

Cortex-M 対応 MULCOS-MK

(MULti Core Operating System – Multilayer Kernel)

EK-RA6M5 サンプルガイド



第1版 2025年4月

改訂履歴

| 版数 | 発行日 | 改訂内容 |
|-------|------------|--------------------------------------------------------------------|
| 第 1 版 | 2025 年 4 月 | Cortex-M 対応 MULCOS-MK μ ITRON4.0 仕様版 EK-RA8M1 サンプルガイドの初版を発行 |
| | | |

はじめに

MULCOS-MK (MULTi Core Operating System - Multilayer Kernel) とは、セキュリティレベルの異なる階層化したリアルタイム・オペレーティング・システム (Real Time Operating System、以降 RTOS と省略) の実行環境を管理する多層化カーネル (以降では、本製品を単に本カーネルと呼ぶ) です。

μ ITRON4.0 仕様版の RTOS (以降では、単に本 RTOS と呼ぶ) は、最も多く使われている仕様の 1 つで、当時の社団法人トロン協会が策定し公開した「 μ ITRON4.0 仕様」(現在は、トロンフォーラムが公開) に準拠しています。

本書は Renesas EK-RA6M5 評価ボードを使用したサンプルプログラムの説明書です。本書以外の説明書として、Cortex-M 対応 MULCOS-MK の μ ITRON4.0 仕様版共通部ユーザーズガイドと、RA デバイス依存部ユーザーズガイドとが用意されています。

(μ ITRON とは Micro Industrial TRON、TRON とは The Realtime Operating system Nucleus の略称)

参考資料

- ・社団法人トロン協会発行「 μ ITRON4.0 仕様 Ver.4.03.00」
- ・Arm Limited 社発行「Arm[®]v8-M Architecture Reference Manual」

目次

| | |
|----------------------------------|----|
| 改訂履歴 | 0 |
| はじめに | 1 |
| 目次 | 2 |
| 第 1 章 環境の説明 | 3 |
| 1-1. ハードウェア設定 | 3 |
| 1-2. フォルダ構成 | 4 |
| 1-3. プロジェクトファイル | 5 |
| 第 2 章 サンプルプログラムの説明と動作 | 6 |
| 2-1. メモリマップ | 6 |
| 2-2. サンプルプログラムの初期化 | 7 |
| 2-2-1. メモリマップに依存した初期化 | 7 |
| 2-2-2. クロックの初期化 | 7 |
| 2-2-3. デバイスとポートのセキュア属性の初期化 | 8 |
| 2-2-4. 割込み要因のセキュア属性の初期化 | 9 |
| 2-3. サンプルプログラムの動作 | 10 |
| 2-3-1. セキュア層の処理 | 11 |
| 2-3-2. 非セキュア層の処理 | 11 |
| 第 3 章 サンプルプログラムの起動 | 12 |
| 3-1. サンプルプログラムのデバッグ | 12 |

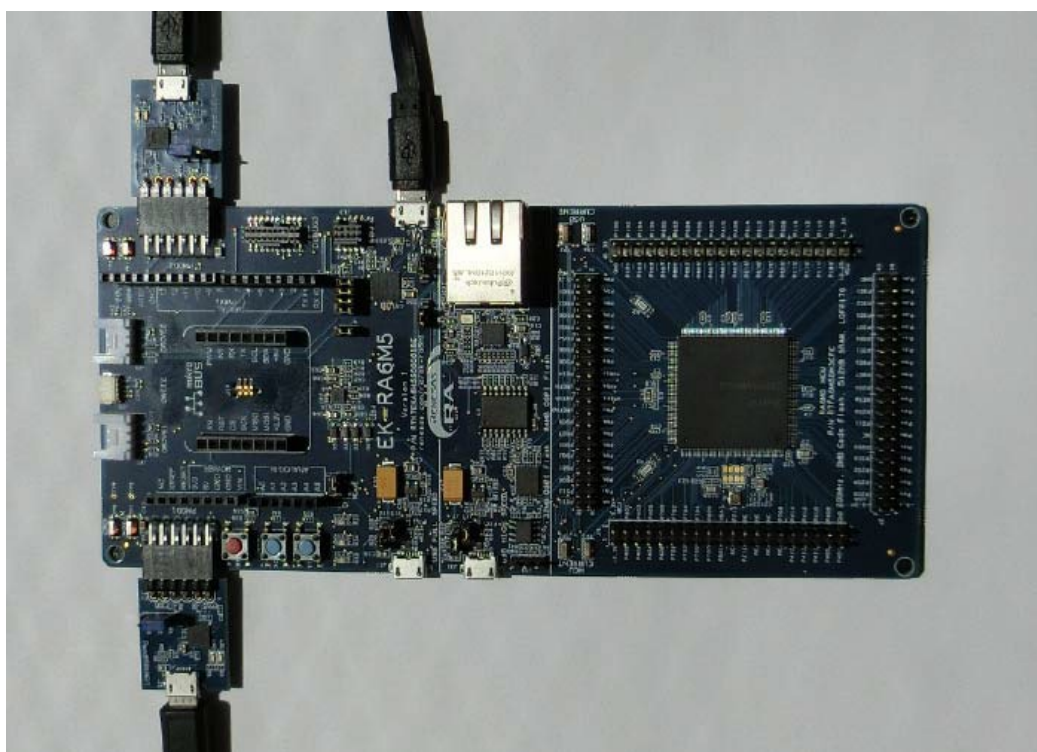
第 1 章 環境の説明

Cortex-M 対応 MULCOS-MK の EK-RA6M5 サンプルプログラムのプロジェクトでは、以下のボードと開発ツールを使用しています。

- [EK-RA6M5 RA6M5 MCU グループ評価キット](#)
- [RA Flexible Software Package \(FSP\)](#)

1-1. ハードウェア設定

プロジェクトは、EK-RA6M5 ボードのオンボードデバッガ (J10) を接続し、Pmod1 を UART の設定にして UART9 で通信するサンプルプログラムです。

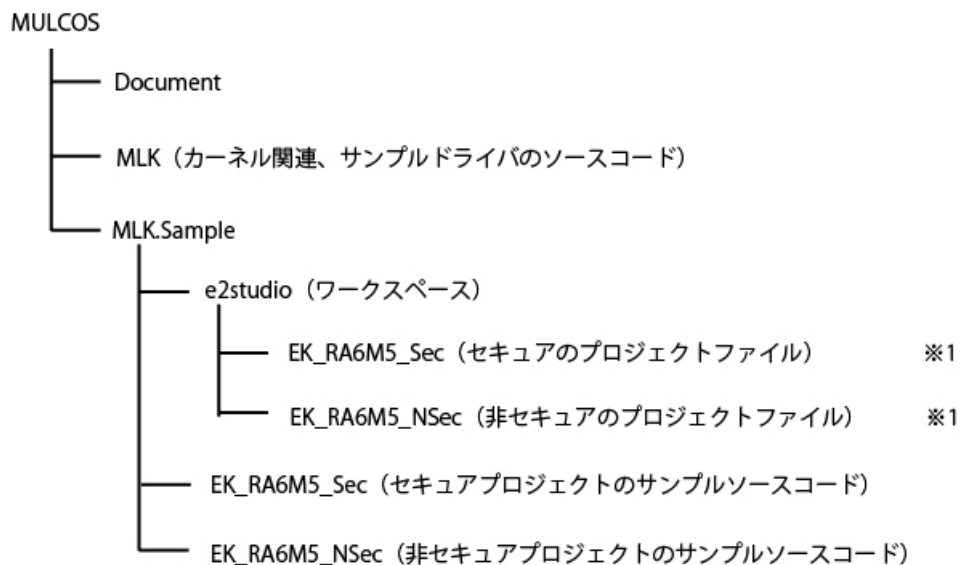


上の写真では、Pmod1 と Pmod2 に以下のモジュールを使用しています。

- [Digilent Pmod USBUART](#)

1-2. フォルダ構成

サンプルプログラムに関連するフォルダは、以下のファイル構成でビルドされています。また、プロジェクトファイルは、ビルドの対象ファイルをコピーせずにリンクすることもできますが、一時的にソース修正することができるように、ファイルはコピーしています。

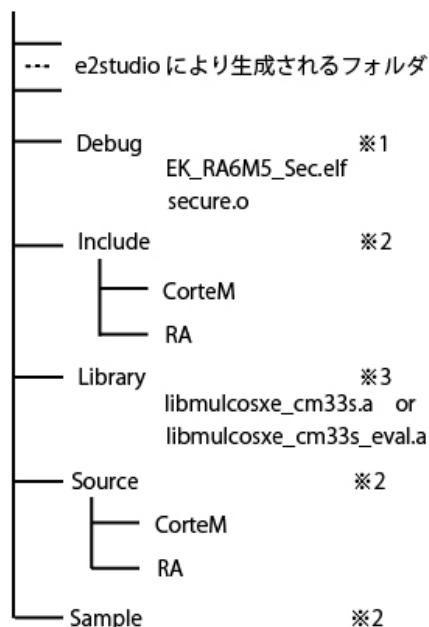


評価版では、プロジェクトファイル（※1）のみの配布となります。

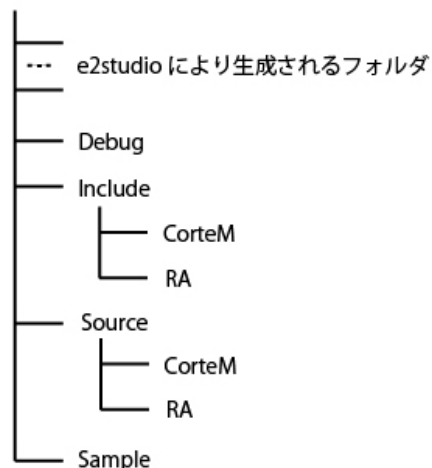
1-3. プロジェクトファイル

サンプルプログラムは、セキュアと非セキュアの 2 つのプロジェクトで構成され、評価版はこれらがエクスポートされています。

EK_RA6M5_Sec (セキュアのプロジェクトファイル)



EK_RA6M5_NSec (非セキュアのプロジェクトファイル)



Debug フォルダ (※1) は e2studio で生成されますが、タイプ 1 評価版では生成できません。フラッシュメモリへの書き込みにはフォルダ内の EK_RA6M5_Sec.elf ファイルを、非セキュアのビルドにはフォルダ内の NSC ライブラリ secure.o を使用します。

Include、Source、Sample フォルダ (※2) は、タイプ 1 評価版には含まれません。

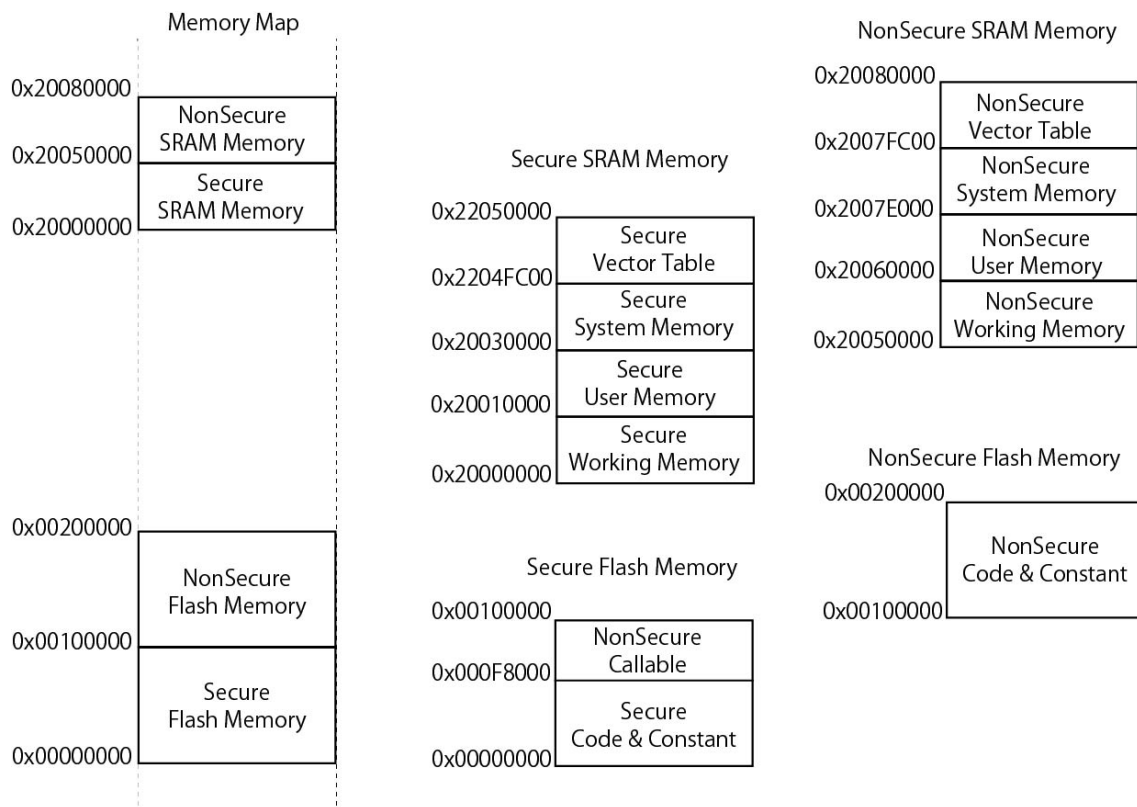
Library フォルダ (※3) には、標準版ではカーネルライブラリ (libmulcosxe_cm33s.a) が、タイプ 2 評価版では制限版のカーネルライブラリ (libmulcosxe_cm33s_eval.a) が含まれています。ただし、タイプ 1 評価版には Library フォルダは含まれません。

第 2 章 サンプルプログラムの説明と動作

EK-RA6M5 ボード用サンプルプログラムのプロジェクトについて説明します。

2-1. メモリマップ

サンプルプログラムでは、メモリ領域を使用状況に合わせて、以下のように定義しています。



RTOS のみが読み書きする特権モードでのみアクセス可能な領域として、

Vector Table 例外ベクタテーブル

System Memory RTOS の制御データとメインスタック領域

アプリケーションを含めて読み書きし、特権モードに限らずアクセス可能な領域として、

User Memory タスクのスタック領域とメモリプール領域

Working Memory アプリケーションが確保し利用する領域

RTOS やアプリケーションのコードと定数の読み出しのみ領域として、

Code & Constant 基本的にコードは RTOS とアプリケーションとを混同した領域

2-2. サンプルプログラムの初期化

SAU や MPU のメモリマップに依存する初期化の他に、デバイスとポートの初期化があります。デバイスとポートでは、タイプ 1 評価版を考慮し、セキュア層で使用する UART と LED 以外は非セキュアで利用できる設定にしています。通常的设计では、それぞれのデバイスやポートをセキュア層で使用するのか、非セキュア層で使用するのかを設計時に決定します。

- メモリマップに依存した初期化
- クロックの初期化
- デバイスとポートのセキュア属性の初期化
- 割込み要因のセキュア属性の初期化

2-2-1. メモリマップに依存した初期化

メモリマップに対応した値で、SAU と MPU は初期化されています。

2-2-2. クロックの初期化

発振器と PLL の初期化により、周波数は以下の設定にしています。

| クロック名 | 周波数 |
|--------|--------|
| CPUCLK | 200MHz |
| PCLKA | 100MHz |
| PCLKB | 50MHz |
| PCLKC | 50MHz |
| PCLKD | 100MHz |

2-2-3. デバイスとポートのセキュア属性の初期化

サンプルプログラムでは、セキュア層で使用している UART9 と LED1-3 に関するデバイスやポート以外では非セキュア属性に初期化しています。

| レジスタ名 | 値 |
|-------|-------------|
| PSARB | 0x780D9B10 |
| PSARC | 0x8C013983 |
| PSARD | 0x1851F830 |
| PSARE | 0xFFFFC030E |
| POSAR | 0xCFFF |
| P1SAR | 0xF07F |
| P2SAR | 0x00FF |
| P3SAR | 0xFFFF |
| P4SAR | 0xBFFF |
| P5SAR | 0xFFFF |
| P6SAR | 0xC9FE |
| P7SAR | 0xFFFF |
| P8SAR | 0xFFFF |
| P9SAR | 0xFFFF |
| PASAR | 0x3FFF |
| PBSAR | 0x00FF |

2-2-4. 割込み要因のセキュア属性の初期化

割込みで使用される例外では、例外番号 16 から 47 までを非セキュアでできるように初期化しています。

| レジスタ名 | 値 |
|------------|------------|
| NVIC_ITNS0 | 0xFFFFFFFF |
| NVIC_ITNS1 | 0x00000000 |
| NVIC_ITNS2 | 0x00000000 |
| ICUSARG | 0xFFFFFFFF |
| ICUSARH | 0x00000000 |
| ICUSARI | 0x00000000 |

NVIC_ITNS0 と ICUSARG、NVIC_ITNS1 と ICUSARH、NVIC_ITNS2 と ICUSARI は同じ値にします。

IRQCRn はすべて非セキュア層で使用する設定で初期化しています。

| レジスタ名 | 値 |
|---------|------------|
| ICUSARA | 0x0000FFFF |

通常的设计では、セキュア層と非セキュア層とで割込みを使う数を、IRQCRn をどちらのセキュアで使用するのかを、設計時に決定します。

2-3. サンプルプログラムの動作

サンプルプログラムはセキュア層と非セキュア層とで、2つのプロジェクトファイルに分割されており、UARTデバイスとLEDがセキュア層に配置すべき処理だと仮定しています。

セキュア層では以下の関数をセキュアライブラリとして定義しています。

| 関数名 | 説明 |
|--------------------------------------------------------------------------|--------------|
| <code>void Led1_On(void);</code> | ユーザ LED1 の点灯 |
| <code>void Led1_Off(void);</code> | ユーザ LED1 の消灯 |
| <code>void Led2_On(void);</code> | ユーザ LED2 の点灯 |
| <code>void Led2_Off(void);</code> | ユーザ LED2 の消灯 |
| <code>void Led3_On(void);</code> | ユーザ LED3 の点灯 |
| <code>void Led3_Off(void);</code> | ユーザ LED3 の消灯 |
| <code>ER ini_uart9(T_UART_DEF *uart_def);</code> | UART9 の初期化 |
| <code>ER ref_uart9(T_UART_REF *uart_ref);</code> | UART9 の状態参照 |
| <code>ER ctr_uart9(UINT cdata);</code> | UART9 の制御 |
| <code>ER fls_uart9(TMO timeout);</code> | UART9 の送出待ち |
| <code>ER putc_uart9(VB data, TMO timeout);</code> | UART9 の一文字送信 |
| <code>ER puts_uart9(VB *data, UINT *count, TMO timeout);</code> | UART9 の文字列送信 |
| <code>ER getc_uart9(VB *data, UB *sts, TMO timeout);</code> | UART9 の一文字受信 |
| <code>ER gets_uart9(VB *data, UB *sts, UINT *count, TMO timeout);</code> | UART9 の文字列受信 |

2-3-1. セキュア層の処理

UART9 を下記で初期化し、セキュアの起動メッセージを表示後はエコーバックを行います。

| | |
|---------|-----------|
| ボーレート | 115200BPS |
| パリティ | パリティなし |
| ストップビット | 1 |

2-3-2. 非セキュア層の処理

UART0 を下記で初期化し、非セキュアの起動メッセージを表示後は以下の処理を行います。

| | |
|---------|-----------|
| ボーレート | 115200BPS |
| パリティ | パリティなし |
| ストップビット | 1 |

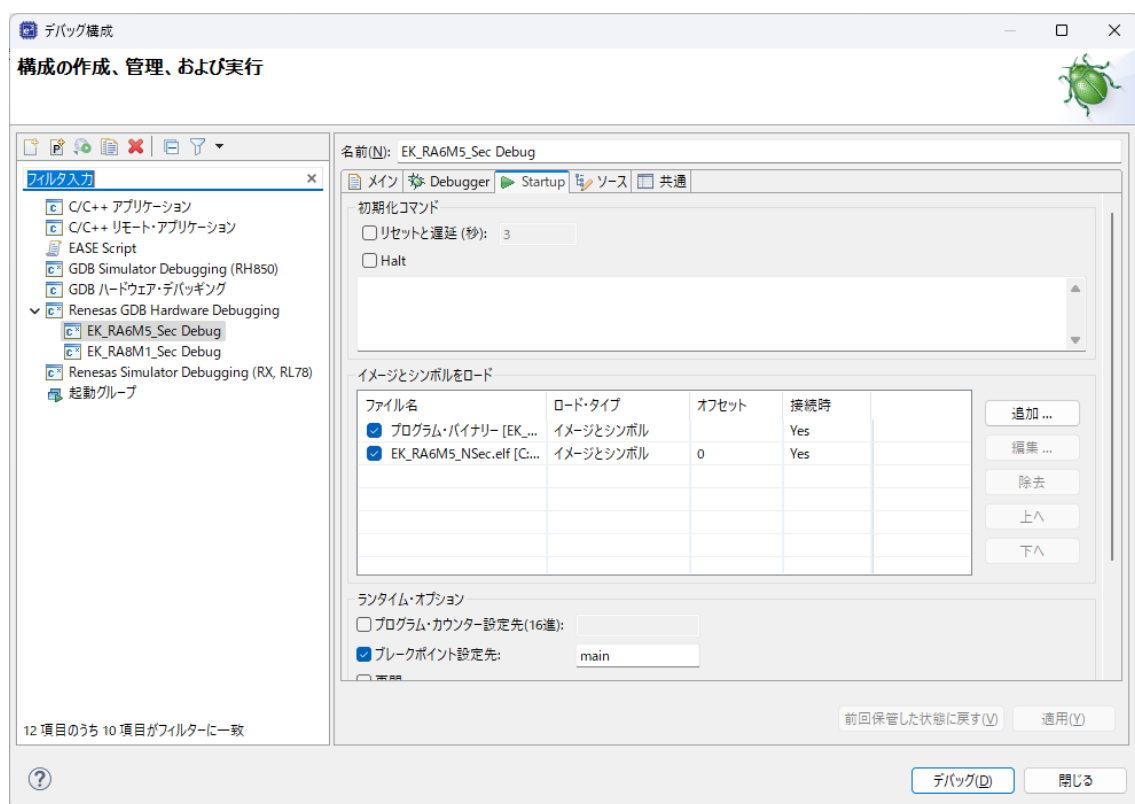
- ユーザ LED1 を 500msec 間隔で 50msec 点灯
- ユーザスイッチ 1 の押下でメッセージを表示し、LED2 を 1000msec 点灯
- ユーザスイッチ 2 の押下でメッセージを表示し、LED3 を 2000msec 点灯

第3章 サンプルプログラムの起動

EK-RA6M5 ボード用サンプルプログラムの起動について説明します。また、使用されるサンプルプロジェクトは、インポートを終えてください。

3-1. サンプルプログラムのデバッグ

e2studio を起動し、メニュー「実行」→「デバッグの構成」より「構成の作成、管理、および実行」画面を表示し、「Renesas GDB Hardware Debugging」→「EK_RA6M5_Sec Debug」を選択し「Startup」タブをクリックします。



初期設定では、セキュアのelfファイルと非セキュアのelfファイルをロードするようになっており、この状態で「デバッグ」をクリックすると実行イメージをロードしデバッガが起動します。「ブレークポイント設定先」で `main` を指定している場合、デバッガ起動後の最初の「再開」クリックで `main0`関数でブレークし、再度の「再開」クリックで続けてプログラムは実行します。

この `main0`関数は、セキュア層と非セキュア層とに存在しますが、処理順序でセキュア層の `main0`関数でブレークします。ただし、セキュア層のelfファイルのロードタイプに「イメージのみ」を選択すると非セキュア層の `main0`関数でブレークします。また、タイプ1評価版にはセキュア層のソースコードが存在しないため、非セキュア層の `main0`関数でブ

レークします。

タイプ 1 評価版では、セキュア層のソースコードを含まないため、セキュア層の elf ファイルは変更されることがありません。そのため、他のプログラムをロードした後の場合のみ、セキュア層の elf ファイルをロードすれば良く、以降はファイル名の左のチェックボックスを外しロードをスキップすることができます。

(余白)

(余白)

Cortex-M 対応 MULCOS-MK

EK-RA6M5 サンプルガイド

2025 年 4 月 第 1 版発行

株式会社スパイラルテック

Mail tech-info@spiral-tech.co.jp