

RL78/G14, RL78/G23

FreeRTOS Blinky デモ

要旨

本アプリケーションノートでは、FreeRTOS を使用して、ルネサス製マイコンボードに搭載された LED を点滅させるデモプログラムについて説明します。

動作確認デバイス

RL78/G14, RL78/G23

目次

1. 概要	4
1.1 FreeRTOS Blinky デモの概要	4
1.2 ソフトウェアスタック図	5
2. 動作確認条件	6
2.1 機材一覧	7
2.1.1 RL78/G14	7
2.1.2 RL78/G23	7
3. ハードウェア説明	8
3.1 RL78/G14 FPB	8
3.1.1 使用端子一覧	8
3.1.2 周辺機能一覧	8
3.2 RL78/G23-128p FPB	8
3.2.1 使用端子一覧	8
3.2.2 周辺機能一覧	8
4. ソフトウェアの説明	9
4.1 FreeRTOS Blinky デモ	9
4.1.1 タスク一覧	9
4.1.2 システム起動からタスク実行までの流れ	10
4.1.3 FreeRTOS Blinky デモの構成	11
4.1.4 FreeRTOS カーネルのコンフィグ設定	12
4.1.5 オプション・バイト設定一覧	14
4.2 フォルダ構成	14
4.3 コードサイズ	15
4.4 システム起動から最初のタスク起動までに使用するスタックサイズ	15
5. FreeRTOS Blinky デモのセットアップ	16
5.1 RL78/G14 FPB	16
5.1.1 プロジェクトをインポート	16
5.1.2 プロジェクトをビルド	16
5.2 RL78/G23-128p FPB	17
5.2.1 プロジェクトをインポート	17
5.2.2 プロジェクトをビルド	17
6. FreeRTOS Blinky デモの実行	18
6.1 RL78/G14 FPB	18
6.1.1 マイコンボードへ電源を供給	18
6.1.2 デバッグコンフィグレーション	18
6.1.3 デバッグ設定	18
6.1.4 FreeRTOS Blinky デモの実行	20
6.2 RL78/G23-128p FPB	21
6.2.1 COM port を使用する場合	21
6.2.1.1 ジャンパピン設定	21
6.2.1.2 マイコンボードへ電源を供給	21

6.2.1.3	デバッグコンフィグレーション	21
6.2.1.4	デバッグ設定	22
6.2.2	エミュレータを使用する場合	23
6.2.2.1	コネクタ実装、ジャンピン設定、パターンカット	23
6.2.2.2	エミュレータをマイコンボードに接続	24
6.2.2.3	デバッグコンフィグレーション	24
6.2.2.4	デバッグ設定	25
6.2.3	FreeRTOS Blinky デモの実行	26
7.	補足情報	27
7.1	ルネサス関数一覧	27
7.2	注意事項	27
8.	ウェブサイトおよびサポート	28
	改訂記録	29

注：

AWS™ および FreeRTOS は Amazon.com, Inc. or its affiliates の商標です。

(<https://aws.amazon.com/trademark-guidelines/>)

GitHub® は GitHub, Inc. のトレードマークです。(<https://github.com/logos>)

1. 概要

本アプリケーションノートでは、FreeRTOS と、RL78 ファミリのマイコンボード RL78/G14 Fast Prototyping Board または RL78/G23-128p Fast Prototyping Board を用いて、ボードに搭載されている LED を点滅制御するデモプログラム（以降、FreeRTOS Blinky デモ）の実行方法を説明します。

FreeRTOS は、マイクロコントローラ向けのオープンソースのリアルタイムオペレーティングシステム (RTOS) です。FreeRTOS の詳しい情報は [FreeRTOS documentation - FreeRTOS™](#) を参照してください。

1.1 FreeRTOS Blinky デモの概要

FreeRTOS Blinky デモは FreeRTOS 上でユーザタスクを 2 つ生成し、ボード上の LED を点滅制御します。詳細は「4.1.3 章 FreeRTOS Blinky デモの構成」を参照してください。

1.2 ソフトウェアスタック図

RL78/G14 のソフトウェアスタック図を以下に示します。

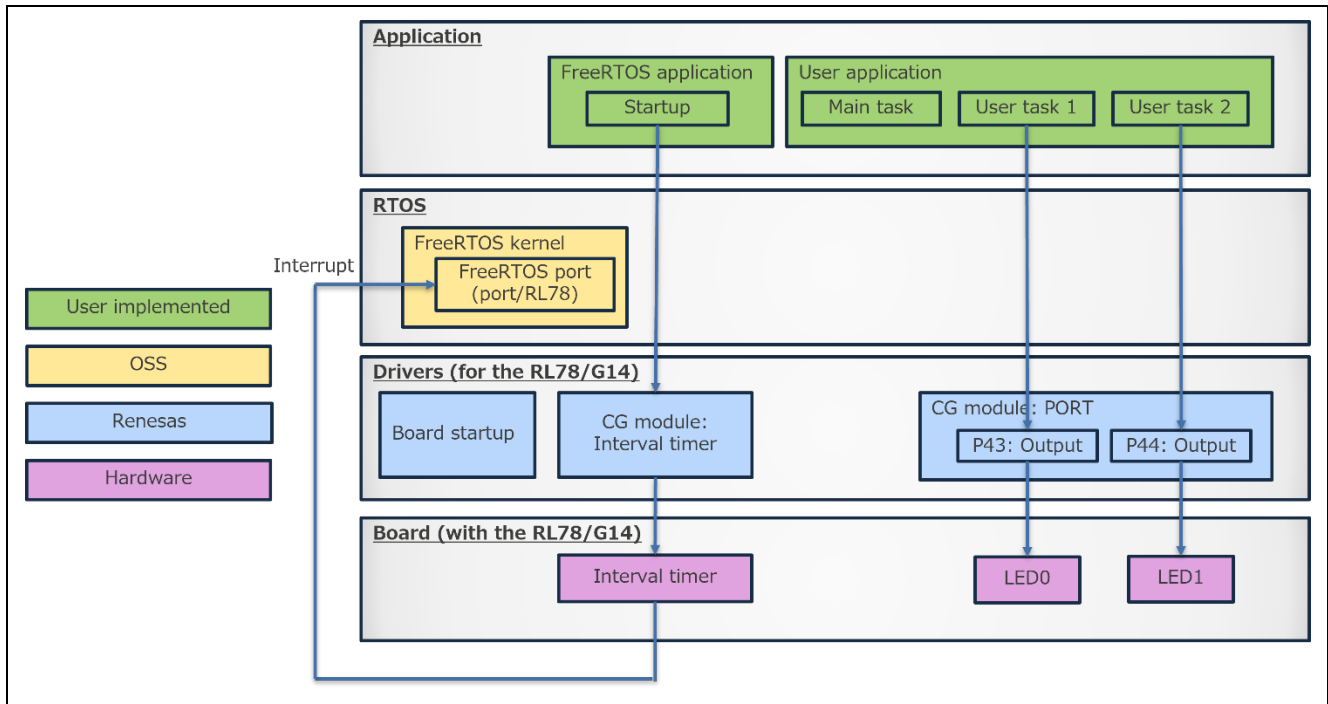


図 1-1 ソフトウェアスタック図 (RL78/G14)

RL78/G23 のソフトウェアスタック図を以下に示します。

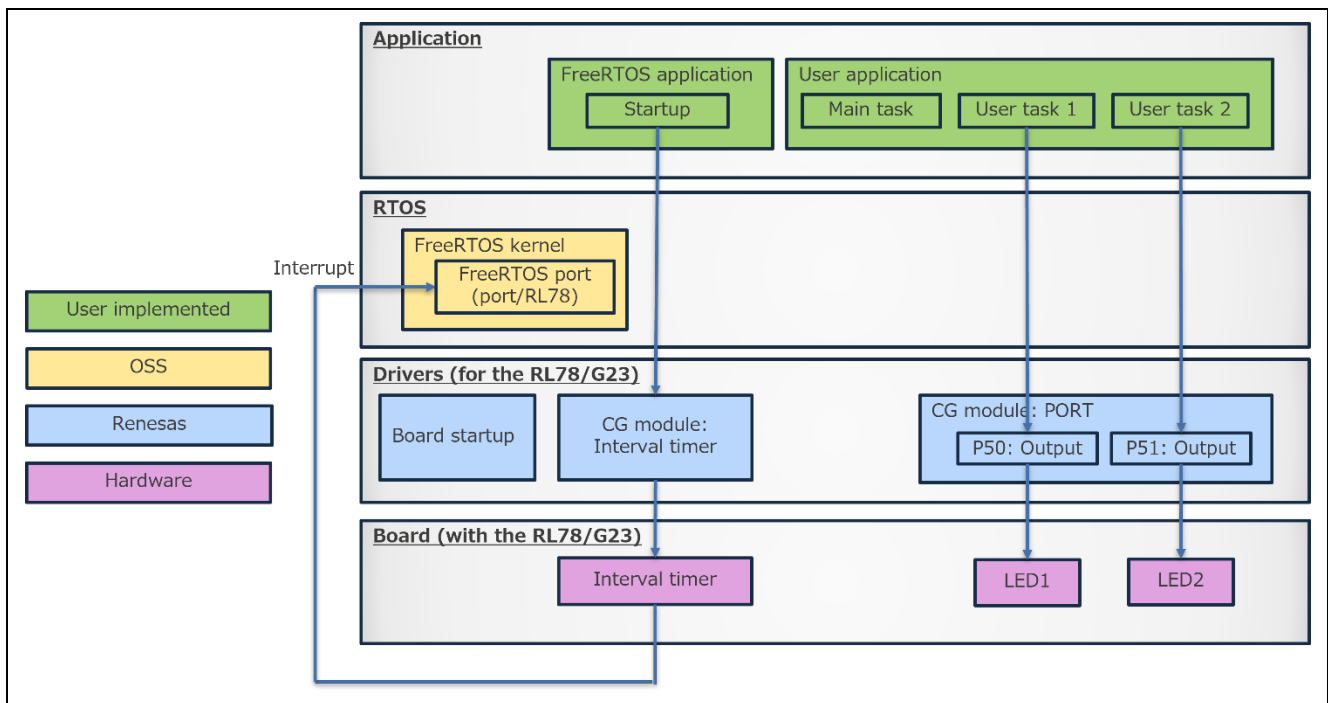


図 1-2 ソフトウェアスタック図 (RL78/G23)

2. 動作確認条件

デモプロジェクトは下記の条件で確認しています。

表 2-1 動作確認条件 (RL78/G14)

項目	内容
使用マイコン	RL78/G14 (R5F104MLA Code Flash 512KB)
使用ボード	RL78/G14 Fast Prototyping Board (RTK5RLG140C00000BJ)
動作周波数	高速オンチップ・オシレータ・クロック: 32MHz
動作電圧	3.3V
統合開発環境	ルネサスエレクトロニクス製 e ² studio 2025-07
C コンパイラ	ルネサスエレクトロニクス製 CC-RL V1.15.01
コードジェネレータ (CG)	Renesas e2 studio Code Generator for RL78 25.7.0.v20250612-0710

表 2-2 動作確認条件 (RL78/G23)

項目	内容
使用マイコン	RL78/G23 (R7F100GSN Code Flash 768KB)
使用ボード	RL78/G23-128p Fast Prototyping Board (RTK7RLG230CSN000BJ)
動作周波数	高速オンチップ・オシレータ・クロック: 32MHz
動作電圧	5V
統合開発環境	ルネサスエレクトロニクス製 e ² studio 2025-07
C コンパイラ	ルネサスエレクトロニクス製 CC-RL V1.15.01
スマート・コンフィグレータ (SC)	Renesas Smart Configurator for RL78 25.7.0.v20250620-0609
ボードサポートパッケージ (BSP)	v1.91 (r_bsp)

2.1 機材一覧

2.1.1 RL78/G14

デモプロジェクトに必要な機材の一覧を示します。

表 2-3 機材一覧 (RL78/G14)

項目	内容
マイコンボード	RL78/G14 Fast Prototyping Board (以降、RL78/G14 FPB) https://www.renesas.com/rtk5rlg140c00000bj
Micro-B USB ケーブル x1	マイコンボードと PC を接続

2.1.2 RL78/G23

デモプロジェクトに必要な機材の一覧を示します。

表 2-4 機材一覧 (RL78/G23)

項目	内容
マイコンボード	RL78/G23-128p Fast Prototyping Board (以降、RL78/G23-128p FPB) https://www.renesas.com/rtk7rlg230csn000bj
Micro-B USB ケーブル x1	マイコンボードと PC を接続
ジャンパピン x3	マイコンボードの電源選択用 (J15, J16, J19)

本デモプロジェクトは COM port デバッグを使用しますが、エミュレータを用いてデバッグをすることも可能です。

表 2-5 デバッグ機材の一覧

項目	内容
エミュレータ	E2 エミュレータ Lite https://www.renesas.com/software-tool/e2-emulator-lite-rte0t0002lkce00000r

注意事項

エミュレータを使用する場合、エミュレータ接続用コネクタの実装と回路変更が必要です。詳細は「6.2.2.1 章 コネクタ実装、ジャンパピン設定、パターンカット」を参照してください。

3. ハードウェア説明

3.1 RL78/G14 FPB

3.1.1 使用端子一覧

FreeRTOS Blinky デモで使用する端子と機能を示します。

表 3-1 使用端子と機能 (RL78/G14)

端子名	入出力	機能
P43	出力	LED0
P44	出力	LED1

3.1.2 周辺機能一覧

FreeRTOS Blinky デモで使用する周辺機能と用途を示します。

表 3-2 周辺機能と用途 (RL78/G14)

周辺機能	用途
12bit インターバルタイマ	FreeRTOS のシステムタイマ

3.2 RL78/G23-128p FPB

3.2.1 使用端子一覧

FreeRTOS Blinky デモで使用する端子と機能を示します。

表 3-3 使用端子と機能 (RL78/G23)

端子名	入出力	機能
P50	出力	LED1
P51	出力	LED2

3.2.2 周辺機能一覧

FreeRTOS Blinky デモで使用する周辺機能と用途を示します。

表 3-4 周辺機能と用途 (RL78/G23)

周辺機能	用途
32bit インターバルタイマ	FreeRTOS のシステムタイマ

4. ソフトウェアの説明

4.1 FreeRTOS Blinky デモ

FreeRTOS Blinky デモは FreeRTOS 上の 2 つのユーザタスクから、LED を点滅制御します。

4.1.1 タスク一覧

FreeRTOS Blinky デモのタスク一覧を示します。

表 4-1 タスク一覧 (RL78/G14 FPB)

Task Entry Function	Task Name	Stack Depth (Byte)	Priority	Memory Allocation	Description
prvIdleTask	IDLE	128 (256)	0	Static	Idle タスク (FreeRTOS idle task - FreeRTOS™)
prvTimerTask	Tmr Svc	128 (256)	4	Static	Timer タスク (FreeRTOS software timers - FreeRTOS™)
main_task	MAIN_TASK	512 (1024)	4	Dynamic	ユーザタスク 1、2 を生成
vUserTask1	UserTask1	256 (512)	4	Dynamic	1 秒間隔で LED0 を点滅
vUserTask2	UserTask2	256 (512)	4	Dynamic	2 秒間隔で LED1 を点滅

表 4-2 タスク一覧 (RL78/G23-128p FPB)

Task Entry Function	Task Name	Stack Depth (Byte)	Priority	Memory Allocation	Description
prvIdleTask	IDLE	128 (256)	0	Static	Idle タスク (FreeRTOS idle task - FreeRTOS™)
prvTimerTask	Tmr Svc	128 (256)	4	Static	Timer タスク (FreeRTOS software timers - FreeRTOS™)
main_task	MAIN_TASK	512 (1024)	4	Dynamic	ユーザタスク 1、2 を生成
vUserTask1	UserTask1	256 (512)	4	Dynamic	1 秒間隔で LED1 を点滅
vUserTask2	UserTask2	256 (512)	4	Dynamic	2 秒間隔で LED2 を点滅

4.1.2 システム起動からタスク実行までの流れ

システム起動からタスク実行までのフローを以下に示します。

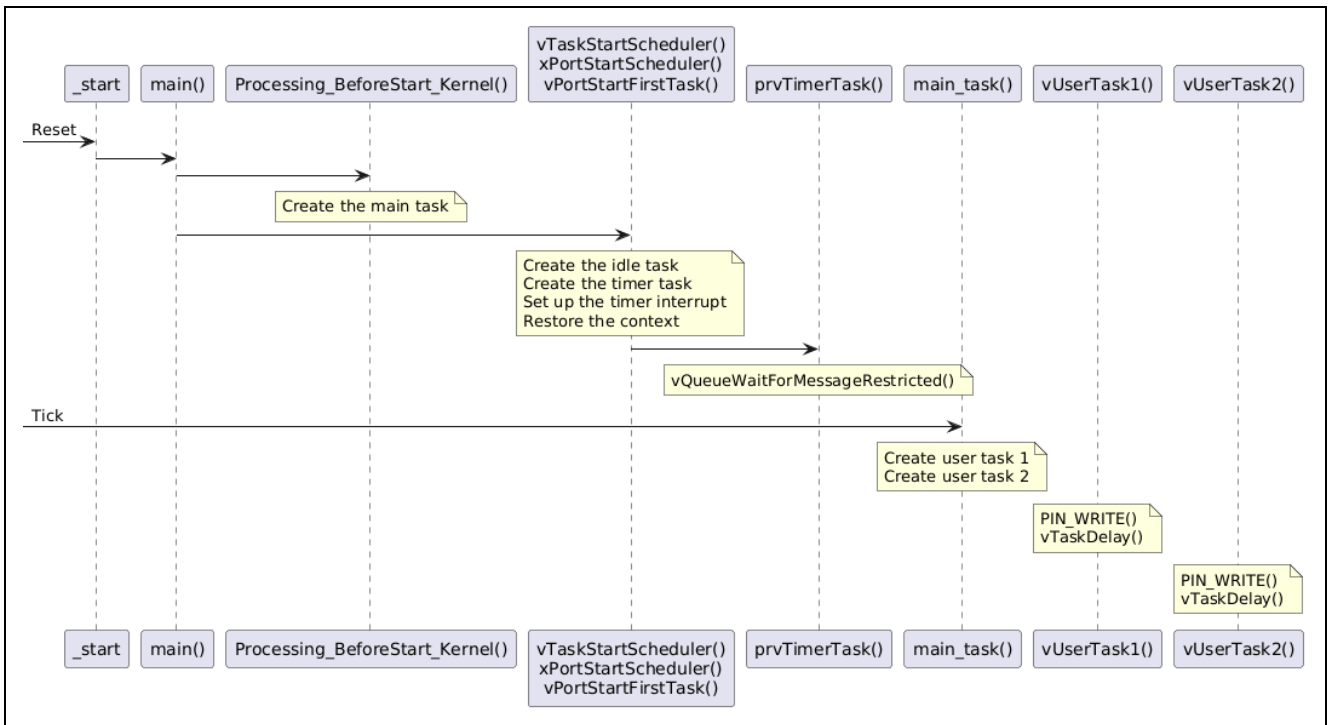


図 4-1 システム起動からタスク実行までのフロー

タスク起動後の各タスクの動作と切り替えのタイミングを以下に示します。

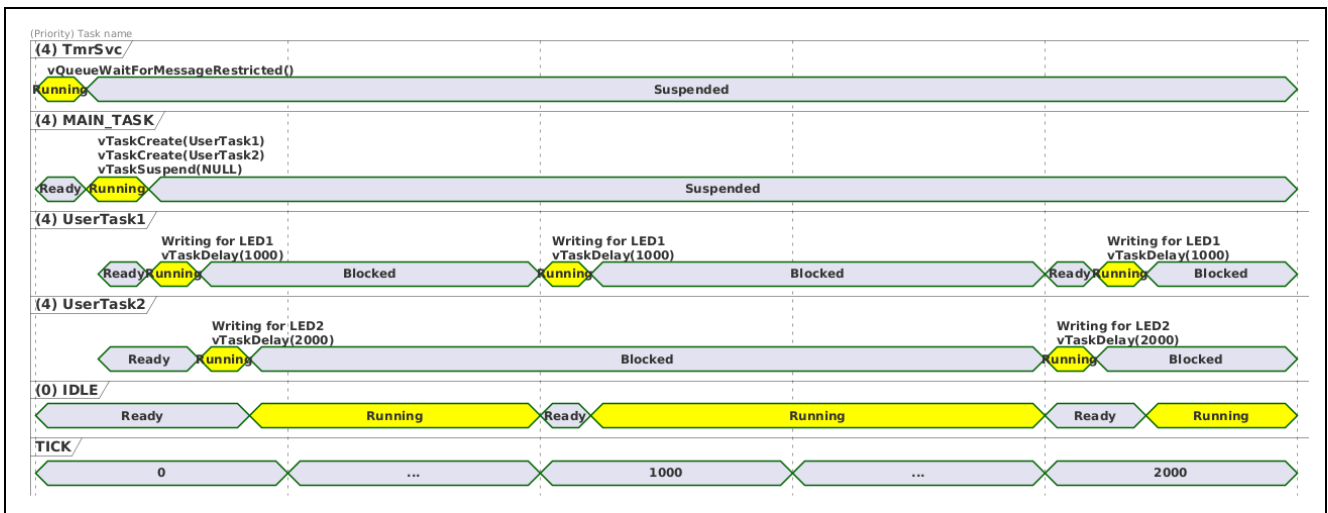


図 4-2 各タスク動作とタスク切り替えのタイミング

4.1.3 FreeRTOS Blinky デモの構成

FreeRTOS Blinky デモのユーザタスク仕様を以下に示します。

表 4-3 デモプロジェクト構成 (RL78/G14 FPB)

プロジェクト名	内容
rl78g14_fpb_blinky	ユーザタスク 1 : LED0 を 1 秒間隔で点滅 ユーザタスク 2 : LED1 を 2 秒間隔で点滅

表 4-4 デモプロジェクト構成 (RL78/G23-128p FPB)

プロジェクト名	内容
rl78g23_fpb_blinky	ユーザタスク 1 : LED1 を 1 秒間隔で点滅 ユーザタスク 2 : LED2 を 2 秒間隔で点滅

4.1.4 FreeRTOS カーネルのコンフィグ設定

FreeRTOSConfig.h の設定値一覧を下表に示します。

表 4-5 FreeRTOSConfig.h 設定値一覧

Configuration Name	Project Name	
	rl78g14_fpb_blinky	rl78g23_fpb_blinky
configCPU_CLOCK_HZ	((unsigned long) 32000000)	((unsigned long) 32000000)
configTICK_RATE_HZ	((TickType_t) 1000)	((TickType_t) 1000)
configUSE_PREEMPTION	1	1
configUSE_TIME_SLICING	1	1
configUSE_PORT_OPTIMISED_TASK_SELECTION	0	0
configUSE_TICKLESS_IDLE	0	0
configMAX_PRIORITIES	5	5
configMINIMAL_STACK_SIZE	128	128
configMAX_TASK_NAME_LEN	16	16
configTICK_TYPE_WIDTH_IN_BITS	TICK_TYPE_WIDTH_32_BITS	TICK_TYPE_WIDTH_32_BITS
configIDLE_SHOULD_YIELD	1	1
configTASK_NOTIFICATION_ARRAY_ENTRIES	1	1
configQUEUE_REGISTRY_SIZE	0	0
configENABLE_BACKWARD_COMPATIBILITY	1	1
configNUM_THREAD_LOCAL_STORAGE_POINTERS	0	0
configUSE_MINI_LIST_ITEM	1	1
configSTACK_DEPTH_TYPE	StackType_t	StackType_t
configMESSAGE_BUFFER_LENGTH_TYPE	size_t	size_t
configHEAP_CLEAR_MEMORY_ON_FREE	1	1
configSTATS_BUFFER_MAX_LENGTH	0xFFFF	0xFFFF
configUSE_NEWLIB_REENTRANT	0	0
configUSE_TIMERS	1	1
configTIMER_TASK_PRIORITY	(configMAX_PRIORITIES - 1)	(configMAX_PRIORITIES - 1)
configTIMER_TASK_STACK_DEPTH	configMINIMAL_STACK_SIZE	configMINIMAL_STACK_SIZE
configTIMER_QUEUE_LENGTH	5	5
configUSE_EVENT_GROUPS	1	1
configUSE_STREAM_BUFFERS	1	1
configSUPPORT_STATIC_ALLOCATION	1	1
configSUPPORT_DYNAMIC_ALLOCATION	1	1
configTOTAL_HEAP_SIZE	4096	4096
configAPPLICATION_ALLOCATED_HEAP	0	0
configSTACK_ALLOCATION_FROM_SEPARATE_HEAP	0	0
configENABLE_HEAP_PROTECTOR	0	0
configUSE_IDLE_HOOK	1	1
configUSE_TICK_HOOK	1	1
configUSE_MALLOC_FAILED_HOOK	1	1
configUSE_DAEMON_TASK_STARTUP_HOOK	1	1
configUSE_SB_COMPLETED_CALLBACK	0	0
configCHECK_FOR_STACK_OVERFLOW	2	2

configGENERATE_RUN_TIME_STATS	0	0
configUSE_TRACE_FACILITY	0	0
configUSE_STATS_FORMATTING_FUNCTIONS	0	0
configUSE_CO_ROUTINES	0	0
configMAX_CO_ROUTINE_PRIORITIES	1	1
configASSERT(x)	do { if((x) == 0) vAssertCalled(); } while(0)	do { if((x) == 0) vAssertCalled(); } while(0)
configUSE_TASK_NOTIFICATIONS	1	1
configUSE_MUTEXES	1	1
configUSE_RECURSIVE_MUTEXES	1	1
configUSE_COUNTING_SEMAPHORES	1	1
configUSE_QUEUE_SETS	1	1
configUSE_APPLICATION_TASK_TAG	0	0
INCLUDE_vTaskPrioritySet	1	1
INCLUDE_uxTaskPriorityGet	1	1
INCLUDE_vTaskDelete	1	1
INCLUDE_vTaskSuspend	1	1
INCLUDE_xResumeFromISR	1	1
INCLUDE_vTaskDelayUntil	1	1
INCLUDE_vTaskDelay	1	1
INCLUDE_xTaskGetSchedulerState	1	1
INCLUDE_xTaskGetCurrentTaskHandle	1	1
INCLUDE_uxTaskGetStackHighWaterMark	0	0
INCLUDE_xTaskGetIdleTaskHandle	0	0
INCLUDE_eTaskGetState	0	0
INCLUDE_xEventGroupSetBitFromISR	1	1
INCLUDE_xTimerPendFunctionCall	0	0
INCLUDE_xTaskAbortDelay	0	0
INCLUDE_xTaskGetHandle	0	0
INCLUDE_xTaskResumeFromISR	1	1

その他の設定情報は以下のドキュメントを参照して下さい。

[Customization - FreeRTOS™](#)

4.1.5 オプション・バイト設定一覧

オプション・バイトの設定を示します。

表 4-6 オプション・バイト設定 (RL78/G14 FPB, RL78/G23-128p FPB)

アドレス	設定値	内容
000C0H/040C0H	11101111B	ウォッチドッグ・タイマ 動作停止 (リセット解除後、カウント停止)
000C1H/040C1H	00111010B	LVD0 オフ (RESET 端子により外部リセットを使用)
000C2H/040C2H	11101000B	HS (高速メイン) モード & 高速オンチップ・オシレータ・クロック (fIH): 32MHz
000C3H/040C3H	10000100B	オンチップ・デバッグ許可

4.2 フォルダ構成

FreeRTOS Blinky デモのフォルダ構成を以下に示します。

表 4-7 FreeRTOS Blinky デモのフォルダ構成 (RL78/G14 FPB)

Folder Name	Description
rl78g14_fpb_blinky	RL78/G14 FPB の FreeRTOS Blinky デモのプロジェクト
├─.settings	
│ └─CodeGenerator	コード生成の設定
├─generate	ボードのスタートアップ
└─src	
├─application	アプリケーションのソースコード
│ └─app	ユーザ用のアプリケーション
│ └─config	FreeRTOS のコンフィグ
│ └─startup	FreeRTOS のスタートアップ
└─kernel	FreeRTOS カーネル

表 4-8 FreeRTOS Blinky デモのフォルダ構成 (RL78/G23-128p FPB)

Folder Name	Description
rl78g23_fpb_blinky	RL78/G23-128p FPB の FreeRTOS Blinky デモのプロジェクト
└─src	
├─application	アプリケーションのソースコード
│ └─app	ユーザ用のアプリケーション
│ └─config	FreeRTOS のコンフィグ
│ └─startup	FreeRTOS のスタートアップ
└─kernel	FreeRTOS カーネル
└─smc_gen	スマート・コンフィグレータによって生成されるコード

4.3 コードサイズ

以下の条件におけるデモプロジェクトのROM、RAMサイズを下表に示します。

CC-RL

- コンパイル・オプション
 - -Odefault : オブジェクト・サイズと実行速度の両方に効果のある最適化
 - -memory_model : 自動選択
 - -far_rom : メモリ・モデルで指定したROMデータに対するnear/far属性をfarに設定する
- リンク・オプション
 - -optimize=symbol_delete : 一度も参照のない変数/関数を削除

表 4-9 ROM/RAM サイズ

デモプロジェクト名	ROM (byte)	RAM (byte)
rl78g14_fpb_blinky	18302	5146 ※
rl78g23_fpb_blinky	18490	5146 ※

※システム起動から最初のタスク起動（Blinky デモの場合、Timer タスク）までに使用するスタックサイズは含まれません。「4.4 章 システム起動から最初のタスク起動までに使用するスタックサイズ」を参考にしてください。

4.4 システム起動から最初のタスク起動までに使用するスタックサイズ

CC-RL が設定する `__STACK_ADDR_START` シンボルと `__STACK_ADDR_END` シンボル間のアドレス領域がシステム起動のスタックとして使用されます。

システム起動から最初のタスク起動までに使用するスタックサイズを下表に示します。

条件は「4.3 章 コードサイズ」と同じです。

表 4-10 システム起動から最初のタスク起動までに使用するスタックサイズ

デモプロジェクト名	スタックサイズ (byte)
rl78g14_fpb_blinky	122
rl78g23_fpb_blinky	122

5. FreeRTOS Blinky デモのセットアップ

5.1 RL78/G14 FPB

5.1.1 プロジェクトをインポート

プロジェクト「rl78g14_fpb_blinky」を e² studio にインポートします。

File > Import > Existing Projects into Workspace > プロジェクト「rl78g14_fpb_blinky」を選択する。

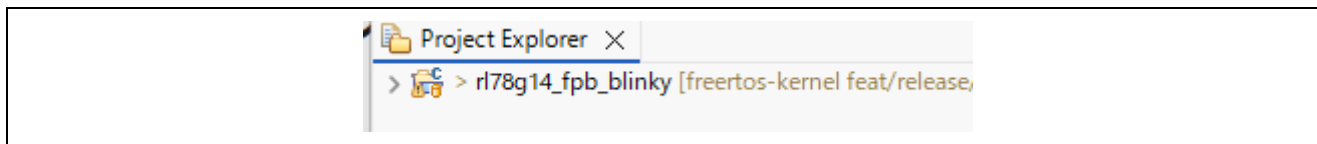


図 5-1 プロジェクト「rl78g14_fpb_blinky」を e² studio にインポート

5.1.2 プロジェクトをビルド

プロジェクトをビルドします。

プロジェクトを選択し、右クリック > Build Project

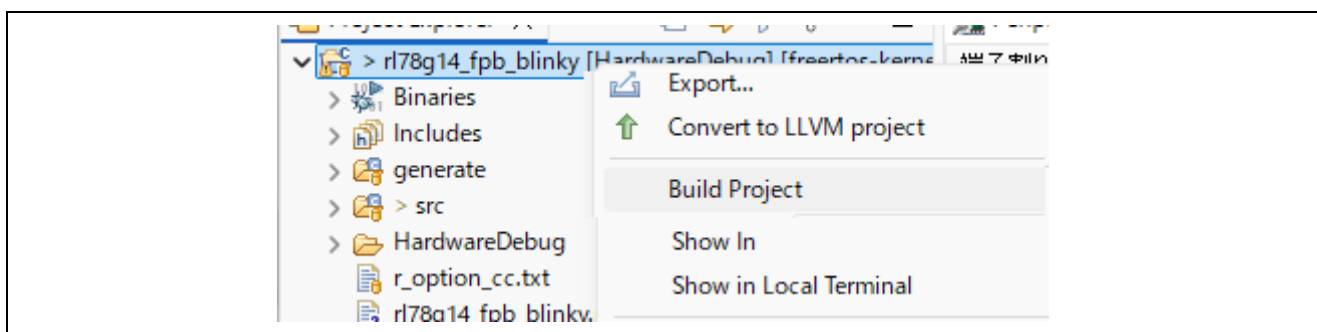


図 5-2 プロジェクト「rl78g14_fpb_blinky」をビルド

5.2 RL78/G23-128p FPB

5.2.1 プロジェクトをインポート

プロジェクト「rl78g23_fpb_blinky」を e² studio にインポートします。

File > Import > Existing Projects into Workspace > プロジェクト「rl78g23_fpb_blinky」を選択する。

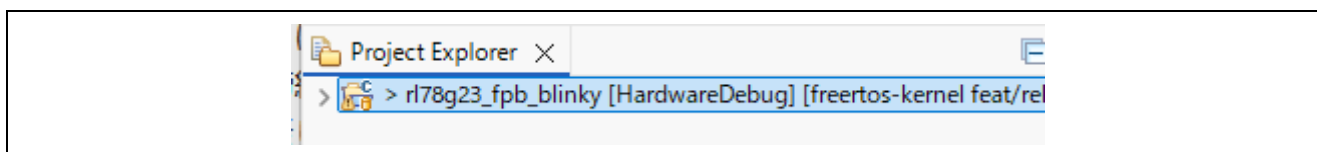


図 5-3 プロジェクト「rl78g23_fpb_blinky」を e² studio にインポート

5.2.2 プロジェクトをビルド

プロジェクトをビルドします。

プロジェクトを選択し、右クリック > Build Project

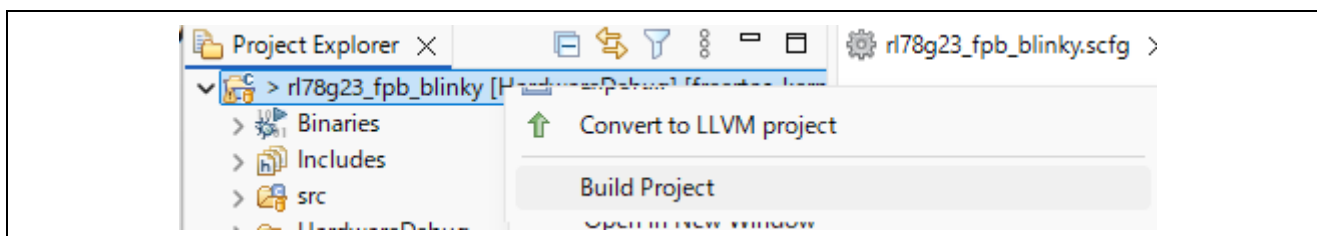


図 5-4 プロジェクト「rl78g23_fpb_blinky」をビルド

6. FreeRTOS Blinky デモの実行

6.1 RL78/G14 FPB

6.1.1 マイコンボードへ電源を供給

マイコンボードと PC を USB ケーブルで接続します。

6.1.2 デバッグコンフィグレーション

実行するコンフィグレーションを選択します。

Debug Configurations... > Renesas GDB Hardware Debugging > rl78g14_fpb_blinky HardwareDebug

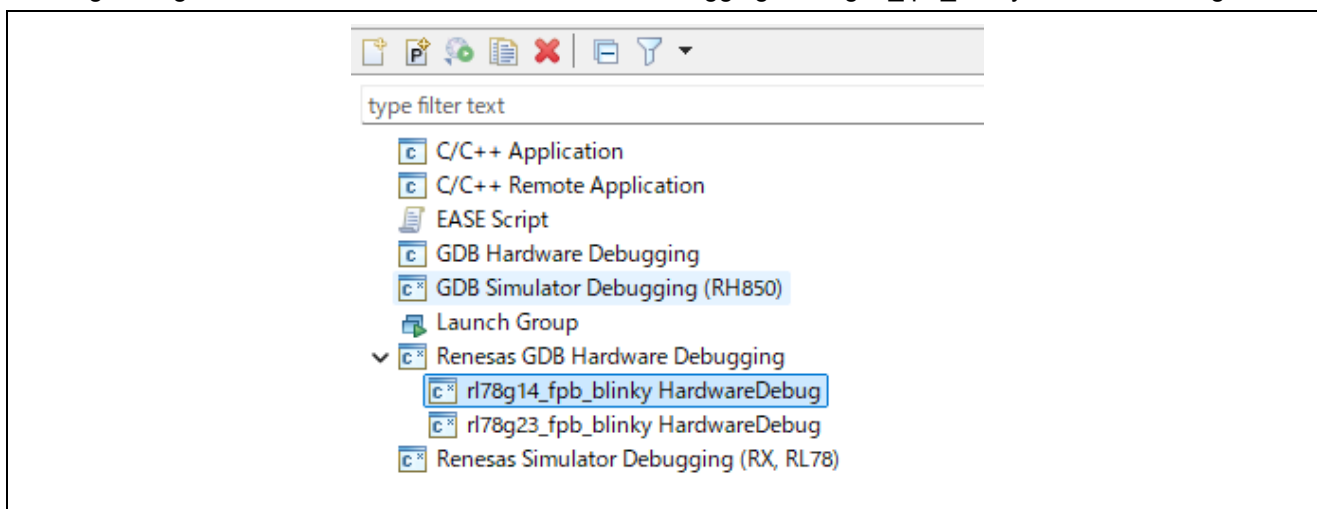


図 6-1 Debug Configuration for the rl78g14_fpb_blinky Project

6.1.3 デバッガ設定

"Debugger" tab を選択します。

Debug hardware: E2 Lite (RL78)

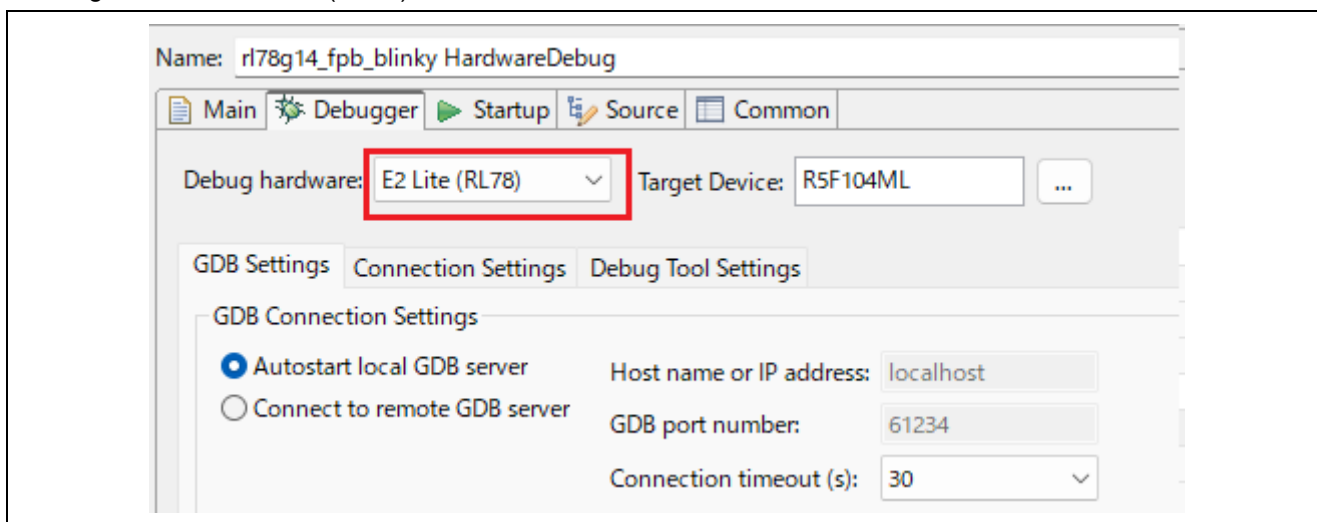


図 6-2 Debug hardware: E2 Lite (RL78)

“Connection Settings” tab > Connection with Target Board を選択します。

Power Target From The Emulator (MAX 200mA): No

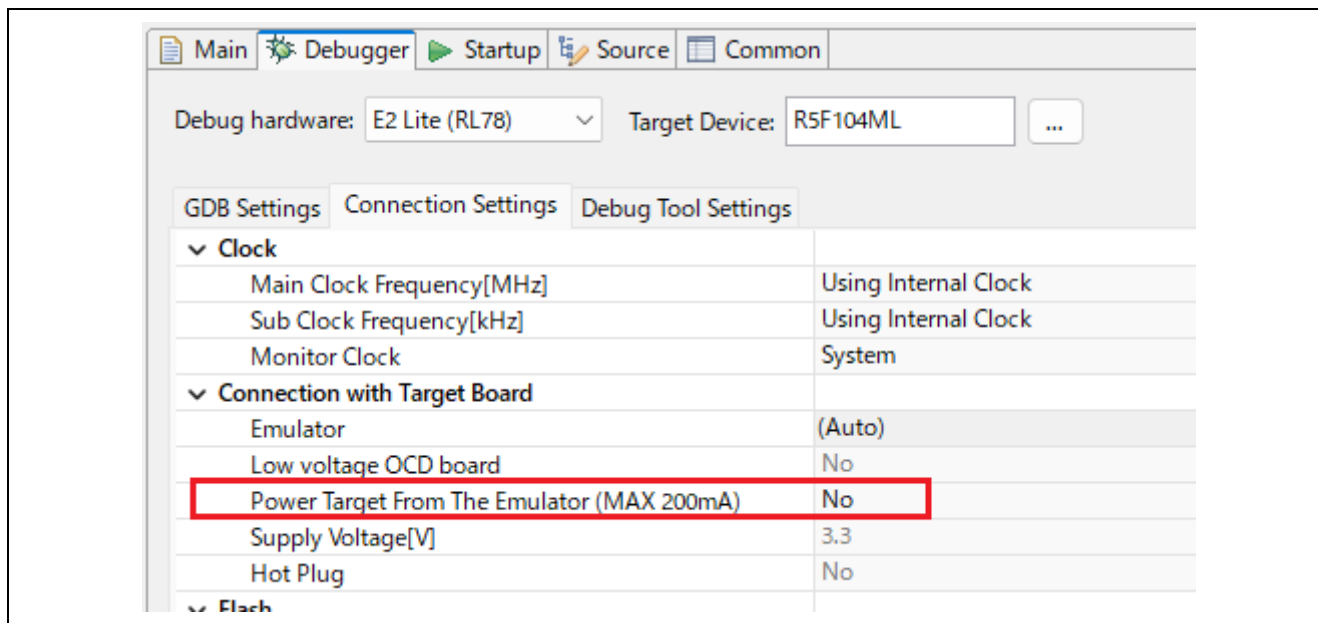
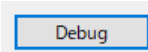


図 6-3 Connection Settings for Using an Emulator



をクリックしてデバッグを開始します。

6.1.4 FreeRTOS Blinky デモの実行



をクリックし、main()で停止することを確認します。

```
#include "FreeRTOS.h"
#include "freertos_user.h"
#include "user_task.h"
int main(void);
void main_task(void *pvParameters);
int main(void)
{
    ...Processing_Before_Start_Kernel();
    ...vTaskStartScheduler();
    ...while(1)
    {
        ...__nop();
    }
}
```

図 6-4 Debug hardware: E2 Lite (RL78)



をクリックすると、ボードの LED0 が 1 秒間隔、LED1 が 2 秒間隔で点滅します。

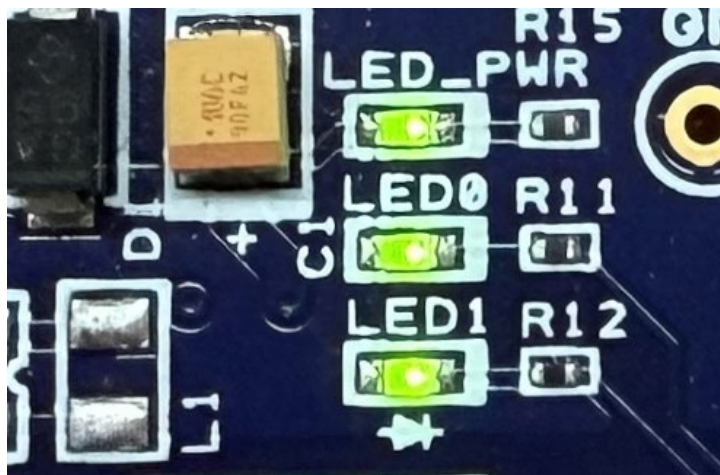


図 6-5 ボードの LED が点滅 (rl78g14_fpb_blinky)

6.2 RL78/G23-128p FPB

6.2.1 COM port を使用する場合

6.2.1.1 ジャンパピン設定

J15: 1-2 Short、J16: 1-2 Short、J19: 1-2 Short にしてください。回路変更を実施していない場合、この操作は不要です。

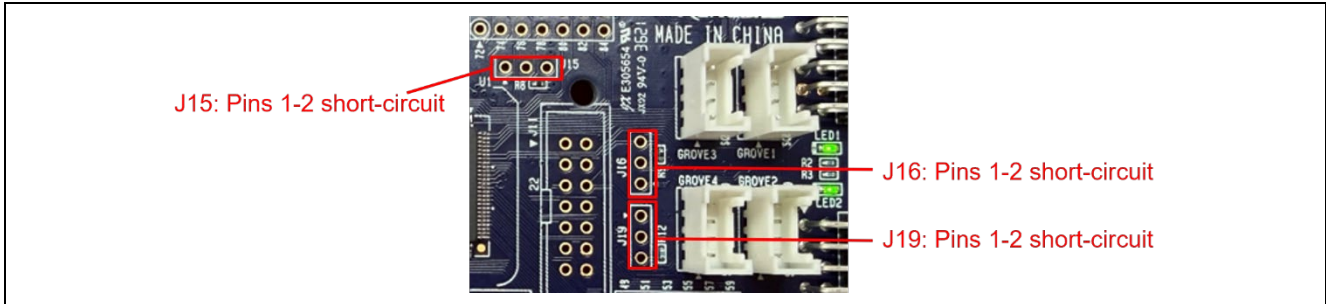


図 6-6 COM port デバッグ使用時設定 (部品面)

6.2.1.2 マイコンボードへ電源を供給

マイコンボードと PC を USB ケーブルで接続します。

6.2.1.3 デバッグコンフィグレーション

実行するコンフィグレーションを選択します。

Debug Configurations... > Renesas GDB Hardware Debugging > rl78g23_fpb_blinky HardwareDebug

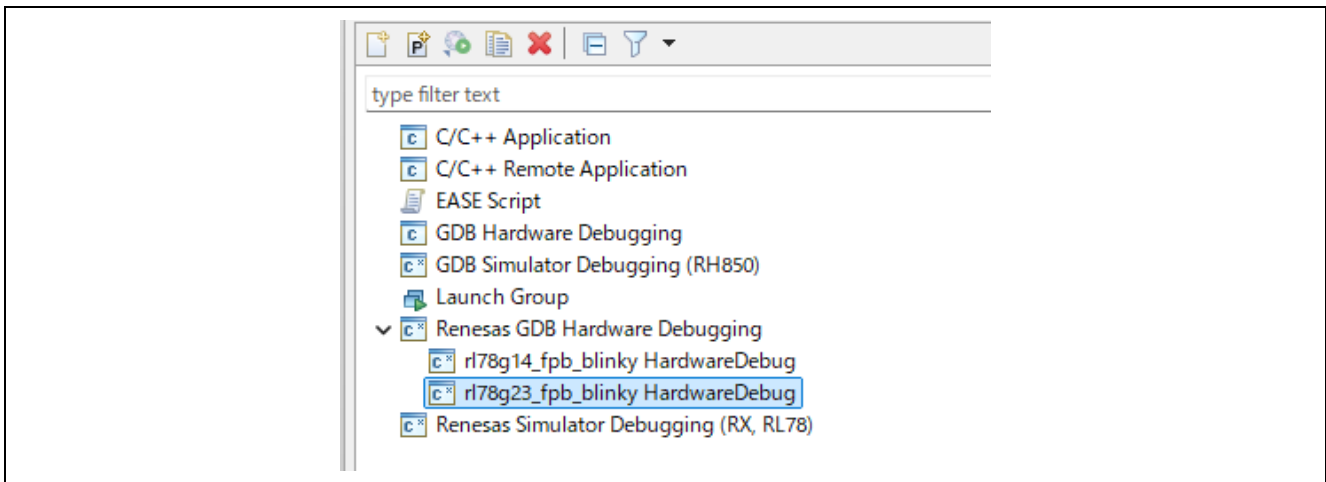


図 6-7 Debug Configuration for the rl78g23_fpb_blinky Project

6.2.1.4 デバッグ設定

”Debugger” tab を選択します。

Debug hardware: COM Port (RL78)

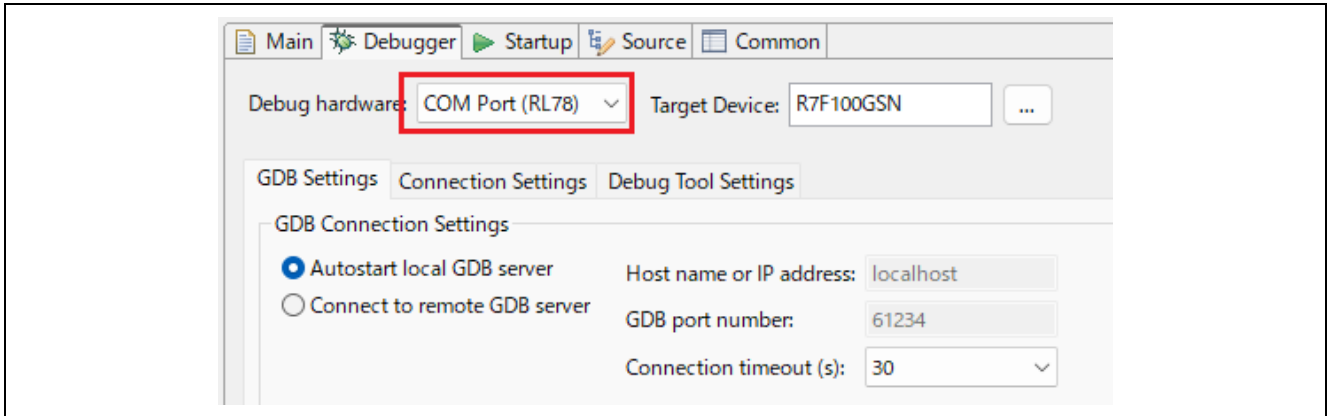


図 6-8 Debug hardware: COM Port (RL78)

“Connection Settings” tab > Connection with Target Board を選択します。COM ポートはマイコンボードが接続されている COM ポートを指定してください。

COM Port: COMxx

Reset control pin: DTR

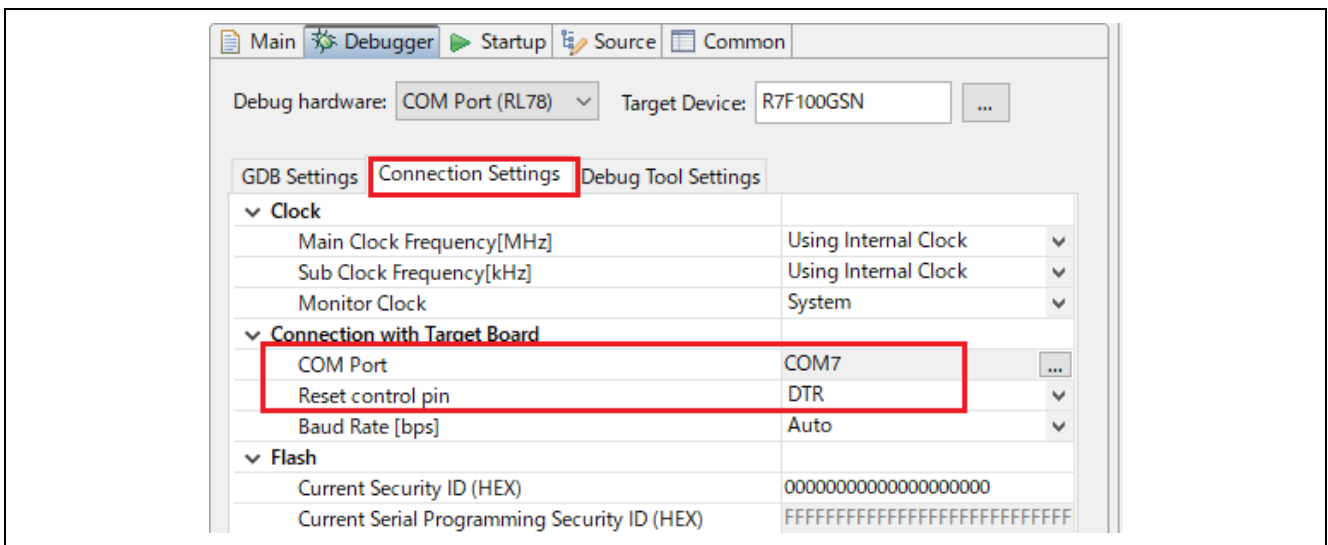
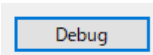


図 6-9 Connection Settings for Using a COM Port



をクリックしてデバッグを開始します。

6.2.2 エミュレータを使用する場合

6.2.2.1 コネクタ実装、ジャンパピン設定、パターンカット

14pin コネクタ (J11) は、ルネサスエレクトロニクス製のプログラミング機能付きオンチップ・デバッグ・エミュレータの E2 エミュレータまたは E2 エミュレータ Lite との接続用コネクタです (コネクタ部品は未実装です)。エミュレータを使用して、評価 MCU のプログラミングおよびデバッグを行います。エミュレータを接続する場合には下図の回路変更が必要となります。詳細は [RL78/G23-128p Fast Prototyping Board ユーザーズマニュアル](#) 5.20 章を参照してください。

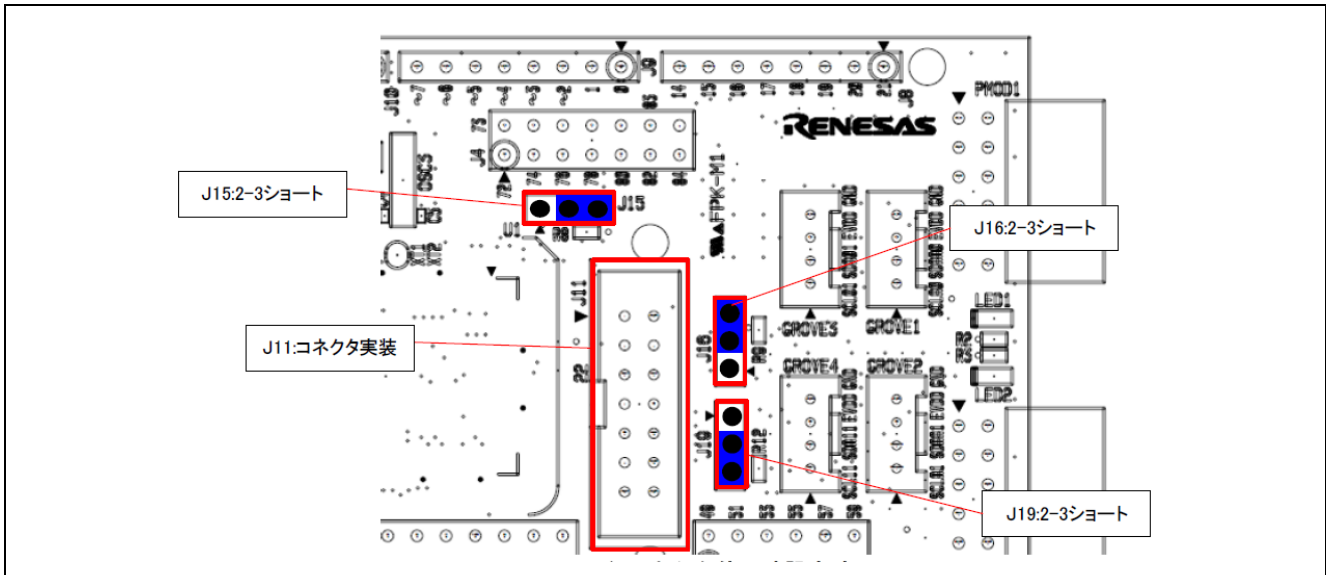


図 6-10 エミュレータコネクタ使用時設定 (部品面)

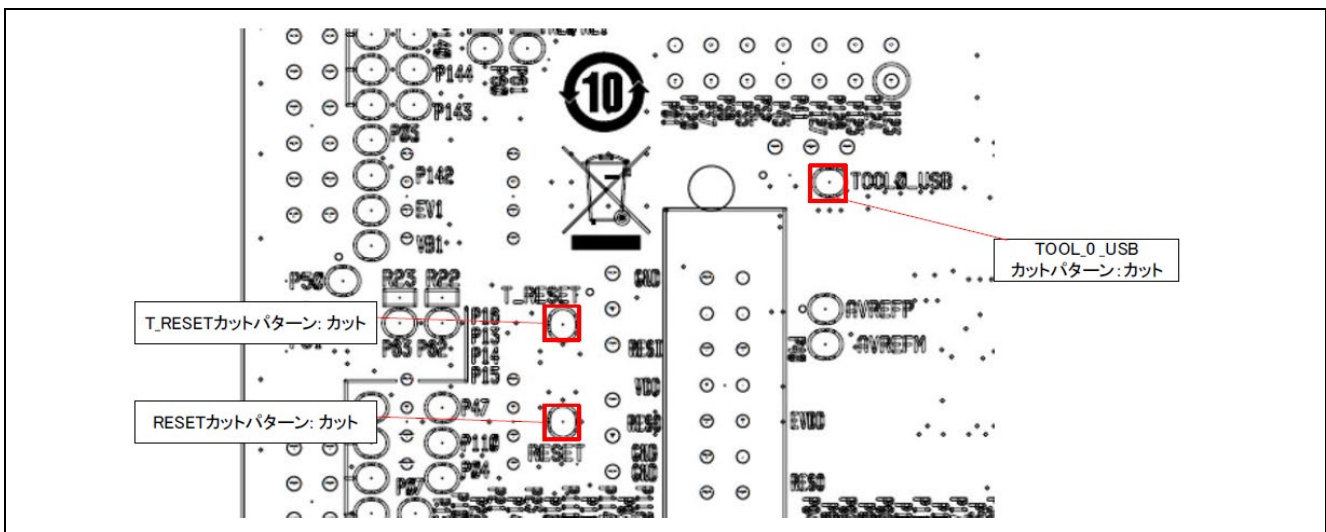


図 6-11 エミュレータコネクタ使用時設定 (ハンダ面)

また、エミュレータの使用方法については、「[E1/E20/E2 エミュレータ, E2 エミュレータ Lite ユーザーズマニュアル別冊 \(RL78 接続時の注意事項\)](#)」 (R20UT1994) を参照してください。

6.2.2.2 エミュレータをマイコンボードに接続

図の通りにエミュレータを接続します。

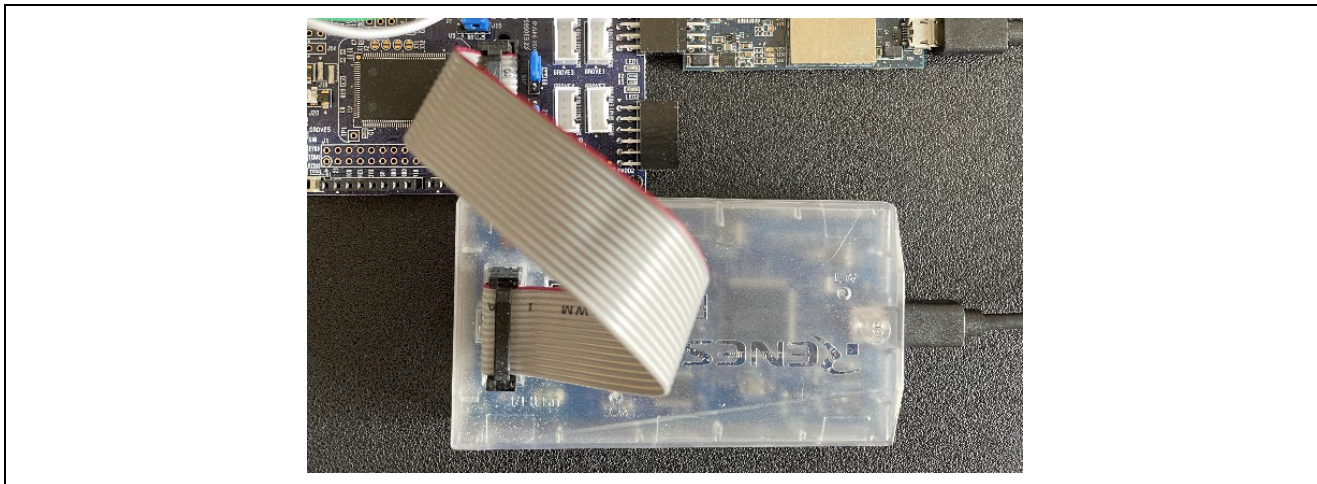


図 6-12 エミュレータをマイコンボードに接続

6.2.2.3 デバッグコンフィグレーション

実行するコンフィグレーションを選択します。

Debug Configurations > Renesas GDB Hardware Debugging > rl78g23_fpb_blinky HardwareDebug

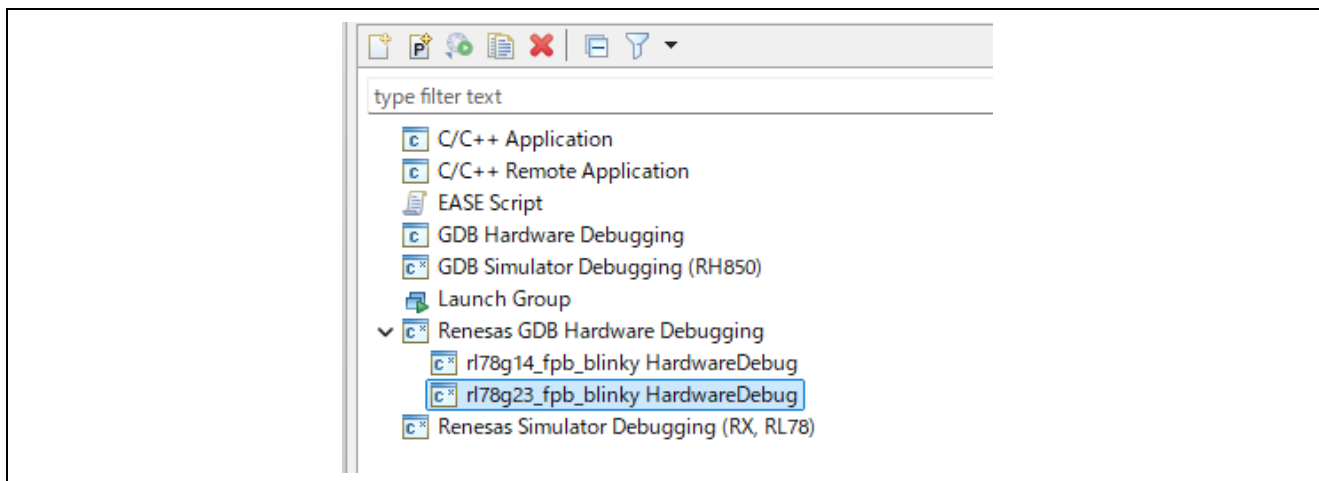


図 6-13 Debug Configuration for the rl78g23_fpb_blinky Project

6.2.2.4 デバッガ設定

”Debugger” tab を選択します。

Debug hardware: E2 Lite (RL78)

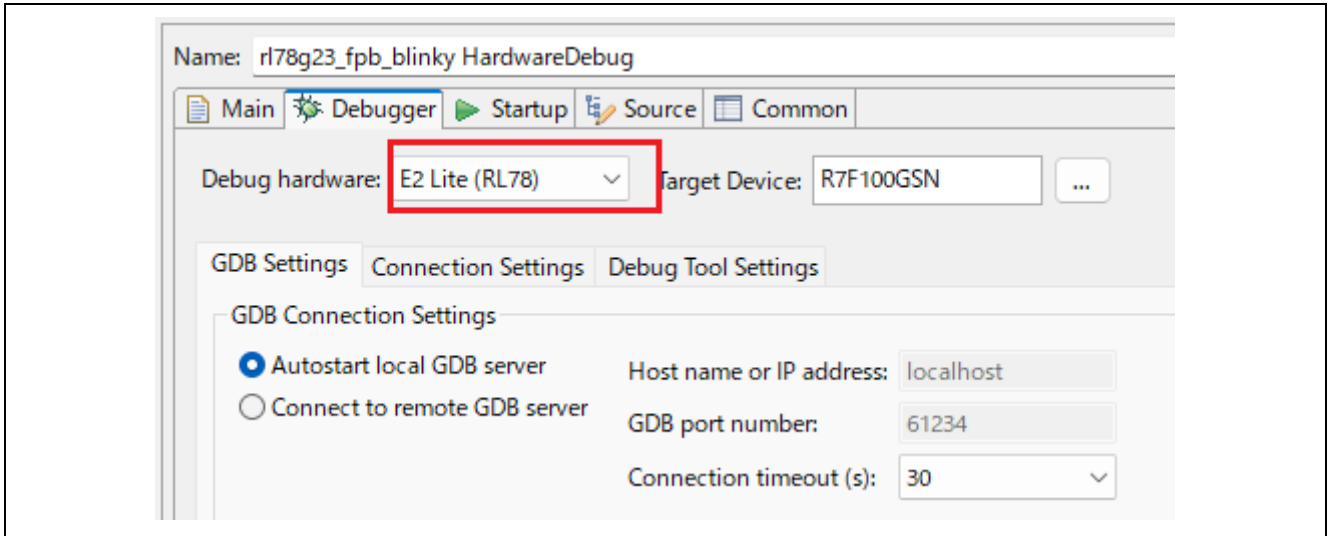


図 6-14 Debug hardware: E2 Lite (RL78)

“Connection Settings” tab > Connection with Target Board を選択します。

Power Target From The Emulator (MAX 200mA): No

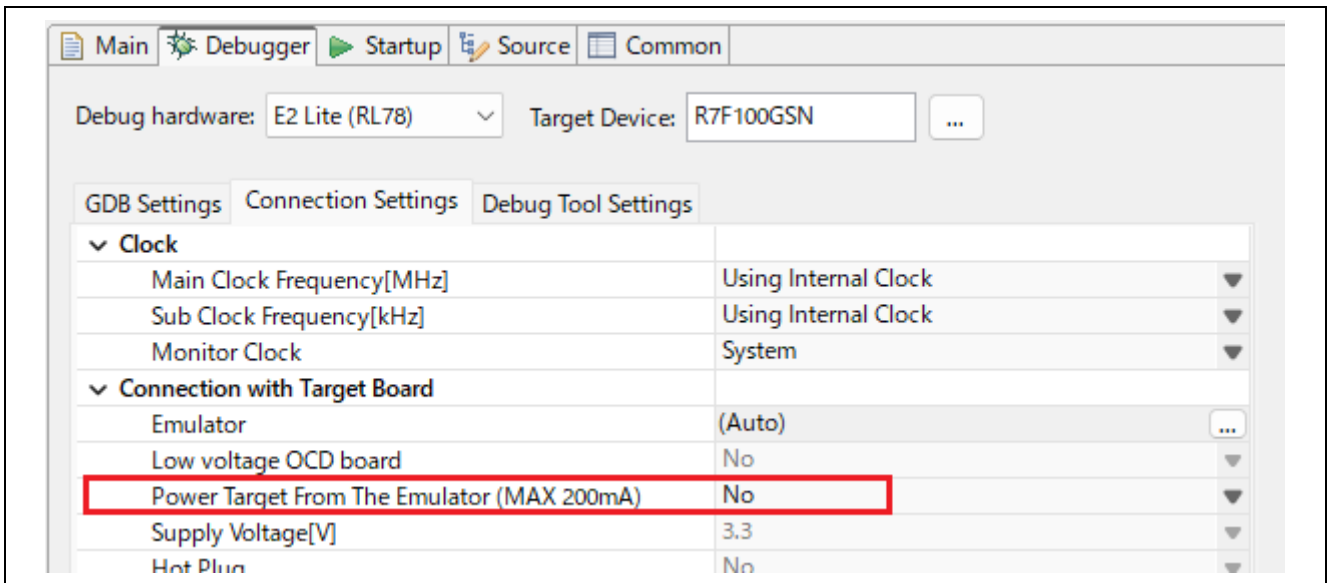
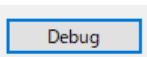


図 6-15 Connection Settings for Using an Emulator



をクリックしてデバッグを開始します。

6.2.3 FreeRTOS Blinky デモの実行



をクリックし、main()で停止することを確認します。

```
#include "FreeRTOS.h"  
#include "freertos_user.h"  
#include "user_task.h"  
  
int main(void);  
void main_task(void *pvParameters);  
  
int main(void)  
{  
    ...Processing_Before_Start_Kernel();  
    ...vTaskStartScheduler();  
  
    ...while(1)  
    {  
        ...__nop();  
    }  
}
```

図 6-16 Debug hardware: E2 Lite (RL78)



をクリックすると、ボードの LED1 が 1 秒間隔、LED2 が 2 秒間隔で点滅します。

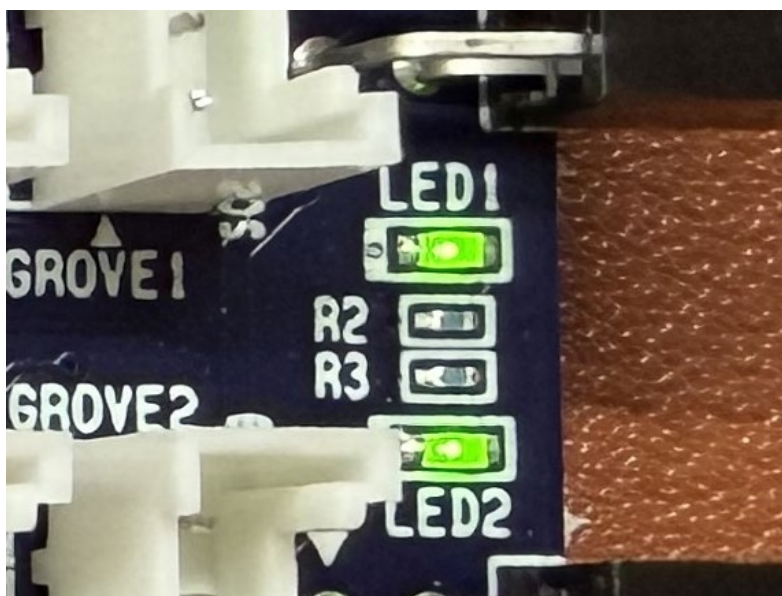


図 6-17 ボードの LED が点滅 (rl78g23_fpb_blinky)

7. 補足情報

7.1 ルネサス関数一覧

表 7-1 ルネサス関数一覧

関数名	説明
vApplicationSetupTimerInterrupt	この関数はスケジューラ開始時に実行されます。システムタイマの初期設定を実行します。
vAssertCalled	この関数は、パラメータチェックなどに使用されるアサーション関数にて、必要な条件が満たされなかった場合に実行されます。この関数が実行されると、クリティカルセクションへ突入後に永久ループします。
Processing_Before_Start_Kernel	この関数は、カーネルスケジューラのスタート前に実行するルネサス独自の関数です。スケジューラのスタート前にRTOSオブジェクト（タスク、メールボックス、セマフォなど）を作成したい場合に任意で使用します。

7.2 注意事項

CC-RL 向けの FreeRTOS には以下の注意事項があります。

レジスタ・バンク（RL78 デバイスの汎用レジスタに割り当てられたメモリ空間）の切り替えに対応していません。

割り込みネスト（[interrupt nesting](#)）に対応していません。

タスクスイッチ要求はソフトウェア割り込みによって実現されているため、タスクスイッチ処理はタスクスイッチ要求後に即座に実行されます。そのため、ユーザアプリケーションでは以下の制限があります。

- ISR 内では、ISR の最後の処理でタスクスイッチを要求（portYIELD_FROM_ISR マクロを実行）してください。この制限の詳細や回避方法については GitHub の [Issue チケット#21](#) を参照してください。
- クリティカルセクション内でタスクスイッチを要求しないでください。

CC-RL におけるメモリ配置の仕様では、near ポインタのサイズを 2byte から 4 byte に拡張すると、下位から 3byte 目は 0x0f になります。仕様の詳細は [CC-RL のユーザズマニュアル](#) 「データの内部表現と領域」章や「メモリ配置領域指定（`__near/__far`）」項をご覧ください。

- この仕様により、例えば、タスクエントリ関数の pvParameters 引数をタスク内で 4byte にキャストした値は xTaskCreate() の pvParameters 引数で指定した値と一致しません。
- したがって、near ポインタを引数に持つ API 関数、たとえば xTaskCreate() を使用する際は上記の仕様に留意してください。具体例としては、タスク内で pvParameters 引数を 4byte 整数として利用する場合、その引数で参照された値を 2byte 整数（int 型）→4byte 整数（long 型）の順に 2 回キャストするなどの対策をユーザアプリケーション側で実施してください。

8. ウェブサイトおよびサポート

- FreeRTOS に関するサポートは FreeRTOS の公式ページにお問合せください。
<https://freertos.org/Why-FreeRTOS/Support-options>
- Blinky デモに関するサポートは弊社の窓口にお問合せください。
<https://www.renesas.com/support>

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Jan.16.26	-	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、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えば、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、ノイズ輻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まれます。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、改変、複製、リパ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、リパ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も、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、不正な改変、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、脆弱性問題に起因した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といたします。
 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（豊洲フォレスト）

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。