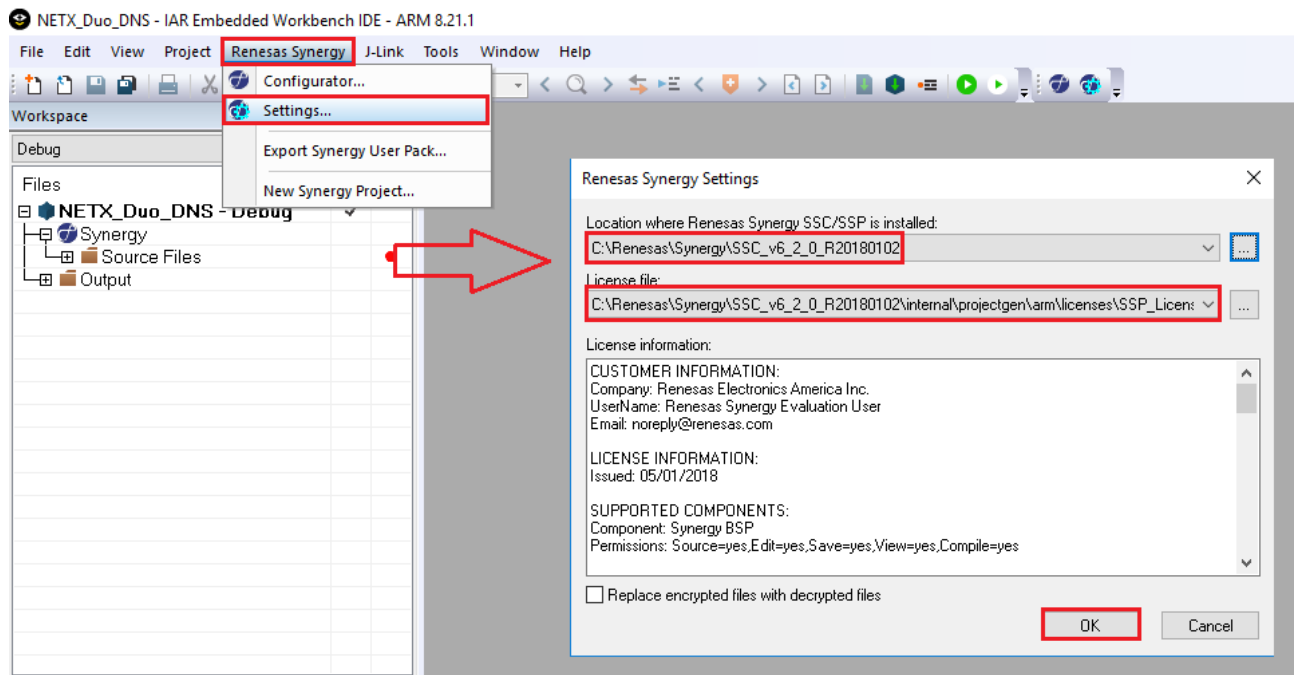


図 1.3 SSC/SSP の設定

2. フォームのうち、[License] (ライセンス) 領域と SSC/SSP の場所が空白であるか、「C:\Renesas\Synergy\SSC」を指していない場合、以下の手順 (A ~ F) を実行してください。これらの設定は、1 回だけ実行する必要があります。
- A. SSC/SSP の場所を指定するために、参照 <…> ボタンをクリックします。IAR EW for Synergy の IDE で、[Open] (開く) ダイアログボックスが表示されます。
- 注記: SSC/SSP をデフォルトの場所にインストールしている場合、SSC/SSP フォルダは「C:\Renesas\Synergy\SSC」ディレクトリにあります。

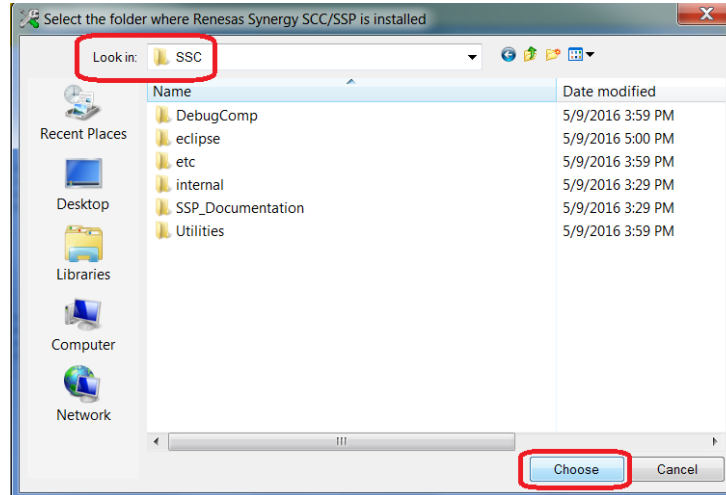


図 1.4 SSC フォルダ

- B. [Choose] (選択) をクリックして、SSC/SSP の場所を設定します。
- C. ライセンスファイルの場所を指定するために、[Browse <…>] (参照…) ボタンをクリックします。IAR EW for Synergy の IDE で、[Open] (開く) ダイアログボックスが表示されます。

注記: SSP ライセンスは、C:\Renesas\Synergy\SSC\internal\projectgen\arm\Licenses ディレクトリに配置されています。

- D. このディレクトリで、「SSP_License_Example_EvalLicence_*.xml」または「SSP_Development_and_Production_License_*.xml」を選択します。

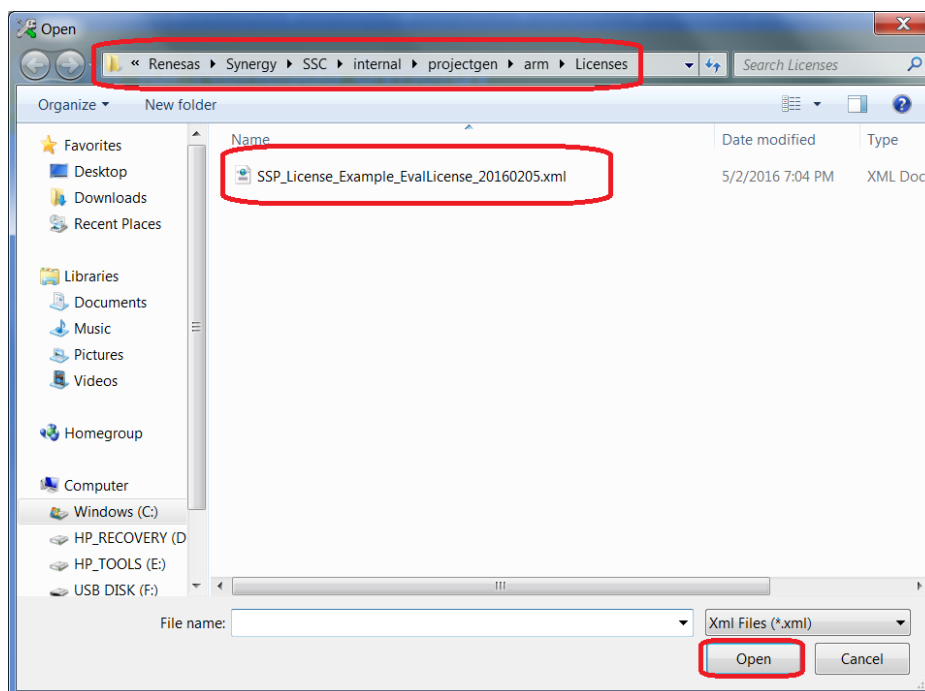


図 1.5 XML SSP ライセンスファイル

- E. **[Open]** (開く) をクリックしてライセンスファイルを設定し、構成ウィンドウで内容を確認して **[OK]** をクリックします。

IDE で、Renesas Synergy Settings と Synergy Configurator のそれぞれに対応するショートカットを利用することもできます。



図 1.6 SCC/SSP の設定に対応するショートカット

3. **[Renesas Synergy]** > **[Configurator]** (コンフィグレータ) をクリックして、Synergy™ Standalone Configurator を開きます。図 1.7 を参照してください。

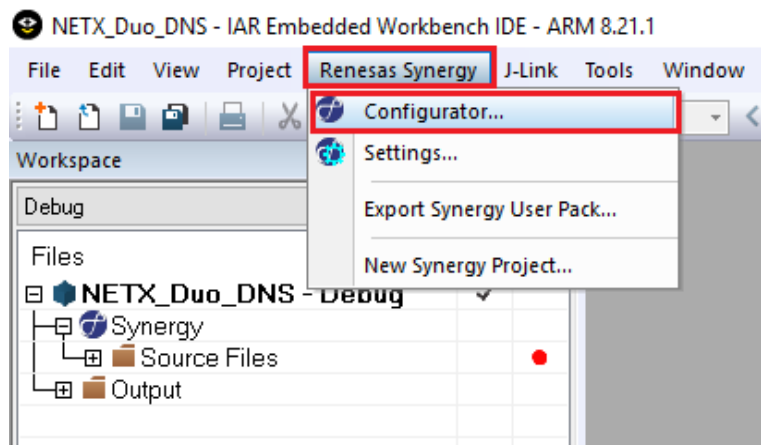


図 1.7 Synergy™ Standalone Configurator

注記: この時点で、「**synergy**」と「**synergy_cfg**」の各フォルダはまだ作成されていません。SSP が生成したファイルは、これら 2 個のフォルダに格納されます。次のステップで、これらのファイルを生成します。

- [Synergy Standalone Configuration] (Synergy スタンドアロン構成) ウィンドウ (図 1.8) で、[Generate Project Content] (プロジェクトコンテンツの生成) ボタンをクリックします。

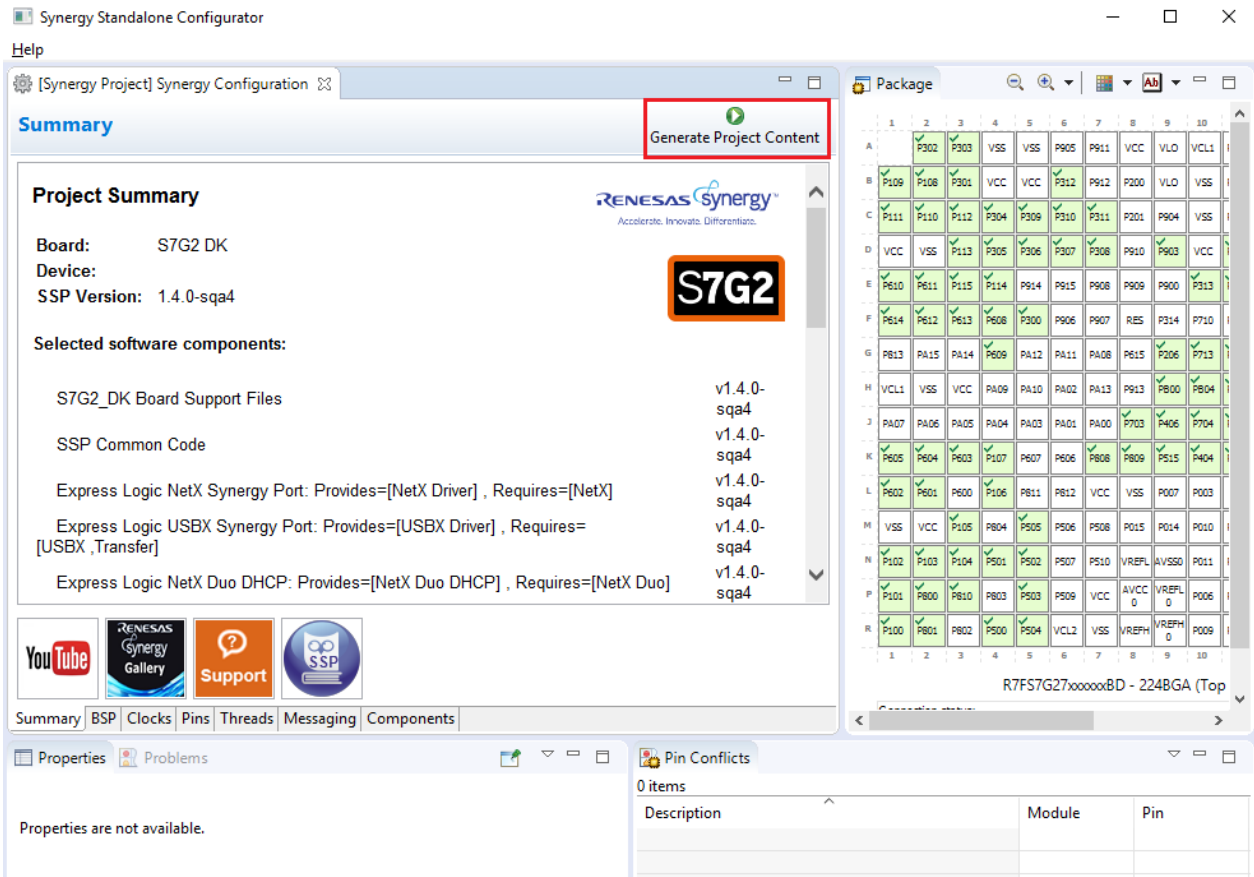


図 1.8 [Generate Project Content] (プロジェクトコンテンツの生成) ボタン

- [Synergy Standalone Configuration] (Synergy スタンドアロン構成) ウィンドウを閉じます。

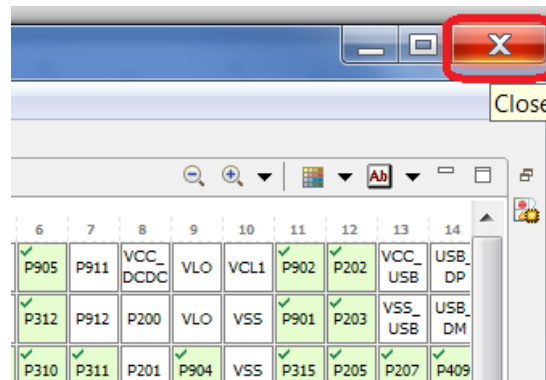


図 1.9 SCC ウィンドウを閉じる

6. プロジェクトは、図 1.10 に示すフォルダ構造になります。

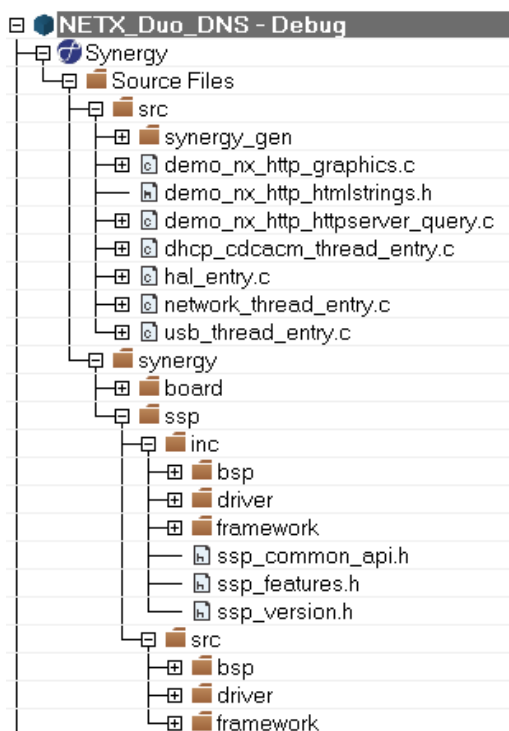


図 1.10 [Synergy Configuration] (Synergy 構成) タブ

1.3 アプリケーションのビルド (Building the Application)

1. メニューバーの **[Make]** (メイク) アイコンをクリックして、プロジェクトをビルドします。**F7** ショートカットキーを使用することもできます。

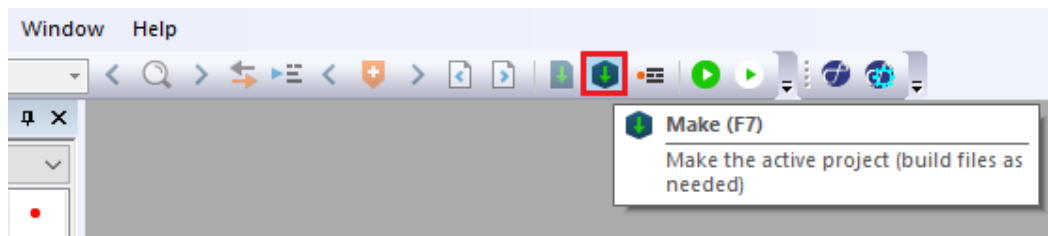


図 1.11 [Build] (ビルド) ボタン

2. ビルドが正常に終了すると、図 1.12 に類似した出力が生成されます。

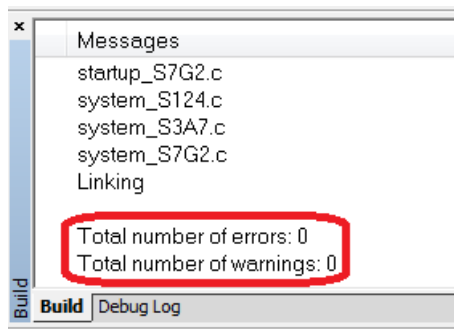


図 1.12 エラーのないビルド

注記：アプリケーションプロジェクトでサードパーティのコードを使用する場合、コンパイル時に警告が発生する可能性があります。それらの警告がプロジェクトの機能に影響を及ぼさない場合、この警告を無視できます。

1.4 コマンドラインからのビルド (Building from the command line)

コマンドラインからプロジェクトをビルドするには、common\bin ディレクトリに配置されている IAR コマンドラインビルドユーティリティ (IAR Command Line Build Utility) (iarbuild.exe) を使用します。通常、連続的な統合 (integration) の目的でテストを自動化する場合にこのユーティリティが役立ちます。

入力として、プロジェクトファイルを使用します。起動の構文は次のとおりです。

```
iarbuild project.ewp [ -clean | -build | -make | -cstat_analyze |
-cstat_clean] config[,config1,config2,...] *[-log
errors|warnings|info|all] [-parallel 数値] [-varfile ファイル名]
```

使用可能なパラメータは、以下のとおりです。

パラメータ	説明
-build	指定したビルド構成内にあるすべてのファイルの再ビルドと再リンクを実行します。
-make	直前のビルドより後に変更されたファイルのみのコンパイル、アセンブル、およびリンクを実行する方法で、指定したビルド構成を最新の状態にします。
-clear	中間ファイル (intermediate file) と出力ファイル (output file) を削除します。
-cstat_analyze	C-STAT を使用してプロジェクトを分析し、メッセージの数に関する情報を生成します。詳細については、『 C-STAT@ Static Analysis Guide 』を参照してください。
-cstat_clean	このプロジェクトに関する C-STAT メッセージデータベースをクリアします。詳細については、『 C-STAT@ Static Analysis Guide 』を参照してください。
config *	パラメータ「config」と、ビルドしようとする構成の名前を指定します。構成の名前は、事前定義の構成である「Debug」と「Release」のどちらか、または開発者が自分で定義した名前のいずれかです。ビルド構成の詳細は、『 IDE Project Management and Building Guide 』の 94 ページ「Projects and build configurations」を参照してください。* (ワイルドカード文字)、「-clean」、「-build」、「-make」の各コマンドは、プロジェクト内で定義されているすべての構成を処理します。
-log errors	ビルドのエラーメッセージを表示します。
-log warnings	ビルドの警告メッセージとエラーメッセージを表示します。
-log info	ビルドの警告メッセージとエラーメッセージ、さらに #pragma メッセージプリプロセッサディレクティブ (message preprocessor directive) を使用して発行したメッセージを表示します。
-log all	ビルドから生成されたすべてのメッセージを表示します。例えば、コンパイラのサインオン情報 (sign-on information) やコマンドライン全体です。
-parallel 数値	CPU 内の複数のコアをより適切に活用してコンパイラを実行できるように、並列プロセス数を指定します。
-varfile ファイル名	使用するファイルを指定する方法で、引数で指定したカスタム定義の変数を、ビルドエンジンから利用できるワークスペースの範囲内で定義済みにします。『 IDE Project Management and Building Guide 』の 87 ページ「Configure Custom Argument Variables dialog box」を参照してください。

コマンドシェル (command shell) からファイルを指定せずにアプリケーションを実行した場合、使用可能なパラメータとそれらの構文を示すサインオンメッセージ (sign-on message) が表示されます。

1.4.1 GUIX Studio のコマンドライン(GUIX Studio Command Line)

GUIX Studio は、いくつかのコマンドラインオプションを用意しています。コマンドライン引数によっては、GUI インタフェースが起動せず、代わりに .gxp プロジェクトをロードし、要求された出力ファイルを生成するのみの動作になります。

コマンドラインの使用法

使用法: `guix_studio [OPTION] [ARGUMENT]`

1. .gxp プロジェクトを開きます。
2. 指定したプロジェクトをロードし、指定した出力ファイルを生成します。

例:

コマンドライン: `demo.gxp`

`demo.gxp` プロジェクトを開きます。

コマンドライン: `guix_studio.exe -p demo.gxp`

`demo.gxp` プロジェクトを開きます。

コマンドライン: `guix_studio.exe -n -p demo.gxp`

`demo.gxp` プロジェクトに対応するすべての出力ファイルを生成します。

コマンドライン: `guix_studio.exe -n -r -p demo.gxp`

`demo.gxp` プロジェクトに対応するすべてのリソースファイルを生成します。

表 1.1 コマンドラインオプション

オプション	説明
-n --nogui	「GUI を使用しない」オプションです。guix_studio.exe の Win32 バージョンに対し、Studio UI インタフェースを起動せず、コマンドラインのみを実行するように指示します。
-o パス名 --log	ログオプション。ログファイルを指定します。
-b --binary	バイナリリソースオプション。C ファイルではなく、バイナリリソースファイルを生成します。
-d display1, display2 --display	表示名オプション。このオプションを指定した場合、生成されるリソースファイルまたは仕様ファイルすべてに、指定した表示名のみが含まれます。このオプションを使用しなかった場合、すべての表示名が含まれます。
-t theme1, theme2 --theme	テーマ名 (theme name) オプション。このオプションを指定した場合、生成されるリソースファイルまたは仕様ファイルすべてに、指定したテーマ名のみが含まれます。このオプションを使用しなかった場合、すべてのテーマ名が含まれます。
-l langage1, language2 --language	言語名 (language name) オプション。このオプションを指定した場合、生成されるリソースファイルまたは仕様ファイルすべてに、指定した言語名のみが含まれます。指定しなかった場合、すべての言語名が含まれます。
-r [filename] --resource	リソースオプション。Studio に対し、すでに指定した表示名、テーマ名、言語名に対応するリソースファイルを生成するように指示します。
-s [filename] --specification	仕様オプションです。Studio に対し、すでに指定した表示名、テーマ名、言語名に対応する仕様ファイルを生成するように指示します。
-p project_pathname --project	プロジェクトのパス名オプションです。ロードする .gxp プロジェクトを指定します。

1.5 アプリケーションの実行(Running the Application)

これで、アプリケーションを対象ハードウェア上で実行する準備ができました。プロジェクト設定はいずれも生成されており、デフォルトのデバッグプロブは J-Link ARM になっています。デバッグを実施するときに、デバッグドライバを変更して、IAR I-jet または I-jet Trace を使用することもできます。

1. デバッグプロブ (図 1.13) を確認するために、[Project] (プロジェクト) > [Options] (オプション) > [Debugger] (デバッガ) > [Setup] (設定) > [Driver] (ドライバ) をクリックします。

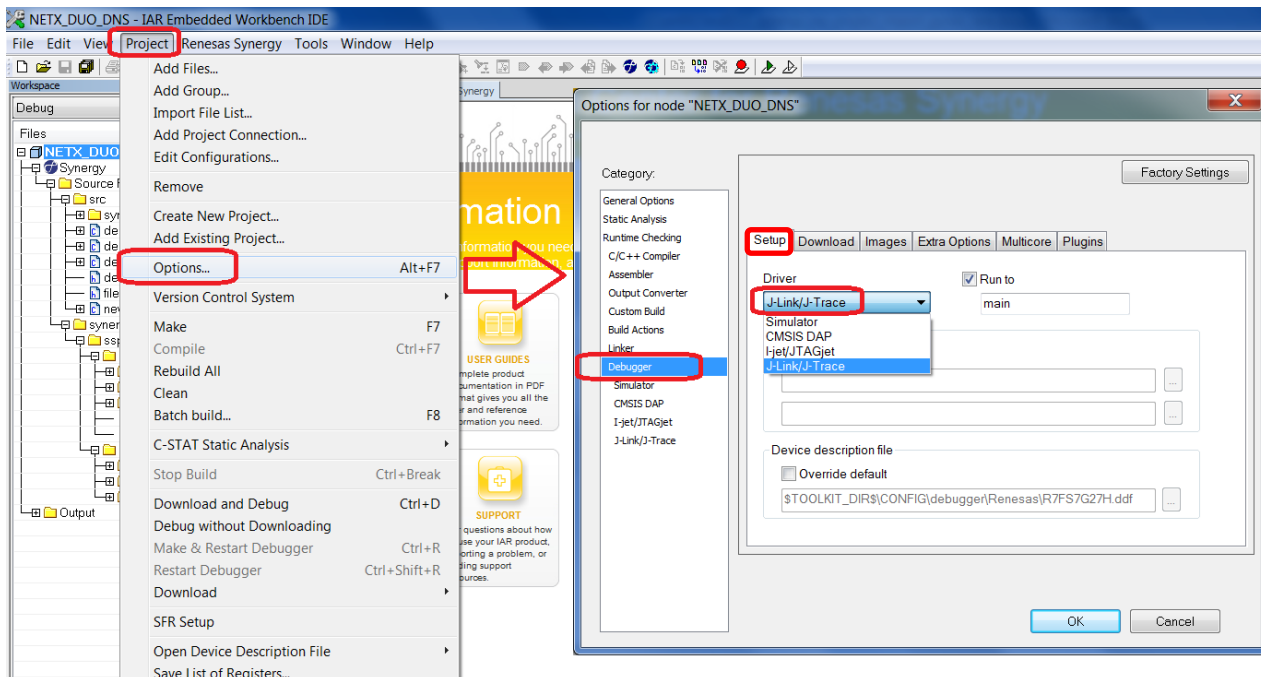


図 1.13 デバッガのセットアップ 1

2. [J-Link] または [IAR I-jet] ドライバが選択されていることを確認するために、[OK] をクリックします。
3. CTRL+D キーを押すか、[Download and Debug] (ダウンロードしてデバッグ) ボタン (図 1.14) をクリックして、デバッグを開始します。

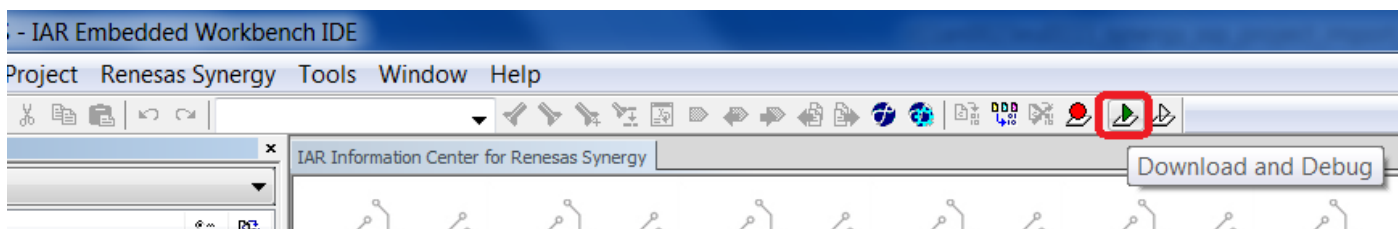


図 1.14 デバッガのセットアップ 2

4. F5 キーを押すか、[Go] ボタン (図 1.15) をクリックして、アプリケーションを起動します。



図 1.15 [Run] (実行) ボタン

注記: この時点で、アプリケーションは対象ハードウェア上で実行されています。各種デバッグコントロール (図 1.16) を使用して、アプリケーションを一時停止、停止、および再開することができます。

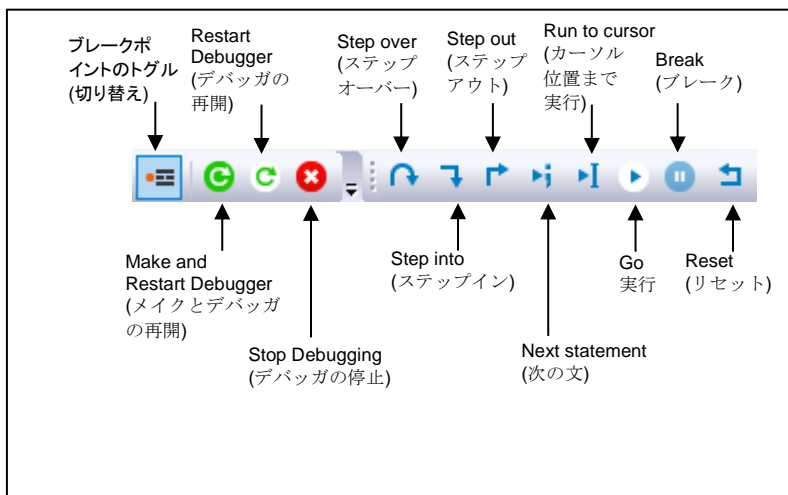


図 1.16 デバッグコントロール

5. **Ctrl + Shift +D** キーを押すか、**[Stop]** ボタンをクリックして、デバッグセッションを終了します。

2. e² studio ISDE でのプロジェクトのインポートとビルド (Importing and Building Projects with e² studio ISDE)

2.1 e² studio の ISDE への既存プロジェクトのインポート (Importing an Existing Project in e² studio ISDE)

1. e² studio を開いて起動します。
2. ファイルのインポート先として使用するワークスペースを開き、ステップ D に進みます。インポート先のワークスペースを開く準備ができていない場合、以下のステップを実行します。
 - A. e² studio の起動が完了した後、図 2.1 のに示す **[Workspace Launcher]** (ワークスペースランチャ) ダイアログボックスが表示されます。

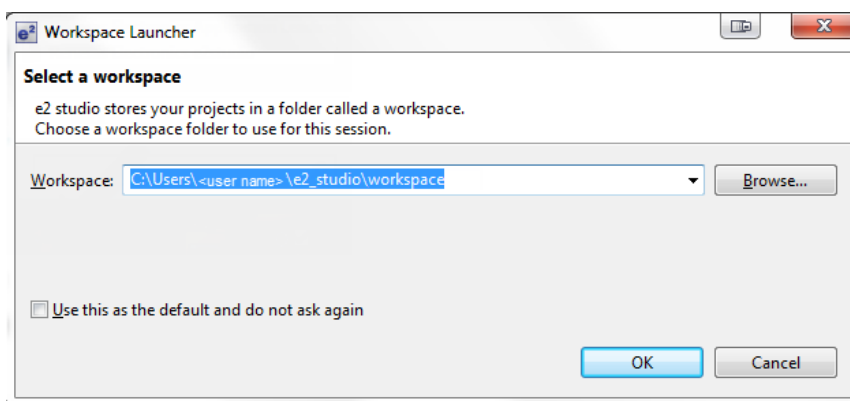


図 2.1 [Workspace Launcher] (ワークスペースランチャ) ダイアログ

- B. このダイアログボックスが表示されない場合、表示をオフにした可能性があります。その場合、希望のプロジェクトを開き、ステップ D に進みます。このダイアログボックスが表示された場合、引き続き以下のステップを実行します。

- C. [Workspace Launcher] (ワークスペースランチャ) ダイアログに、新しいワークスペースの名前を入力します (図 2.2)。

e² studio は、入力された名前を使用して新しいワークスペースを作成します。

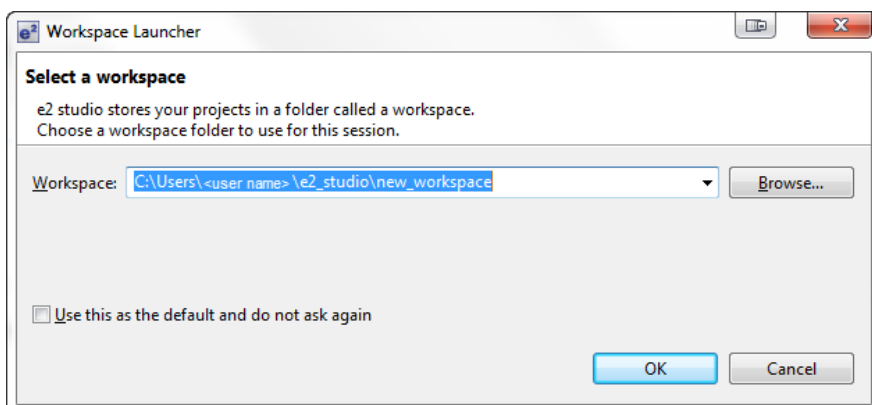


図 2.2 [Workspace Launcher] (ワークスペースランチャ) ダイアログ

- D. [OK] をクリックします。
E. ワークスペースが開いた時点で、[Welcome] (ようこそ) ウィンドウが表示されることがあります。その場合、[Workbench] (ワークベンチ) 矢印 (ボタン) をクリックし、[Welcome] (ようこそ) ウィンドウから先へ進みます (図 2.3)。

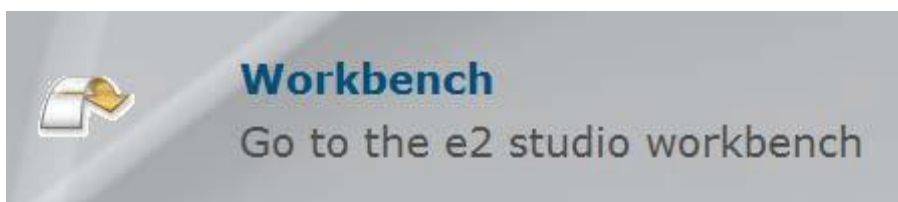


図 2.3 [Workbench] (ワークベンチ) 矢印 (ボタン)

3. この時点で、プロジェクトのインポート先として使用するワークスペースが開いています。メニューバーの [File] (ファイル) をクリックします (図 2.4)。

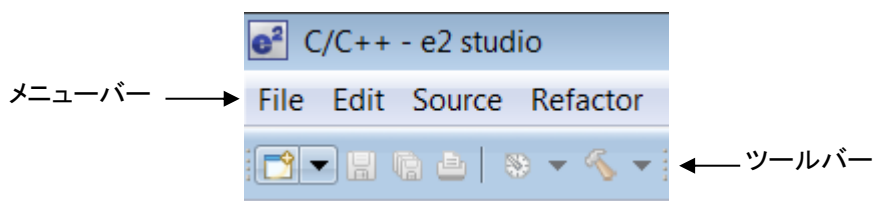


図 2.4 メニューとツールバー

4. **[File]** (ファイル) ドロップダウンメニューの **[Import]** (インポート) をクリックします (図 2.5)。

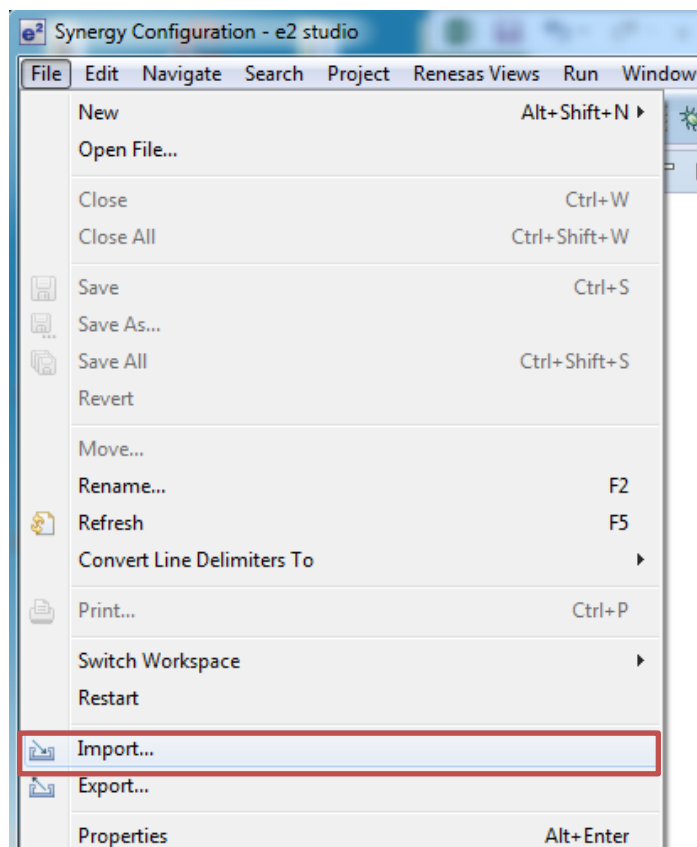


図 2.5 [File] (ファイル) ドロップダウンメニュー

5. 図 2.6 に示す [Import] (インポート) ダイアログボックスで、**[General]** (一般) オプション、次に **[Existing Projects into Workspace]** (既存プロジェクトをワークスペースへ) を選択し、現在のワークスペースにインポートするプロジェクトを選択します。

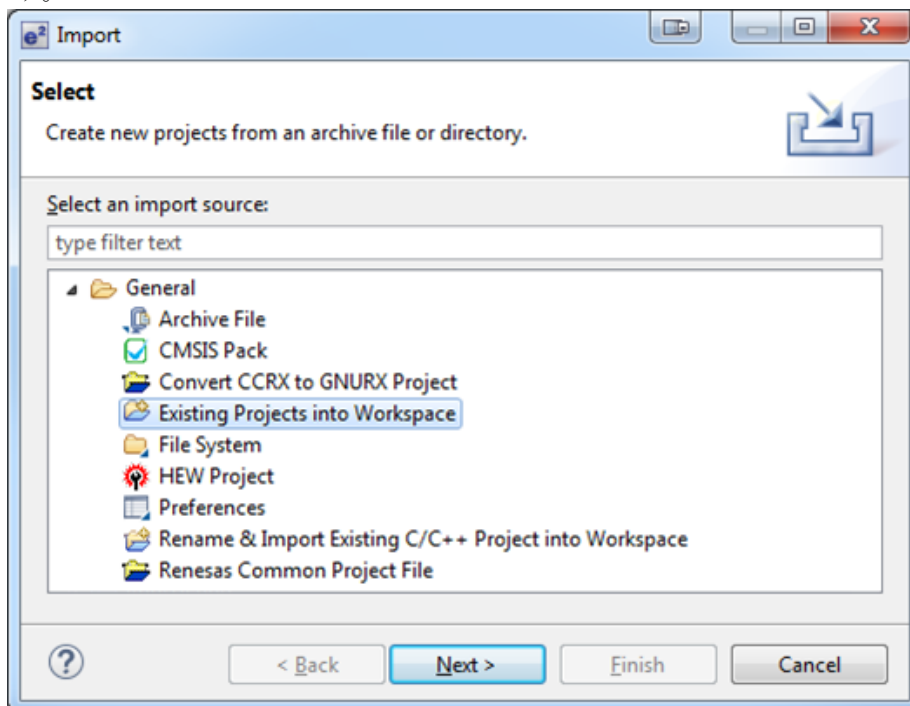


図 2.6 [Project Import] (プロジェクトのインポート) ダイアログで [Existing Projects into Workspace] (既存のプロジェクトをワークスペースに) オプションを選択している状態

6. **[Next]** (次へ) をクリックします。
7. **[Select archive file]** (アーカイブファイルの選択) をクリックします (図 2.7)。

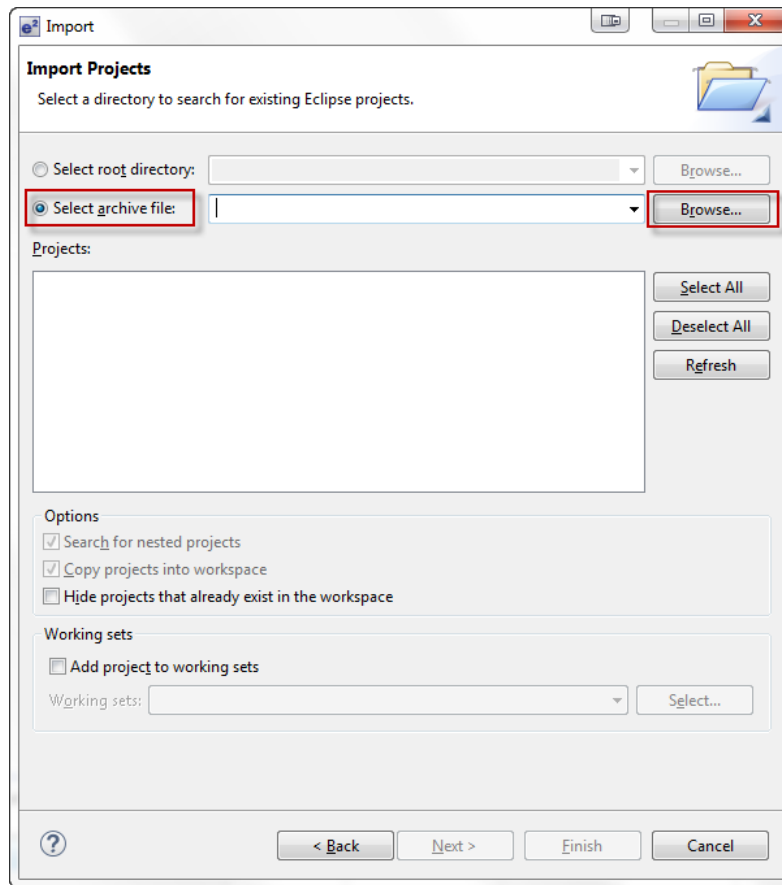


図 2.7 既存プロジェクトのインポート用ダイアログ 1

8. **[Browse]** (参照) をクリックします。
9. インポートするプロジェクトに対応する zip ファイルを格納しているフォルダを参照します。
10. インポートするファイルを選択します。この例では、「NETX_DNS_DK-S7G2.zip」を使用します。
11. **[Open]** (開く) をクリックします。
12. **[Projects]** (プロジェクト) リストで、インポートするプロジェクトを選択します (図 2.8)。

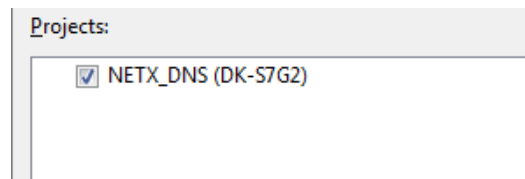


図 2.8 既存プロジェクトのインポート用ダイアログ 2

13. **[Finish]** (終了) をクリックしてプロジェクトをインポートします。

2.2 Synergy ライセンスのインストール (Installing the Synergy License)

サンプルアプリケーションをビルドして実行するには、e² studio に Synergy ライセンスをインストールする必要があります。このライセンスをまだインストールしていない場合、サンプルアプリケーションをインポートした後、ISDE の右下隅に黄色いウィンドウが表示されます。

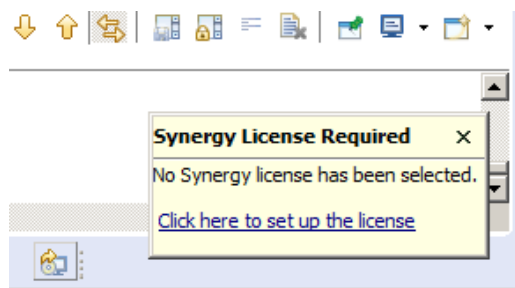


図 2.9 [Synergy License Required] (Synergy ライセンスが必要) ウィンドウ

2. ライセンスをインストールするには、[Click here to set up the license] (ライセンスをセットアップするにはここをクリック) リンクを選択します。その結果、[Synergy License] (Synergy ライセンス) セットアップウィンドウが表示されます。

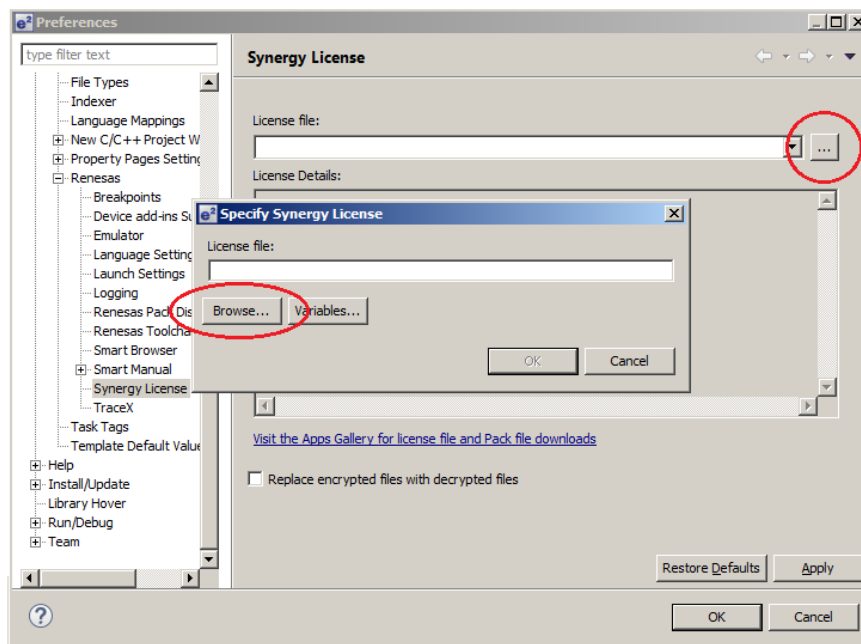


図 2.10 Synergy ライセンスファイル

[Browse] (参照) または [...] ボタンを合計 2 回クリックすると、デフォルトの Synergy License が格納されている場所が表示されます。このファイルを選択し、ライセンスをインストールします。インストールが完了した時点で、黄色いウィンドウは表示されなくなり、ビルドを開始できます。

2.3 ISDE を使用したプロジェクトファイルの生成 (Generating the Project Files in the ISDE)

ここまでで、プロジェクトを正常にインポートできました。ハードウェアに合わせてプロジェクトの構成を開始できます。

1. 図 2.11 の [Project Explorer] (プロジェクトエクスプローラ) で、左にある矢印をクリックし、プロジェクトを展開します。



図 2.11 折りたたまれた状態の [Project Explorer] (プロジェクトエクスプローラ)

2. [Synergy Configuration] (Synergy 構成)を開きます。[Synergy Configuration] (Synergy 構成)がまだ開いていない場合は、[Project Explorer] (プロジェクトエクスプローラ) ウィンドウで **configuration.xml** ファイルをダブルクリックして開きます (図 2.12)。

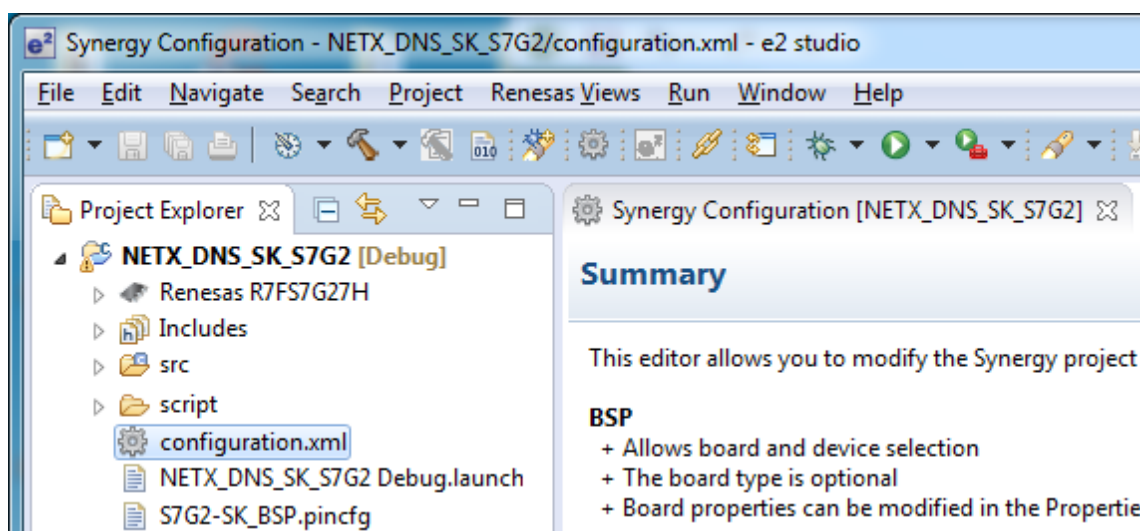


図 2.12 [Project Explorer] (プロジェクトエクスプローラ)

注記: この時点で、「**synergy**」と「**synergy_cfg**」の各フォルダはまだ作成されていません。e² studio と SSP が生成したファイルは、これら 2 個のフォルダに格納されます。次のステップで、これらのファイルを生成します。

3. [Synergy Configuration] (Synergy 構成) ウィンドウ (図 2.13) で、[Generate Project Content] (プロジェクトコンテンツの生成) ボタンをクリックします。

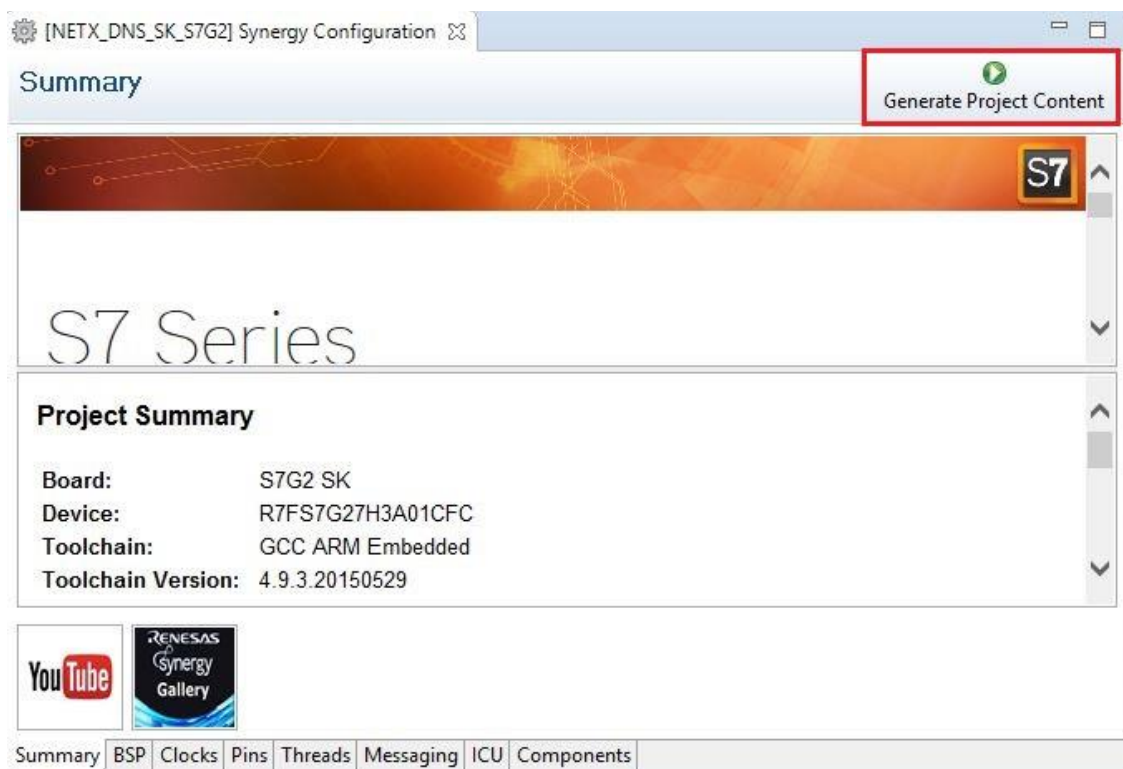
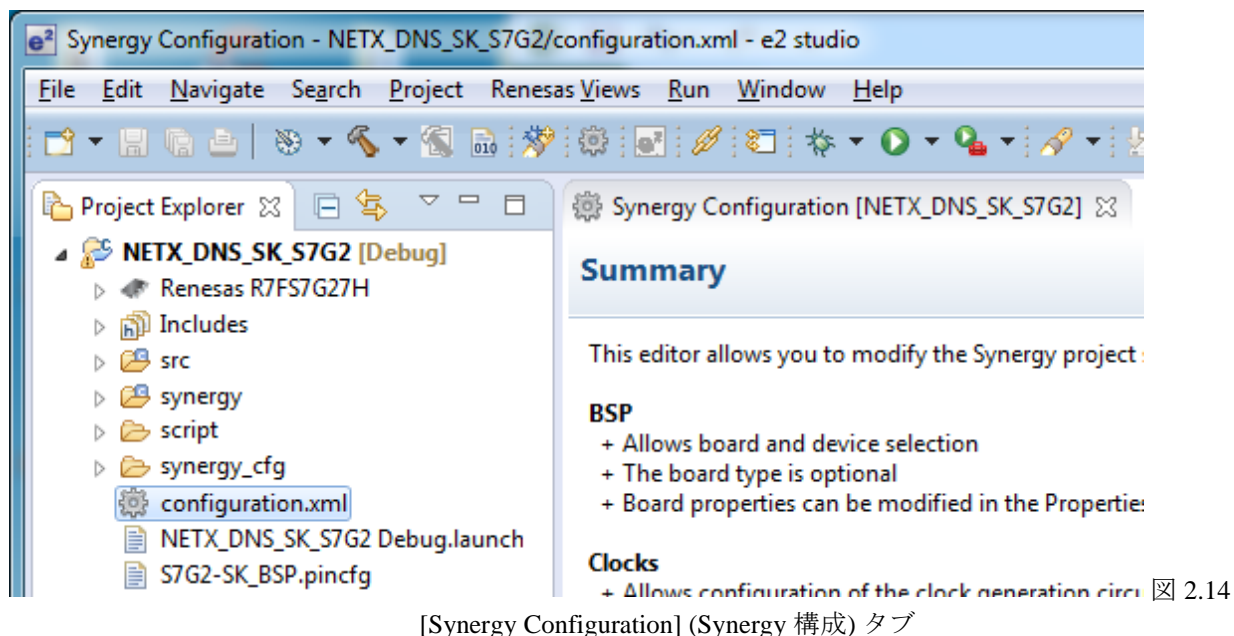


図 2.13 [Generate Project Content] (プロジェクトコンテンツの生成) ボタン

4. プロジェクトは、図 2.14 に類似したフォルダ構造になります。



[Synergy Configuration] (Synergy 構成) タブ

2.4 アプリケーションのビルド(Building the Application)

ツールバーのハンマーアイコン (図 2.15) をクリックして、プロジェクトをビルドします。

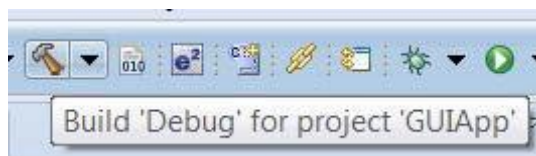


図 2.15 [Build] (ビルド) ボタン

ビルドが正常に終了すると、図 2.16 に類似した出力が生成されます。

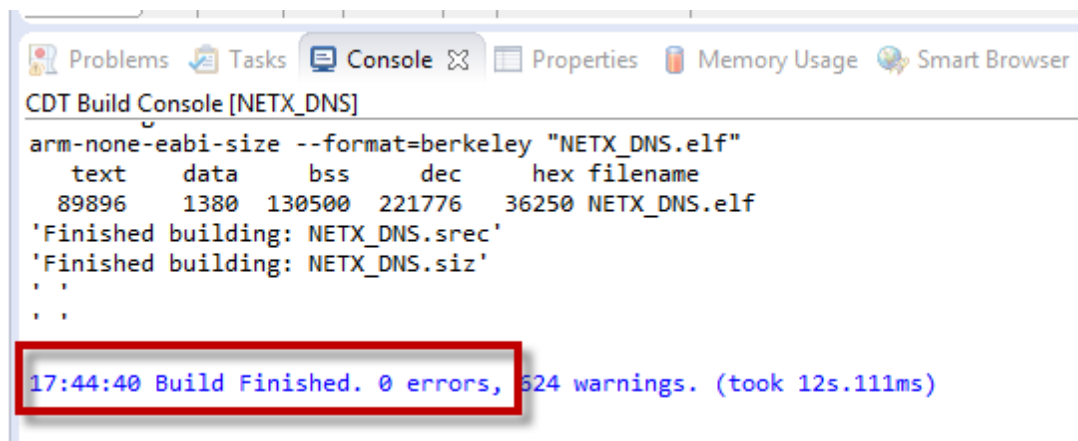


図 2.16 エラーのないビルド

2.5 コマンドラインからのビルド(Building from the command line)

ヘッドレスビルド (Headless build) とは基本的に、e² studio UI を使用せずにコマンドラインからプロジェクトをビルドすることを意味します。Hudson または Jenkins のような連続的統合ツールを使用してビルドを自動化しようとする場合に、この方法が役立ちます。

2.5.1 プロジェクトのビルド方法 (How to build projects)

ヘッドレスビルドを使用してプロジェクトをビルドするには、eclipsesec.exe という eclipse のコマンドラインバージョンが必要です。

まず、ヘッドレスビルドがプロジェクトをビルドできるように、既存のプロジェクトをワークスペースインポートしてビルドする必要があります。この作業を実行するには、次のコマンドを使用します。

```
eclipsesec.exe -nosplash -debug -consolelog -application  
org.eclipse.cdt.managedbuilder.core.headlessbuild -data c:\work\project\test -  
import [プロジェクトの位置] -build all
```

このコマンドは、test というワークスペースにプロジェクトをインポートするほか、すべての構成のプロジェクトもビルドできます。1 つのワークスペース内にあるすべてのプロジェクトをインポートする場合は、「-import」スイッチを「-importAll」に差し替えて、すべてのプロジェクトが存在しているパスを定義するだけです。

「-consolelog」スイッチを指定したので、コンパイル出力やメイク出力など、すべてのビルド出力はコンソールに表示されます。ヘッドレスビルドで使用できるスイッチのリストは、表 2.1 を参照してください。

表 2.1 ヘッドレスビルドのオプションに関する説明

オプション	説明
-nosplash	e ² studio/eclipse のスプラッシュ画面を省略します
-consolelog	すべてのログ出力を System.out に送ります。通常、これはコマンドシェル自体を指します
-debug	すべての情報メッセージ、警告メッセージ、またはエラーメッセージを、コンソールに出力します
-application	アプリケーションの識別子 (identifier) を定義します。この場合、cdt managedbuilder になります。
-import {[uri:]/path/to/project}	指定したパス内で定義されているプロジェクトをインポートします
-importall {[uri:]/path/to/projectTreeURI}	URI の下にあるすべてのプロジェクトをインポートします
-build {project_name_reg_ex{/config_reg_ex} all}	選択した構成に基づき、定義済みのプロジェクトをビルドします
-cleanBuild {project_name_reg_ex{/config_reg_ex} all}	構成に基づき、定義済みのプロジェクトのクリーンとビルドを実施します
-I {include_path}	ツールに追加する付加的な include_path を指定します
-include {include_file}	ツールに渡す付加的な include_file を指定します
-D {preproc_define}	ツールに渡す付加的なプリプロセッサ用 define を指定します
-E {var=value}	すべてのツールを実行するときに、環境変数の値の置換/追加を行います
-Ea {var=value}	すべてのツールを実行するときに、環境変数に値を追加します
-Er {var}	特定の環境変数の削除/設定解除 (unset) を行います
-Ep {var=value}	すべてのツールを実行するときに、環境変数の先頭に値を追加 (prepend) します
-T {toolid} {optionid=value}*	ビルドする構成ごとに、ツールオプションの値を置換します
-Ta {toolid} {optionid=value}*	ビルドする構成ごとに、ツールオプションに値を追加します
-Tp {toolid} {optionid=value}*	ビルドする構成ごとに、ツールオプションの先頭に値を追加します
-Tr {toolid} {optionid=value}*	ビルドする構成ごとに、ツールオプションを削除します

注記: * ツールオプションで指定した値は、オプションの種類に基づき、1 個の文字列、複数の文字列をカンマで区切ったリスト、またはブール値 (Boolean) として解析されます。

Hudson/Jenkins にヘッドレスビルドを追加する方法

Hudson または Jenkins ビルドシステムにヘッドレスビルド機能を追加するには、「Execute Windows batch command」(Windows バッチコマンド実行) というビルドステップを追加する必要があります。すべての出力が Hudson または Jenkins のビルド出力に表示されるように、「-consolelog」を有効にしてあることを確認してください。このビルド出力に対し、Hudson/Jenkins で利用できる多くのプラグインのいずれかを使用し、出力の解析やカラーコード化を行うことで、カスタマイズできます。

図 2.17 に、Hudson 3.1.0 ジョブに対して新しいビルドステップを追加した様子を示します。ジョブ内で環境変数を使用することで、ビルドステップをさらにカスタマイズすることができます。

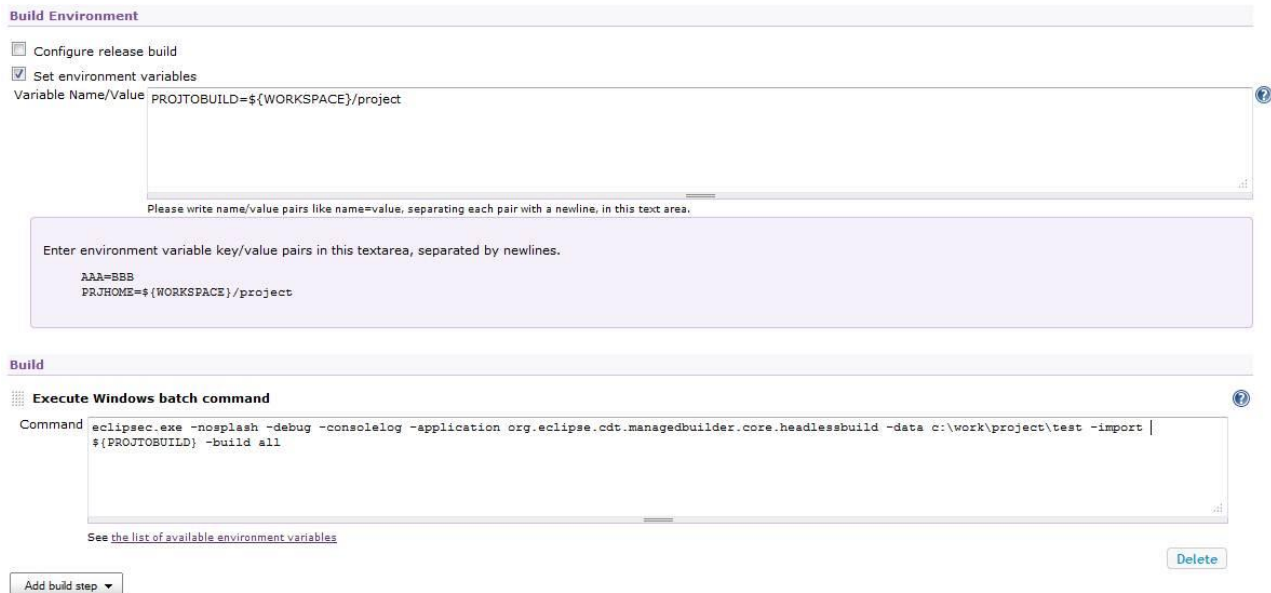


図 2.17 Hudson 内のヘッドレスビルドジョブの定義

GUIX Studio のコマンドライン

GUIXStudio コマンドラインの詳細については、「1.4 コマンドラインからのビルド (Building from the command line)」を参照してください。

2.6 アプリケーションの実行 (Running the Application)

これで、アプリケーションを対象ハードウェア上で実行する準備ができました。

1. デバッグアイコンをクリックし、対応するドロップダウンメニューを表示します (図 2.18)。

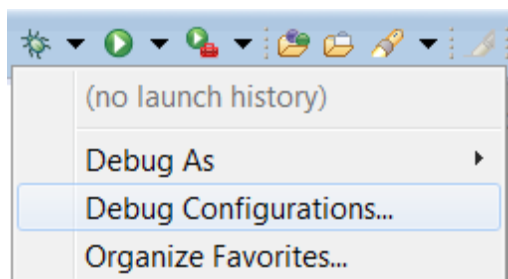


図 2.18 デバッグオプション

2. **[Debug Configurations...]** (デバッグ構成...) オプションを選択します。
3. **[Renesas GDB Hardware Debug]** (Renesas GDB ハードウェアデバッグ) セクションで、プロジェクトの名前を選択します。この例では、**NETX_DNS_SK_S7G2 Debug** です。
4. 「.elf」が付いているファイル名が、デバッグフォルダ内にあるプロジェクトによって生成されたファイルに一致していることを確認します。

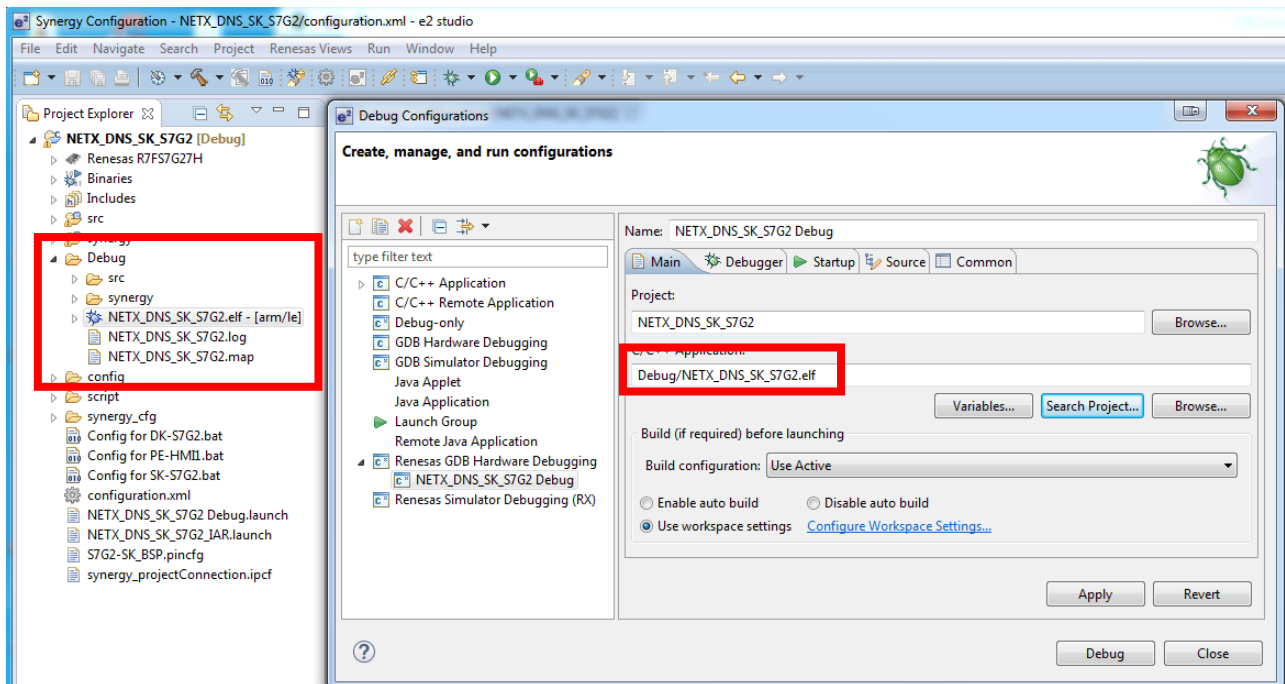


図 2.19 [Debug Configurations] (デバッグ構成) ウィンドウ

5. ダイアログボックスの右側にある **[Debugger]** (デバッガ) タブをクリックします。
6. **[Debug hardware]** (デバッグハードウェア) 項目が、**[J-Link ARM]** に設定されていることを確認します。この値になっていない場合、ドロップダウンメニューを使用して、この値に変更します。
7. **[Target Device]** (対象デバイス) 項目が対象ハードウェアに一致していることを確認します。一致していない場合、**[...]** ボタンをクリックし、Synergy デバイスリストから正しい対象デバイスを選択します。

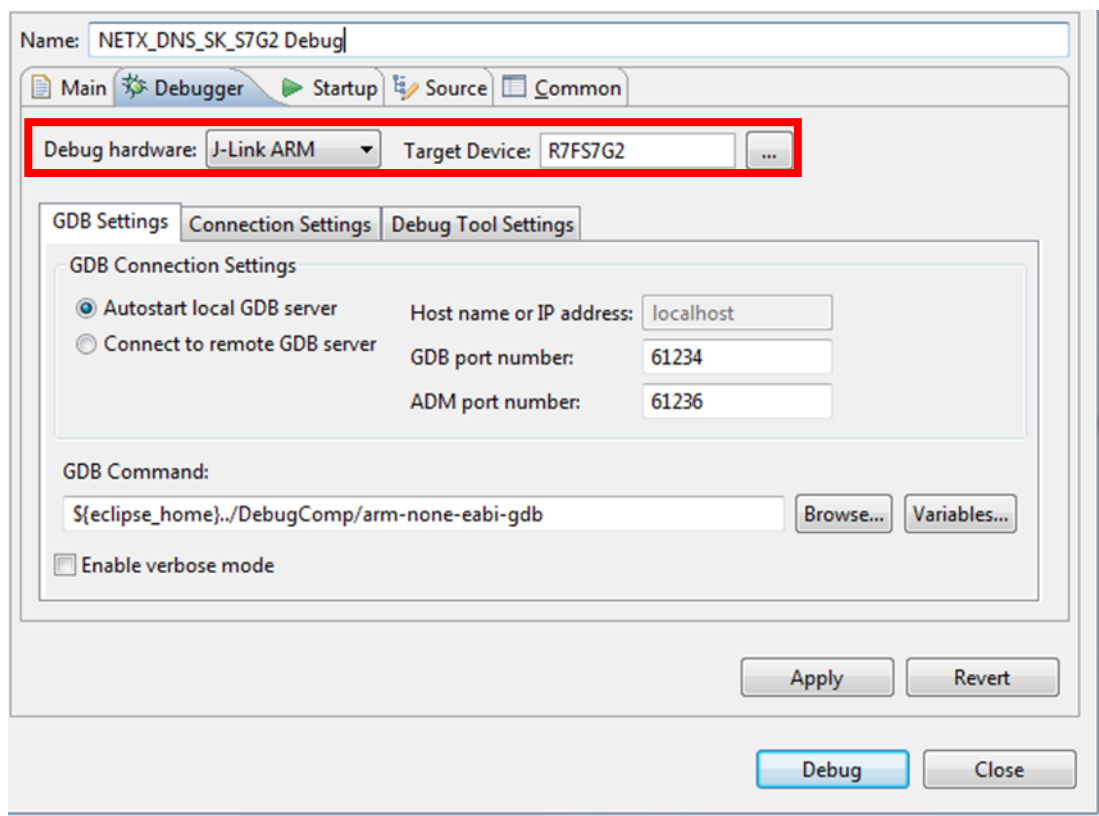


図 2.20 [Debug Configurations] (デバッグ構成) の [Debugger] (デバッガ) 設定

8. **[Debug]** (デバッグ) ボタンをクリックしてデバッグを開始します。



図 2.21 **[Debug]** (デバッグ) ボタン

9. **[Yes]** (はい) をクリックして、**[Debug]** (デバッグ) パースペクティブを開きます。

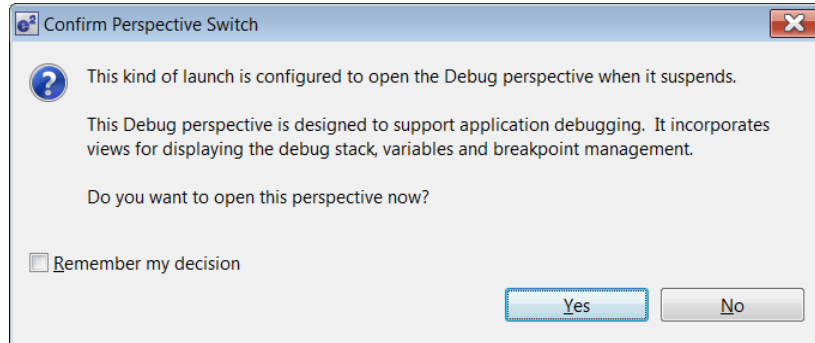


図 2.22 **[Perspective Switch]** (パースペクティブへの切り替え) ダイアログ

10. **F8** キーを押すか、**[Resume]** (再開) ボタンをクリックして、アプリケーションを起動します。

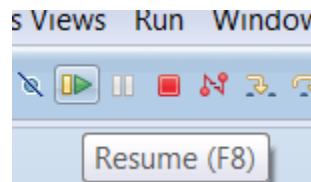


図 2.23 **[Resume]** (再開) ボタン

11. **F8** キーを押すか、メインウィンドウにある **[Resume]** (再開) ボタンをクリックして、実行を続行します。

注記: この時点で、アプリケーションは対象ハードウェア上で実行されています。各種デバッグコントロール (図 2.24) を使用して、アプリケーションを一時停止、停止、および再開することができます。

12. **Ctrl + F2** キーを押すか、**[Stop]** ボタンをクリックして、デバッグセッションを終了します。
13. **[Synergy Configuration]** (Synergy 構成) ボタンをクリックし、Synergy パースペクティブに戻ります。

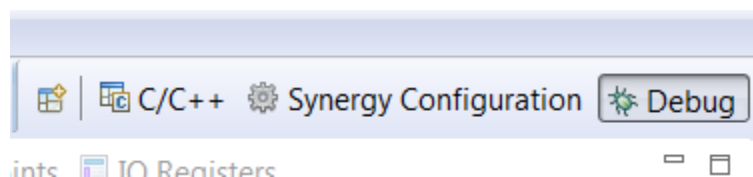


図 2.24 **[Perspective]** (パースペクティブ) のオプション

3. IAR コンパイラを使用してビルドを行うための e² studio の構成 (Configuring e² studio to build with the IAR compiler)

(注) 本章の詳細は下記ドキュメントを参照してください。

(<https://www.renesas.com/jp/ja/doc/products/renesas-synergy/apn/r11an0272eu0103-synergy-e2studio-compiler-install.pdf>)

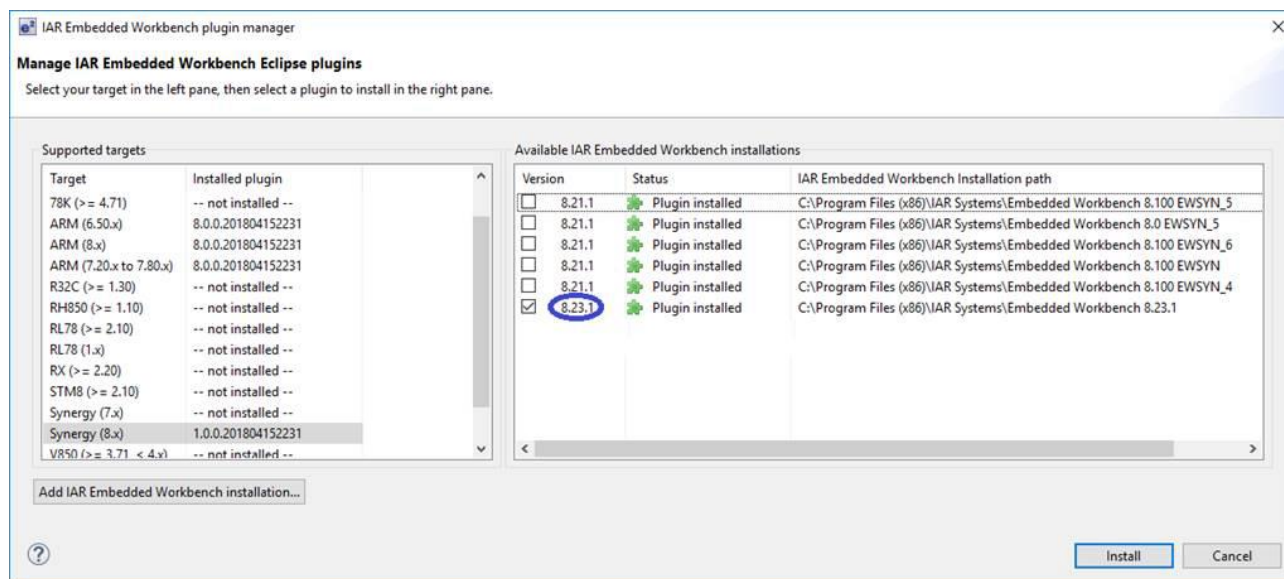
e² studio の ISDE は、デフォルトで GNU コンパイラを使用してビルドを行います。ただし、代わりに IAR コンパイラを使用するように e² studio を構成することも可能です。使用できるのは、Embedded Workbench for Synergy に付属しているコンパイラのみです。

組み合わせで動作させることができる 2 つの IDE には、以下のバージョンがあります。

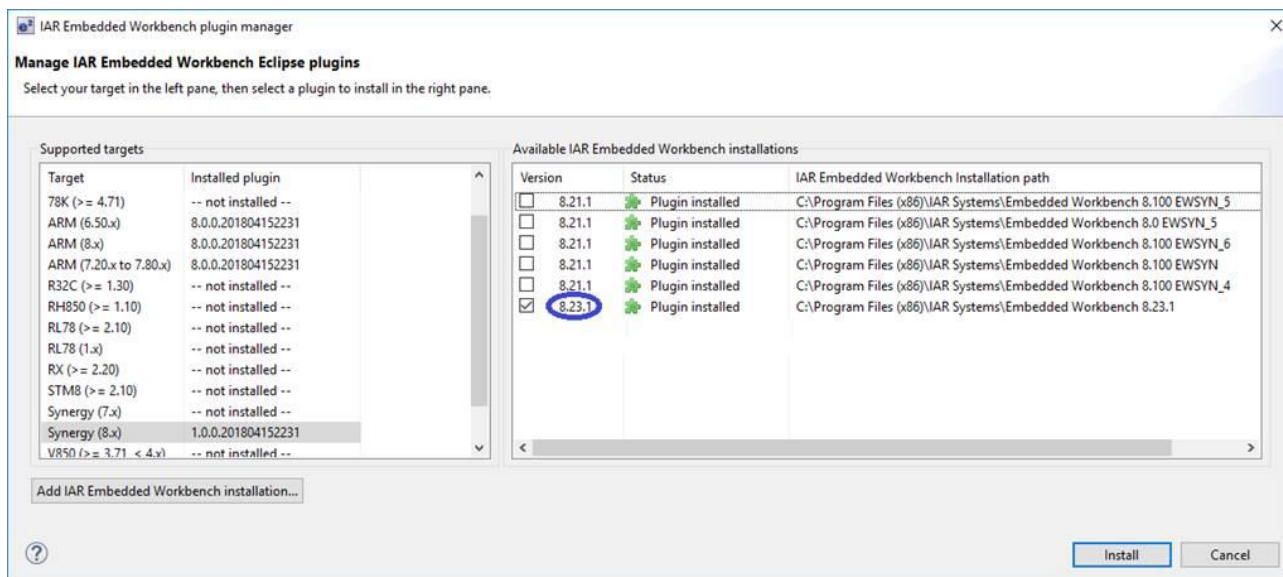
e ² studio	IAR EW for Synergy
v6.2.0	v7.71.3 / v8.21.1
v6.2.1	V8.23.1

IAR コンパイラを使用してビルドを行うために e² studio を構成する方法は、以下のとおりです。

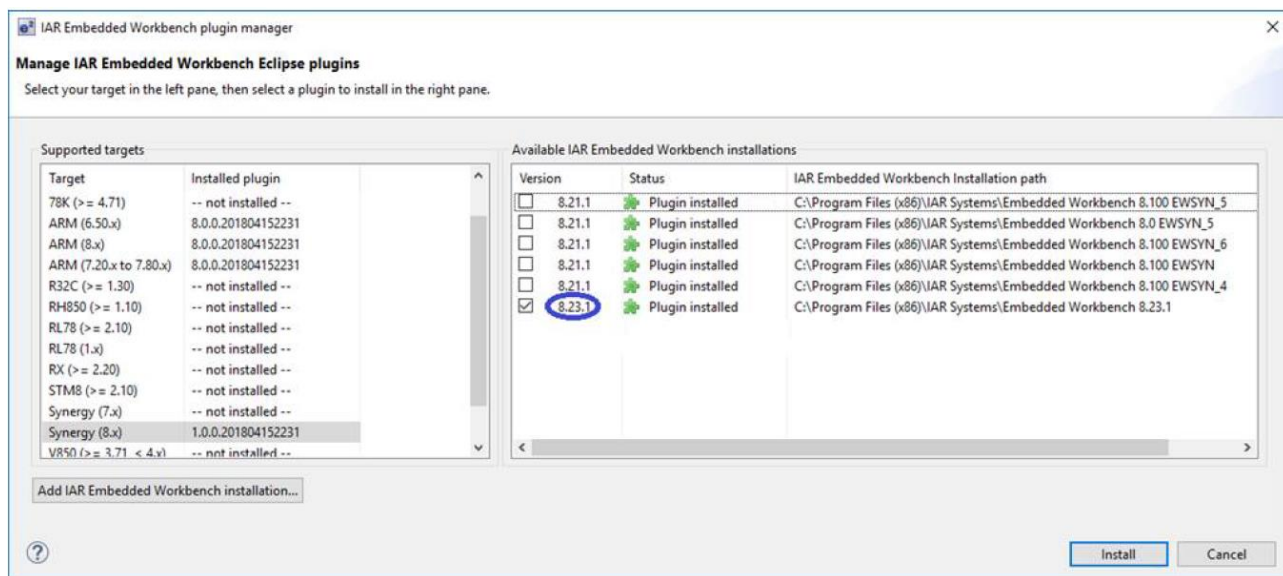
1. IAR EW for Synergy をダウンロードし、インストールします
2. e² studio を開き、[help] (ヘルプ) にアクセスし、[IAR Embedded workbench plugin manager] (IAR Embedded Workbench プラグインマネージャ) を選択します
3. Synergy 用の IAR プラグインをインストールします



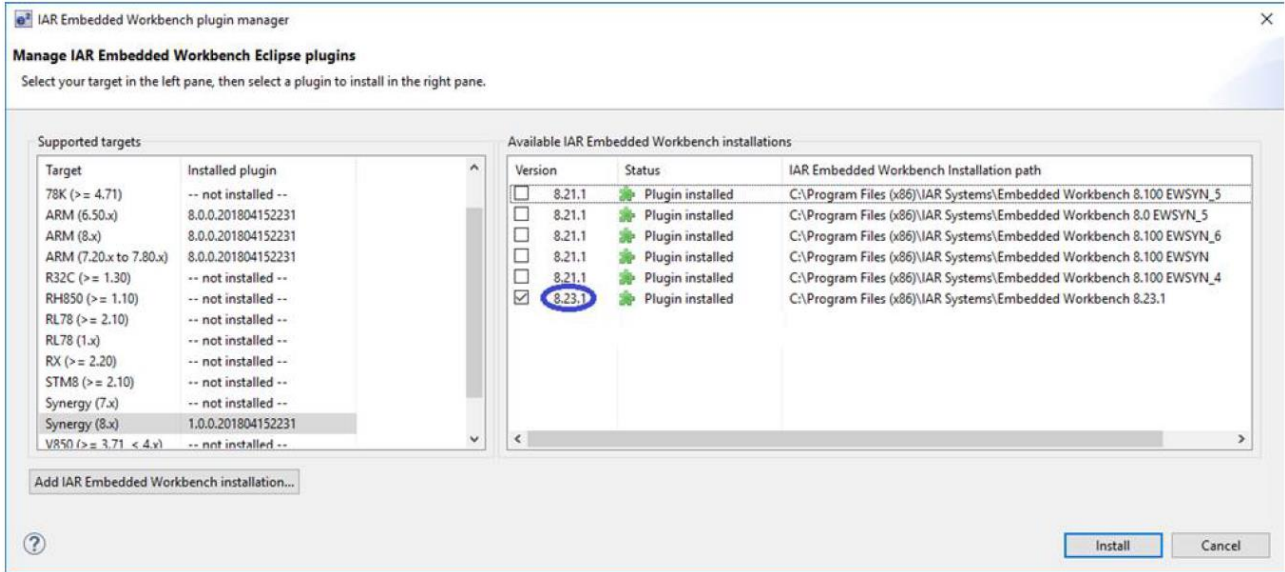
4. [Window] (ウィンドウ) □ [Preferences] (設定) □ [IAR Embedded Workbench Setup] (IAR Embedded Workbench セットアップ) を選択し、[Apply] (適用) をクリックして、以下のように IAR EW for Synergy の新しいパスを設定します。



e² studio を再起動する必要があります。



5. これ以降、新しい Synergy プロジェクトを作成するときに、使用可能なツールチェーンのリストから IAR コンパイラを選択することができます。その結果、IAR コンパイラを使用してプロジェクトがビルドされます。



Web サイトおよびサポート

以下の URL で、Synergy プラットフォームの詳細の確認、関連するドキュメントのダウンロード、サポートの活用ができます。

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

改訂履歴

Rev.	発行日	説明	
		ページ	ポイント
1.21	2019.4.04		<ul style="list-style-type: none">・英語版(R11AN0023EU0121、2018.10.22発行)を翻訳・P1 ソフトウェア入手先をRenesas Synergy 日本語WEBサイトに変更・P9 パラメータ表に"clear"を追加

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。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、
家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、
金融端末基幹システム、各種安全制御装置等
当社製品は、データシート等により高信頼性、Harsh environment向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じて、当社は一切その責任を負いません。
6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。
注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。

(Rev.4.0-1 2017.11)



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

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)

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<https://www.renesas.com/contact/>