

Renesas Synergy™ プラットフォーム

# TraceX を使用した ThreadX® RTOS アプリケーションのデバッグ手順

R20AN0404JJ0110

Rev.1.10

2017 年 8 月 1 日

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。

## はじめに

ThreadX® は、高性能組み込みカーネルを持った、Express Logic 社の RTOS です。このドキュメントでは、Renesas Synergy™ e2 studio Integrated Solution Development Environment (以下 e2 studio) でのアプリケーション開発において、ThreadX のスレッドとオブジェクト (リソースと呼ぶ) の状態を確認する手順について説明します。さらに TraceX を起動する手順についても説明します。ThreadX の仕様と機能については、Express Logic 社の Web サイト (<http://rtos.com/>) を参照してください。TraceX の仕様と機能については、Renesas Synergy™ ギャラリー Web サイト (<https://synergygallery.renesas.com/>) の [Development Tools] の下で [TraceX] を選択し、TraceX のページに移動して、TraceX のユーザマニュアルを参照してください。

このアプリケーションノートでは、Renesas Synergy™ ソフトウェアパッケージ (SSP) v1.3.0 以降のバージョンをインストールし、Blinky with ThreadX と呼ばれるプロジェクトを使用した例を説明しています。Blinky with ThreadX の操作手順については、e<sup>2</sup> studio の一般的な使用方法を説明している『e<sup>2</sup> studio Getting Started Guide for Renesas Synergy (以下 Getting Started Guide)』を参照してください。『Getting Started Guide』については、Renesas Synergy™ ギャラリー Web サイト (<https://synergygallery.renesas.com/>) を参照してください。

このアプリケーションノートは、SSP v1.3.0 以降のバージョンと e<sup>2</sup> studio の v5.4.0.018 以降のバージョンをサポートしています。

## 動作確認環境

以下の環境で動作を確認しました。

- Renesas Synergy™ ソフトウェアパッケージ (SSP) v1.3.0
- e<sup>2</sup> studio v5.4.0.018
- ThreadX のソースファイルを閲覧するには SSP の開発・量産ライセンスが必要です(「5.1」、「ThreadX のライセンス」を参照)
- DK-S7G2

---

**目次**

1. ThreadX を用いた Blinky プロジェクト:.....	3
2. TraceX の使用 .....	6
2.1 TraceX のインストール.....	6
2.2 TraceX の起動手順.....	7
3. RTOS リソースビュー機能の使用 .....	11
3.1 RTOS リソースビューの表示.....	11
4. Debug ビューでの実行された関数の表示.....	15
4.1 Debug ビューでの実行された関数の表示を設定する .....	16
4.2 実行された関数の表示の確認 .....	17
5. 付録 .....	17
5.1 ThreadX のライセンス .....	17
5.1.1 SSP の開発および実運用ライセンスの登録および保存 .....	18
5.1.2 e <sup>2</sup> studio へのライセンスファイルの登録 .....	20
ホームページとサポート窓口.....	22

## 1. ThreadX を用いた Blinky プロジェクトの作成

ThreadX を用いた Blinky プロジェクトは、e2 studio上でスレッドの監視と[Debug] ビューによるデバッグ機能を提供します。新しい Blinky with ThreadX プロジェクトを作成するには、このセクションの手順に従ってください。

1. ファイル(F) > 新規(N) > Synergy C Project を選択します。

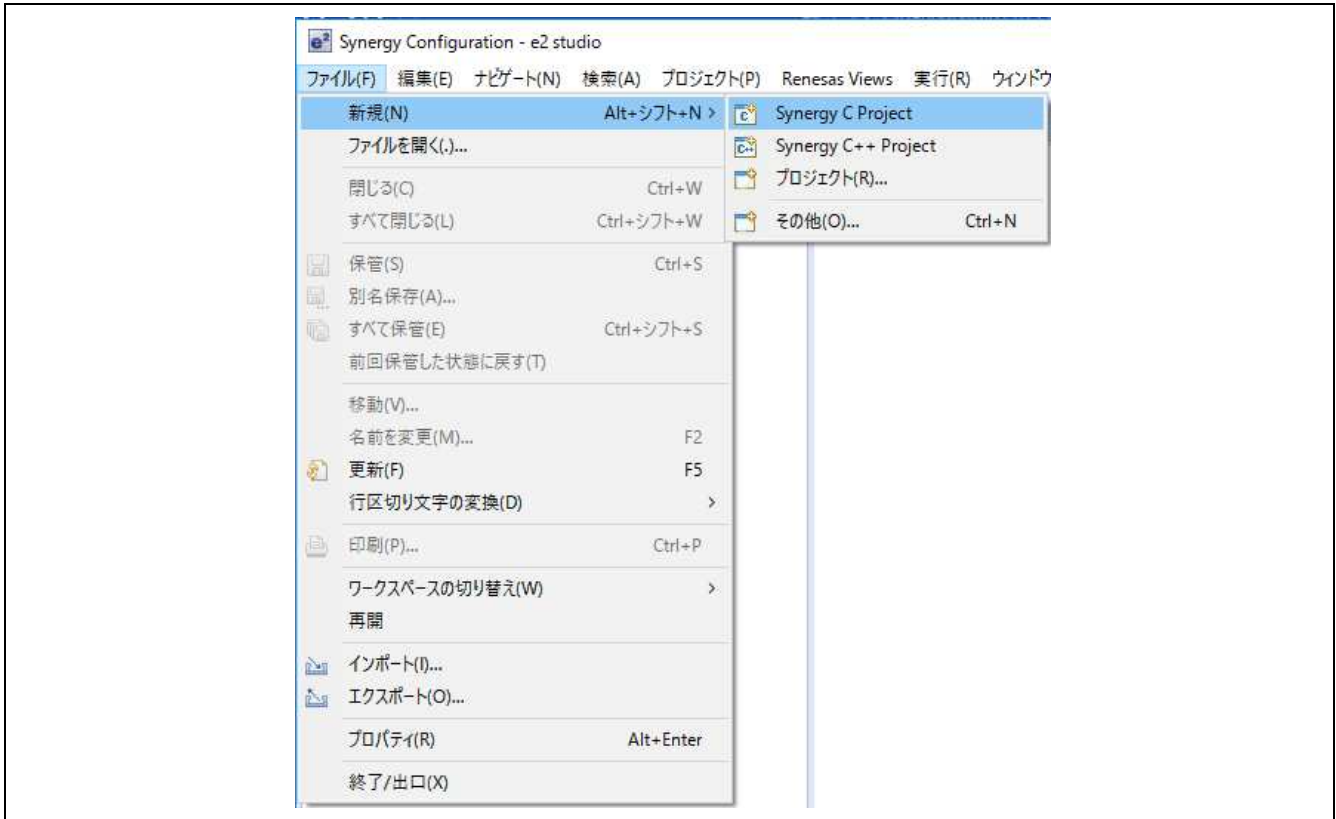


図 1 新しいプロジェクトの作成

- プロジェクト名を入力し、[次へ] をクリックします。ライセンスが新しい SSP 1.3.0 であることを確認してください。

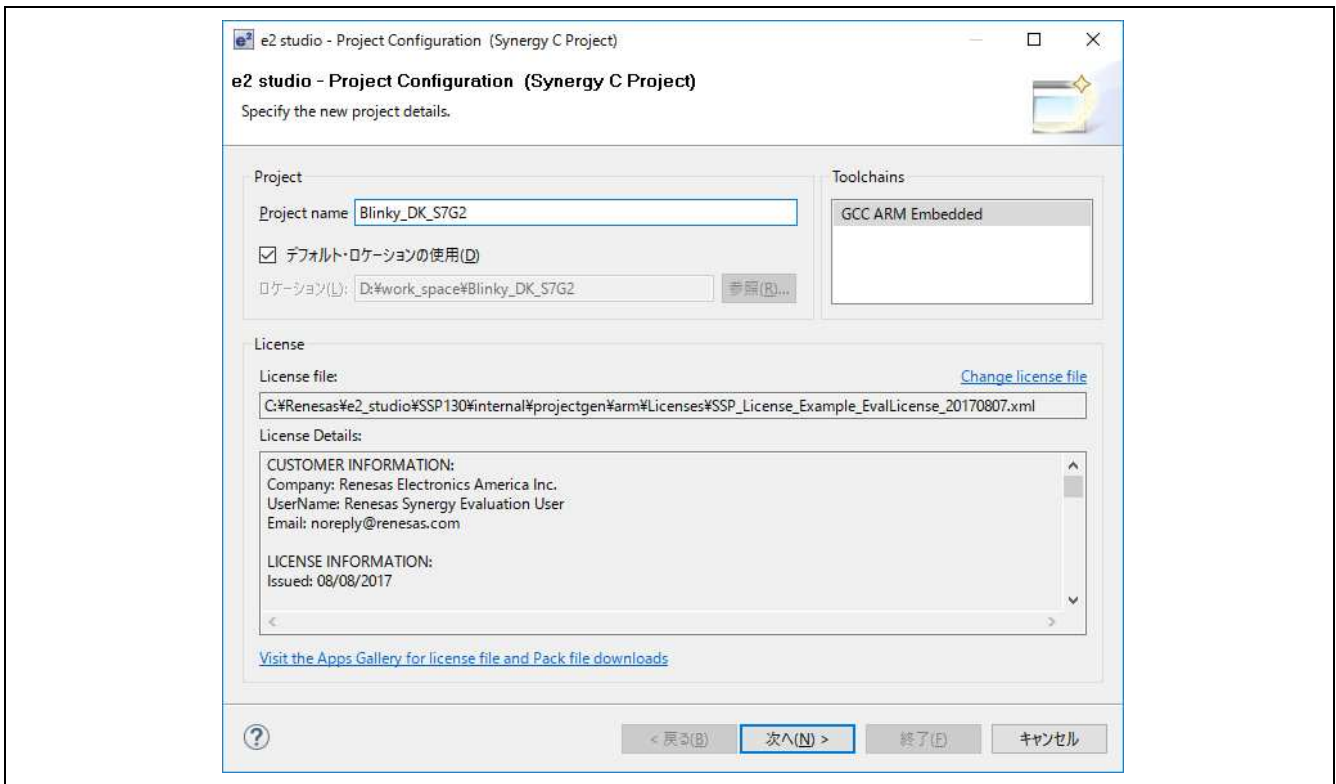


図 2 プロジェクト名

3. [Device Selection] で SSP バージョンとボードの値が正しいことを確認します。[次へ] をクリックします。

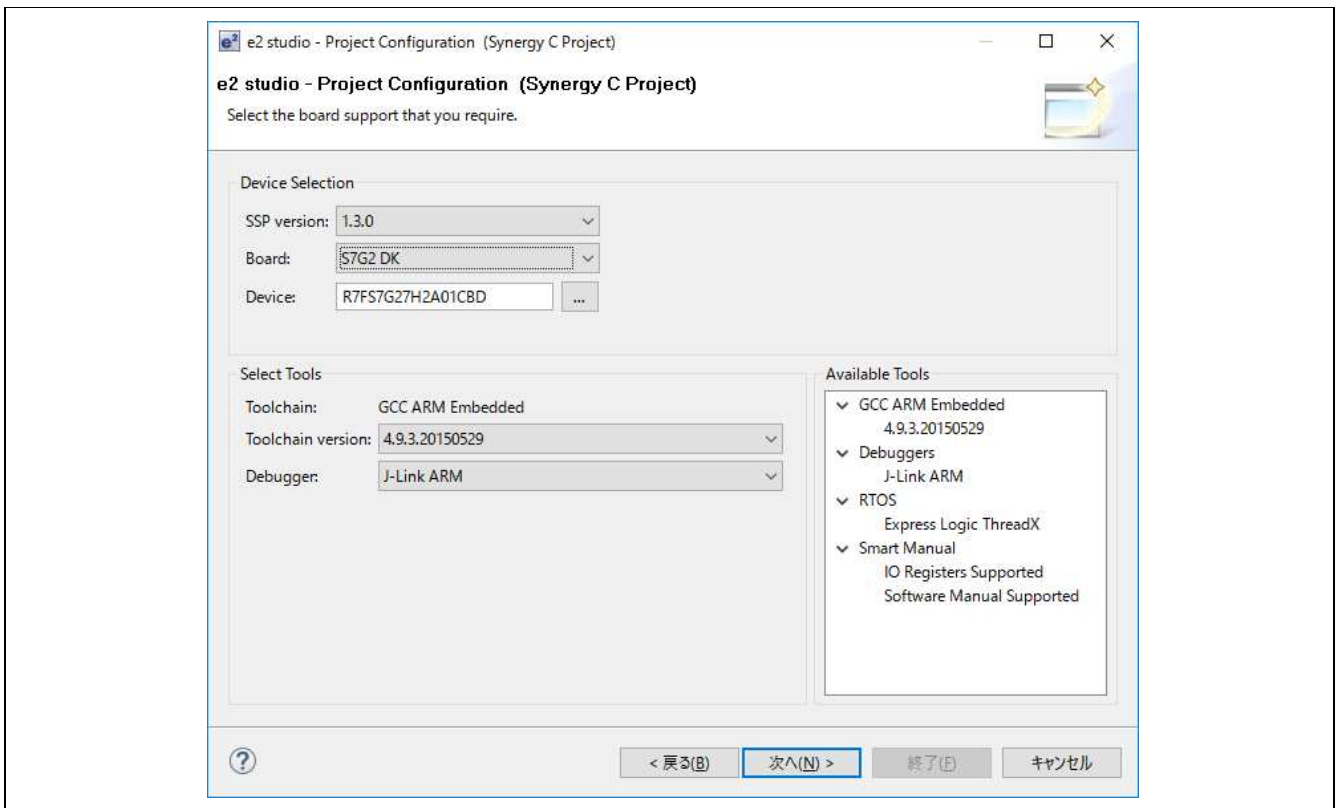


図 3 ボードとバージョンの選択

4. [Project Template Selection] でタイプを [Blinky with ThreadX] として選択します。[終了] をクリックします。

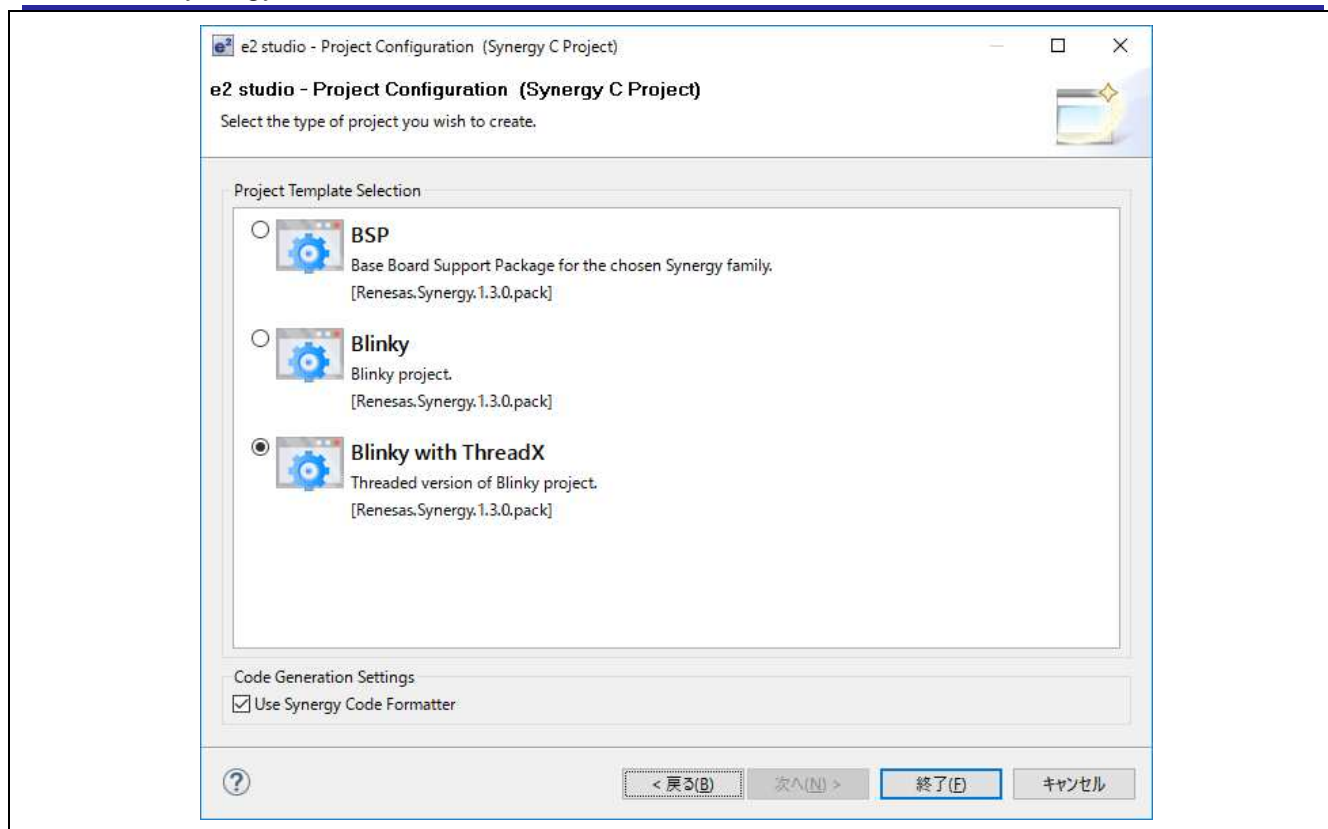


図 4 テンプレートの選択

プロジェクトエクスプローラーウィンドウに、新しいプロジェクトが指定した名前で追加されます。

## 2. TraceX の使用

TraceX®はPCで動作するThreadX®用のスレッドの遷移などを表示するソフトウェアです。TraceX は、e<sup>2</sup> studio から直接起動できます。この章では、TraceX を起動および使用する手順について説明します。

### 2.1 TraceX のインストール

1. TraceX をダウンロードするには、Renesas Synergy™ ギャラリー Web サイト (<https://synergycastle.renesas.com/>) の [Development Tools] の下で [TraceX] を選択し、その後、TraceX ソフトウェアをダウンロードするページに移動します。

注記: ダウンロード後に、ファイルを解凍します。TraceX ソフトウェアをインストールする場合は、必ず、**管理者**が実行してください。そのためには、セットアップアプリケーションを右クリックして、[Run as Administrator] を選択します。インストーラセットアップを管理者が実行しないと、ファイルの欠落などのエラーが発生します。

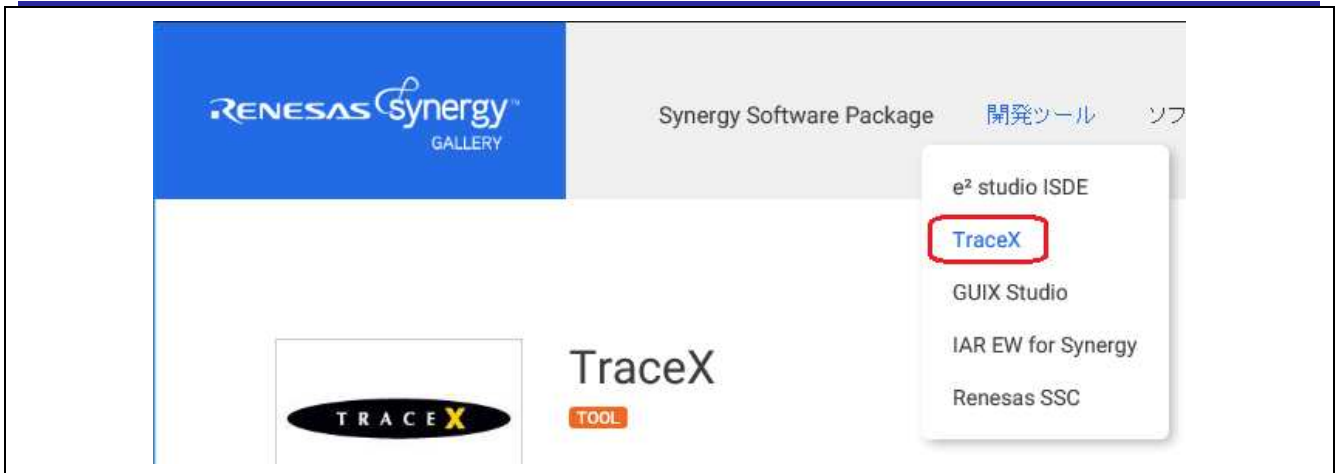


図 5 TraceX のダウンロード用ページへの移動

2. ダウンロードページで、[ダウンロード] をクリックして TraceX を PC にダウンロードしてインストールします。



図 6 TraceX のダウンロードページ

## 2.2 TraceX の起動手順

このセクションでは、e<sup>2</sup> studio から TraceX を起動する手順について説明します。

1. 図 7 および図 8 に示すコンフィグレーション・エディタを使用して、ThreadX のソースコードファイルを e<sup>2</sup> studio に登録します。[Threads] タブページを開いて、項目 (1) ~ (5) を順番に指定し、ソースファイルを登録します。



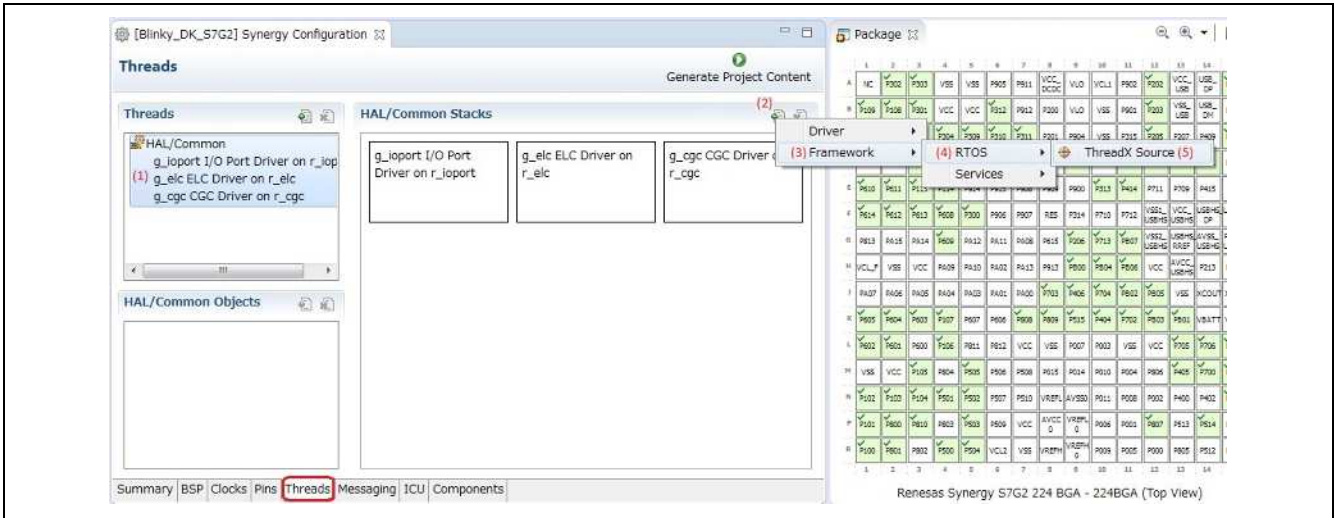


図 7 ThreadX のソースファイルの登録 (v5.3.1 以前のバージョンの e<sup>2</sup> studio)

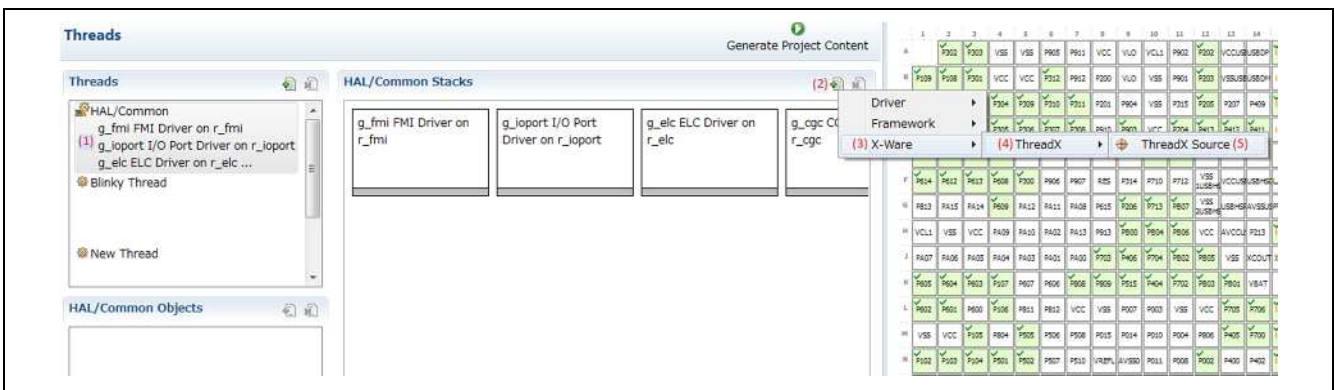
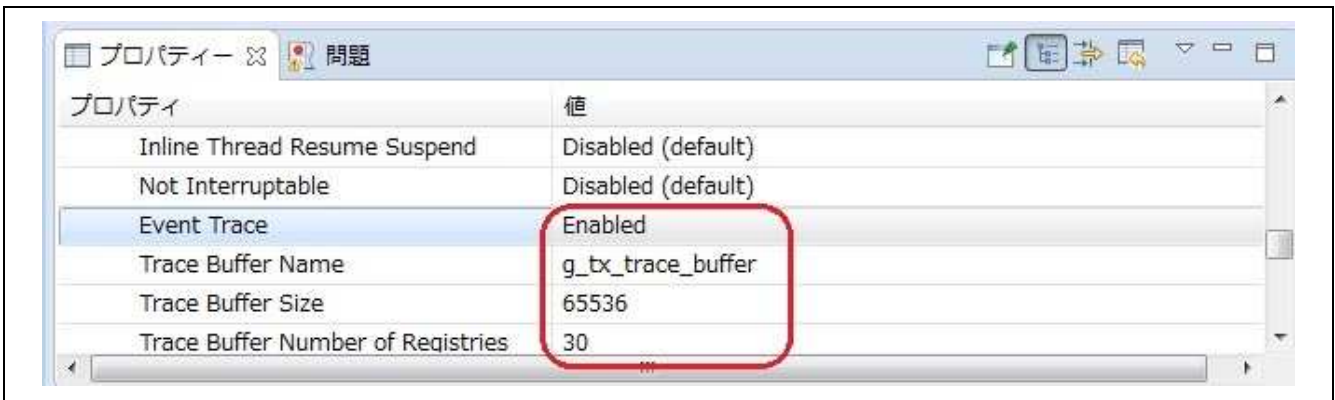


図 8 ThreadX のソースファイルの登録 (v5.3.1 以降のバージョンの e<sup>2</sup> studio)

- プロパティタブページを確認し、[Event Trace] を [Enabled] に設定します。[Trace Buffer Name]、[Trace Buffer Size]、および [Trace Buffer Number Registries] を図 9 「Event Trace の設定」に示すように設定し、その後、プロジェクトを更新してビルドします。





## 図 9 Event Trace の設定

- 【注意】 S1 デバイスの Trace Buffer Size は 16 KB 未満で、アプリケーションとは別に使用可能な RAM 領域であることが前提です。
- ビルドが完了したら、デバッガを起動し、「実行(R) > 再開(M)」メニューからプログラムを実行します。このプロセスが 2 回繰り返されると、LED が 1 秒間隔で点滅します。「実行(R) > 中断(S)」メニューを選択すると、プログラムの実行は一時停止します。
  - TraceX を使用してプログラムの実行結果を確認します。TraceX の起動方法を設定します。「実行(R) > Launch TraceX Debugging...」メニューを選択すると、TraceX の起動に必要なパラメータを設定するためのダイアログボックスが開きます。



図 10 TraceX デバッグの起動の設定

- 【注意】 TraceX を起動するためのパラメータは、図 9 「Event Trace の設定」で指定した値です。

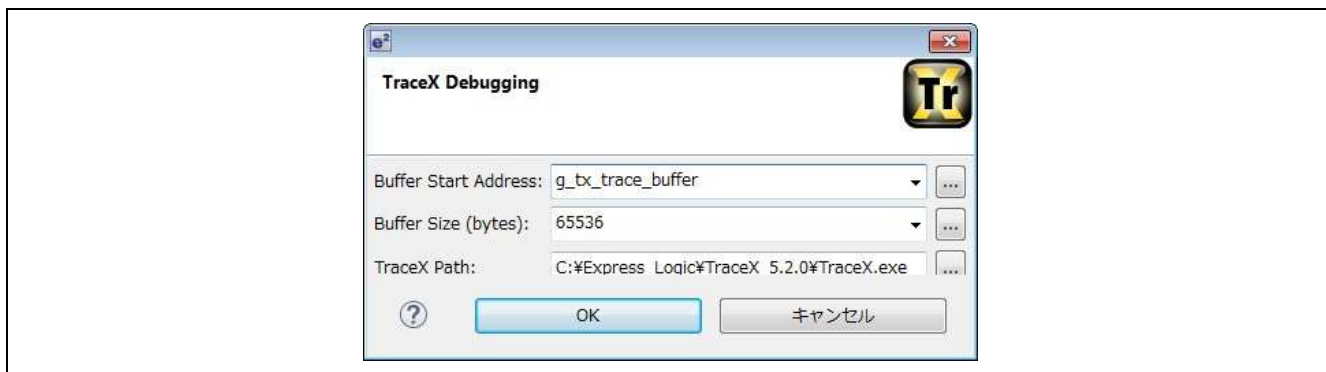


図 11 TraceX を起動するためのパラメータの設定

- [OK] ボタンをクリックします。TraceX が起動し、プログラムの実行以降の遷移が表示されます。別の方法として、e<sup>2</sup> studio の「ウィンドウ(W) > 設定(P)」メニューで設定ダイアログボックスを開き、【C】 - 【Renesas】 - 【TraceX】を選択することでも、TraceX へのパスを指定できます。

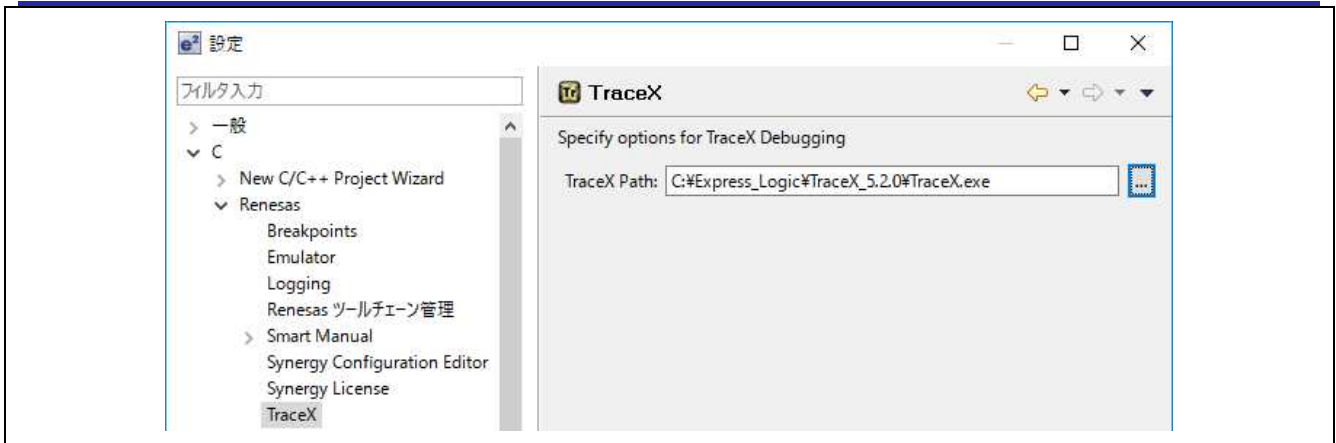


図 12 TraceX パスの設定

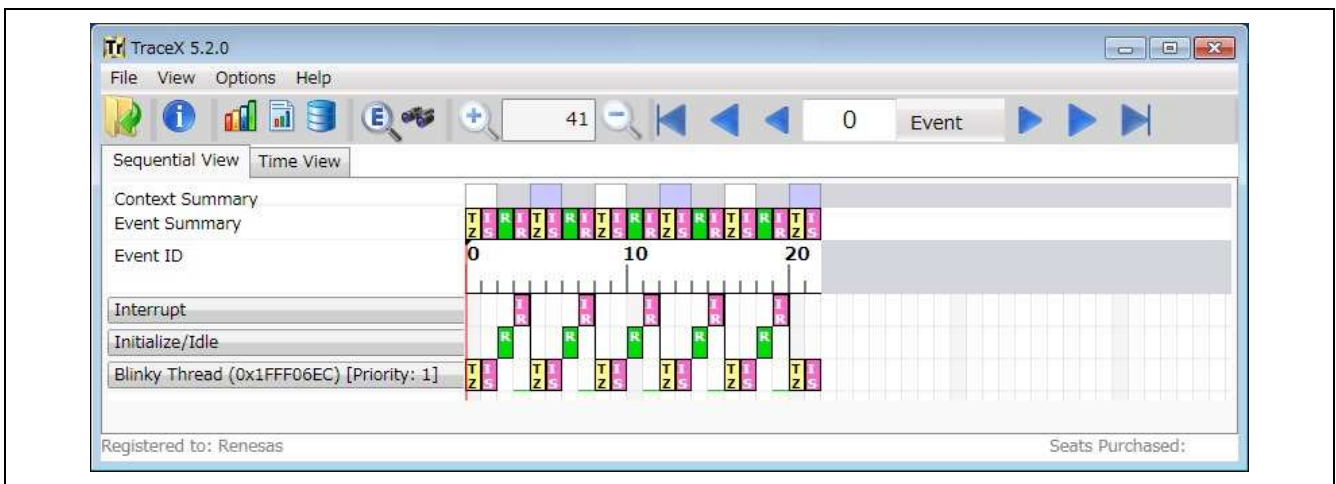


図 13 TraceX の起動

- 「実行(R) > 再開(M)」メニューからプログラムを実行した後、「実行(R) > 中断(S)」メニューからプログラムの実行を一時停止し、その後、TraceX を開始してデータが更新されたことを確認します。TraceX の 2 回目以降の起動では、パラメータを再度設定する必要はありません。「実行(R) > Update Trace Data」メニューから TraceX を起動します。

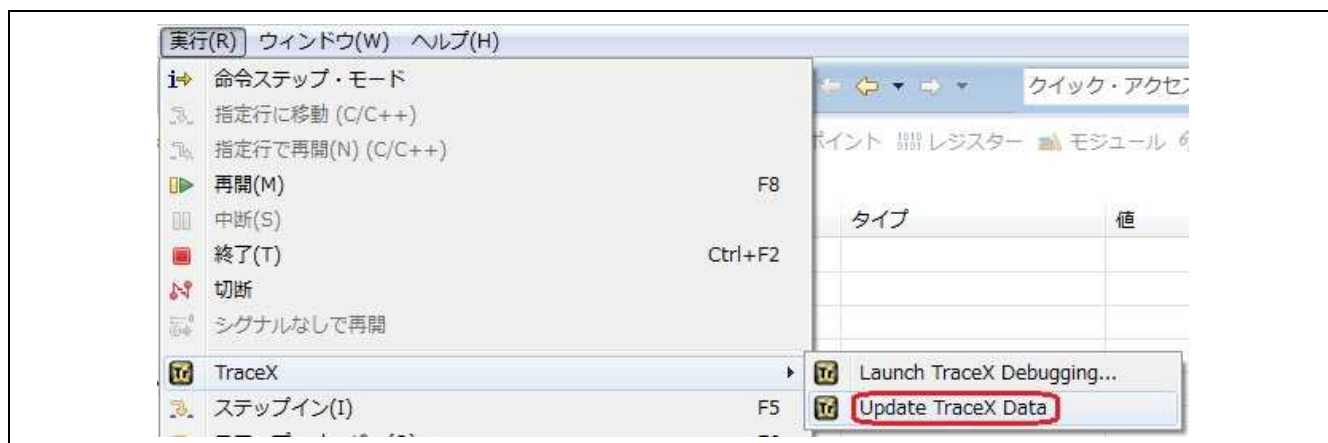


図 14 TraceX データの更新

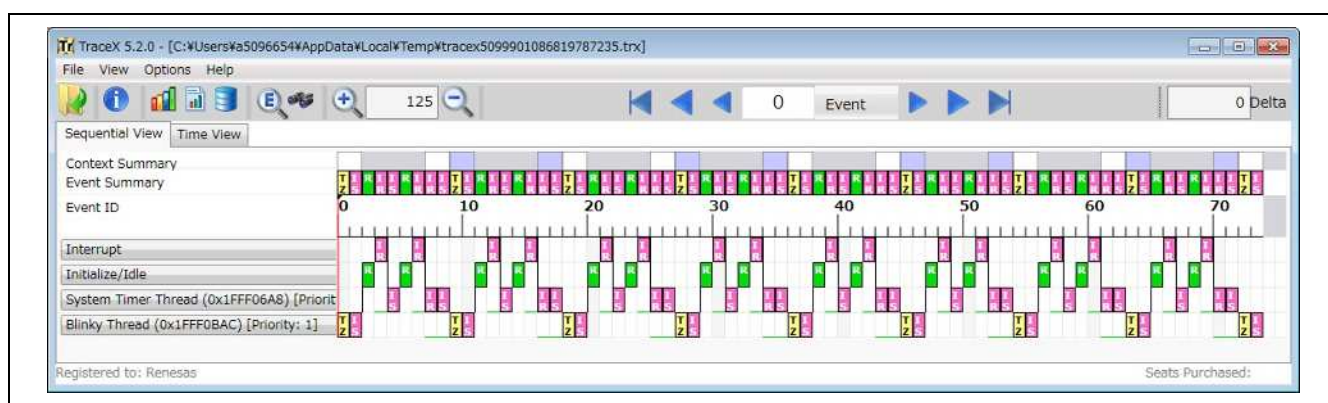


図 15 TraceX データの更新後の表示

### 3. RTOS リソースビュー機能の使用

e<sup>2</sup> studio には、ThreadX のリソースの状態を表示する RTOS リソースビュー機能があります。このセクションでは、RTOS リソースビューを使用する手順について説明します。ただし、v5.3.1 以降のバージョンの e<sup>2</sup> studio でこの機能を使用する場合は、セクション 4.1 「Debug ビューでの実行された関数の表示を設定する」を参照し、[RTOS Integration in Debug View] を「いいえ」に設定してください。

#### 3.1 RTOS リソースビューの表示

RTOS リソースビューはデバッガの実行中のみ機能するため、最初にセクション 2.2 に示すように、**Blinky with ThreadX** などのプロジェクトをビルドします。その後、デバッガを起動して、Renesas Views > パートナー OS > RTOS リソース」を選択します。OS 選択ダイアログボックスが表示されたら、[ThreadX (R)] を選択します。RTOS リソースビューが表示されます。

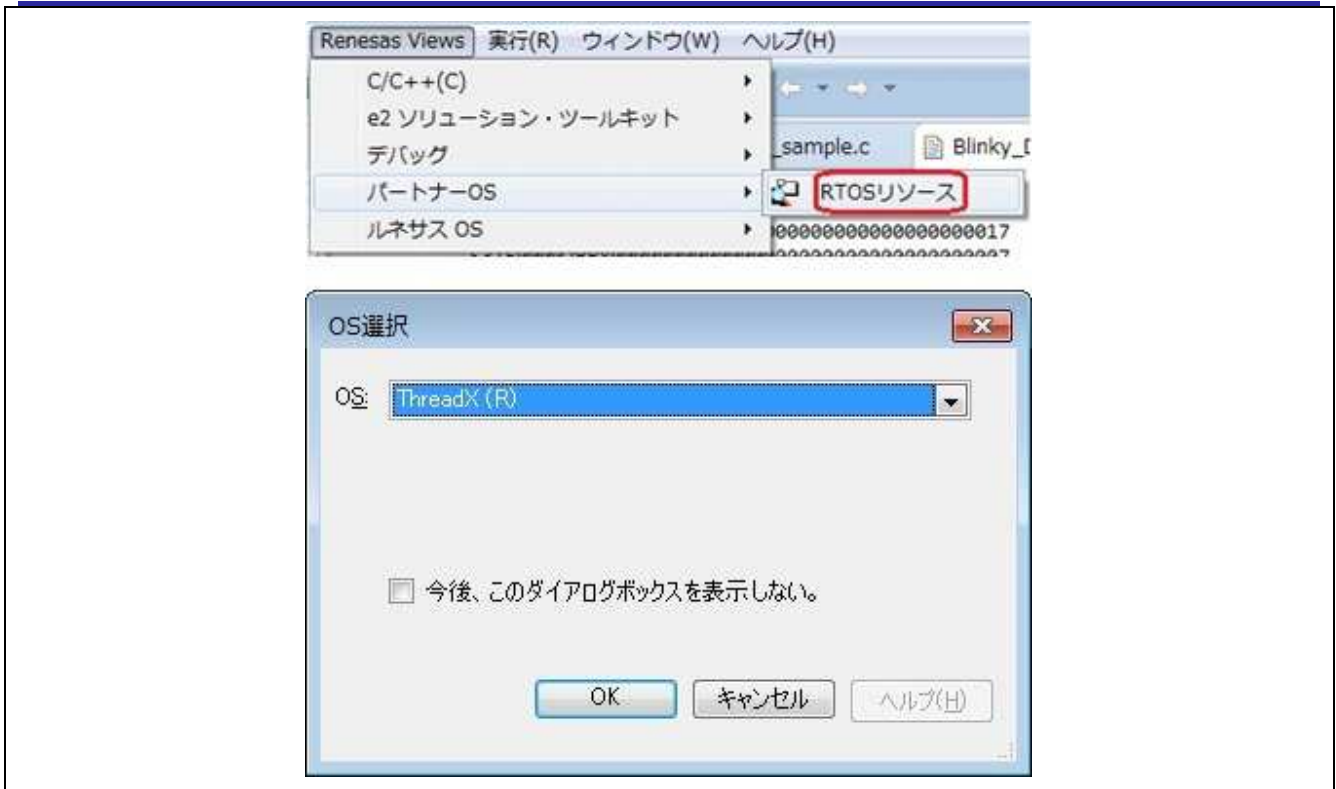


図 16 RTOS リソースと OS の指定

v5.3.1 以降のバージョンの e<sup>2</sup> studio の場合は、ダイアログボックスではなく [RTOS Resources] ビューで OS を選択します。

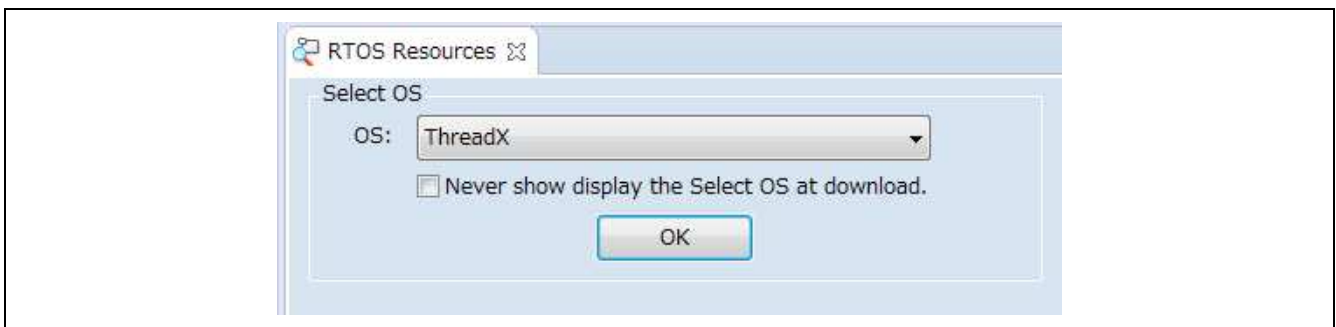


図 17 v5.3.1 以降のバージョンの e<sup>2</sup> studio で OS を選択する

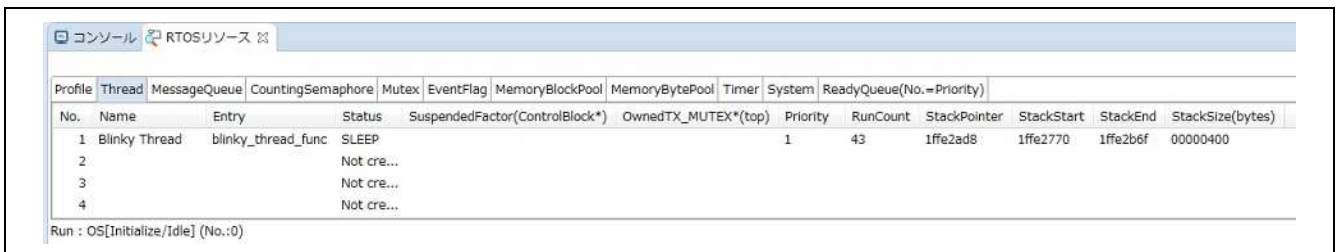


図 18 RTOS Resources ビュー

[Thread] から [ReadyQueue(No.=Priority)] までのタブ付きページに、各リソースの状態が示されます。リソースの状態を確認するには、対象のリソースのタブを選択します。

注記: [Profile] タブは、将来の拡張用です。

表 1 各タブ付きウィンドウの内容

リソースタブ名	表示される情報と選択項目	表示される情報
Thread (e2 studio v5.0.0 以前のバージョン)	Name	スレッドの名前
	Entry	各スレッドを開始する関数
	Status	スレッドの状態
	Suspended Factor (Control Block*)	一時停止の原因となっているリソース
	OwnedTX_MUTEX*(top)	獲得した上位のミューテックス
	Priority	優先度
	RunCount	スレッドが実行される回数
	StackPointer	現在のスタックポインタ
	StackStart	スタックの開始アドレス
	StackEnd	スタックの終了アドレス
	StackSize(bytes)	スタックのサイズ
Thread (e2 studio v5.3.1 以降のバージョン)	Name	スレッドの名前
	Entry	各スレッドを開始する関数
	Status	スレッドの状態
	Suspended Factor (Control Block*)	一時停止の原因となっているリソース
	OwnedTX_MUTEX*(top)	獲得した上位のミューテックス
	Priority	優先度
	RunCount	スレッドが実行された回数
Stack (e2 studio v5.3.1 以降のバージョン)	Name	スレッドの名前
	Entry	各スレッドを開始した関数
	StackPointer	現在のスタックポインタ
	StackStart	スタックの開始アドレス
	StackEnd	スタックの終了アドレス
	StackSize(bytes)	スタックのサイズ
	MaxStackUsage(bytes)	現在使用されているスタックの最大数

リソース名	表示される情報と選択項目	表示される情報
MessageQueue	Name	メッセージキューの名前
	UsedCount	使用中のメッセージキューの数
	FreeCount	使用可能なメッセージキューの数
	TotalCount	メッセージキューの合計数
	MessageSize	メッセージのサイズ
	SuspendedTX_THREAD*(top)	キュー内で待機中のスレッドのうち最上位のスレッド
	SuspendedCount	一時停止されたスレッドの数
	StartAddress	メッセージキューの開始アドレス
	EndAddress	メッセージキューの終了アドレス
CountingSemaphore	Name	セマフォの名前
	SemaphoreCount	セマフォの数
	SuspendedTX_THREAD*(top)	キュー内で待機中のスレッドのうち最上位のスレッド
	SuspendedCount	一時停止されたスレッドの数
Mutex	Name	ミューテックスの名前
	OwnerTX_THREAD*	獲得しているスレッド
	OwnerCount	所有者の数
	SuspendedTX_THREAD*(top)	キュー内で待機中のスレッドのうち最上位のスレッド
	SuspendedCount	一時停止されたスレッドの数
EventFlag	Name	イベントフラグの名前
	Flag	現在のフラグパターン
	SuspendedTX_THREAD*(top)	キュー内で待機中のスレッドのうち最上位のスレッド
	SuspendedCount	一時停止されたスレッドの数
MemoryBlockPool	Name	メモリブロックの名前
	FreeCount	使用可能なブロックの数
	TotalCount	ブロックの合計数
	BlockSize(bytes)	ブロックのサイズ
	TotalSize(bytes)	メモリブロックプールの合計サイズ
	SuspendedTX_THREAD*(top)	キュー内で待機中のスレッドのうち最上位のスレッド
	SuspendedCount	一時停止されたスレッドの数
	StartAddress	メモリブロックプールの上位アドレス
MemoryBytePool	Name	メモリプールの名前
	Free(bytes)	使用可能なバイトの数
	Total(bytes)	メモリバイトプールの合計サイズ
	FragmentCount	フラグメントの数
	SuspendedTX_THREAD*(top)	キュー内で待機中のスレッドのうち最上位のスレッド
	SuspendedCount	一時停止されたスレッドの数
	StartAddress	メモリバイトプールの開始アドレス
Timer	Name	タイマーの名前
	Remaining Tick	残り時間
	Re-initialization Tick	サイクル時間
System	SystemClock	システムクロック



リソースタブ名	表示される情報と選択項目	表示される情報
ReadyQueue(No.=Priority)	QueuedTX_THREAD*(top)	上位の使用可能なスレッド

#### 4. デバッグビューへ実行した関数を表示する

V5.3.1 以降のバージョンの e<sup>2</sup> studio には、実行した関数をスレッド毎に表示する機能が追加されています。

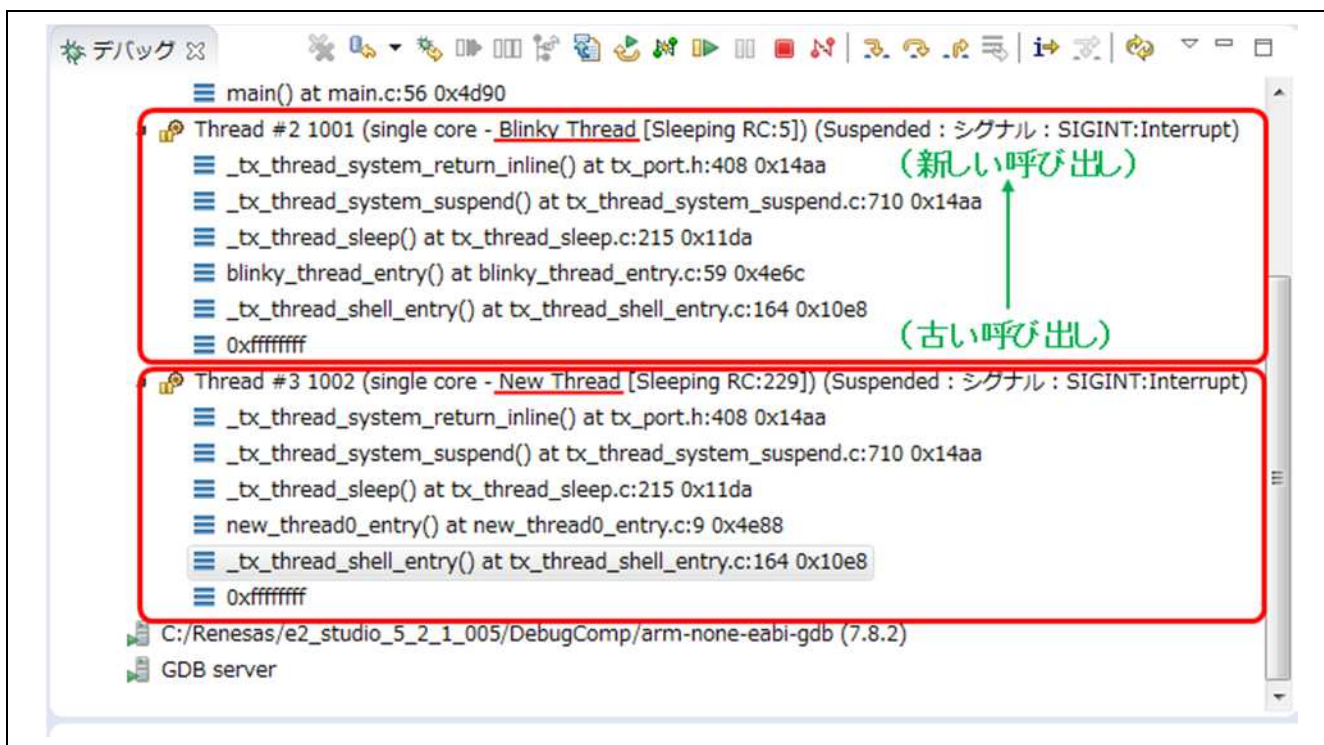


図 19 スレッド単位での関数の表示

##### 4.1 デバッグビューへ実行した関数を表示する機能の設定方法

この機能は、デバッグ構成 ダイアログで設定されます。

- 「実行(R)>デバッグの構成(B)...」メニューを選択して、デバッグの構成 ダイアログを開きます。
- [Debugger] タブ、さらに [Debug Tool Settings] タブを選択します。
- [RTOS Integration in Debug View] を [はい] に設定します。[いいえ] を選択すると、この機能を使用できません。この選択は、デフォルトで [はい] に設定されています。



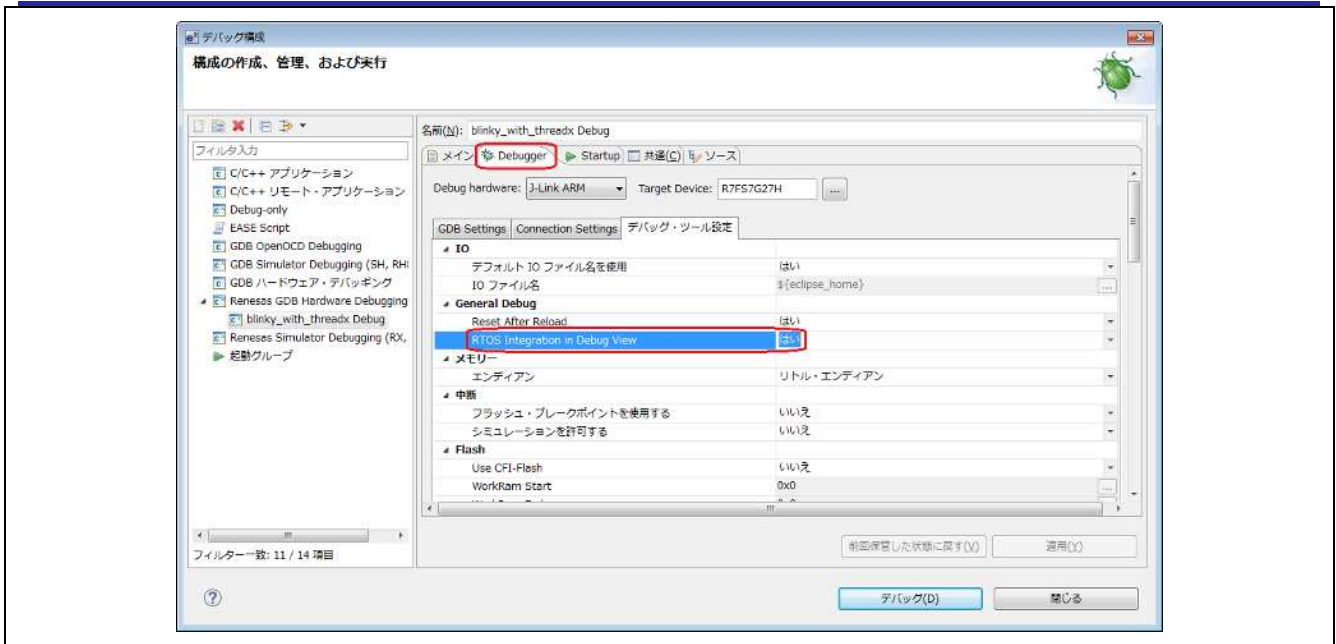


図 20 実行された関数の表示の設定

## 4.2 実行された関数の表示の確認

この機能は以下の項目に関する情報を表示します。

- 関数名
- 引数とローカル変数
- レジスタ値

図 18 は、**デバッグ ビュー**で `blinky_thread_entry()` 関数を選択し、**変数 ビュー**では関数が実行された後のローカル変数が示され、**レジスター ビュー**にはレジスタ値が示された例です。

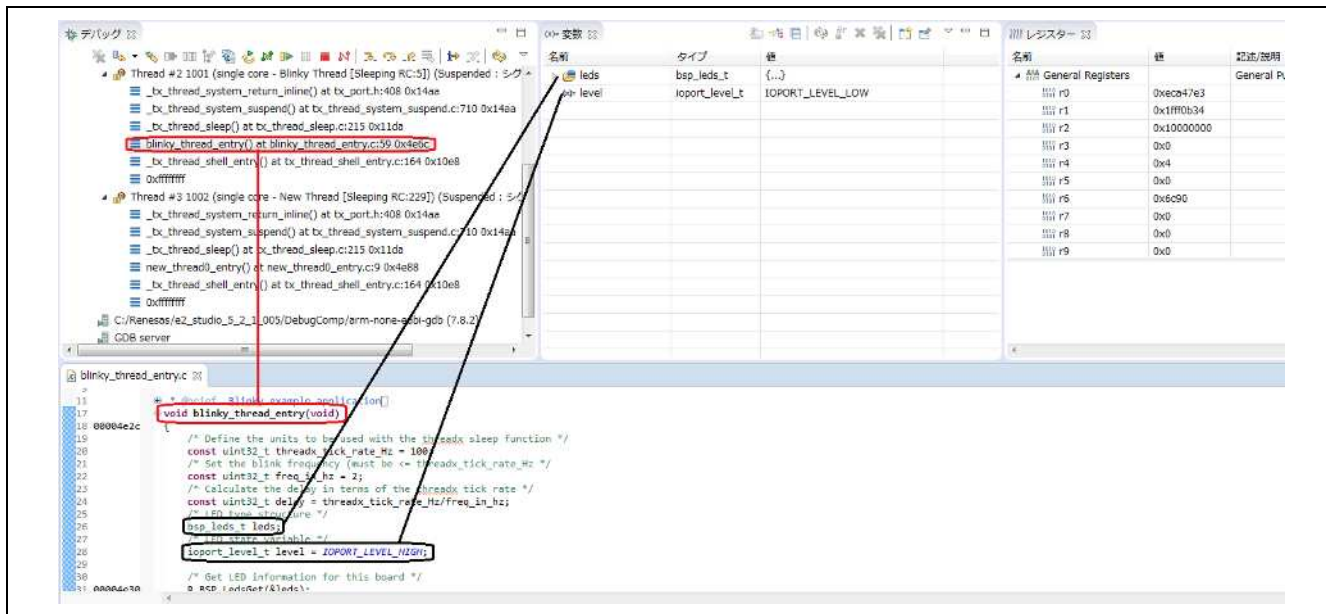


図 21 関数が実行中のローカル変数とレジスタの表示

## 5. 付録

### 5.1 ThreadX のライセンス

SSP の評価ライセンスでも RTOS リソース関数と TraceX は使用できますが、ThreadX のソースファイルにはアクセスできません。ThreadX のソースファイルにアクセスするには、SSP の開発・量産ライセンスの契約が必要です。このセクションでは、そのライセンスを登録する手順について説明します。

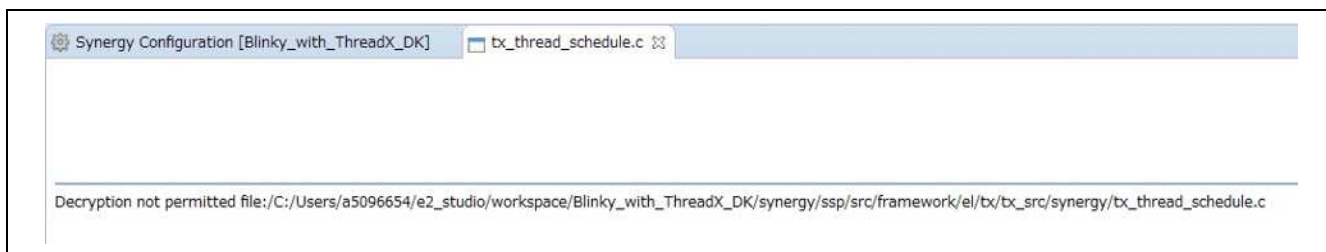


図 22 評価ライセンスでは ThreadX のソースファイルにアクセスできない

#### 5.1.1 SSP の開発・量産ライセンスの登録および保存

Renesas Synergy Web サイト (<https://synergygallery.renesas.com/>) の SSP ページで **[開発・生産ライセンスを作成する]** をクリックして、SSP の開発・生産ライセンスを登録します。



図 23 ライセンスの確認

ライセンスを登録したら、その情報をライセンスファイルに保存します。

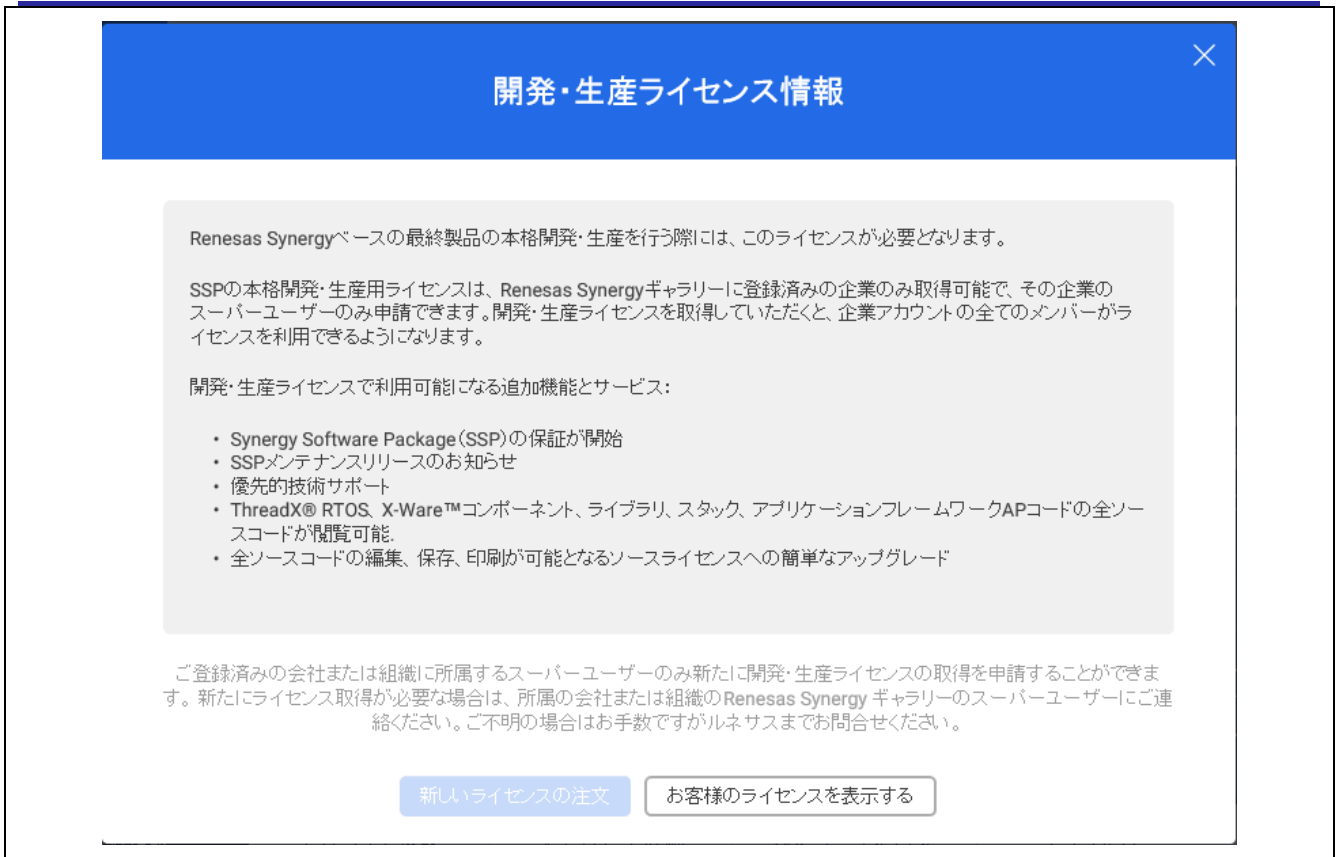


図 24 ライセンス情報の表示

ライセンスファイルを保存するには、対象の SSP の XML ファイルを PC に保存します。これらのファイルは、e<sup>2</sup> studio がインストールされている次のディレクトリにコピーすることをお勧めします。

<e<sup>2</sup> studio installation directory>%internal%projectgen%arm%Licenses



図 25 ライセンスファイルの保存

### 5.1.2 e<sup>2</sup> studio へのライセンスファイルの登録

保存されたライセンスファイルを e<sup>2</sup> studio に登録します。登録が完了すると、ThreadX のソースファイルにアクセスできるようになります。新しいプロジェクトを作成する場合は、プロジェクトウィザードを使用してライセンスファイルを登録します。プロジェクトが既に作成されている場合は、ライセンスファイルを「ウィンドウ(W) > 設定(P)」メニューから登録します。このセクションでは、既に作成されている **Blinky with ThreadX** プロジェクトの登録について説明します。

1. 「ウィンドウ(W) > 設定(P)」メニューを選択して、**設定** ダイアログを開き、C > Renesas > Synergy License」を選択します。

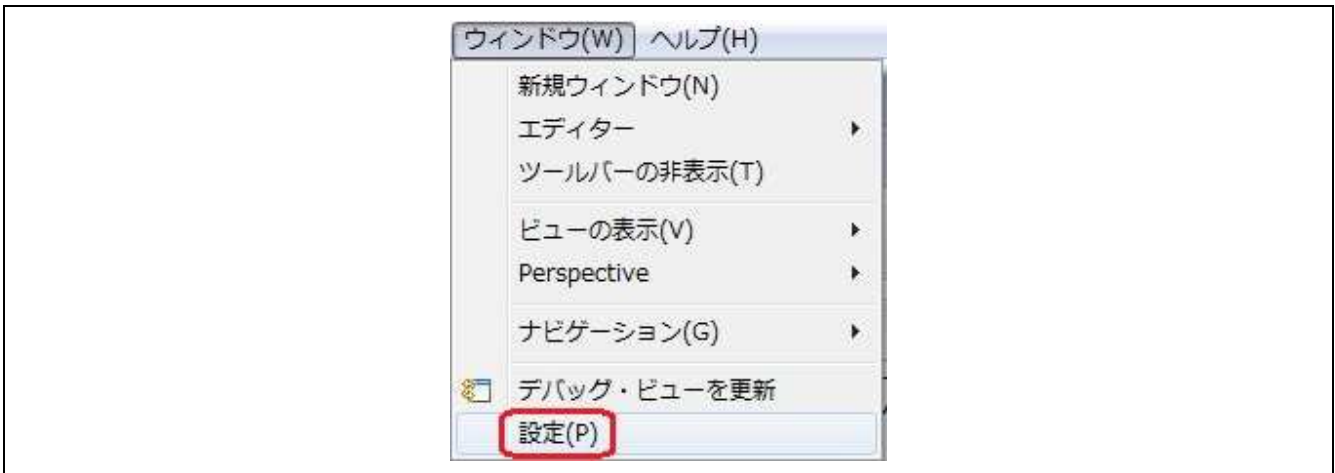


図 26 [Window] > [Preferences] メニューの選択

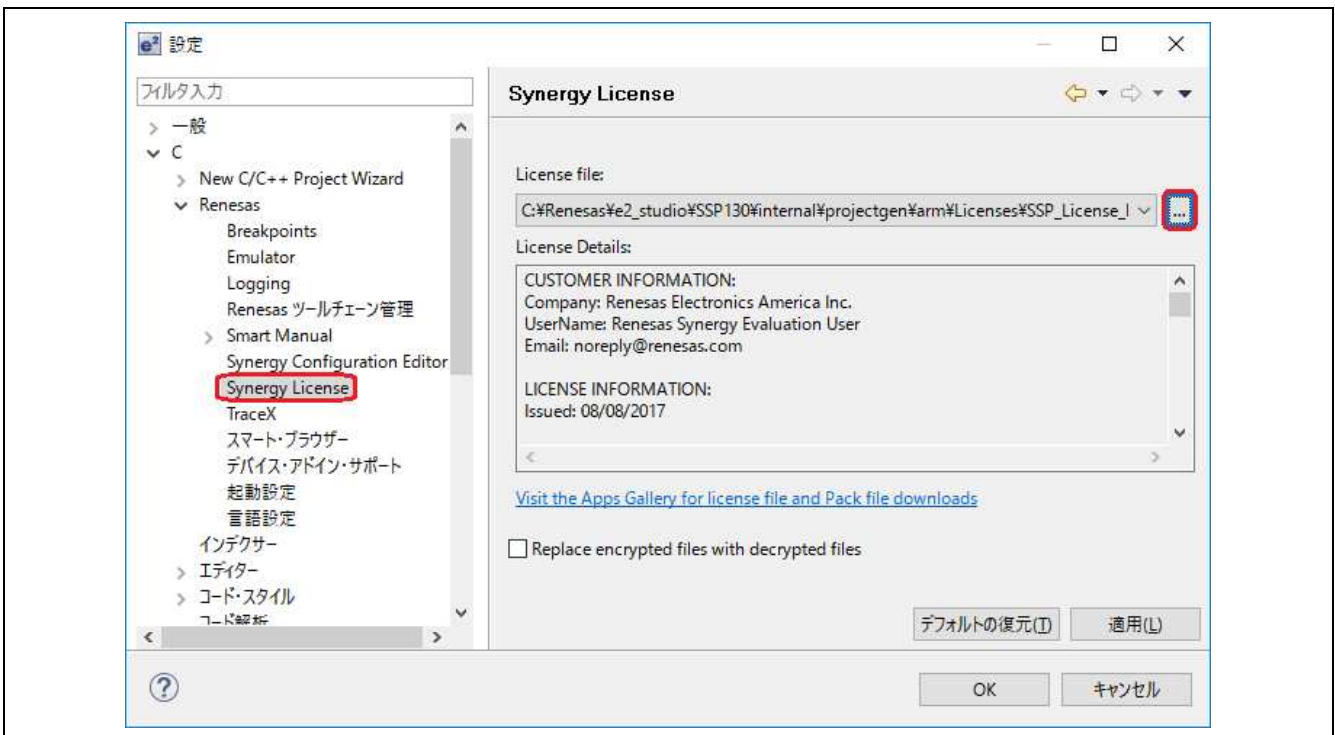


図 27 ライセンスを設定するためのダイアログボックス

2. **[Browse...]** ボタンをクリックし、セクション 5.1.1 で保存したライセンスファイルを選択します。**[OK]** をクリックすると、ライセンスファイルが登録されます。

---

## ホームページとサポート窓口

サポート: <https://synergygallery.renesas.com/support>

技術的なお問い合わせ先:

- 米国: [https://renesas.zendesk.com/anonymous\\_requests/new](https://renesas.zendesk.com/anonymous_requests/new)
- 欧州: <https://www.renesas.com/en-eu/support/contact.html>
- 日本: <https://www.renesas.com/ja-jp/support/contact.html>

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。



## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2016年8月24日	-	初版
1.01	2016年11月14日	11	第3章の追加とその他の訂正。
1.02	2017年4月6日	全	SSP および e <sup>2</sup> studio ISDE のサポートされるバージョンを更新。
1.03	2017年6月29日	全	SSP v1.3.0 および e2studio 5.4.0.018 向けに更新
1.04	2017年7月12日	全	新しいプロジェクト向けの更新とマイナーな変更。
1.10	2017年8月1日	-	初期リリース

## 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、その他の不適切に使用しないでください。かかる改造、改変、複製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、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、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り、これらの用途に使用することはできません。たとえ、意図しない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
  6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
  7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
  8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
  9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術は、(1)核兵器、化学兵器、生物兵器等の大量破壊兵器およびこれらを運搬することができるミサイル（無人航空機を含みます。）の開発、設計、製造、使用もしくは貯蔵等の目的、(2)通常兵器の開発、設計、製造または使用の目的、または(3)その他の国際的な平和および安全の維持の妨げとなる目的で、自ら使用せず、かつ、第三者に使用、販売、譲渡、輸出、賃貸もしくは使用許諾しないでください。  
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. お客様の転売、貸与等により、本書（本ご注意書きを含みます。）記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は一切その責任を負わず、お客様にかかる使用に基づく当社への請求につき当社を免責いただきます。
  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. 本資料に記載された情報または当社製品に関し、ご不明点がある場合には、当社営業にお問い合わせください。
- 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その総株主の議決権の過半数を直接または間接に保有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.3.0-1 2016.11)



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 (豊洲フォレスト)

■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/>