RX ファミリ

Renesas FreeRTOS

R01AN4307JJ0104
Rev.1.04
2020.08.31

要旨

このアプリケーション・ノートは、FreeRTOS をベースとする Renesas FreeRTOS モジュールについて説明します。

対象デバイス

- RX72M Group
- RX72N Group
- RX72T Group
- RX71M Group
- RX66T Group
- RX66N Group
- RX65N, RX651 Group
- RX64M Group
- RX23W Group
- RX231 Group
- RX230 Group
- RX130 Group

関連ドキュメント

- RX ファミリに関する『ボードサポートパッケージモジュール Firmware Integration Technology (R01AN1685)』
- 『RX ファミリ CMT モジュール Firmware. Integration Technology (R01AN1856)』
- 『Renesas e^2 studio スマート・コンフィグレータユーザーガイド (R20AN0451)』

参考資料

- FreeRTOS customization: https://www.freertos.org/a00110.html [1]
- FreeRTOS Memory Management: https://www.freertos.org/a00111.html [2]

目次

1. 概要

1.1 Renesas FreeRTOS モジュール

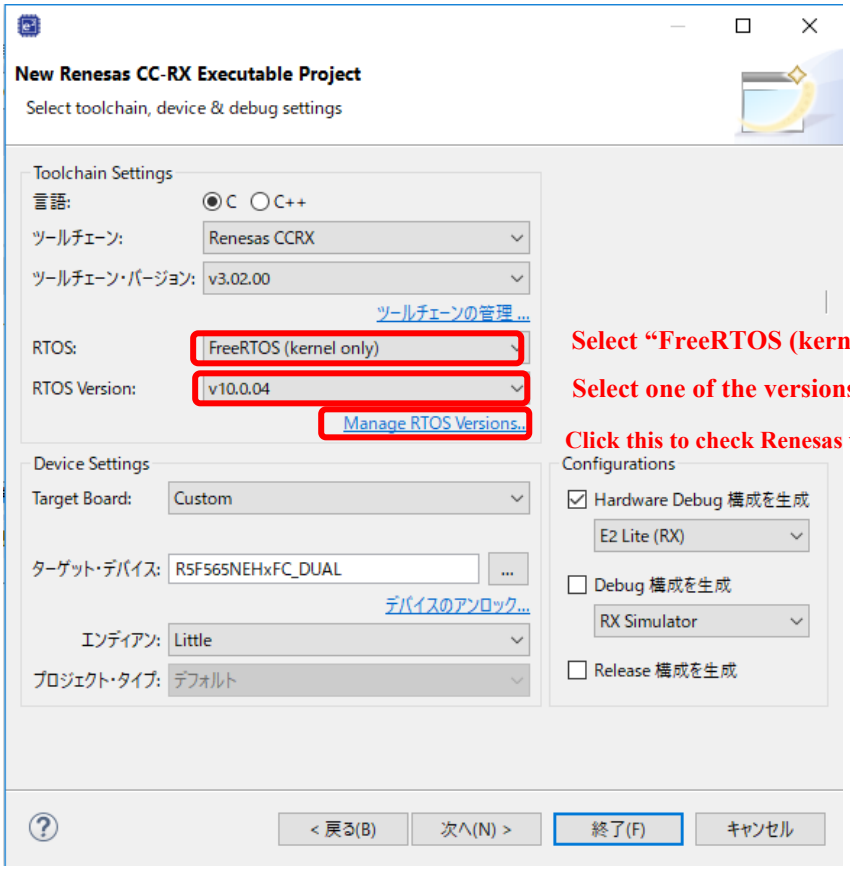
Renesas FreeRTOS モジュールは、FreeRTOS カーネルの RX 対応バージョンです。

1.2 Renesas FreeRTOS を使用した RTOS プロジェクトの作成

1.2.1 CCRX プロジェクトの作成

RTOS プロジェクトの作成をサポートする e2studio を使用して、Renesas FreeRTOS プロジェクトを作成できます。プロジェクトの作成を開始する時点で、ユーザは FreeRTOS パッケージのバージョンを選択します。また、選択したバージョンが、プロジェクトに自動的にインポートされます。この方法はユーザにとって使いやすく、FreeRTOS の設定やアプリケーション・コードの作成に集中することができます。

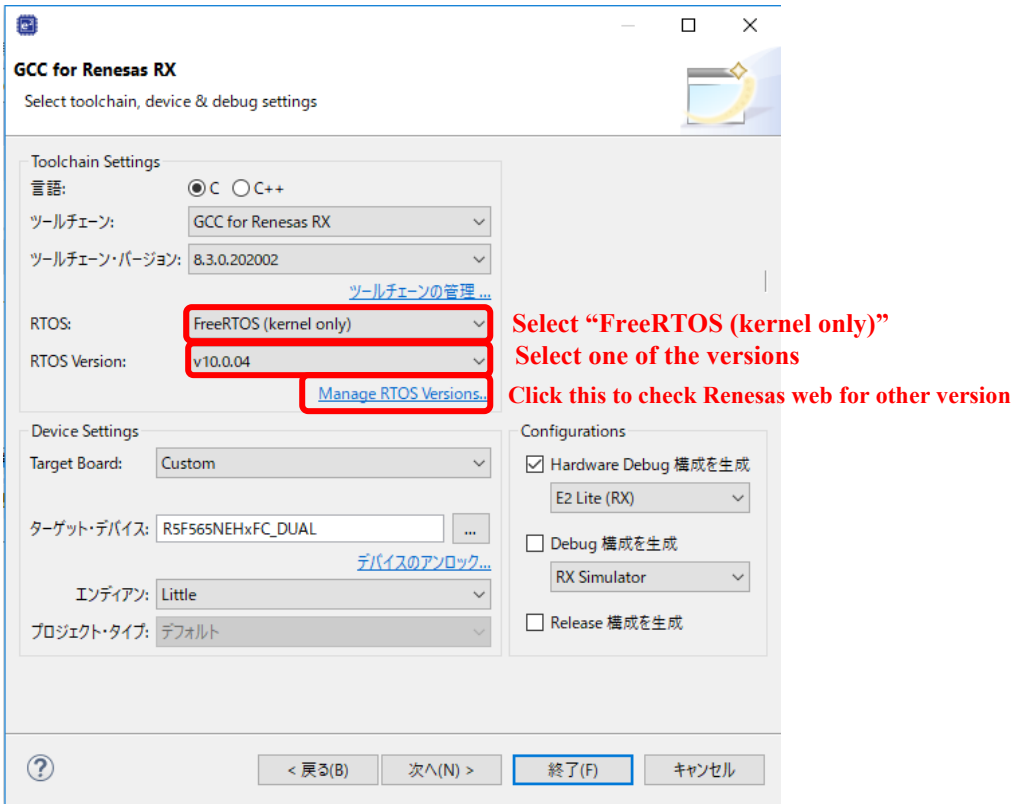
次の図に、プロジェクト作成に使用する RTOS を選択する方法を示します。



1.2.2 GCC プロジェクトの作成

CCRX 同様に GCC を使用したプロジェクトの生成もサポートしています。はじめてプロジェクト生成をするときに、Renesas FreeRTOS のパッケージバージョンを選択し、インポートします。

GCC プロジェクト生成方法は以下を参照してください。



1.3 BSP 設定とタイマのソース

1.3.1 BSP 設定

セクション 1.3 で説明する方法を使用する場合、以下の BSP の設定が自動的に実行されます。

- BSP_CFG_RTOS_USED が「1」に設定されます
- BSP_CFG_RTOS_SYSTEM_TIMER は「0」に設定されます (これは CMT チャネル 0 を意味します)

デフォルトでは、CMT チャネル 0 をシステム・タイマとして使用するように RTOS カーネルが設定されます。e2studio の将来のバージョンでは、ユーザは使用可能な CMT チャネルのいずれかを選択できるようになる予定です。

1.3.2 CMT モジュールの使用方法

Renesas FreeRTOS カーネルでは、選択した CMT チャネルの排他的使用が必要です。ユーザが、同じプロジェクト内でカーネル以外の目的で CMT モジュールを使用することを希望する場合、カーネルが使用している CMT チャネルの動作に干渉しないように注意を払う必要があります。したがって、このような状況では、RTOS をサポートする CMT FIT モジュールを使用することを推奨します。現時点で、CMT FIT をサポートしている RTOS のバージョンは、v3.30 です。

2. 要件

この FIT モジュールは、以下の条件下で動作するように設定済みです。

2.1 ハードウェア要件

使用する MCU は、以下の機能をサポートしている必要があります。

- CMT

2.2 ソフトウェア要件

この Renesas FreeRTOS モジュールは、以下の FIT モジュールに依存しています。

- Renesas ボード・サポート・パッケージ (r_bsp)

2.3 サポートするツールチェーン

このドライバは、「5.1 動作確認環境」に一覧を示すツールチェーンとの組み合わせで動作することを確認済みです。

3. 設定の概要

3.1 ヒープ・メモリの管理

3.1.1 heap_1.c

これは、すべての中で最も簡単な実装です。この関数は、一度割り当てたメモリの解放を許可しません。このような特性はありますが、heap_1.c は多数の組み込みアプリケーションに適しています。奥深くに組み込まれた小型アプリケーションの多くは、必要なタスク、キュー、セマフォなどのすべてをシステム・ブート時に作成し、その後、(アプリケーションが再度オフに切り替えられるか、再起動されるまで) プログラムの寿命に達するまでこれらのオブジェクトすべてを使用するからです。つまり、どのオブジェクトも削除されません。

詳細については、[2] を参照してください。

3.1.2 heap_2.c

この方式は、最適近似アルゴリズムを使用し、方式 1 とは異なり、既に割り当てたブロックの解放を許可します。この方式は、隣接している複数のブロックを結合して単一の大きいブロックにすることはありません。複数の空きブロックの結合を実行する実装については、heap_4.c を参照してください。使用可能なヒープ領域の合計量は、FreeRTOSConfig.h 内で定義する configTOTAL_HEAP_SIZE を使用して設定します。FreeRTOSConfig.h 内の configAPPLICATION_ALLOCATED_HEAP 設定定数は、メモリ内の特定のアドレスにヒープを配置できるようにする目的で用意してあります。

詳細については、[2] を参照してください。

3.1.3 heap_3.c

この実装は、C の標準ライブラリ関数である malloc() と free() に対するシンプルなラッパーであり、ほとんどの場合、ユーザが選択したコンパイラにこれらの関数が付属しています。このラッパーは単純に、malloc() と free() の各関数をスレッドセーフにします。

詳細については、[2] を参照してください。

3.1.4 heap_4.c

この方式は近似アルゴリズムを使用しますが、方式 2 とは異なり、隣接している複数の空きメモリ・ブロックを結合して単一の大きいブロックにします (結合アルゴリズムを内包しています)。
使用可能なヒープ領域の合計量は、FreeRTOSConfig.h 内で定義する configTOTAL_HEAP_SIZE を使用して設定します。FreeRTOSConfig.h 内の configAPPLICATION_ALLOCATED_HEAP 設定定数は、メモリ内の特定のアドレスにヒープを配置できるようにする目的で用意してあります。

詳細については、[2] を参照してください。

3.2 カスタム設定

Renesas FreeRTOS をカスタマイズするために、FreeRTOSConfig.h という設定ファイルを使用します。Renesas FreeRTOS のどのアプリケーションも、プリプロセッサ・インクルード・パス内に FreeRTOSConfig.h ヘッダー・ファイルを含める必要があります。FreeRTOSConfig.h は、ビルドするアプリケーションに合わせて、RTOS カーネルをカスタマイズします。したがって、このファイルは RTOS 固有ではなくアプリケーション固有であり、RTOS カーネルのソース・コード・ディレクトリ内ではなくアプリケーション・ディレクトリ内に配置する必要があります。

以下に、典型的な FreeRTOSConfig.h 定義のいくつかを示します。いくつかのマクロ設定が BSP 内のパラメータを参照する必要があることに注意してください。誤った値を設定した場合、RTOS カーネルは期待どおりに動作しなくなります。

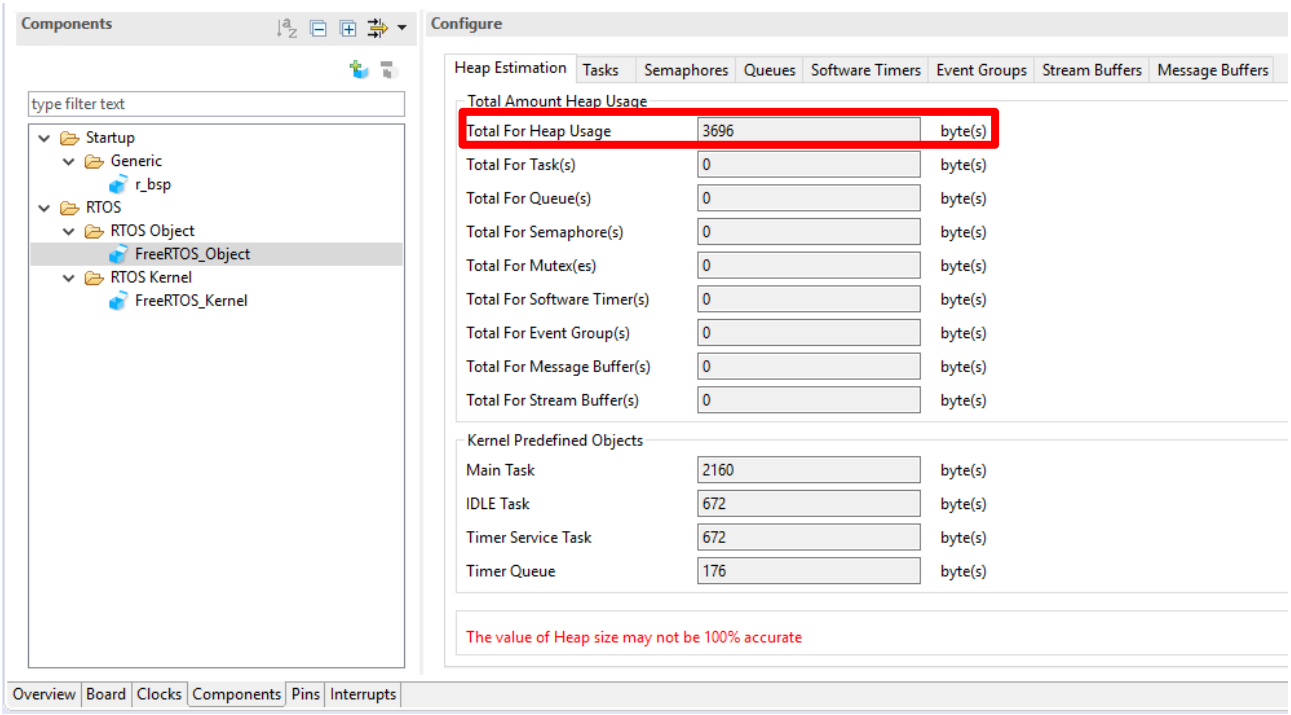
FreeRTOSConfig.h 内の設定オプション	
configUSE_PREEMPTION	[1] を参照してください。
configUSE_IDLE_HOOK	[1] を参照してください。
configUSE_TICK_HOOK	[1] を参照してください。
configCPU_CLOCK_HZ	このマクロは、MCU のシステム・クロック速度を Hz 単位で定義します。タイマ・ペリフェラルを正しく設定するには、このマクロが必須です。このマクロは、BSP の mcu_info.h 内で定義されている **BSP_ICLK_HZ** に設定する必要があります。
configPERIPHERAL_CLOCK_HZ	CMT が使用するペリフェラル・モジュール・クロックの周波数。RX の場合、クロックの周波数は PCLKB になります。したがって、このマクロは、mcu_info.h 内で定義されている **BSP_PCLKB_HZ** に設定する必要があります。
configTICK_RATE_HZ	RTOS カーネルのティック割り込みの周波数。ティック割り込みは、時間測定に使用します。したがって、より高いティック周波数を使用すると、より高い分解能で時間を測定できます。ただし、高いティック周波数を使用すると、RTOS カーネルがより多くの CPU 時間を使用するようになるので、効率は低下します。通常、この値は 1,000 です。
configMINIMAL_STACK_SIZE	[1] を参照してください。
configTOTAL_HEAP_SIZE_N	3.3 を参照してください。
configMAX_TASK_NAME_LEN	[1] を参照してください。
configUSE_TRACE_FACILITY	[1] を参照してください。
configUSE_16_BIT_TICKS	[1] を参照してください。
configIDLE_SHOULD_YIELD	[1] を参照してください。
configUSE_CO_ROUTINES	[1] を参照してください。
configUSE_MUTEXES	[1] を参照してください。
configGENERATE_RUN_TIME_STATS	[1] を参照してください。
configCHECK_FOR_STACK_OVERFLOW	[1] を参照してください。
configUSE_RECURSIVE_MUTEXES	[1] を参照してください。
configQUEUE_REGISTRY_SIZE	[1] を参照してください。
configUSE_MALLOC_FAILED_HOOK	[1] を参照してください。
configUSE_APPLICATION_TASK_TAG	[1] を参照してください。
configUSE_QUEUE_SETS	[1] を参照してください。
configUSE_COUNTING_SEMAPHORES	[1] を参照してください。
configMAX_PRIORITIES	[1] を参照してください。
configMAX_CO_ROUTINE_PRIORITIES	[1] を参照してください。
configUSE_TIMERS	[1] を参照してください。
configTIMER_TASK_PRIORITY	[1] を参照してください。
configTIMER_QUEUE_LENGTH	[1] を参照してください。
configTIMER_TASK_STACK_DEPTH	[1] を参照してください。
configKERNEL_INTERRUPT_PRIORITY	[1] を参照してください。
configMAX_SYSCALL_INTERRUPT_PRIORITY	[1] を参照してください。
configTICK_VECTOR	このマクロは、CMT チャネルが使用するベクタの番号と同じ値に設定する必要があります。使用可能な値は、以下のとおりです。 _CMT0_CMI0: CMT0 を使用する場合。 _CMT1_CMI1: CMT1 を使用する場合。 _CMT2_CMI2: CMT2 を使用する場合。 _CMT3_CMI3: CMT3 を使用する場合。
configUSE_TASK_NOTIFICATIONS	[1] を参照してください。
configRECORD_STACK_HIGH_ADDRESS	[1] を参照してください。

configNUM_THREAD_LOCAL_STORAGE_POINTERS	[1] を参照してください。
configSUPPORT_DYNAMIC_ALLOCATION	[1] を参照してください。
configSUPPORT_STATIC_ALLOCATION	[1] を参照してください。

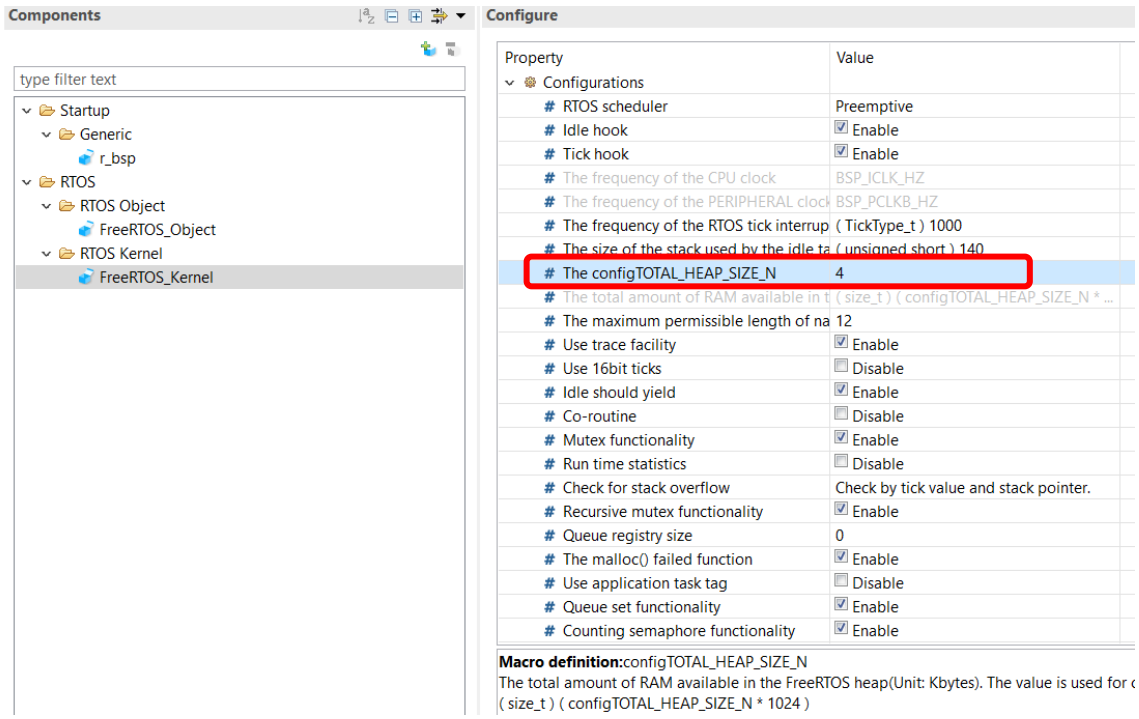
3.3 ヒープ使用量予測機能

スマート・コンフィグレータの FreeRTOS_Object 内の Heap Estimation タブで、ヒープの使用量を予測する機能をサポートしました。ヒープ領域で使用可能なメモリサイズは(configTOTAL_HEAP_SIZE)はタスクやキューなどの FreeRTOS オブジェクトで使用するメモリサイズで決定します。ユーザが FreeRTOS オブジェクトの追加変更を行うと使用するヒープサイズも変わります。Heap Estimation タブでは使用するヒープ領域のサイズを見積もることができ、ヒープ領域の節約が可能になります。

以下はスマート・コンフィグレータの Heap Estimation タブです。

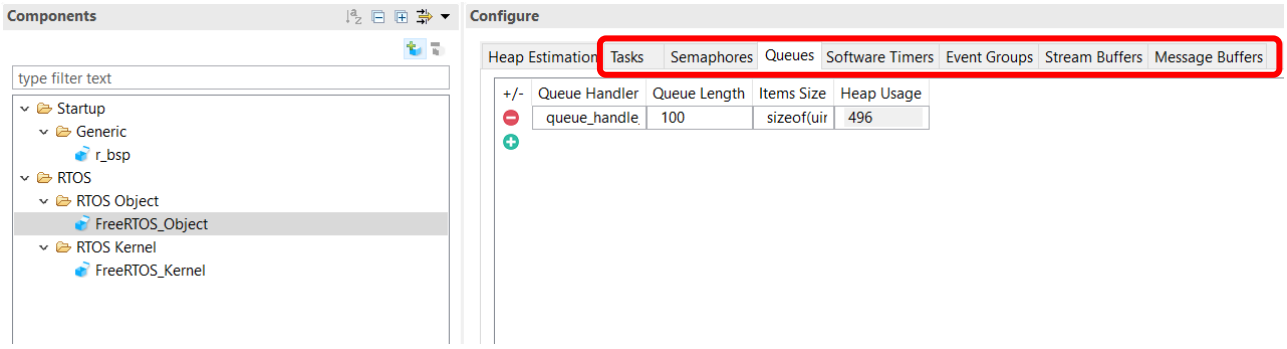


FreeRTOS_Kernal の“The configTOTAL_HEAP_SIZE_N” は configTOTAL_HEAP_SIZE_N*1024 Byte のヒープサイズを確保します。
Heap Estimation タブで見積もったヒープサイズを KByte で切り上げた数値を入力してください。



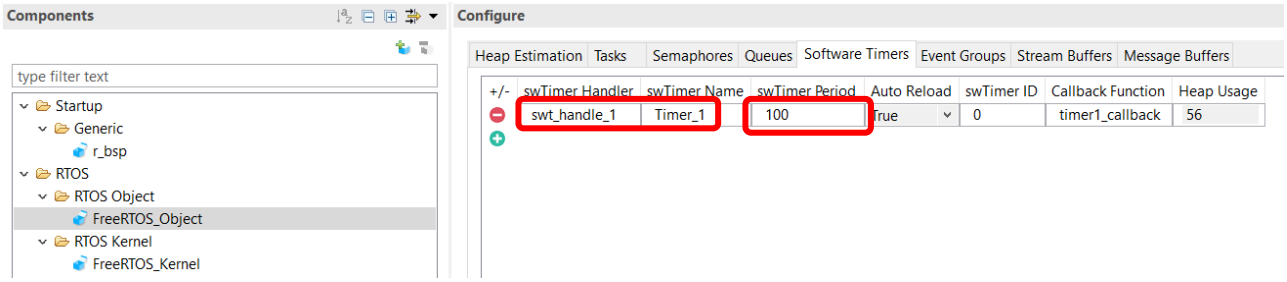
3.4 オブジェクトの追加と削除

スマート・コンフィグレータの FreeRTOS_Object コンポーネントを使用することで、FreeRTOS Object を追加、削除することが可能です。



変更するオブジェクトを選択し、追加の場合は ⊕、削除の場合は ⊖ ボタンを押します。

下図のように各パラメータをデフォルト値からシステムに合わせて変更してください。



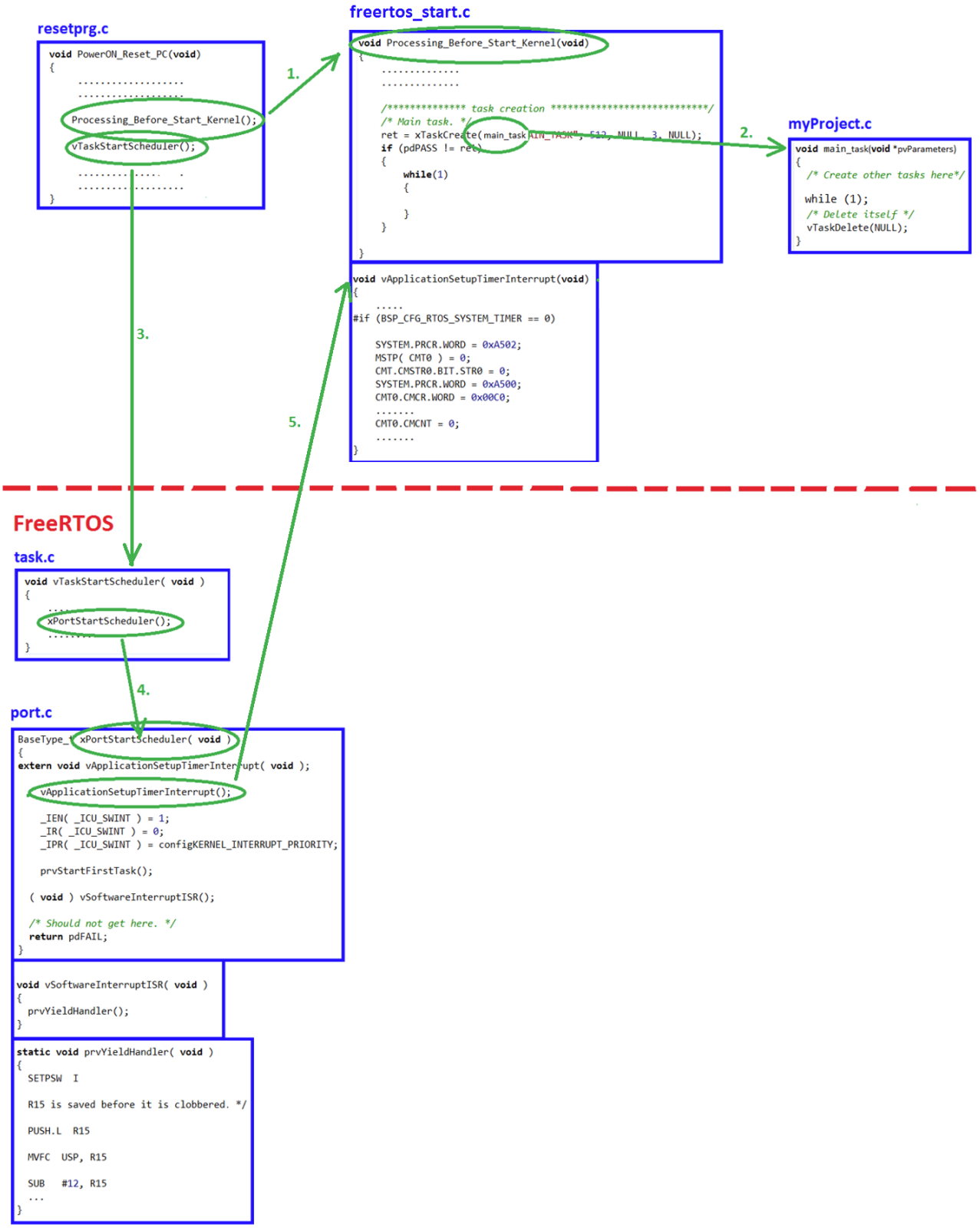
生成したオブジェクトは数にある freerots_start.c に生成されます

使用するヒープサイズはオブジェクトのコード生成後に“Total Amount Heap Usage”に表示されます。使用可能なヒープ領域の合計を超えるオブジェクトのヒープの使用は、ビルドエラーが発生します。

より詳細な使い方は e2studio のヘルプの以下の章を参照して下さい。
e2studio ユーザガイド>ビルドに関する機能>スマート・コンフィグレータ>
RTOS FreeRTOS コンフィグレーション　4.2.4 Heap 使用量予測機能

4. ユーザスタートコード

次の図に、Renesas FreeRTOS プロジェクトのプログラムの流れを示します。複数の BSP ルーチンを開始し、FreeRTOS カーネルを起動した後、ユーザのアプリケーション・コードを実行します。



4.1 freertos_start.c, freertos_start.h

RTOS カーネルを起動する前に、ユーザのスタートアップ関数と CMT 設定ルーチンを実行することができます。これらの関数は、以下のような形式で提供されています。

4.1.1 RTOS システム・タイマの初期化 - void vApplicationSetupTimerInterrupt(void)

提供されているこの関数は、RTOS カーネルのシステム・ティックとして使用する目的で、選択した CMT チャネル (#define BSP_CFG_RTOS_SYSTEM_TIMER) を設定します。ユーザはなにも変更を加えずに、この関数を簡単に使用することができます。

4.1.2 アイドル・フック関数 - void vApplicationIdleHook(void)

RTOS スケジューラを開始するときにアイドル・タスクが自動的に作成され、実行可能なタスクが常に少なくとも 1 つ確実に存在するようになります。このタスクは使用可能な最小の優先順位で作成され、より高い優先順位のアプリケーション・タスクが実行可能な状態になっているときは、このアイドル・タスクは確実に CPU 時間を消費しません。このアイドル・タスクはオプションで、アプリケーションが定義したフック関数を呼び出すことができます。この関数は、アイドル・フック関数です。このアイドル・タスクは非常に低い優先順位で動作するので、アイドル・フック関数が実行されるのは、それより高い優先順位でなおかつ実行可能なタスクが存在していない場合のみです。

アイドル・フック関数が呼び出されるのは、FreeRTOSConfig.h 内で configUSE_IDLE_HOOK が 1 に設定されている場合のみです。

4.1.3 ティック・フック関数 - void vApplicationTickHook(void)

このティック割り込みはオプションで、アプリケーションが定義したフック関数を呼び出すことができます。この関数は、ティック・フック関数です。ティック・フックは、タイマ機能を実装するうえで便利な場所を提供します。

ティック・フックが呼び出されるのは、FreeRTOSConfig.h 内で configUSE_TICK_HOOK が 1 に設定されている場合のみです。

4.1.4 Malloc が失敗した場合のフック関数 - void vApplicationMallocFailedHook(void)

heap_1.c、heap_2.c、heap_3.c、heap_4.c、heap_5.c の形で実装すメモリ割り当て方式はオプションで、malloc() が失敗した場合のフック関数を定義することもできます。pvPortMalloc() が NULL を返した場合は必ずこのフック関数を呼び出すように設定できます。

malloc() が失敗した場合のフックを定義すると、ヒープ・メモリの不足が原因で発生した問題を識別しやすくなります。特に、API 関数内で pvPortMalloc() の呼び出しに失敗した場合です。

malloc が失敗した場合のフックが呼び出されるのは、FreeRTOSConfig.h 内で configUSE_MALLOC_FAILED_HOOK が 1 に設定されている場合のみです。

4.1.5 void Processing_Before_Start_Kernel(void)

この関数内で、メイン・タスク「main_task()」が作成されます。必要な場合、ユーザは FreeRTOS の各種オブジェクト (メールボックス、セマフォ、ミューテックス) を作成することもできます。

4.2 void main_task(void *pvParameters)

このタスクは、Processing_Before_Start_Kernel() 内で作成されます。ユーザは、他のすべてのアプリケーション・タスクもこの関数内で作成する必要があります。

5. 付録

5.1 動作確認環境

このセクションでは、Renesas FreeRTOS モジュールの動作確認環境について説明します。

表 5.1 動作確認環境

項目	内容
統合開発環境（IDE）	ルネサスエレクトロニクス製 e² studio バージョン 7.8
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler Package for RX Family V.3.02 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
	GCC for Renesas RX 4.8.4.201902 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99 リンカオプション：統合開発環境のデフォルト設定に以下のオプションを追加 最適化で-Osを使用する場合以下の設定を追加 -Wl,--no-gc-sections これはGCCリンカがFITモジュールで宣言された割り込み関数を誤って破棄しないための対策です。
エンディアン	ビッグエンディアン／リトルエンディアン
モジュールのリビジョン	Renesas FreeRTOS バージョン 10.0.03
使用ボード	Renesas Starter Kit+ for RX72N (product No.: RTK5572NNxSxxxxBE) Target Board for RX23W (product No.: RTK5RX23W0CxxxxxBJ) Renesas Starter Kit+ for RX130 512KB (product No.: RTK5051308CxxxxxBR) Renesas Starter Kit+ for RX72M (product No.: RTK5572MNDCxxxxxBJ) Renesas Starter Kit+ for RX72T (product No.: RTK5572TKCCxxxxxSE) Renesas Starter Kit+ for RX71M (product No.: R0K50571MSxxxBE) Renesas Starter Kit+ for RX66T (product No.: RTK50566T0CxxxxxBE) Renesas Starter Kit+ for RX65N-2M (product No.: RTK50565N2SxxxxxBE) Renesas Starter Kit+ for RX64M (product No.: R0K50564MSxxxBE) Renesas Starter Kit+ for RX231 (product No.: R0K505231CxxxBE)

5.2 注意事項

5.2.1 マクロ configTOTAL_HEAP_SIZE_N について

- configTOTAL_HEAP_SIZE_N は KByte サイズの値です。初期値は 4 に設定しています。
  Heap サイズの初期値は 4KB になります。

5.2.2 本アプリケーションノート対応 e²studio について

本アプリケーションノートは、e²studio V.7.8 もしくは e²stduio 2020-04 以降のバージョンで使用可能です。

ホームページとサポート窓口

ルネサス エレクトロニクス Web サイト

http://www.renesas.com/

お問い合わせ先

http://www.renesas.com/contact/

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2019.01.21	−	初版
1.01	2019.10.08	-	サポートデバイス追加 RX130グループ、RX230グループ、RX231グループ、 RX66Tグループ、RX72Tグループ、RX72Mグループ
1.02	2020.01.30	-	サポートデバイス追加 RX23Wグループ
1.03	2020.04.30	-	サポートデバイス追加 RX72Nグループ、RX66Nグループ
		-	config_TOTAL_HEAP_SIZEを config_TOTAL_HEAP_SIZE_Nに変更
		P.5	1.2 GCCプロジェクトの生成　追加
1.04	2020.08.31	P.16	5.1 動作確認環境　更新
		P.17	5.2.2 本アプリケーション対応e^{2}studioについて　追加

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、VIL（Max.）から VIH（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、VIL（Max.）から VIH（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。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、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。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)

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

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/

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。