

# ルネサス RA ファミリ

## Azure RTOS TraceX を用いた Azure RTOS ThreadX のデバッグ

### 要旨

Azure RTOS ThreadX は Microsoft Corporation の RTOS で、高性能の組み込み型カーネルをベースとしています。

Azure RTOS TraceX は Windows で動作する解析ツールです。リアルタイムシステムイベントをグラフィック表示する機能により、リアルタイムシステムの動作を可視化して、組み込み開発者がより良く理解できるようにします。

本アプリケーションノートでは、e<sup>2</sup> studio でのアプリケーション開発における Azure RTOS ThreadX のスレッドとオブジェクト状態（リソース）をチェックする手順を説明します。Azure RTOS TraceX のインストールや起動についても説明します。

### 動作確認デバイス

RA6M3 MCU グループ (R7FA6M3AH)

### 動作環境

|             |                                                                                   |
|-------------|-----------------------------------------------------------------------------------|
| ターゲットボード    | EK-RA6M3                                                                          |
| 統合開発環境（IDE） | e <sup>2</sup> studio バージョン 2021-04 および FSP v3.0.0                                |
| トレースツール     | Microsoft Azure RTOS TraceX v6.1.6.0                                              |
| OS          | Microsoft Azure RTOS ThreadX v6.1.6                                               |
| ツールチェーン     | GNU Arm Embedded Toolchain: 9-2020-q2-update<br>(GNU ARM Embedded 9.3.1.20200408) |

【注】 あらかじめ、以下の URL からツールをダウンロードしてインストールしてください。

- RA 向け e<sup>2</sup> studio クイックスタートガイドのダウンロードサイト：  
[Quick Start Guide for e<sup>2</sup> studio for RA](#)
- プラットフォームインストーラ（FSP with e<sup>2</sup> studio installer）のダウンロードサイト：  
<https://github.com/renesas/fsp/releases>
- Microsoft Azure RTOS ThreadX ユーザガイドのサイト：  
[Azure RTOS ThreadX に関するガイド | Microsoft Docs](#)
- Microsoft Azure RTOS TraceX ユーザガイドのサイト：  
[Azure RTOS TraceX の概要 | Microsoft Docs](#)
- Microsoft Azure RTOS TraceX のダウンロードサイト：  
[Azure RTOS TraceX - Microsoft Store](#)

## 目次

|                                                                 |    |
|-----------------------------------------------------------------|----|
| 1. プラットフォームインストーラ（FSP with e <sup>2</sup> studio）によるインストール..... | 3  |
| 2. Azure RTOS TraceX のインストール.....                               | 3  |
| 3. e <sup>2</sup> studio プロジェクトの作成 .....                        | 3  |
| 4. FSP の構成.....                                                 | 7  |
| 4.1 Blinky スレッドの設定.....                                         | 7  |
| 4.2 プロジェクトのビルド .....                                            | 9  |
| 5. PC と EK-RA6M3 ボードの接続.....                                    | 10 |
| 5.1 デバッグ設定.....                                                 | 10 |
| 6. RTOS リソースビューの使用.....                                         | 11 |
| 6.1 RTOS リソースビューの表示 .....                                       | 11 |
| 6.2 コンテキストメニュー .....                                            | 12 |
| 6.3 スタック設定.....                                                 | 13 |
| 6.3.1 スタックデータロードの有効化とスタックしきい値の設定 .....                          | 13 |
| 6.4 タブメニュー.....                                                 | 14 |
| 7. Azure RTOS TraceX を使用したプロジェクトデバッグの開始 .....                   | 16 |
| 7.1 e <sup>2</sup> studio でのデバッガの起動 .....                       | 16 |
| 7.2 Azure RTOS TraceX の起動.....                                  | 17 |
| 7.3 分析モード.....                                                  | 18 |
| 改訂記録.....                                                       | 20 |

### 1. プラットフォームインストーラ（FSP with e<sup>2</sup> studio）によるインストール

Azure RTOS ThreadX は Flexible Software Package（FSP）に含まれていますので、インストールが必要なのは FSP と Azure RTOS TraceX のみです。最初に、Renesas e<sup>2</sup> studio 2021-04 or higher User's Manual: Quick Start Guide の「2.1 Installing the FSP with e<sup>2</sup> studio Installer」を参照して FSP をインストールしてください。

### 2. Azure RTOS TraceX のインストール

Azure RTOS TraceX ユーザガイドの「第 2 章 Azure RTOS TraceX のインストールと使用方法」を参照してインストールしてください。

### 3. e<sup>2</sup> studio プロジェクトの作成

e<sup>2</sup> studio にはプロジェクト生成ウィザードが用意されており、プロジェクト名、関連づけるデバイスとボード、ドライバを指定して RA プロジェクトを生成することができます。e<sup>2</sup> studio を起動して、起動画面でワークスペースフォルダを選択してください。新規 RA プロジェクトは、以下の手順で作成します。

1. [ファイル]メニュー → [新規] → [Renesas C/C++ Project] → [Renesas RA]の順に選択します。
2. "Renesas RA C/C++ Project"テンプレートを選択します。[次へ]で次に進みます。

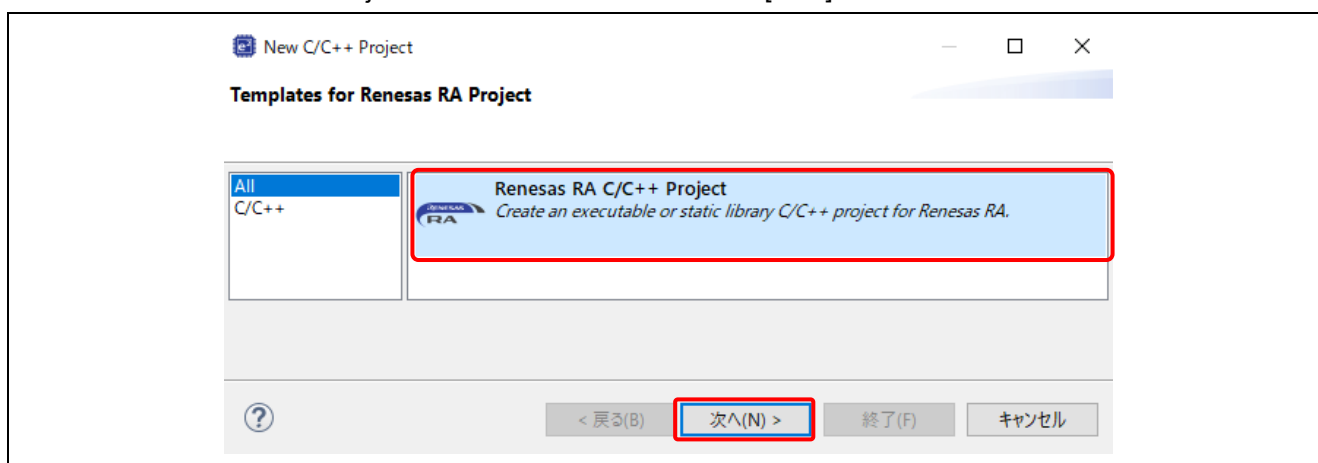


図 1 テンプレートの選択

3. 表示されたダイアログボックスでプロジェクト名を入力し、[次へ]をクリックします。

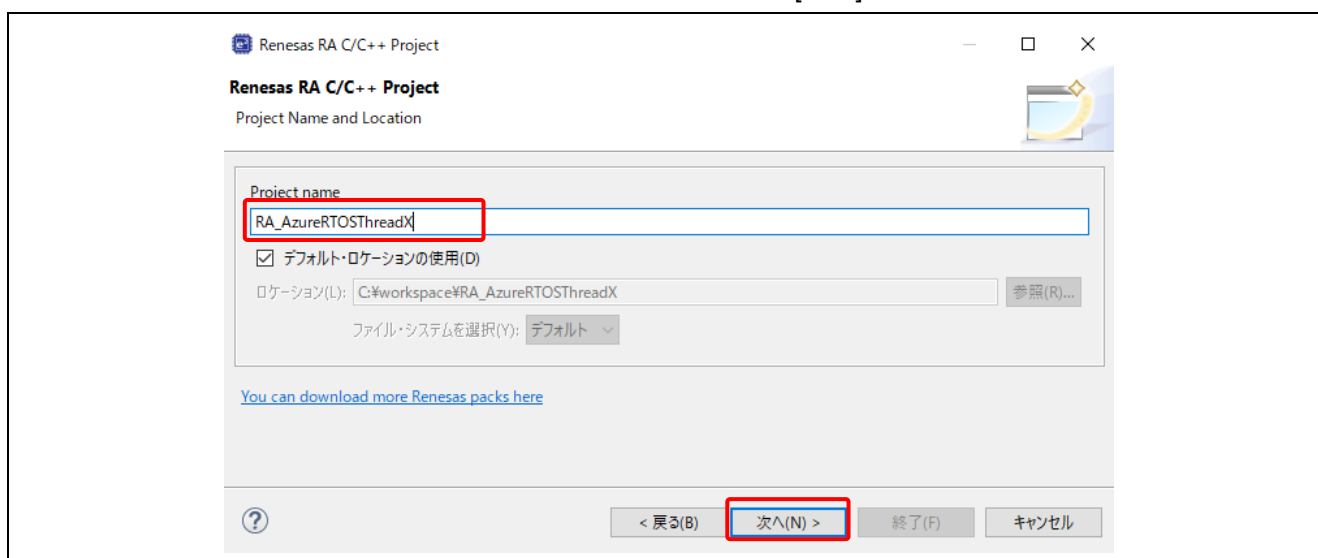


図 2 プロジェクト名と作成場所

### 4. デバイス選択の画面で、デバイスとツールの情報を次のように入力します。

- FSP Version : **3.0.0**
- Board : **EK-RA6M3**
- Device : 自動的に選択されます。
- Language : **C**
- Toolchains : ルネサス RA ファミリ向けに認定された最新の GNU Arm Embedded Toolchain を選択します。(例 : GCC ARM Embedded 9.3.1.20200408)
- Debugger : **J-Link ARM**
- [次へ]で次に進みます。

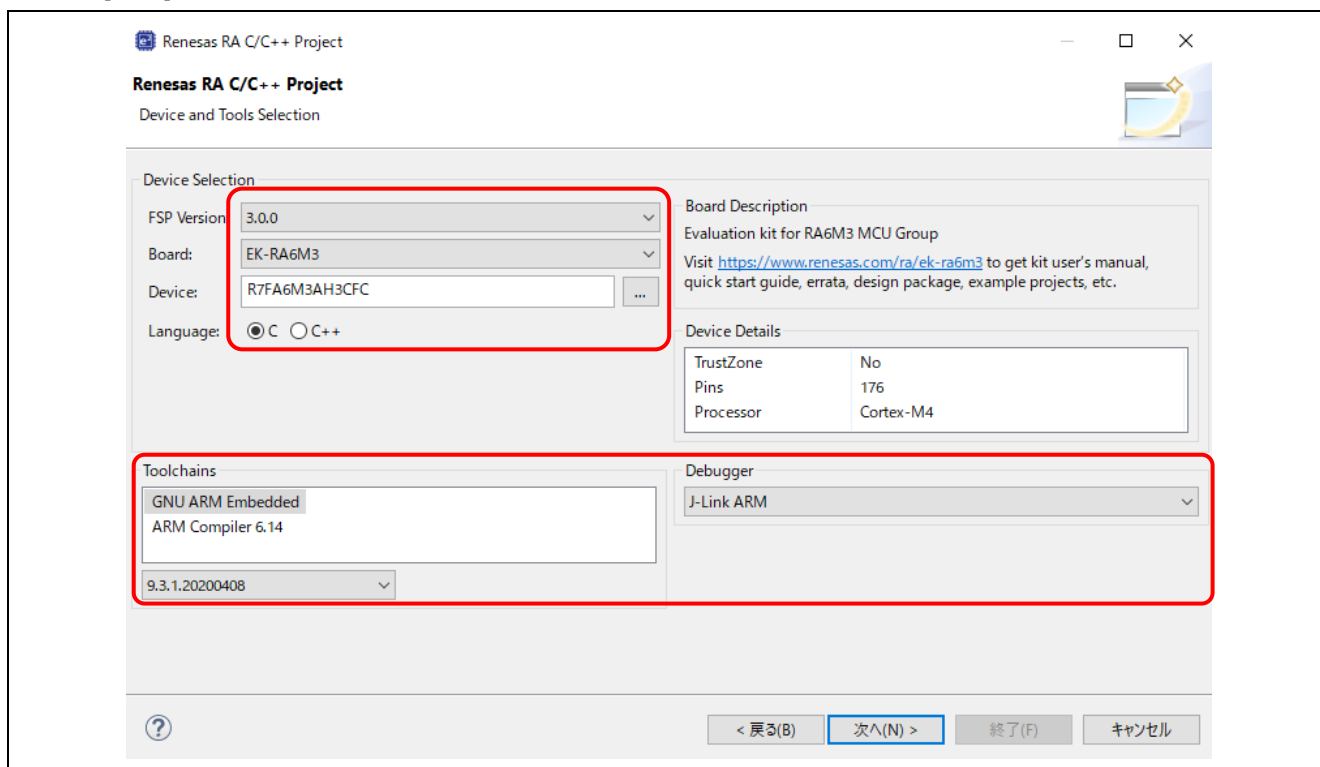


図3 EK-RA6M3 用の新規プロジェクトの作成

5. Build Artifact Selection : **Executable** を選択します。  
RTOS Selection : **Azure RTOS ThreadX** を選択します。

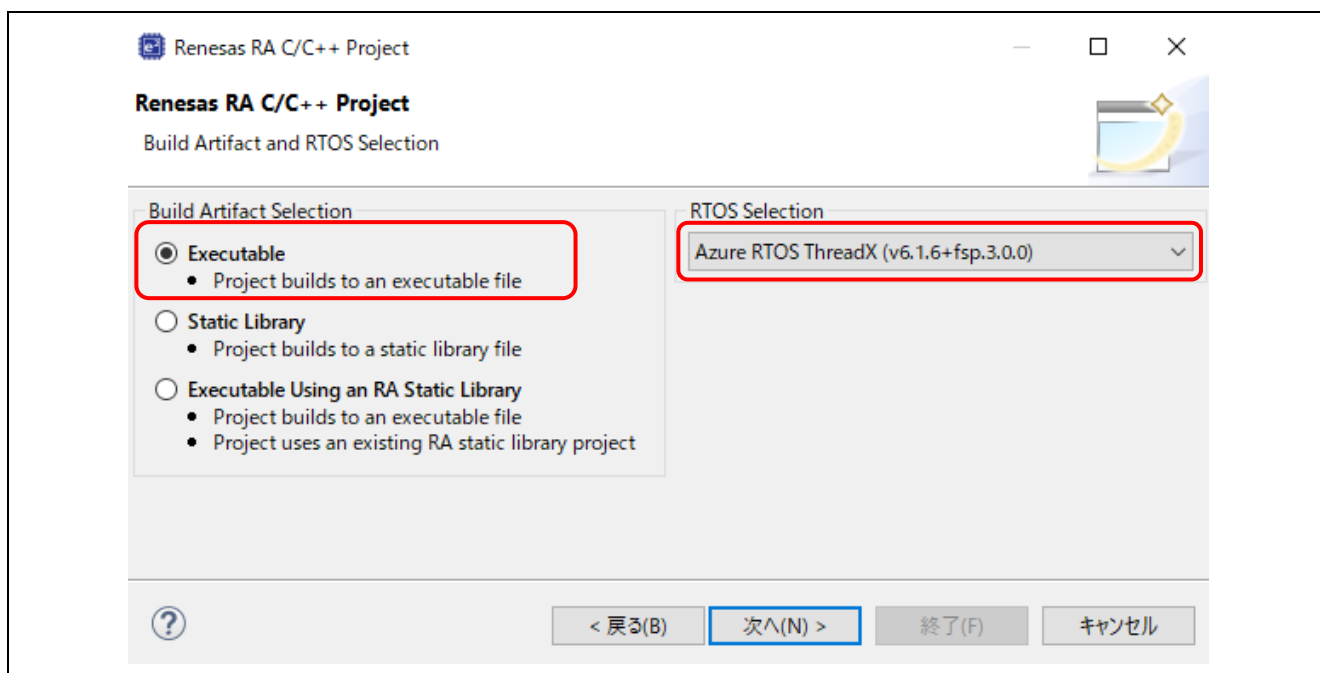


図 4 Build Artifact と RTOS の選択

6. プロジェクトテンプレート選択の画面で、"Azure RTOS ThreadX – Blinky"を選択し、[終了]をクリックします。

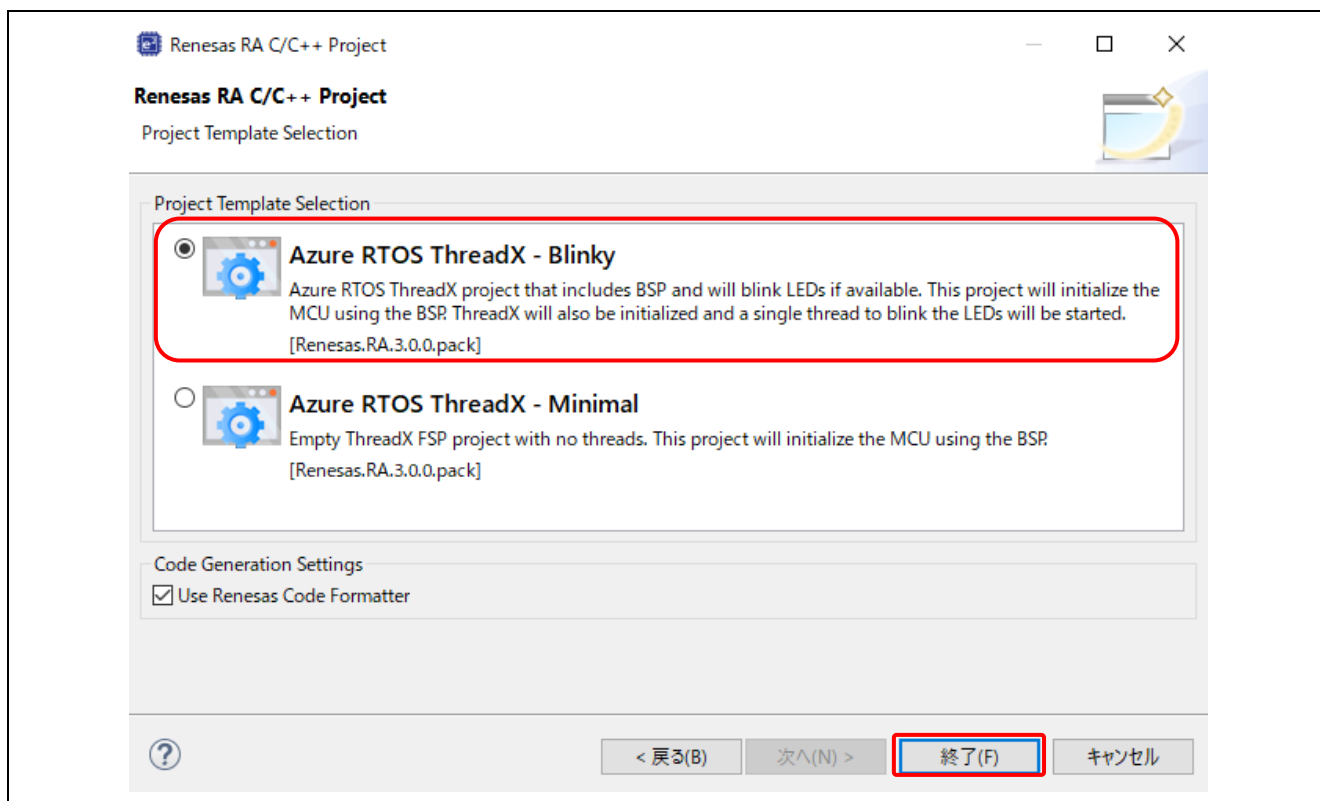


図 5 プロジェクトテンプレートの選択

## ルネサス RA ファミリ Azure RTOS TraceX を用いた Azure RTOS ThreadX のデバッグ

7. ここまでの手順が完了すると、e<sup>2</sup> studio は新規プロジェクトを作成します。[FSP Configuration] パースペクティブが開き、プロジェクトの設定が可能となります。

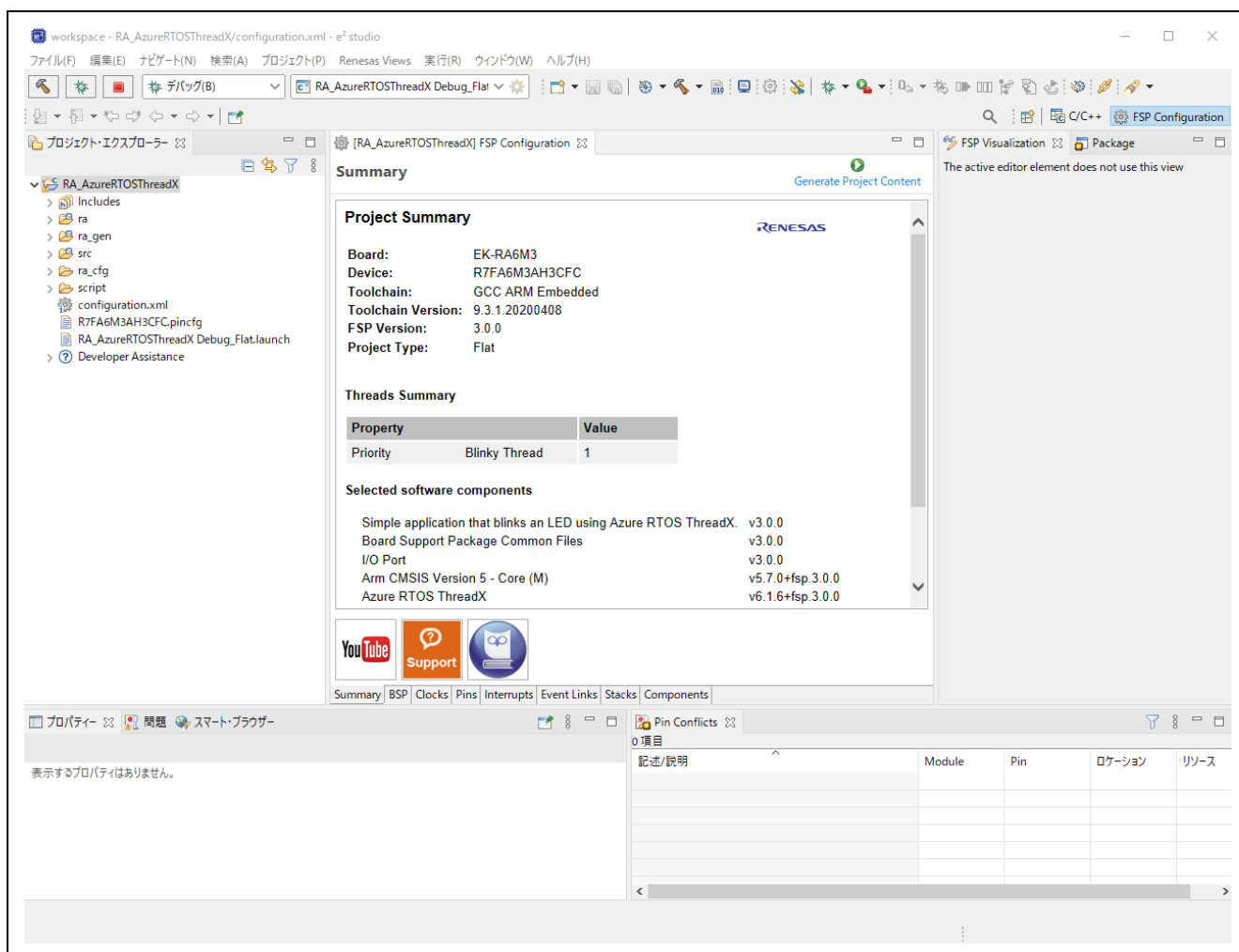


図 6 EK-RA6M3 用の新規プロジェクト

## 4. FSP の構成

### 4.1 Blinky スレッドの設定

トレースバッファのデフォルト設定は、図 7 の緑の四角で囲んだ部分です。

図に示すように、Blinky スレッドのプロパティを変更します。

スレッドのプロパティの定義名と Azure RTOS ThreadX の定義名の対応を以下に示します。

表 1 定義名の対応

| プロパティ                        | 値の変更                    | Azure RTOS ThreadX の定義名                |
|------------------------------|-------------------------|----------------------------------------|
| Event Trace                  | Enabled に<br>変更<br>(必須) | TX_ENABLE_EVENT_TRACE                  |
| Block Pool Performance Info  | (任意)                    | TX_BLOCK_POOL_ENABLE_PERFORMANCE_INFO  |
| Byte Pool Performance Info   | (任意)                    | TX_BYTE_POOL_ENABLE_PERFORMANCE_INFO   |
| Event Flags Performance Info | (任意)                    | TX_EVENT_FLAGS_ENABLE_PERFORMANCE_INFO |
| Mutex Performance Info       | (任意)                    | TX_MUTEX_ENABLE_PERFORMANCE_INFO       |
| Queue Performance Info       | (任意)                    | TX_QUEUE_ENABLE_PERFORMANCE_INFO       |
| Semaphore Performance Info   | (任意)                    | TX_SEMAPHORE_ENABLE_PERFORMANCE_INFO   |
| Thread Performance Info      | (任意)                    | TX_THREAD_ENABLE_PERFORMANCE_INFO      |
| Timer Performance Info       | (任意)                    | TX_TIMER_ENABLE_PERFORMANCE_INFO       |

【注】 詳細は、「動作環境」に記載の Azure RTOS ThreadX ユーザガイドの「第 2 章 Azure RTOS ThreadX のインストールと使用」の“構成オプション”を参照してください。

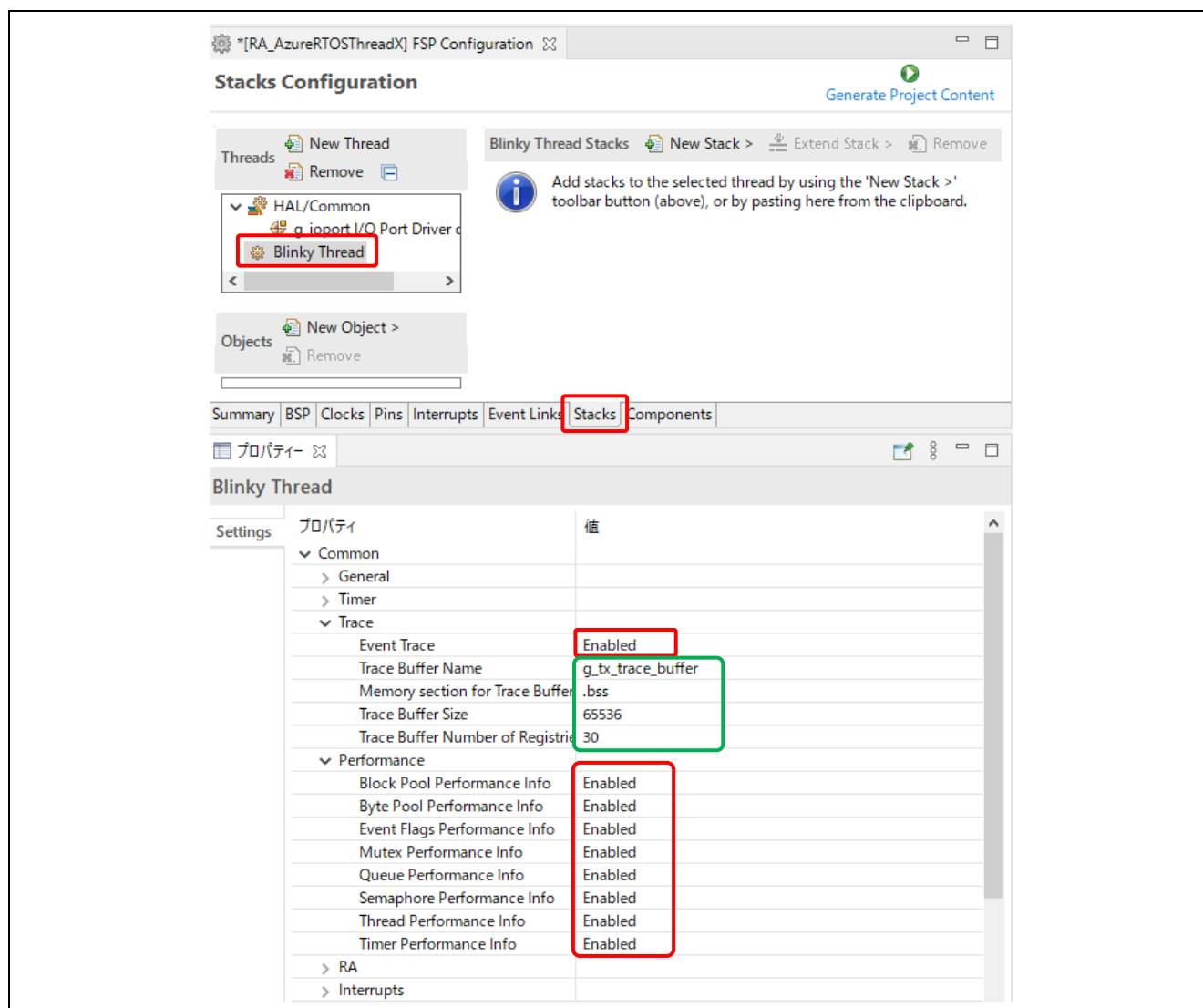
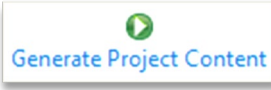


図 7 Blinky スレッドのプロパティ

プロジェクト内容を生成するには、

- 
 ボタンをクリックしてソースファイルを生成します。

## 4.2 プロジェクトのビルド

プロジェクトを右クリックして、[プロジェクトのビルド]を選択します。エラーがないことを確認してください。

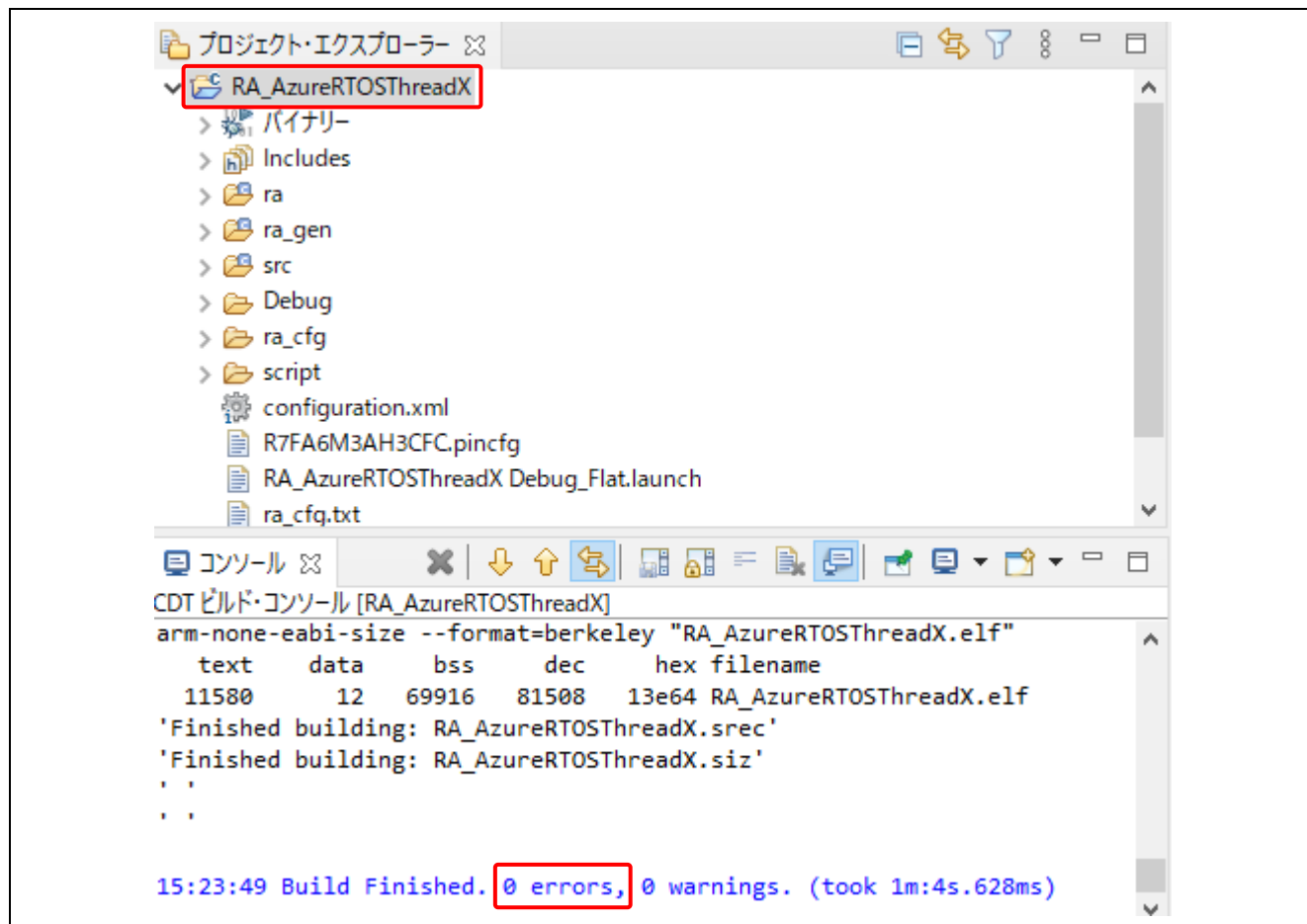


図 8 プロジェクトのビルド

## 5. PC と EK-RA6M3 ボードの接続

図 9 にホスト PC と EK-RA6M3 ボードの接続を示します。

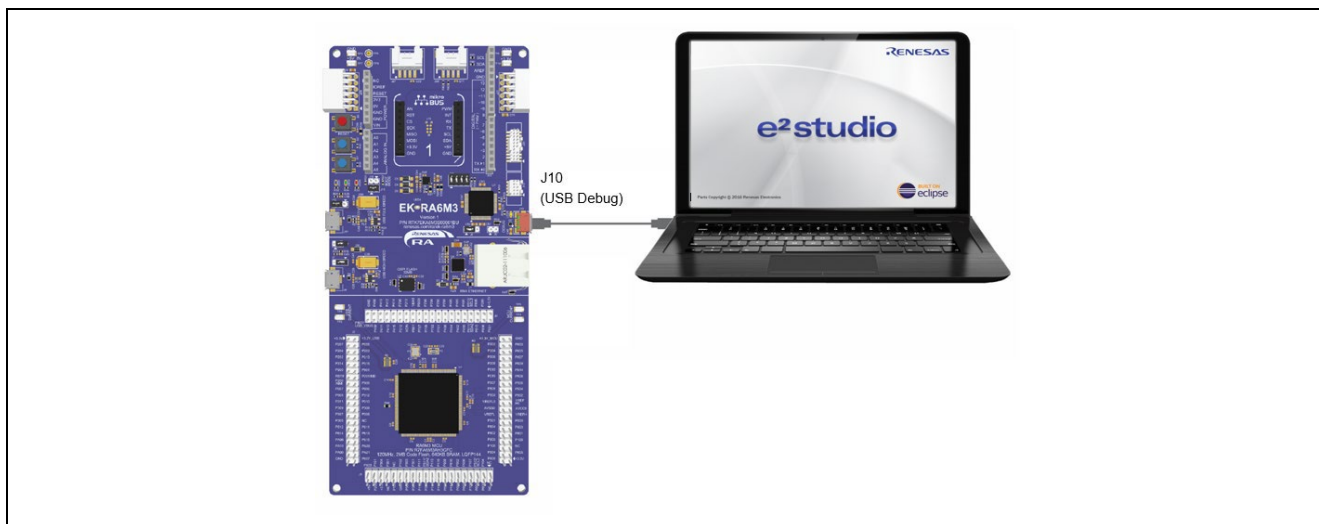


図 9 EK-RA6M3 ボードの接続

### 5.1 デバッグ設定

ハードウェア設定を以下に示します。

表 2 デバッグモード別のジャンパ接続一覧

| デバッグモード   | J8                | J9   | J29                           |
|-----------|-------------------|------|-------------------------------|
| オンボードデバッグ | ピン 1-2 にジャンパを取り付け | オープン | ピン 1-2、3-4、5-6、7-8 にジャンパを取り付け |

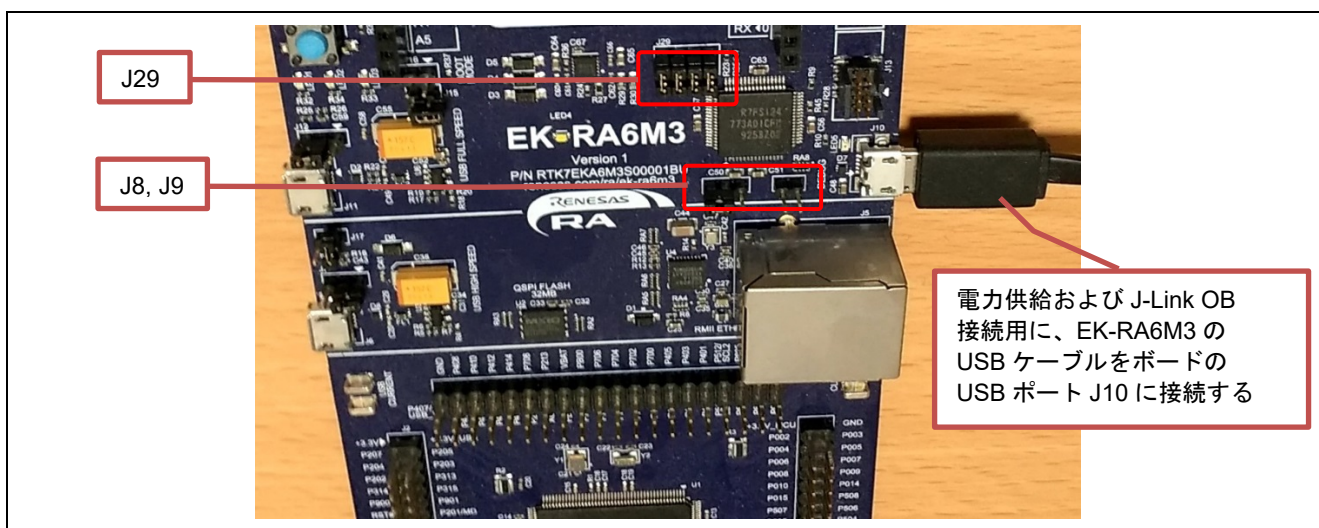


図 10 ホスト PC と EK-RA6M3 ボードの接続

## 6. RTOS リソースビューの使用

e<sup>2</sup> studio には[RTOS リソース]ビュー機能があり、Azure RTOS ThreadX のリソースの状態を表示できます。この章では、[RTOS リソース]ビューの使用手順を説明します。

### 6.1 RTOS リソースビューの表示

[RTOS リソース]ビュー機能が使用できるのはデバッグ実行中のみです。デバッグを起動してから [Renesas Views] → [パートナーOS] → [RTOS リソース]の順に選択します。[OS 選択]ダイアログボックスが表示されたら図 11 のように"ThreadX"を選択してください。図 12 のように[RTOS リソース]ビューが開きます。

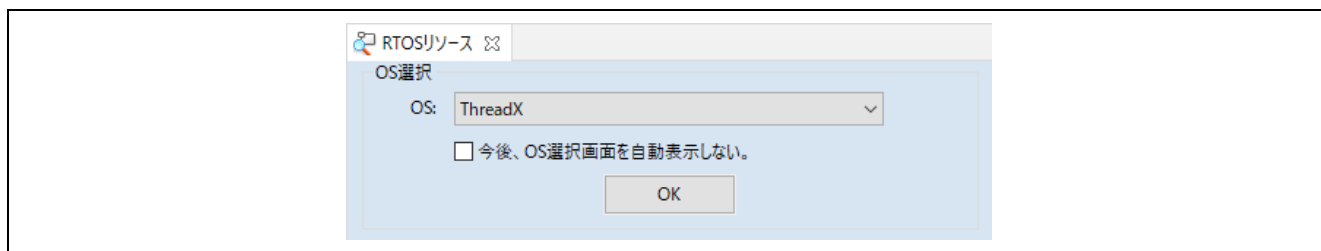


図 11 OS の選択

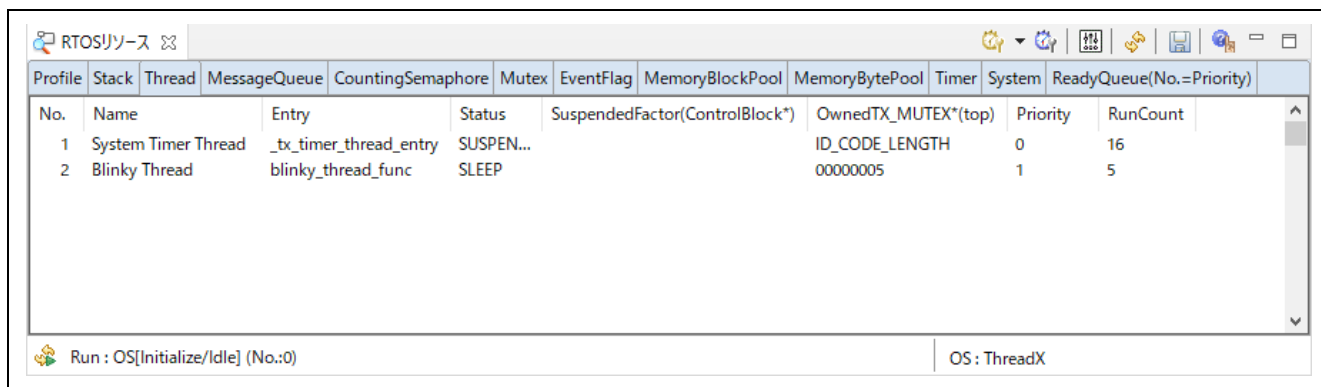


図 12 [RTOS リソース]ビュー

## 6.2 コンテキストメニュー

[RTOS リソース]ビュー上でマウスの右ボタンをクリックすると、コンテキストメニューが開きます。

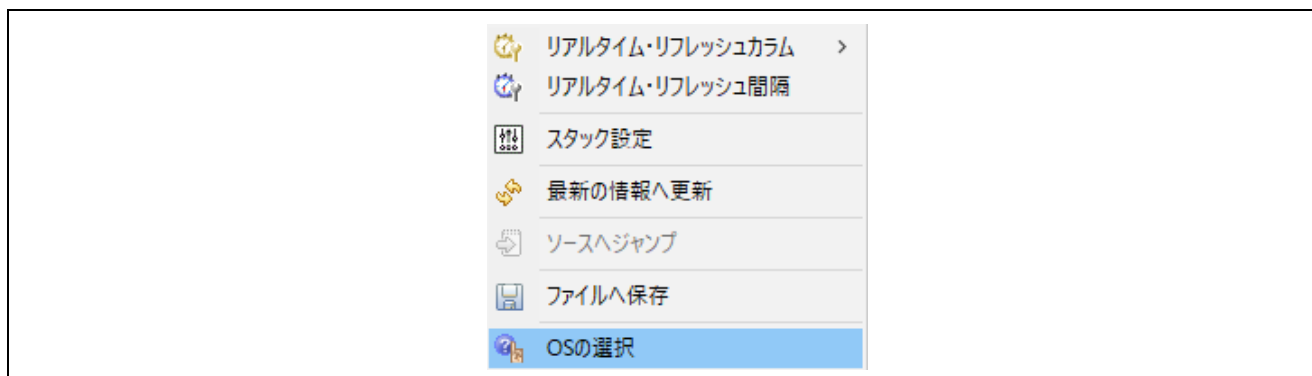


図 13 コンテキストメニュー

- **リアルタイム・リフレッシュカラム**  
表示されている項目の内容をリアルタイムに更新できます。  
このメニューは、プログラム実行中はグレー表示で選択できません。
- **リアルタイム・リフレッシュ間隔**  
リアルタイム表示を更新する間隔を指定します。設定できる間隔は 500 ms から 10000 ms の範囲です。  
このメニューは、プログラム実行中はグレー表示で選択できません
- **スタック設定**  
スタックデータロードを有効または無効にし、スタック警告のしきい値を設定します。  
このメニューは、プログラム実行中はグレー表示で選択できません
- **最新の情報へ更新**  
表示を最新の情報に更新します。
- **ソースヘジャンプ**  
[エディタ]ビューを開き、タスク/スレッドまたはハンドラのソースコードを表示します。[エディタ]ビューは、タスク/スレッドまたはハンドラをダブルクリックしても開くことができます。  
このメニューは、プログラム実行中はグレー表示で選択できません
- **ファイルへ保存**  
現在選択されているタブのデータをテキストファイル (\*.txt) に保存します。  
このメニューは、プログラム実行中はグレー表示で選択できません
- **OS の選択**  
[OS 選択]ダイアログボックスを開きます。  
このメニューは、プログラム実行中はグレー表示で選択できません

## 6.3 スタック設定

### 6.3.1 スタックデータロードの有効化とスタックしきい値の設定

1. コンテキストメニューを開き、[スタック設定]を選択します。
2. スタックデータを[RTOS リソース]ビューにロードするには、[スタック設定]ダイアログボックスの"スタックデータロードを有効にする"のチェックボックスを選択します。これを有効にしていないと、次のデバッグセッションでスタックデータがロードされません。

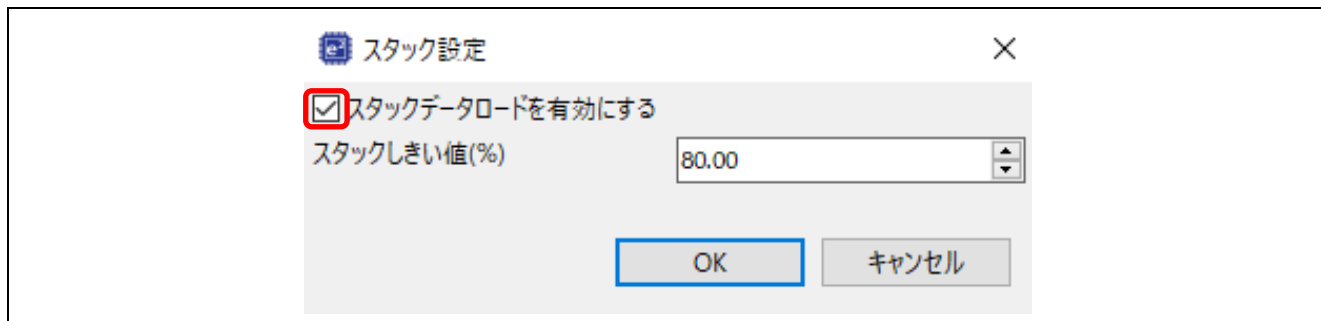


図 14 スタックデータロードの有効化

3. 任意のスタックしきい値を[スタックしきい値(%)]テキストボックスに設定できます。[OK]で設定を保存します。

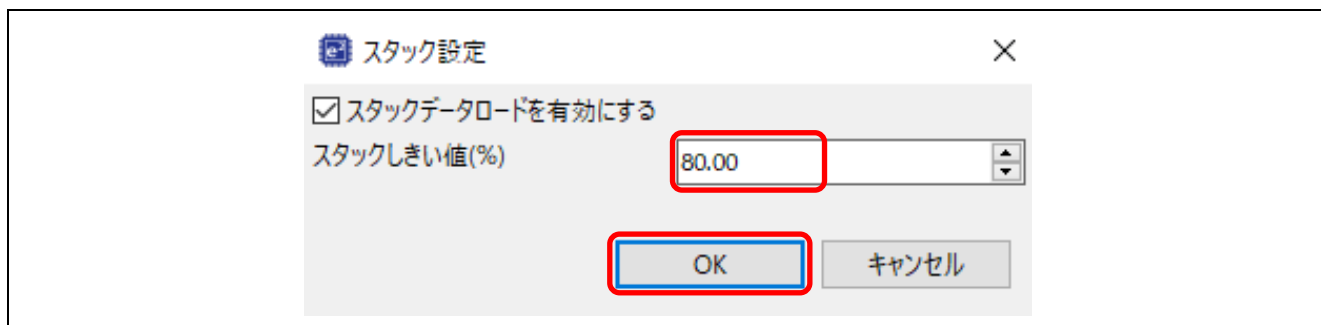


図 15 スタックしきい値の設定

## 6.4 タブメニュー

各タブに表示する項目を表 3 に示します。

表 3 各タブに表示する内容

| [RTOS リソース]<br>ビューのタブの名称 | 表示する情報と選択項目の名称                    | 表示内容            |
|--------------------------|-----------------------------------|-----------------|
| Profile                  | [Profile]タブの機能は、<br>現在使用できません。    | —               |
| Stack                    | Name                              | スレッド名           |
|                          | Entry                             | 各スレッドを開始した関数名   |
|                          | 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)                  | スタックサイズ         |
|                          | MaxStackUsage(bytes)              | 現在使用中の最大スタックサイズ |
| Thread                   | Name                              | スレッド名           |
|                          | Entry                             | 各スレッドを開始した関数名   |
|                          | Status                            | スレッドの状態         |
|                          | Suspended Factor (Control Block*) | 中断の要因となったリソース   |
|                          | OwnedTX_MUTEX*(top)               | 獲得した上位のミューテックス  |
|                          | Priority                          | 優先度             |
|                          | RunCount                          | スレッドを実行した回数     |
| MessageQueue             | Name                              | メッセージキュー名       |
|                          | UsedCount                         | 使用中のメッセージキュー数   |
|                          | FreeCount                         | 使用可能なメッセージキュー数  |
|                          | TotalCount                        | メッセージキューの総数     |
|                          | MessageSize                       | メッセージサイズ        |
|                          | SuspendedTX_THREAD*(top)          | キューに待機中の先頭スレッド  |
|                          | SuspendedCount                    | 中断中のスレッド数       |
|                          | 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                              | セマフォ名           |
|                          | SemaphoreCount                    | セマフォ数           |
|                          | SuspendedTX_THREAD*(top)          | キューに待機中の先頭スレッド  |
|                          | SuspendedCount                    | 中断中のスレッド数       |
| Mutex                    | Name                              | ミューテックス名        |
|                          | OwnerTX_THREAD*                   | 獲得しているスレッド      |
|                          | OwnerCount                        | 所有者の数           |
|                          | SuspendedTX_THREAD*(top)          | キューに待機中の先頭スレッド  |
| EventFlag                | SuspendedCount                    | 中断中のスレッド数       |
|                          | Name                              | イベントフラグ名        |
|                          | Flag                              | 現在のフラグパターン      |
|                          | SuspendedTX_THREAD*(top)          | キューに待機中の先頭スレッド  |

|                            |                          |                   |
|----------------------------|--------------------------|-------------------|
| MemoryBlockPool            | Name                     | メモリブロック名          |
|                            | FreeCount                | 使用可能なブロック数        |
|                            | TotalCount               | ブロックの総数           |
|                            | BlockSize(bytes)         | ブロックサイズ           |
|                            | TotalSize(bytes)         | メモリブロックプールの合計サイズ  |
|                            | SuspendedTX_THREAD*(top) | キューに待機中の先頭スレッド    |
|                            | SuspendedCount           | 中断中のスレッド数         |
|                            | StartAddress             | メモリブロックプールの先頭アドレス |
| MemoryBytePool             | Name                     | メモリプール名           |
|                            | Free(bytes)              | 使用可能なバイト数         |
|                            | Total(bytes)             | メモリバイトプールの合計サイズ   |
|                            | FragmentCount            | フラグメント数           |
|                            | SuspendedTX_THREAD*(top) | キューに待機中の先頭スレッド    |
|                            | SuspendedCount           | 中断中のスレッド数         |
|                            | StartAddress             | メモリバイトプールの先頭アドレス  |
| Timer                      | Name                     | タイマ名              |
|                            | 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          |
| ReadyQueue(No. = Priority) | QueuedTX_THREAD*(top)    | 上位の使用可能なスレッド      |

## 7. Azure RTOS TraceX を使用したプロジェクトデバッグの開始

### 7.1 e<sup>2</sup> studio でのデバッガの起動

メニューから[実行] → [デバッグ]の順に選択してデバッガを起動します。

以下の手順で、Azure RTOS TraceX のデータファイルを作成します。

1. "tx\_user.h"の"g\_tx\_trace\_buffer"を"監視する式"に追加して、Azure RTOS TraceX のデータバッファアドレスを取得します。

図 16 の例では、アドレスは 0x1ffe0044 です。

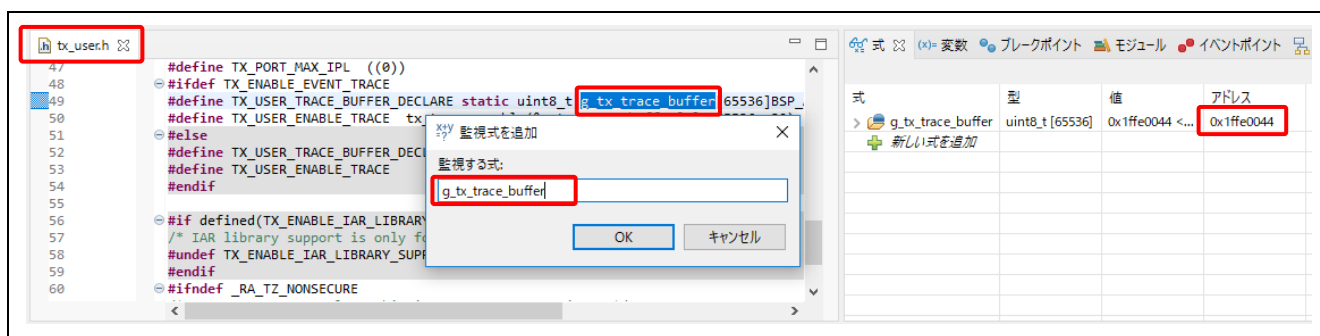


図 16 データバッファアドレスの取得

2. TraceX のデータファイルをエクスポートします。

- (1) [メモリー]タブを選択します。
- (2) [メモリー]タブの右にある[エクスポート]ボタンをクリックします。
- (3) [メモリーをエクスポート]ダイアログボックスが開きます。
- (4) "Format"に、"RAW Binary"を選択します。
- (5) "Start address"に、手順 1 で取得した TraceX データバッファアドレス 0x1ffe0044 を入力します。
- (6) "Length"に、TraceX データバッファ"g\_tx\_trace\_buffer"のサイズとして 65536 を入力します。
- (7) "File name"に、拡張子.trx の任意のファイル名を指定します。
- (8) [OK]をクリックします。(TraceX のデータファイルが(7)で指定したファイル名でエクスポートされます。)

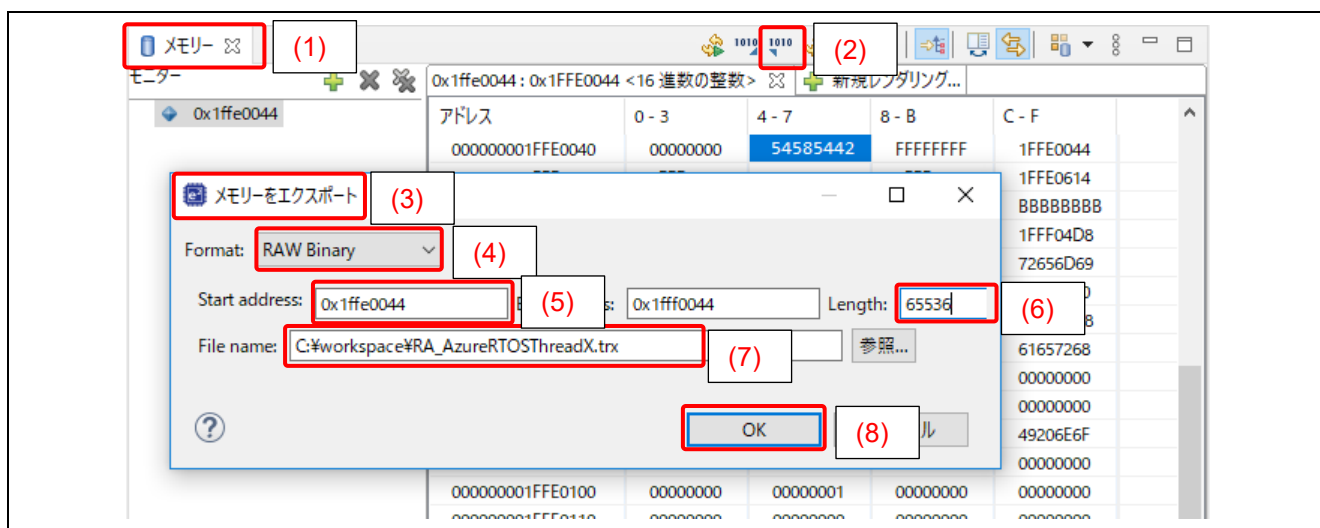


図 17 データファイルのエクスポート

## 7.2 Azure RTOS TraceX の起動

ホスト PC にインストールした Azure RTOS TraceX を起動します。

Azure RTOS TraceX のメニューから[File] → [Open]の順にクリックします。

7.1 章の手順 2 でエクスポートした TraceX データファイル"\*\*\*\*.trx"を選択します。

TraceX 上にトレース情報が表示されます。

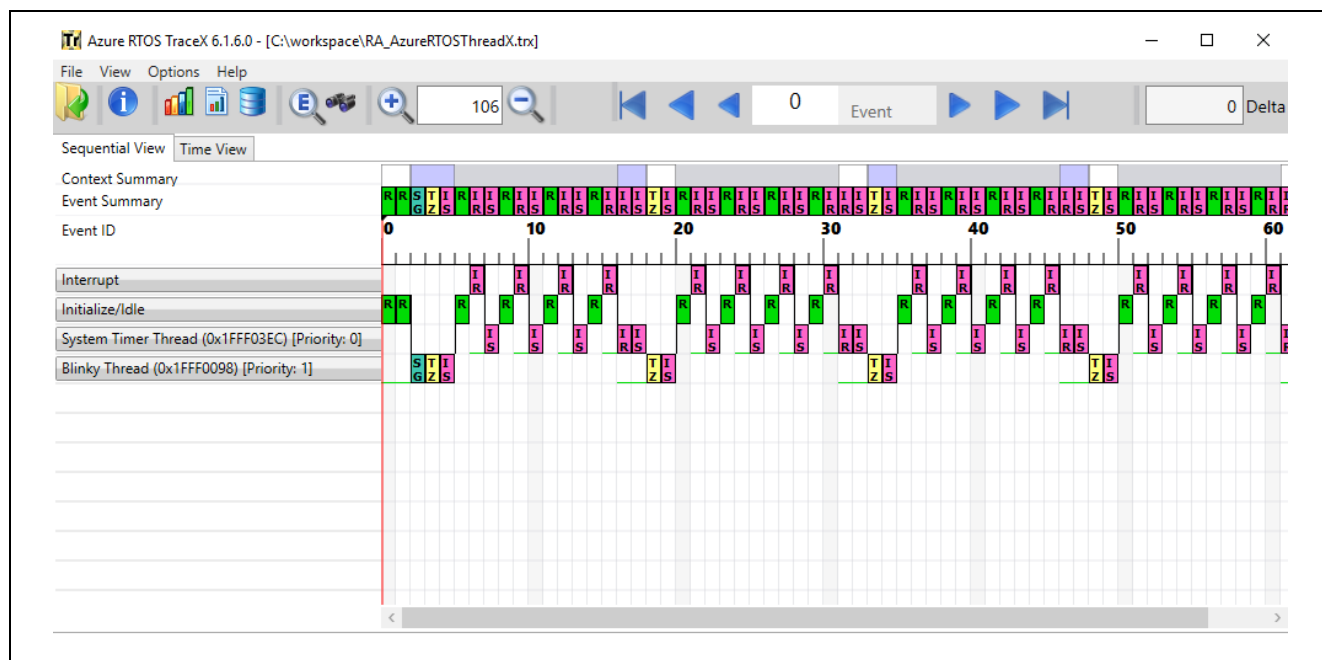


図 18 Azure RTOS TraceX の起動

## 7.3 分析モード

各種の分析モードが使用できます。詳細は、[Help] → [Manual]の順に選択して情報を参照してください。

TraceX ユーザガイド | Microsoft Docs の「[第 3 章 Azure RTOS TraceX の説明](#)」の以下の説明も参照してください。

- シーケンシャルビュー（Sequential View）モード
- タイムビュー（Time View）モード

TraceX ユーザガイド | Microsoft Docs の「[第 4 章 Azure RTOS TraceX のパフォーマンス分析](#)」の以下の説明も参照してください。

- 実行プロファイル（Execution Profile）
- 人気のあるサービス（Popular Services）
- スレッド スタックの使用状況（Thread Stack Usage）
- パフォーマンス統計（Performance Statistics）
- FileX 統計（FileX Statistics）
- NetX 統計（NetX Statistics）
- トレース ファイル情報（Trace File Information）

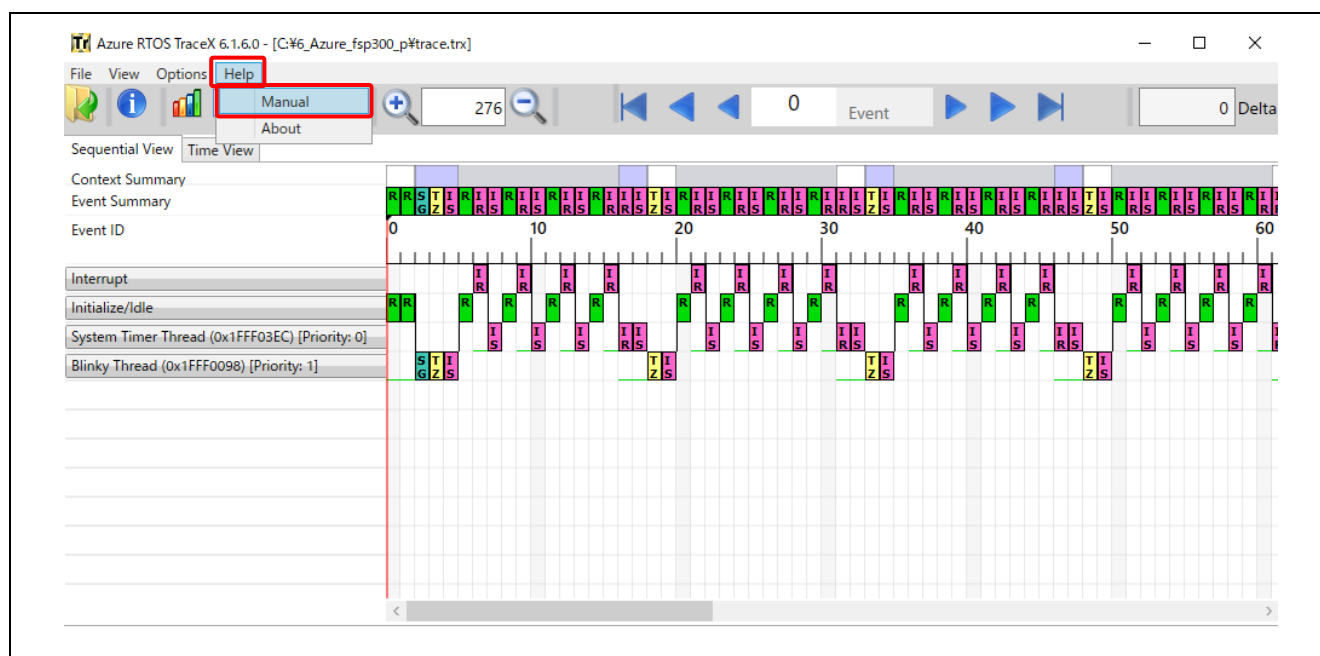


図 19 トレース情報の表示

【注】 e<sup>2</sup> studio から TraceX ファイルを出力せずに TraceX v5.2.0 を使用方法もあります。詳細は、[TraceX® by Microsoft Corp. | Renesas](#) を参照してください。この方法はルネサス RA ファミリでも対応しています。

## ホームページとサポート窓口

RA ファミリの主な機能、ダウンロードコンポーネント、関連ドキュメント、サポートについては、下記のサイトにアクセスしてください。

|                                   |                                                                        |
|-----------------------------------|------------------------------------------------------------------------|
| RA ファミリ製品情報                       | <a href="http://www.renesas.com/ra">www.renesas.com/ra</a>             |
| RA ファミリのサポートフォーラム                 | <a href="http://www.renesas.com/ra/forum">www.renesas.com/ra/forum</a> |
| RA ファミリ Flexible Software Package | <a href="http://www.renesas.com/FSP">www.renesas.com/FSP</a>           |
| ルネサスサポートサイト                       | <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 | Sep.08.21 | —    | 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うと、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、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)

## 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/)