

RL78/G14, RL78/G23

FreeRTOS Blinky Demo

Introduction

This application note describes a demo program for making the LEDs mounted on a microcontroller board from Renesas Electronics blink with the use of FreeRTOS.

Target Device

RL78/G14, RL78/G23

Contents

1. Overview	4
1.1 Overview of the FreeRTOS Blinky Demo.....	4
1.2 Software Stack Diagrams	5
2. Conditions under Which Operation Was Verified	6
2.1 Lists of Equipment.....	7
2.1.1 RL78/G14	7
2.1.2 RL78/G23	7
3. Hardware.....	8
3.1 RL78/G14 FPB	8
3.1.1 List of Pins Used.....	8
3.1.2 List of Peripheral Function.....	8
3.2 RL78/G23-128p FPB	8
3.2.1 List of Pins Used.....	8
3.2.2 List of Peripheral Function.....	8
4. Software	9
4.1 FreeRTOS Blinky Demo.....	9
4.1.1 List of Tasks	9
4.1.2 Flow from Starting the System to Execution of the Tasks.....	10
4.1.3 Configuration of the FreeRTOS Blinky Demo	11
4.1.4 Configuration Settings for the FreeRTOS Kernel.....	12
4.1.5 List of Settings of the Option Bytes	14
4.2 Structure of the Folders.....	14
4.3 Code Sizes	15
4.4 Stack Size Used during the Period from Starting up the System to the Start of the First Task	15
5. Setting Up the FreeRTOS Blinky Demo	16
5.1 RL78/G14 FPB	16
5.1.1 Importing the Project	16
5.1.2 Building the Project	16
5.2 RL78/G23-128p FPB	17
5.2.1 Importing the Project	17
5.2.2 Building the Project	17
6. Running the FreeRTOS Blinky Demo	18
6.1 RL78/G14 FPB	18
6.1.1 Power Supply to the Microcontroller Board.....	18
6.1.2 Debug Configuration	18
6.1.3 Settings for the Debugger	18
6.1.4 Running the FreeRTOS Blinky Demo	20
6.2 RL78/G23-128p FPB	21
6.2.1 For Using a COM Port.....	21
6.2.1.1 Setting Jumper Pins	21
6.2.1.2 Power Supply to the Microcontroller Board.....	21
6.2.1.3 Debug Configuration	21

6.2.1.4	Settings for the Debugger	22
6.2.2	For Using an Emulator	23
6.2.2.1	Mounting a Connector, Setting Jumper Pins, and Patterns for Cutting	23
6.2.2.2	Connecting an Emulator to the Microcontroller Board	24
6.2.2.3	Debug Configuration	24
6.2.2.4	Settings for the Debugger	25
6.2.3	Running the FreeRTOS Blinky Demo	26
7.	Supplementary Information.....	27
7.1	List of Renesas Functions	27
7.2	Usage Note.....	27
8.	Website and Support	28
	Revision History	29

Note:

AWS™ and FreeRTOS are trademarks of Amazon.com, Inc. or its affiliates.

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

GitHub® is a trademark of GitHub, Inc. (<https://github.com/logos>)

1. Overview

This application note describes how to run a demo program that uses FreeRTOS and an RL78 family microcontroller board, either the RL78/G14 Fast Prototyping Board or the RL78/G23-128p Fast Prototyping Board, to control the LEDs mounted on the board so that they blink on and off. Both programs are hereinafter referred to as the FreeRTOS Blinky demo. FreeRTOS is a popular real-time operating system (RTOS) for microcontrollers. For details on FreeRTOS, refer to [FreeRTOS documentation - FreeRTOS™](#).

1.1 Overview of the FreeRTOS Blinky Demo

The FreeRTOS Blinky demo generates two user tasks under FreeRTOS for controlling the LEDs mounted on the board to make them blink on and off. For details, see section 4.1.3 Configuration of the FreeRTOS Blinky Demo.

1.2 Software Stack Diagrams

The figure below shows the software stack for the demo on the RL78/G14.

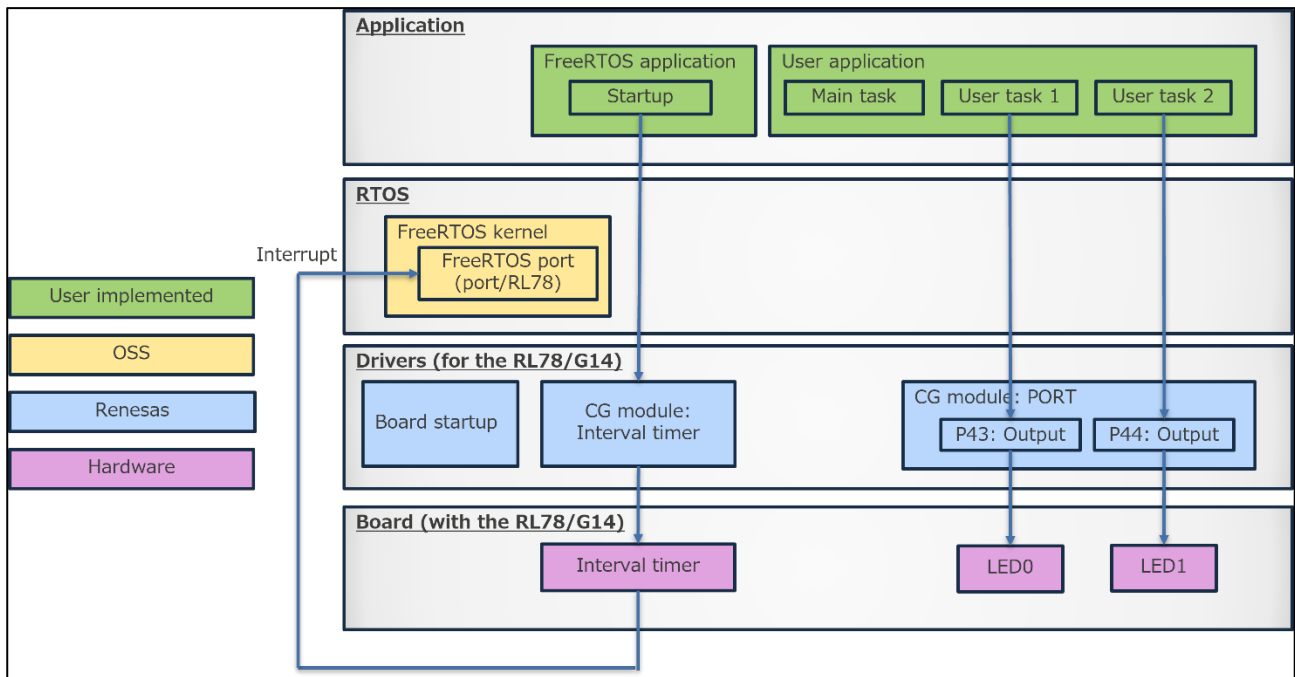


Figure 1-1 Software Stack for the Demo on the RL78/G14

The figure below shows the software stack for the demo on the RL78/G23.

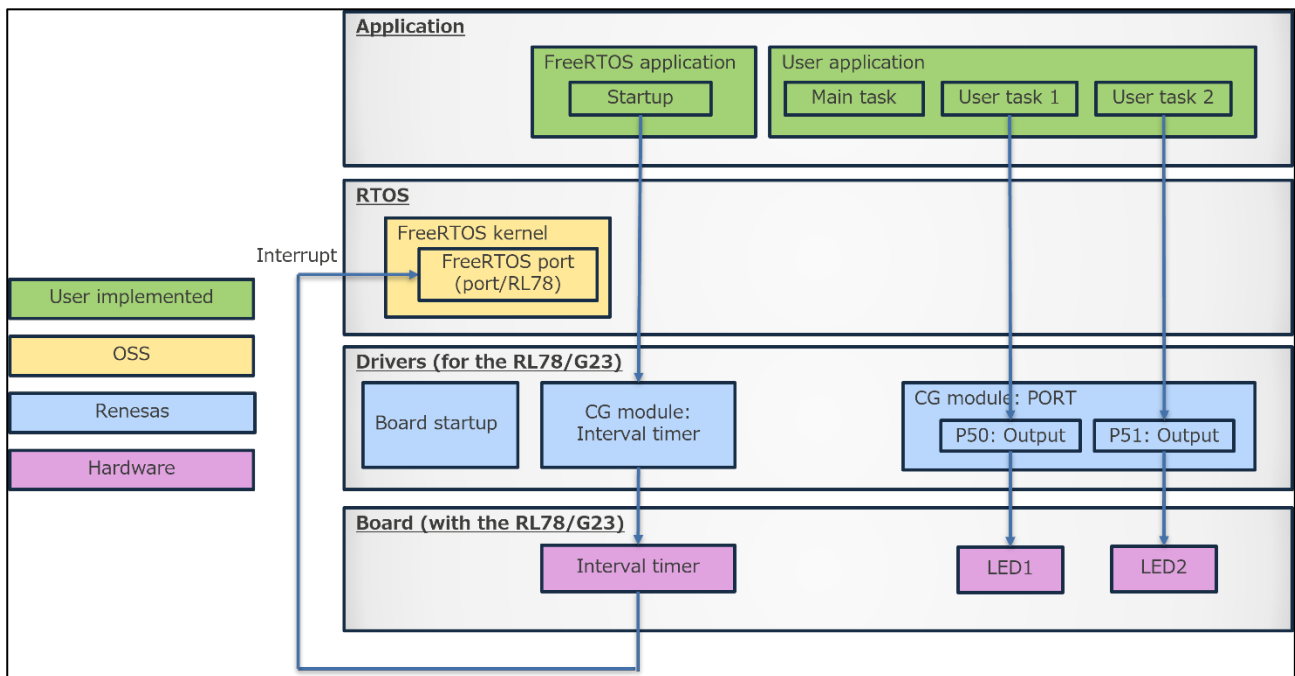


Figure 1-2 Software Stack for the Demo on the RL78/G23

2. Conditions under Which Operation Was Verified

The demo projects were verified under the specific sets of conditions listed below.

Table 2-1 Conditions under Which Operation Was Verified for the RL78/G14

Item	Detail
MCU used	RL78/G14, R5F104MLA with 512 Kbytes of code flash memory
Board used	RL78/G14 Fast Prototyping Board (RTK5RLG140C00000BJ)
Operating frequency	High-speed on-chip oscillator clock at a frequency of 32 MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics e ² studio 2025-07
C compiler	Renesas Electronics CC-RL V1.15.01
Code Generator (CG)	Renesas e2 studio Code Generator for RL78 25.7.0.v20250612-0710

Table 2-2 Conditions under Which Operation Was Verified for the RL78/G23

Item	Detail
MCU used	RL78/G23, R7F100GSN with 768 Kbytes of code flash memory
Board used	RL78/G23-128p Fast Prototyping Board (RTK7RLG230CSN000BJ)
Operating frequency	High-speed on-chip oscillator clock at a frequency of 32 MHz
Operating voltage	5 V
Integrated development environment	Renesas Electronics e ² studio 2025-07
C compiler	Renesas Electronics CC-RL V1.15.01
Smart Configurator (SC)	Renesas Smart Configurator for RL78 25.7.0.v20250620-0609
Board Support Package (BSP)	v1.91 (r_bsp)

2.1 Lists of Equipment

2.1.1 RL78/G14

The equipment required for this demo project is listed below.

Table 2-3 List of Equipment for the RL78/G14

Item	Detail
Microcontroller board	RL78/G14 Fast Prototyping Board, hereinafter referred to as the RL78/G14 FPB https://www.renesas.com/rtk5rlg140c00000bj
One micro-B USB cable	For connecting the microcontroller board and the PC

2.1.2 RL78/G23

The equipment required for this demo project is listed below.

Table 2-4 List of Equipment for the RL78/G23

Item	Detail
Microcontroller board	RL78/G23-128p Fast Prototyping Board, hereinafter referred to as the RL78/G23-128p FPB https://www.renesas.com/rtk7rlg230csn000bj
One micro-B USB cable	For connecting the microcontroller board and the PC
Three jumper pins	J15, J16, and J19 for selecting the power-supply source for the microcontroller board

Although these demo projects were developed with the use of COM port debugging, note that an emulator can also be used for debugging.

Table 2-5 List of Equipment for Use in Debugging

Item	Detail
Emulator	E2 emulator Lite https://www.renesas.com/software-tool/e2-emulator-lite-rte0t0002lkce00000r

Usage Note

If you wish to use an emulator, a connector for connecting the emulator must be mounted, and the relevant circuit must be changed. For details, see section 6.2.2.1 Mounting a Connector, Setting Jumper Pins, and Patterns for Cutting.

3. Hardware

3.1 RL78/G14 FPB

3.1.1 List of Pins Used

The pins used for the FreeRTOS Blinky demo and their functions are listed below.

Table 3-1 Pins Used and Their Functions for the RL78/G14

Pin Name	I/O	Function
P43	Output	LED0
P44	Output	LED1

3.1.2 List of Peripheral Function

The peripheral function used for the FreeRTOS Blinky demo and its usage is listed below.

Table 3-2 Peripheral Function and Its Usage for the RL78/G14

Peripheral Function	Usage
12-bit interval timer	System timer for FreeRTOS

3.2 RL78/G23-128p FPB

3.2.1 List of Pins Used

The pins used for the FreeRTOS Blinky demo and their functions are listed below.

Table 3-3 Pins Used and Their Functions for the RL78/G23

Pin Name	I/O	Function
P50	Output	LED1
P51	Output	LED2

3.2.2 List of Peripheral Function

The peripheral function used for the FreeRTOS Blinky demo and its usage is listed below.

Table 3-4 Peripheral Function and Its Usage for the RL78/G23

Peripheral Function	Usage
32-bit interval timer	System timer for FreeRTOS

4. Software

4.1 FreeRTOS Blinky Demo

The FreeRTOS Blinky demo controls blinking of the LEDs through two user tasks running under FreeRTOS.

4.1.1 List of Tasks

The tasks in the FreeRTOS Blinky demo are listed below.

Table 4-1 List of Tasks for the RL78/G14 FPB

Task Entry Function	Task Name	Stack Depth (Bytes)	Priority	Memory Allocation	Description
prvIdleTask	IDLE	128 (256)	0	Static	An idle task (FreeRTOS idle task - FreeRTOS™)
prvTimerTask	Tmr Svc	128 (256)	4	Static	A timer task (FreeRTOS software timers - FreeRTOS™)
main_task	MAIN_TASK	512 (1024)	4	Dynamic	Generating user tasks 1 and 2
vUserTask1	UserTask1	256 (512)	4	Dynamic	Making LED0 blink every 1 second
vUserTask2	UserTask2	256 (512)	4	Dynamic	Making LED1 blink every 2 seconds

Table 4-2 List of Tasks for the RL78/G23-128p FPB

Task Entry Function	Task Name	Stack Depth (Bytes)	Priority	Memory Allocation	Description
prvIdleTask	IDLE	128 (256)	0	Static	An idle task (FreeRTOS idle task - FreeRTOS™)
prvTimerTask	Tmr Svc	128 (256)	4	Static	A timer task (FreeRTOS software timers - FreeRTOS™)
main_task	MAIN_TASK	512 (1024)	4	Dynamic	Generating user tasks 1 and 2
vUserTask1	UserTask1	256 (512)	4	Dynamic	Making LED1 blink every 1 second
vUserTask2	UserTask2	256 (512)	4	Dynamic	Making LED2 blink every 2 seconds

4.1.2 Flow from Starting the System to Execution of the Tasks

The flow from starting the system to execution of the tasks is shown below.

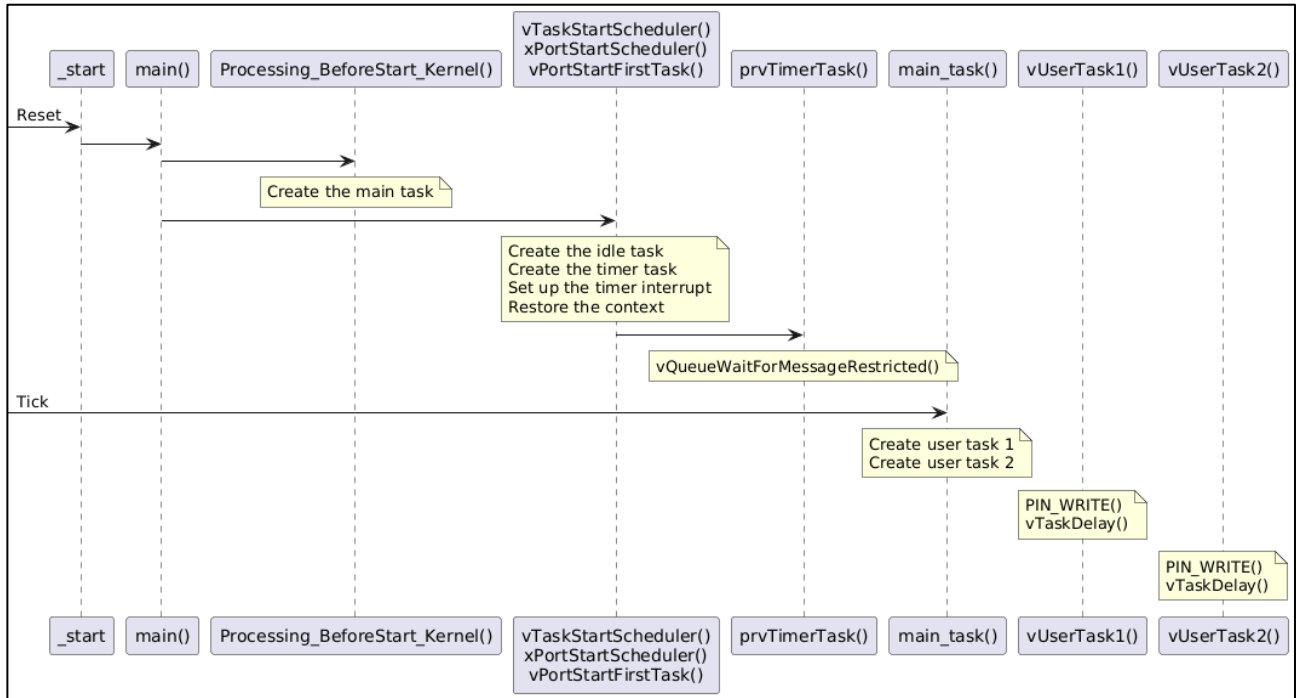


Figure 4-1 Flow from Starting the System to Execution of the Tasks

The operation of the respective tasks after having been activated and the times of their switching are shown below.

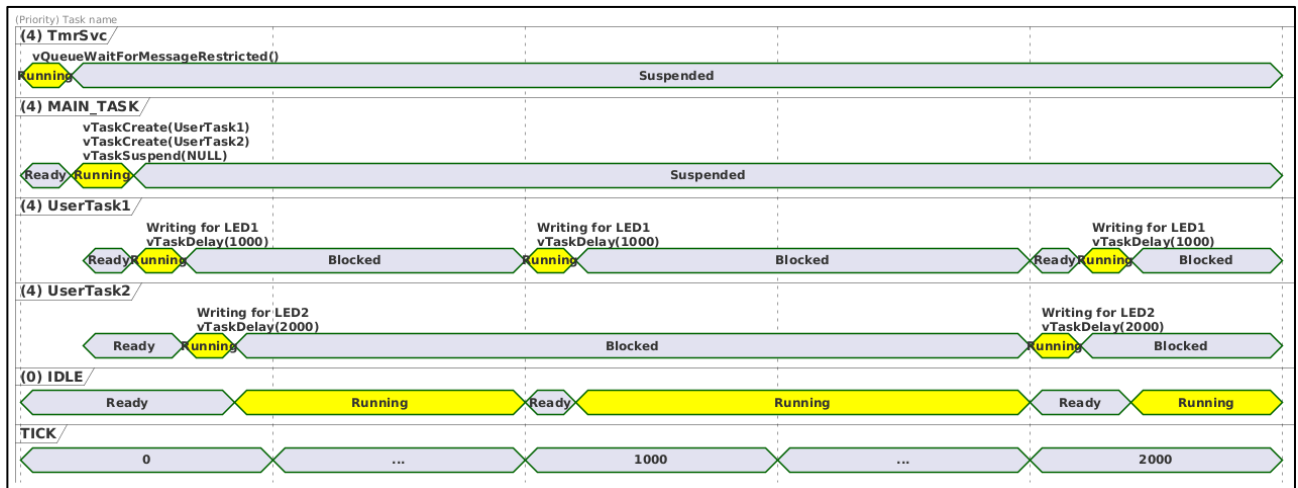


Figure 4-2 Operation of the Respective Tasks and Times of Their Switching

4.1.3 Configuration of the FreeRTOS Blinky Demo

The specifications of the user tasks in the FreeRTOS Blinky demo are listed below.

Table 4-3 Demo Project Configuration for the RL78/G14 FPB

Project Name	Details
rl78g14_fpb_blinky	User task 1: Making LED0 blink every 1 second User task 2: Making LED1 blink every 2 seconds

Table 4-4 Demo Project Configuration for the RL78/G23-128p FPB

Project Name	Details
rl78g23_fpb_blinky	User task 1: Making LED1 blink every 1 second User task 2: Making LED2 blink every 2 seconds

4.1.4 Configuration Settings for the FreeRTOS Kernel

The setting values for FreeRTOSConfig.h are listed in the following table.

Table 4-5 List of the Setting Values for FreeRTOSConfig.h

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_BI TS	TICK_TYPE_WIDTH_32_BI TS
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_POINTER S	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_SIZ E	configMINIMAL_STACK_SIZ E
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_H EAP	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

For further information on settings, refer to the following document.

[Customization - FreeRTOS™](#)

4.1.5 List of Settings of the Option Bytes

The settings of the option bytes are listed below.

Table 4-6 Settings of the Option Bytes for the RL78/G14 FPB and RL78/G23-128p FPB

Address	Setting Value	Details
000C0H/040C0H	11101111B	Stops watchdog timer operation (counting is stopped after a reset)
000C1H/040C1H	00111010B	LVD0 off setting (external reset input from the RESET pin is used)
000C2H/040C2H	11101000B	HS (high-speed main) mode, and the frequency of the high-speed on-chip oscillator clock (fIH) is 32 MHz
000C3H/040C3H	10000100B	Enables on-chip debugging.

4.2 Structure of the Folders

The structures of the folders for the FreeRTOS Blinky demo are described below.

Table 4-7 Structure of the Folders for the FreeRTOS Blinky Demo for the RL78/G14 FPB

Folder Name	Description
rl78g14_fpb_blinky	Project for the FreeRTOS Blinky demo for the RL78/G14 FPB
├ .settings	
├ └ CodeGenerator	Settings for code generation
├ └ generate	Starting up the board
├ └ src	
├ └ └ application	Source code of the application
├ └ └ └ app	User application
├ └ └ └ config	Configuration of FreeRTOS
├ └ └ └ startup	Starting up FreeRTOS
├ └ └ kernel	FreeRTOS kernel

Table 4-8 Structure of the Folders for the FreeRTOS Blinky Demo for the RL78/G23-128p FPB

Folder Name	Description
rl78g23_fpb_blinky	Project for the FreeRTOS Blinky demo for the RL78/G23-128p FPB
└ src	
├ └ application	Source code of the application
├ └ └ app	User application
├ └ └ config	Configuration of FreeRTOS
├ └ └ startup	Starting up FreeRTOS
├ └ kernel	FreeRTOS kernel
├ └ smc_gen	Code generated by the Smart Configurator

4.3 Code Sizes

The ROM and RAM sizes in the demo projects under the following conditions are listed below.

CC-RL

- Compiler options
 - -Odefault: Applied optimization is effective for both the object size and execution speed.
 - -memory_model: Automatically selected
 - -far_rom: Setting the near/far attribute for the ROM data specified by the memory model to far
- Linker option
 - -optimize=symbol_delete: Deleting variables or functions not referred to even once

Table 4-9 ROM and RAM Sizes

Demo Project Name	ROM (Bytes)	RAM (Bytes)
rl78g14_fpb_blinky	18302	5146 ^{Note}
rl78g23_fpb_blinky	18490	5146 ^{Note}

Note: This amount does not include the stack size used during the period from starting up the system to the start of the first task, that is, a timer task for the Blinky demo. For reference, see section 4.4 Stack Size Used during the Period from Starting up the System to the Start of the First Task.

4.4 Stack Size Used during the Period from Starting up the System to the Start of the First Task

The address area between the `__STACK_ADDR_START` symbol and `__STACK_ADDR_END` symbol is used as the stack for starting up the system. The symbols are specified by CC-RL. The stack sizes respectively used during the period from starting up the system to the start of the first task are listed below. The conditions for section 4.3 Code Sizes also apply.

Table 4-10 Stack Sizes Respectively Used during the Period from Starting up the System to the Start of the First Task

Demo Project Name	Stack Size (Bytes)
rl78g14_fpb_blinky	122
rl78g23_fpb_blinky	122

5. Setting Up the FreeRTOS Blinky Demo

5.1 RL78/G14 FPB

5.1.1 Importing the Project

Import the rl78g14_fpb_blinky project into the e² studio.

File > Import > Existing Projects into Workspace > Select the rl78g14_fpb_blinky project.

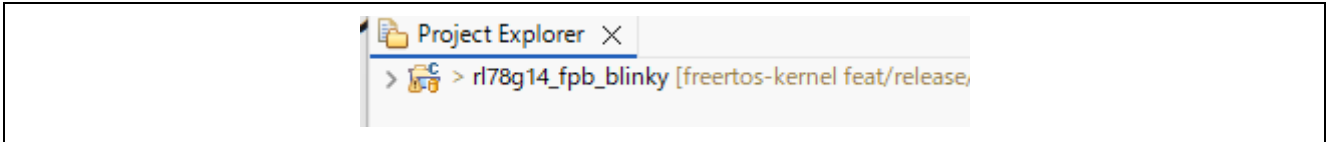


Figure 5-1 Importing the rl78g14_fpb_blinky Project into the e² studio

5.1.2 Building the Project

Build the project.

Select the project, then right-click on the project name > Build Project

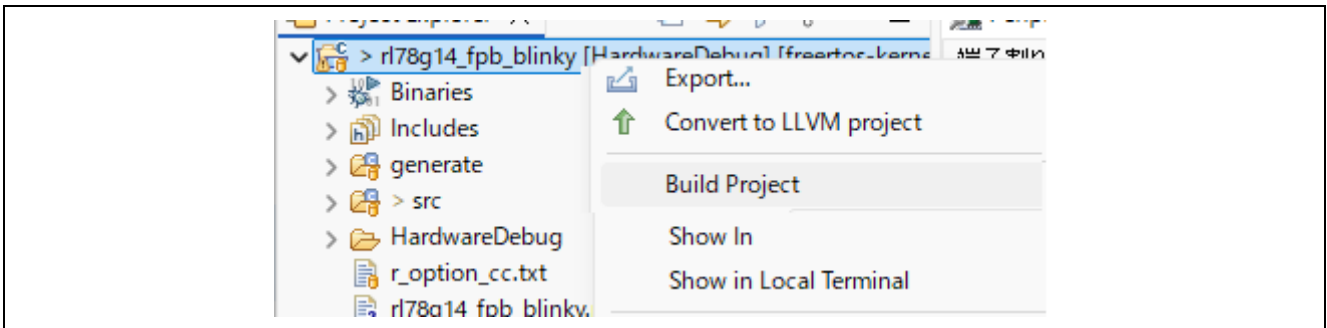


Figure 5-2 Building the rl78g14_fpb_blinky Project

5.2 RL78/G23-128p FPB

5.2.1 Importing the Project

Import the rl78g23_fpb_blinky project into the e² studio.

File > Import > Existing Projects into Workspace > Select the rl78g23_fpb_blinky project.



Figure 5-3 Importing the rl78g23_fpb_blinky Project into the e² studio

5.2.2 Building the Project

Build the project.

Select the project, then right-click on the project name > Build Project

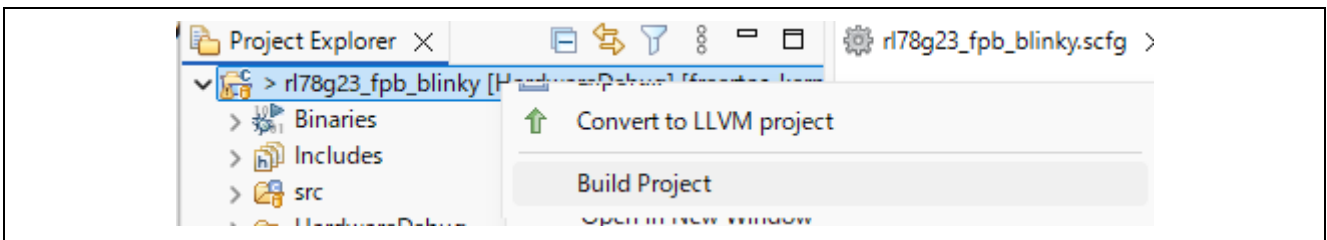


Figure 5-4 Building the rl78g23_fpb_blinky Project

6. Running the FreeRTOS Blinky Demo

6.1 RL78/G14 FPB

6.1.1 Power Supply to the Microcontroller Board

Connect the microcontroller board and the PC via a USB cable.

6.1.2 Debug Configuration

Select the configuration to be executed.

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

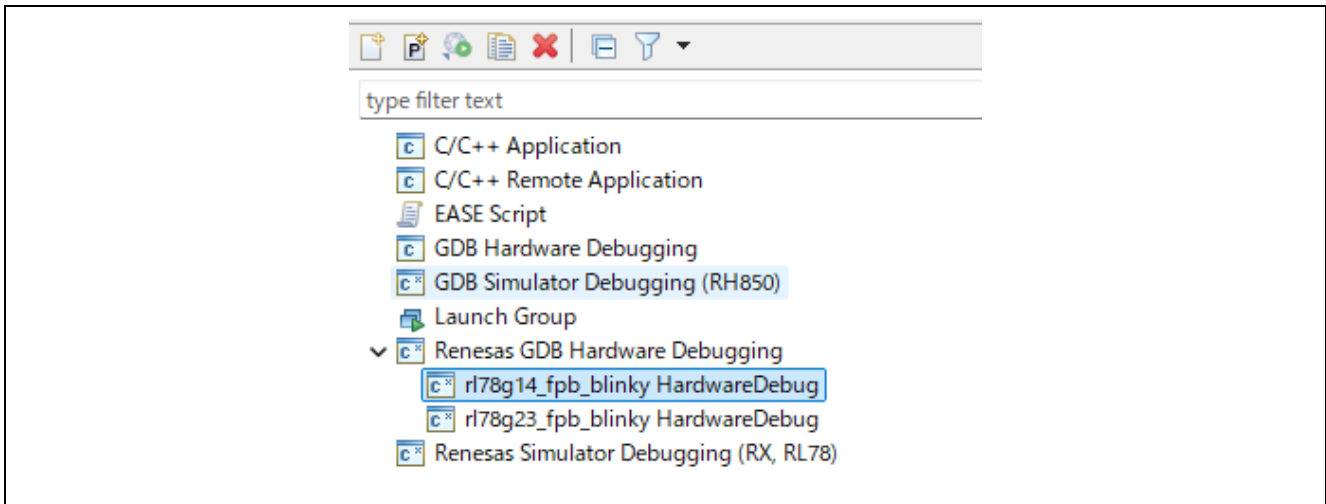


Figure 6-1 Debug Configuration for the rl78g14_fpb_blinky Project

6.1.3 Settings for the Debugger

Select the Debugger tab.

Debug hardware: E2 Lite (RL78)

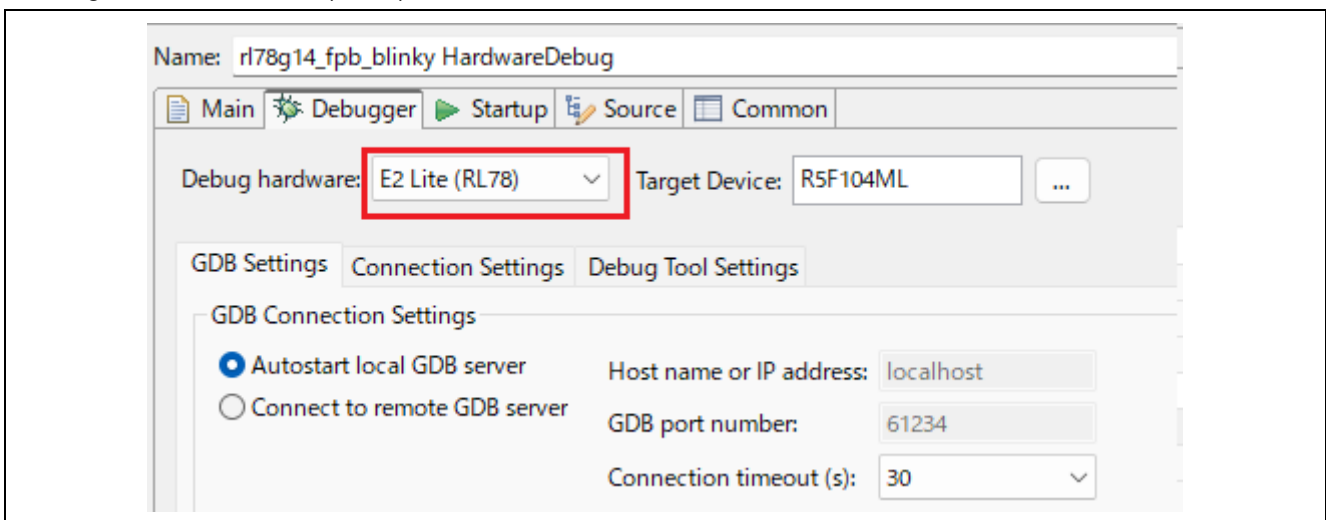


Figure 6-2 Debug hardware: E2 Lite (RL78)

Select the Connection Settings tab > Connection with Target Board.

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

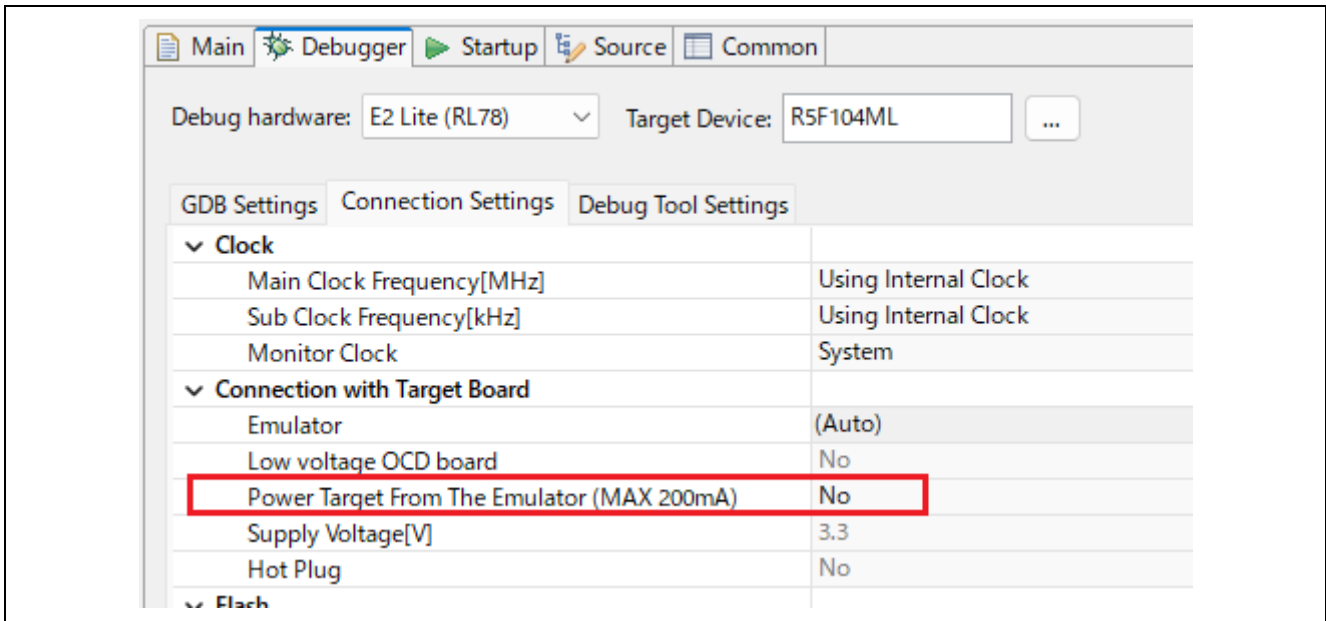



Figure 6-3 Connection Settings for Using an Emulator

Click on  to start debugging.

6.1.4 Running the FreeRTOS Blinky Demo

Click on  and then confirm that execution stops in main().

```


#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();
... }
}
    
```

Figure 6-4 Debug hardware: E2 Lite (RL78)

Click on  again. LED0 and LED1 on the board start blinking, respectively every 1 and 2 seconds.

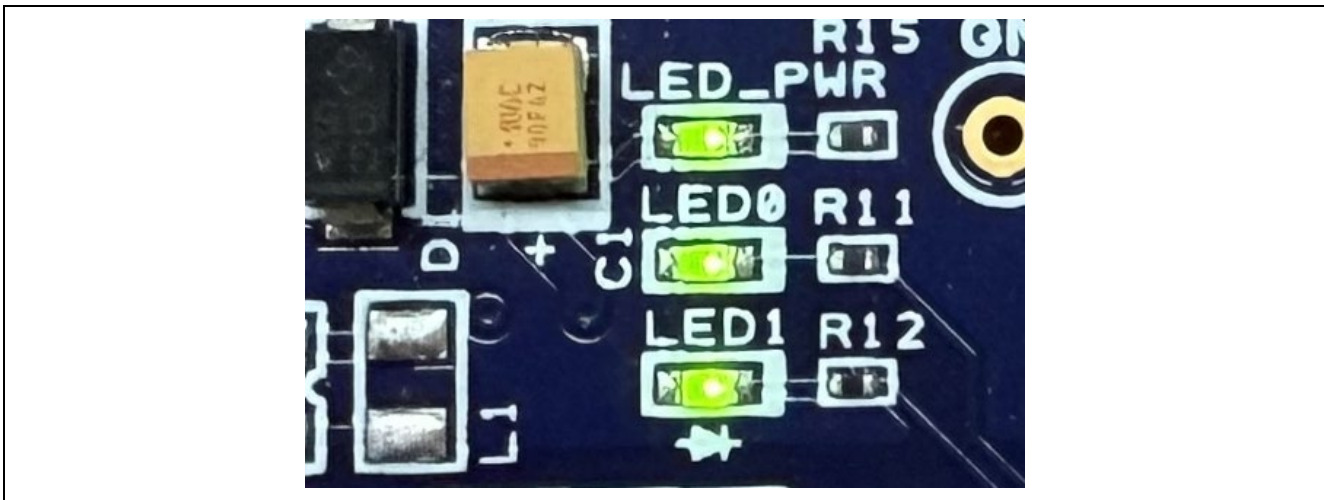


Figure 6-5 LEDs on the Board Blinking with r178g14_fpb_blinky

6.2 RL78/G23-128p FPB

6.2.1 For Using a COM Port

6.2.1.1 Setting Jumper Pins

If any changes to the circuit have been made, make the following settings.

J15, J16, and J19: Pins 1-2 are short-circuit

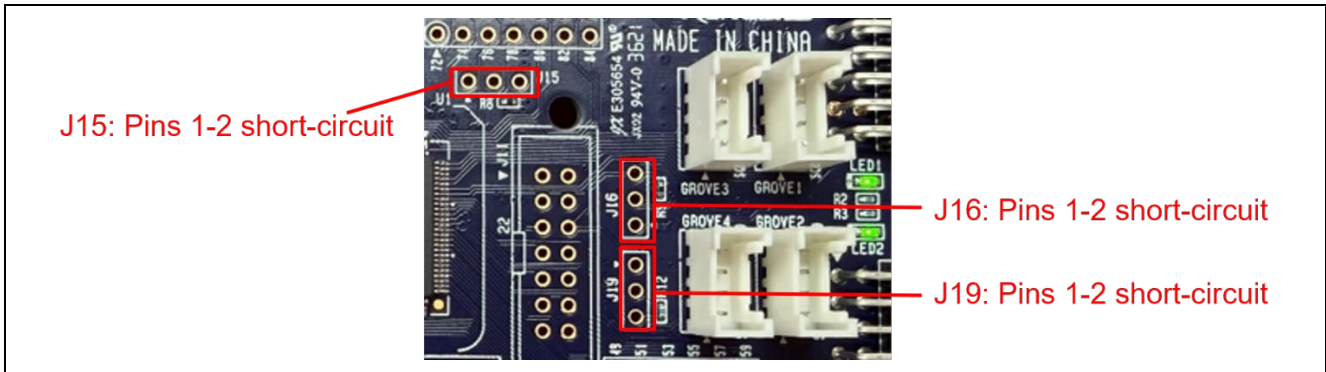


Figure 6-6 Settings for Use with COM Port Debugging on the Top Side

6.2.1.2 Power Supply to the Microcontroller Board

Connect the microcontroller board and the PC via a USB cable.

6.2.1.3 Debug Configuration

Select the configuration to be executed.

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

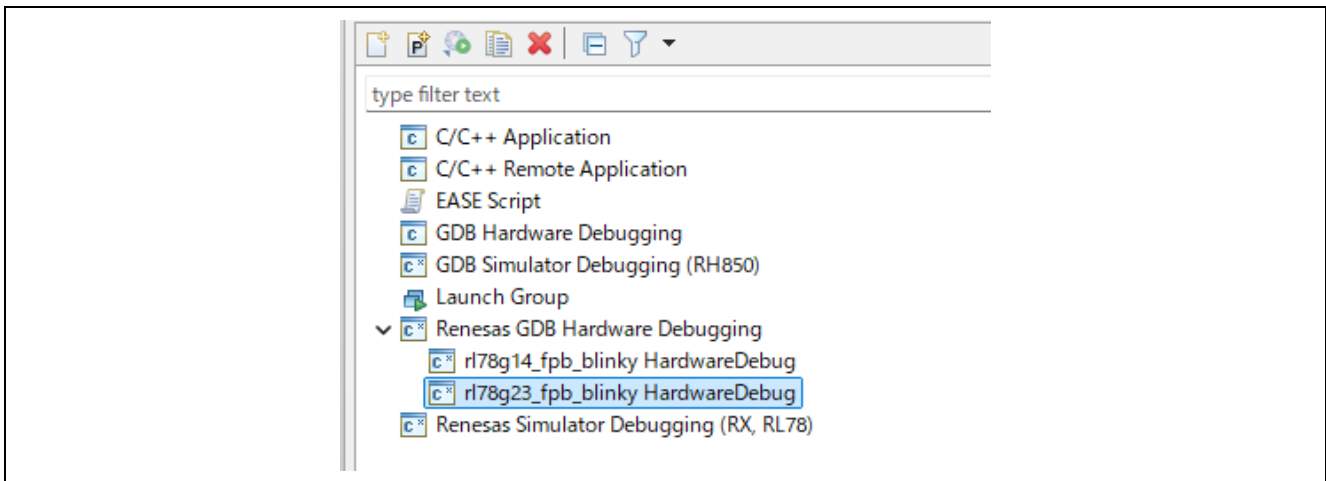


Figure 6-7 Debug Configuration for the rl78g23_fpb_blinky Project

6.2.1.4 Settings for the Debugger

Select the Debugger tab.

Debug hardware: COM Port (RL78)

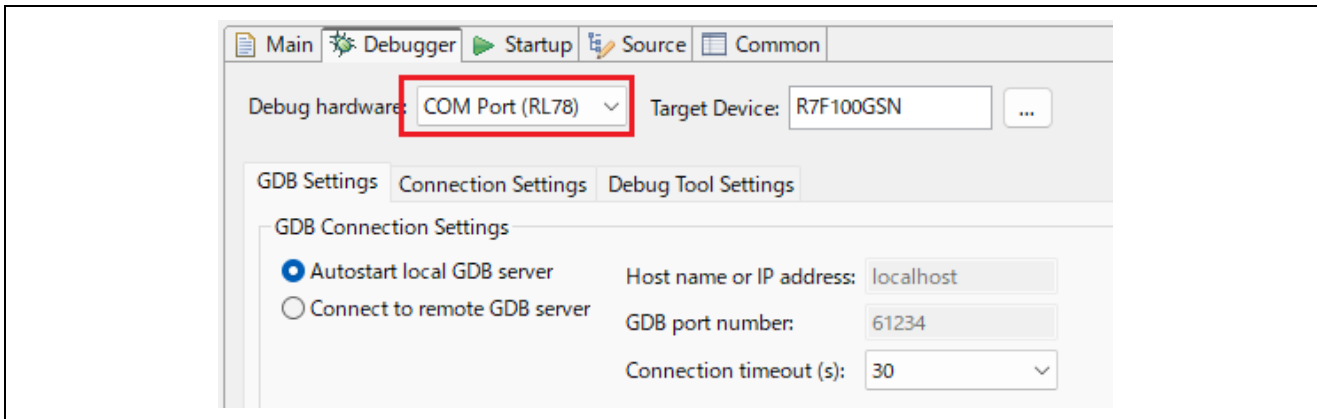


Figure 6-8 Debug hardware: COM Port (RL78)

Select the Connection Settings tab > Connection with Target Board. As the COM Port, select the one connected to the microcontroller board.

COM Port: COMxx

Reset control pin: DTR

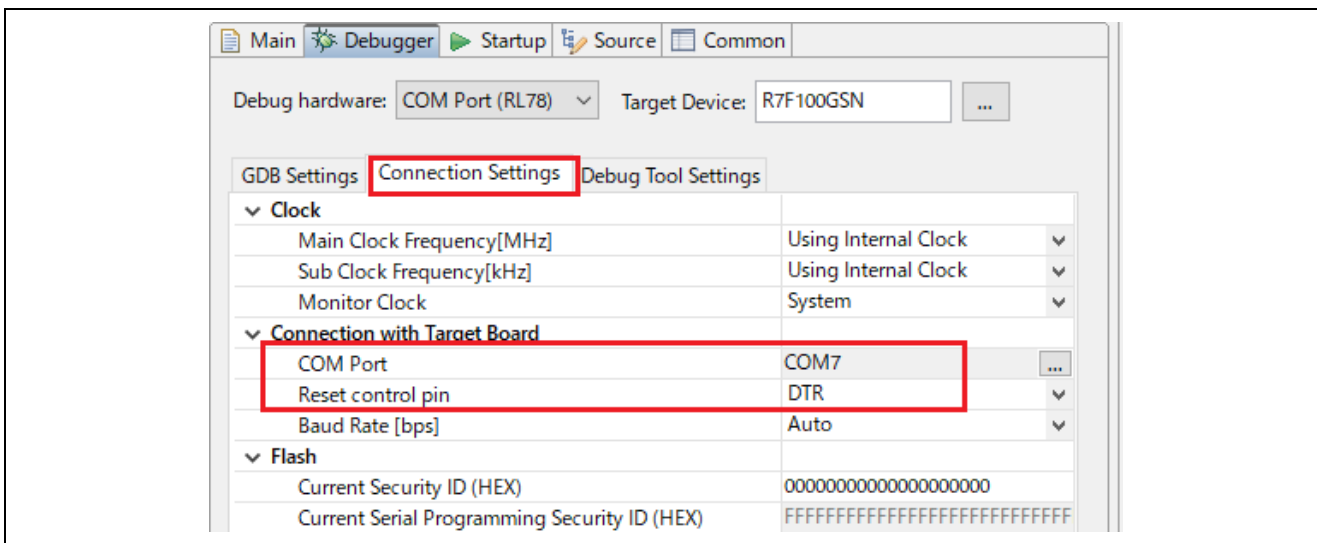


Figure 6-9 Connection Settings for Using a COM Port

Click on  to start debugging.

6.2.2 For Using an Emulator

6.2.2.1 Mounting a Connector, Setting Jumper Pins, and Patterns for Cutting

A 14-pin connector (on J11) is used to connect this product to an on-chip debugging E2 emulator or E2 emulator Lite, from Renesas Electronics, each of which also incorporates programming facilities. Note that the connector is not mounted on the board as shipped. The emulator is used for programming or debugging the evaluation MCU. To connect the emulator, change the circuit as follows. For details, refer to section 5.20 in the [RL78/G23-128p Fast Prototyping Board User's Manual](#).

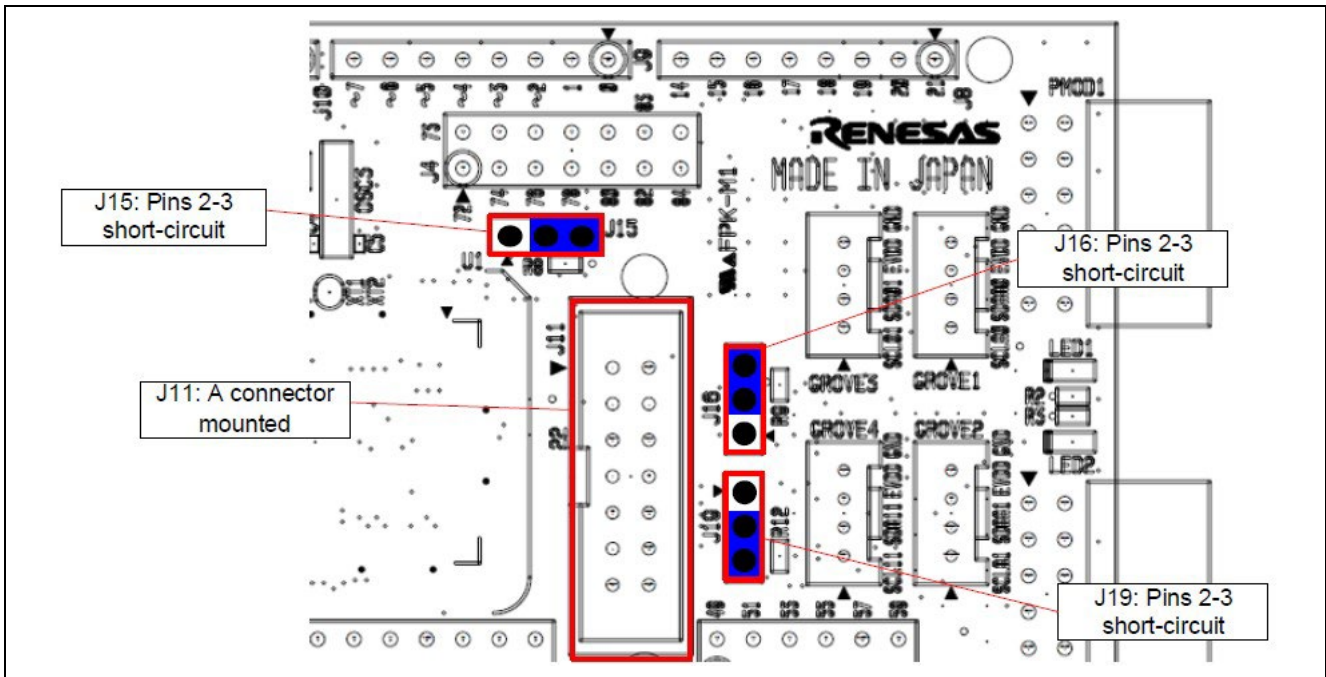


Figure 6-10 Settings for Use with the Emulator Connector on the Top Side

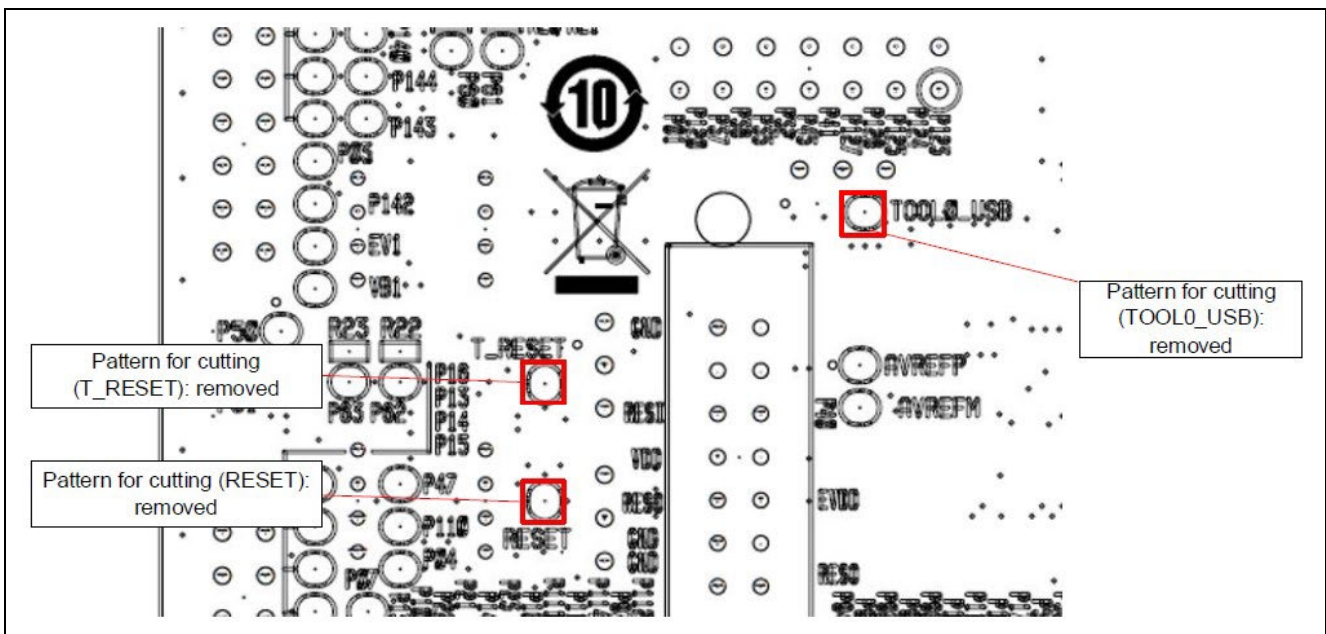


Figure 6-11 Settings for Use with the Emulator Connector on the Soldered Side

For the usage of the selected emulator, refer to the [E1/E20/E2 Emulator, E2 emulator Lite Additional Document for User's Manual \(Notes on Connection of RL78\)](#) (R20UT1994).

6.2.2.2 Connecting an Emulator to the Microcontroller Board

Connect the emulator to be used as shown below.

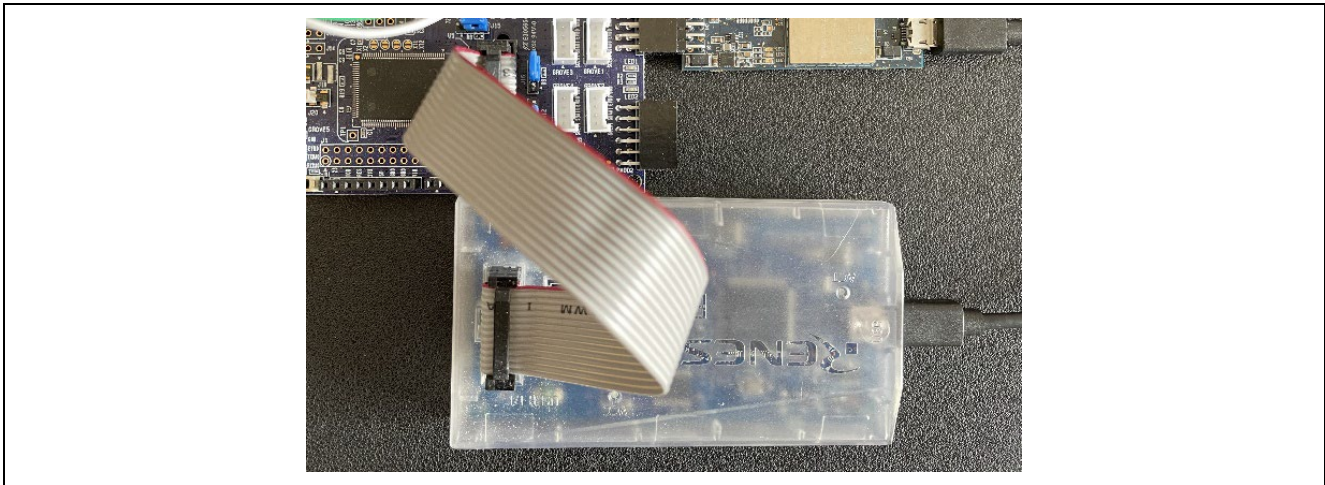


Figure 6-12 Connecting an Emulator to the Microcontroller Board

6.2.2.3 Debug Configuration

Select the configuration to be executed.

Debug Configurations > Renesas GDB Hardware Debugging > rl78g23_fpb_blinky HardwareDebug

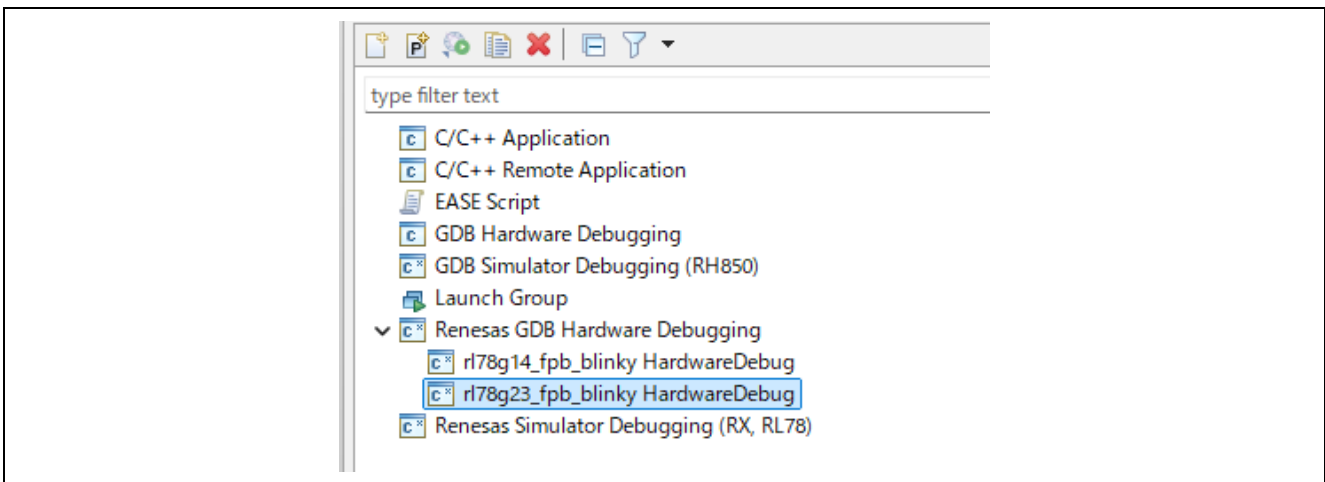


Figure 6-13 Debug Configuration for the rl78g23_fpb_blinky Project

6.2.2.4 Settings for the Debugger

Select the Debugger tab.

Debug hardware: E2 Lite (RL78)

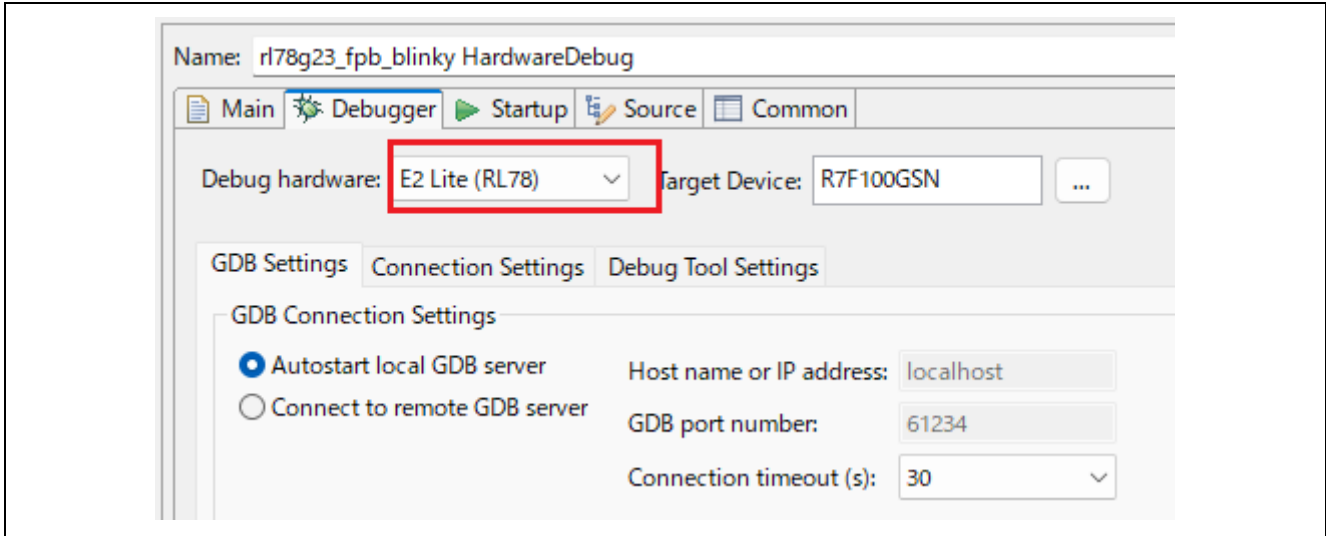


Figure 6-14 Debug hardware: E2 Lite (RL78)

Select the Connection Settings tab > Connection with Target Board.

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

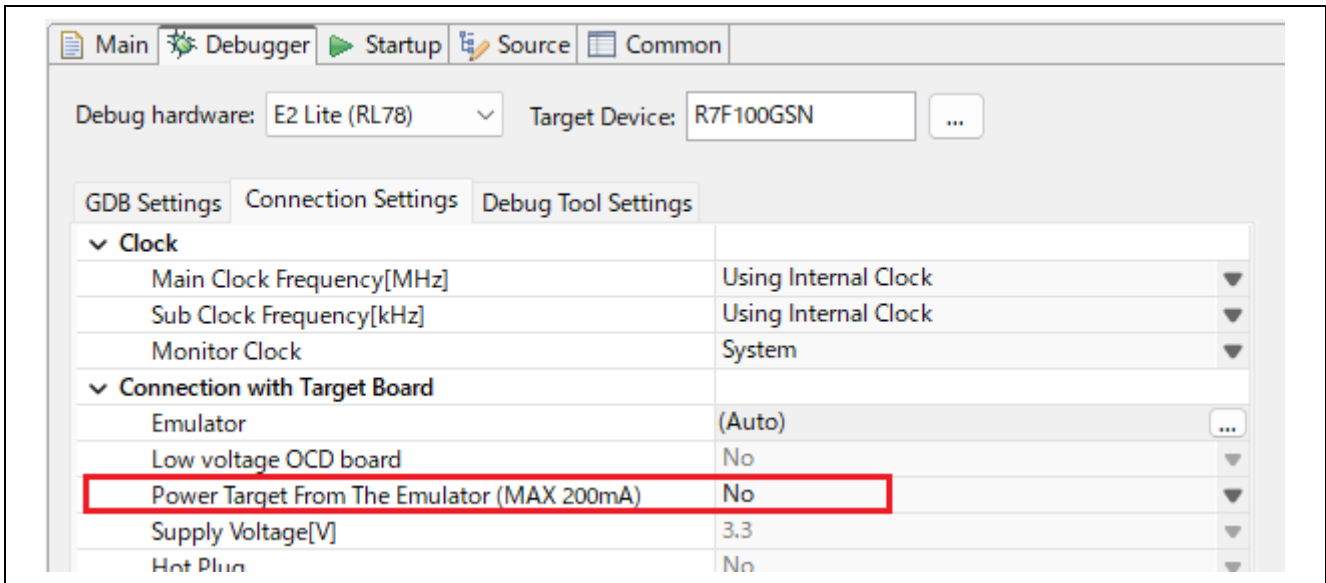
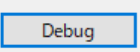



Figure 6-15 Connection Settings for Using an Emulator


Click on  to start debugging.

6.2.3 Running the FreeRTOS Blinky Demo

Click on  and then confirm that execution stops in main().

```
#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();  
    }  
}
```

Figure 6-16 Debug hardware: E2 Lite (RL78)

Click on  again. LED1 and LED2 on the board start blinking, respectively every 1 and 2 seconds.

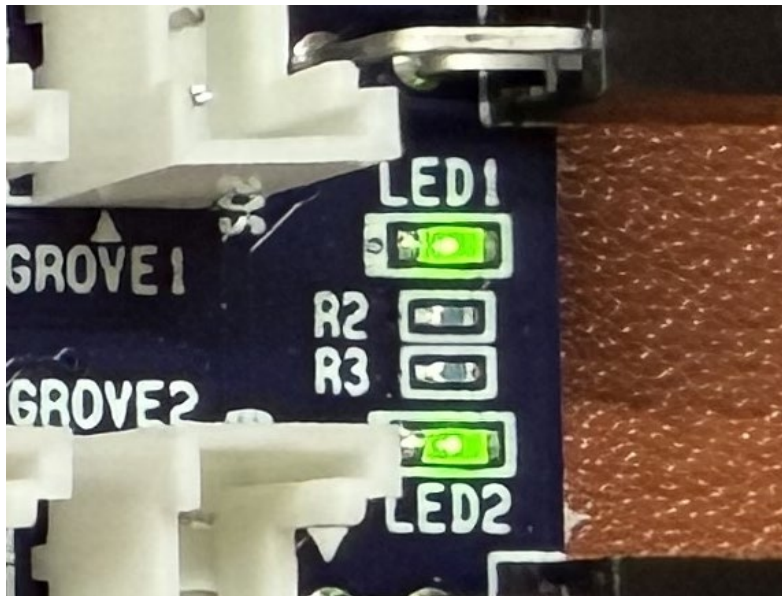


Figure 6-17 LEDs on the Board Blinking with rl78g23_fpb_blinky

7. Supplementary Information

7.1 List of Renesas Functions

Table 7-1 List of Renesas Functions

Function Name	Description
vApplicationSetupTimerInterrupt	This function is executed when the scheduler is started and is for executing the initial settings of the system timer.
vAssertCalled	This function is executed when necessary conditions are not met in assertion functions used for checking parameters and so on. Its execution results in entry into a critical section, and then to an infinite loop.
Processing_Before_Start_Kernel	This function is a Renesas-original function which is executed before the kernel scheduler is started. The function can be used as required for generating RTOS objects such as tasks, mailboxes, or semaphores before the scheduler is started.

7.2 Usage Note

The following notes apply to FreeRTOS for CC-RL.

Switching between register banks (the memory space allocated to the general-purpose registers of RL78 devices) is not supported.

The [interrupt nesting](#) is not supported.

Because the task switch request is implemented using the software interrupt, the task switch process is executed immediately after the request. Therefore, the following restrictions apply to user application:

- Within an ISR, request a task switch (execute the portYIELD_FROM_ISR macro) at the end of the ISR process. For details on this restriction and possible workarounds, refer to GitHub [issue ticket #21](#).
- Do not request a task switch within a critical section.

In accord with a specification for memory allocation by CC-RL, when the size of a near pointer is extended from 2 bytes to 4 bytes, the third byte from the least significant is set to 0x0f. For details on the specification, refer to section 4.1.3 Internal representation and value area of data, and "Specifying memory allocation area (__near / __far)", in the [CC-RL User's Manual](#).

- An example of a result of this specification is that when the pvParameters argument in the task entry function is cast to 4 bytes in the task, the resulting value does not match the value specified as the pvParameters argument in xTaskCreate().
- Therefore, the above specification must be considered when using an API function with near pointers, such as xTaskCreate(). As a more specific example, if the pvParameters argument is to be used as a 4-byte integer in a task, the user application requires a countermeasure such as casting the value referred to by the argument twice, firstly to an int type 2-byte integer, and then to a long type 4-byte integer.

8. Website and Support

- For support on FreeRTOS, visit the official FreeRTOS website.
<https://freertos.org/Why-FreeRTOS/Support-options>
- For support on the Blinky demo, visit our contact web page.
<https://www.renesas.com/support>

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jan. 16, 2026	-	First edition issued

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity.

Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on

The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state

Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

4. Handling of unused pins

Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

5. Clock signals

After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

6. Voltage application waveform at input pin

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).

7. Prohibition of access to reserved addresses

Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.