RL78 Family Real-Time OS RI78V4 V2.00.00
Release Note

R20UT3404EJ0102
Rev.1.02
2021.04.28

Contents

1. Packaged Tools

Agreement type and contents are different according to the product.

- Trial

Product Name	Agreement Type
RTRRL7800TR01ERRZZ	Trial License, Limited 1 host

The provided tools are" Real-Time OS RI78V4 Kernel Object for trial (with restriction) " and "Command-line Configurator "CF78V4". These tools can be downloaded from a website of the Renesas Electronics.

And if you use the plug-in for "CS+ for CC", please download from a website of the Renesas Electronics.

- Evaluation License, Mass-production License

Product Name	Agreement Type	Contents
RTRRL7800TR01ERR	Evaluation License, Limited 1 host	A
RTRRL7800TR01ERRLU	Evaluation License, Unlimited hosts	A
RTRRL7800TR01RRRUL	Mass-production License, 3000 copies	A
RTRRL7800TR01RRRUU	Mass-production License, Unlimited copies	A
RTRRL7800TR01SRRUU	Mass-production License, Unlimited copies, With source code	B

The following tools are provided.

Contents		Tool Name	Version No.
B	A	Real-Time OS RI78V4 Kernel Object	V2.00.00
		Command-line Configurator CF78V4	V2.01.00.01
		Plug-ins for CS+ for CC	
		Realtime OS Build Tool Plug-in (Common)	V3.01.00.01
		Realtime OS Build Tool Plug-in (RI78V4)	V1.00.00.04
		Realtime OS Analysis Control Plug-in (Common)	V3.00.00.03
		Realtime OS Analysis Control Plug-in (µITRON4)	V3.01.00.01
		Realtime OS Analysis Control Plug-in (RI78V4)	V1.00.00.04
		Realtime OS Resource Information Displaying Plug-in (Common)	V3.01.00.01
		Realtime OS Resource Information Displaying Plug-in (µITRON4)	V3.01.00.01
		Realtime OS Task Analyzer Plug-in (Common)	V3.00.01.01
		Realtime OS Task Analyzer Plug-in (Panel)	V3.00.00.03
		Realtime OS Task Analyzer Plug-in (RI78V4)	V1.00.00.03
		Real-Time OS RI78V4 Kernel Source	V2.00.00

2. User’s Manual

The following user's manuals are included with this version. Please read these manuals together with this document.

Manual Name	Document Number
RI Series Real-Time Operating System User's Manual: Start	R20UT0751EJ0104
RI78V4 V2.00.00 Real-Time Operating System User's Manual: Coding	R20UT3375EJ0100
RI78V4 V2.00.00 Real-Time Operating System User's Manual: Debug	R20UT3374EJ0100
RI78V4 V2.00.00 Real-Time Operating System User's Manual: Analysis	R20UT3373EJ0100
RI Series Real-Time Operating System User's Manual: Message	R20UT0756EJ0104

These PDF files are provided in this package and are available for downloading at the Renesas Electronics World-Wide Web site.

3. Target Devices

The following devices are supported by the product.

- RL78 Family (S2 core, S3 core)

4. Operating Environment

The operating environment for this product is described below.

4.1. Hardware environment

The following hardware environments are supported.

- Processor: At least 1 GHz (support for hyper threading/multicore CPU)
- Main memory: At least 1 GB (2 GB or higher for Windows 10 and other 64-bit editions of Windows OS),
  2 GB or higher recommended
- Display: Resolution at least 1,204 x 768; at least 65,536 colors

4.2. Software environment

The following software environments are supported.

- Windows 8.1 (32-bit and 64-bit editions)
- Windows 10 (32-bit and 64-bit editions)
- Microsoft .NET Framework 4.5.2
- Microsoft Visual C++ 2015 Update 3 runtime libraries (x86)

Remark For any of these, we recommend having the latest service pack installed.

4.3. Support Tools

Tool Name	Manufacturer	Version
CS+ integrated development environment	Renesas Electronics	V8.05.00 or later
CC-RL C Compiler	Renesas Electronics	V1.10.00 or later

5. Installation Cautions

This section provides cautions for installation and uninstallation.

5.1. Cautions for Installation

5.1.1. Cautions for administrator privileges

Windows® administrator privileges are required to install the software.

5.1.2. Cautions for execution environment

The .NET Framework and the Visual C++ runtime libraries are required to run the installer.

5.1.3. Cautions for network drives

The software cannot be installed from a network drive. It also cannot be installed to a network drive.

5.1.4. Cautions for installation folder name

The available characters for specifying the installation folder are the same as for Windows®.

The 11 characters / * : < > ? | " \ ; , cannot be used. Folder names also cannot start or end with a space.

Specify folders as absolute paths. Do not use relative paths.

Use the backslash character (\) as the path separator for the installation folder. Do not use the forward slash (/).

5.1.5. Cautions for modifying and repairing functions

To modify or repair the function of a tool that has already been installed, have the tool's installer package on hand, and run the installation program. The program maintenance program will start; select Modify or Repair.

Uninstall or change a program dialog boxes will cause an error.

5.1.6. Cautions for required files after installation

The following folder is created after installation. Do not delete it, because it contains files that are necessary for the tools to run.

- If Windows® is 32bit and the installation drive is C:
  `C:\Program Files\Common Files\Renesas Electronics CubeSuite+\`
- If Windows® is 64bit and the installation drive is C:
  `C:\Program Files (x86)\Common Files\Renesas Electronics CubeSuite+\`

5.1.7. Cautions for version of installed tools

If the newer version tool is already installed, the older version tool may not be installed.

5.1.8. Cautions for starting installer

If the installer is started on a non-Japanese version of Windows®, then if the path contains multi-byte characters it will cause an error, and the installer will not start.

5.1.9. Enable Plug-ins

Plug-ins of this product may be disabled immediately after installation of this product. Please enable Plug-ins of this product. For details, refer to ”9.2 Enable Real-Time OS Plug-ins”.

5.2. Cautions for Uninstallation

5.2.1. Cautions for administrator privileges

Windows® administrator privileges are required to uninstall the software.

5.2.2. Cautions for uninstallation folder name

Depending on the order in which tools are uninstalled, the folders may not be completely deleted. If this happens, remove any remaining folders via Explorer or the like.

5.2.3. Cautions for adding/repairing via other than the installer

If you added or modified files to the folders in which tools and manuals were installed using other means than the installers, they cannot be deleted during uninstallation.

5.2.4. Key Word for Uninstallation

There are two ways to uninstall this product.

- Use the integrated uninstaller (uninstalls CS+ for CC)
- Use separate uninstaller (uninstalls this product only)

To use the separate uninstaller, select the following from the Control Panel:

- Programs and Features

After the applet appears, delete the followings.

- CS+ Realtime OS Common Plugins
- CS+ Realtime OS RI78V4 Plugins
- CS+ Realtime OS RI78V4 Object Release or CS+ Realtime OS RI78V4 Source Release

6. Changes from Previous Version (V1.00.02)

6.1. The kernel version information

Keyword	Before	After
TKERNEL_PRVER	0x0102	0x0200

6.2. Support C Compiler

In RI78V4 V1.00.02, the support C Compile is CA78K0R, in RI78V4 V2.00.00, the support C Compile is CC-RL.

6.3. Target Devices

In RI78V4 V1.00.02, the target device are 78K0R microcontrollers and RL78 Family, in RI78V4 V2.00.00, the target device in only RL78 Family.

6.4. Changes the supporting memory model

It is large model supporting only in RI78V4 V1.00.02, but RI78V4 2.00.00 is medium model supporting only because CC-RL is not support large model, support medium model and small model. Further, it can be run the small model application with RI78V4 V2.00.00.

6.5. Changes in the way to write hexadecimal in an assembly language

In RI78V4 V1.00.02 the way to write hexadecimal is suffix form (e.g. 012ab), in RI78V4 V2.00.00 the way to write hexadecimal is prefix form (e.g. 0x12ab).

6.6. The structure packing function can be use

In RI78V4 V1.00.02 the structure packing function (-rc option of CA78K0R) cannot be used, but in RI78V4 V2.00.00 it can be used.

6.7. Changes in the kernel initialization address

V1.00.02	V2.00.00
__urx_start	__kernel_start

6.8. Changes in the section name and addition with the section name

The memory area name of RI78V4 is changed (In V1.00.02, the memory area is called “segment”, In V2.00.00 the memory area is called “section”). The memory area adds newly in V2.00.00 for function substantiality.

A comparison table of the memory areas name in V1.00.02 and V2.00.00 is as follows.

V1.00.02	V2.00.00
k_system	.kernel_system
k_info	.kernel_info
k_const	.kernel_const／.kernel_const_f
k_data	.kernel_data
k_stack	.kernel_stack
k_work0, k_work1, k_work2, k_work3	.kernel_work0, .kernel_work1, .kernel_work2, .kernel_work3

The memory areas name added in V2.00.00 and something to allocate in the area is as follows.

Area name	Description
.kernel_system_timer_n	The area for system timer and FAR branch information.
.kernel_data_init	The area for Initialization data of the kernel.
.kernel_data_trace_n	The area for trace data.
.kernel_const_trace_f	The data area for trace taking.
.kernel_system_trace_f	The code area for trace taking.
.kernel_sbss	The SADDR area for the kernel. Note

Note There are SADDR area where RI78V4 V1.00.02 is used, but its area is made a section in RI78V4 V2.00.00.

6.9. Adds the data queues function

In RI78V4 V2.00.00, adds the data queues function. The following service call is added.

Service call name	Function
snd_dtq	Send to data queue
psnd_dtq	Send to data queue (polling)
ipsnd_dtq	Send to data queue (polling, non-task context)
tsnd_dtq	Send to data queue (with timeout)
fsnd_dtq	Forced send to data queue
ifsnd_dtq	Forced send to data queue (non-task context)
rcv_dtq	Receive from data queue (waiting forever)
prcv_dtq	Receive from data queue (polling)
trcv_dtq	Receive from data queue (with timeout)
ref_dtq	Reference data queue state

The definition of the following data queue is added in the system configuration file.

```
CRE_DTQ (
    dtqid  {                    // ID
        dtqatr,                 // Attribute
        dtqcnt [:sec_nam] ,     // Data count, memory area name
        dtq                     // Reserved for future use
    }
)
```

6.10. Adds the PHS attribution for cyclic handler

In RI78V4 V2.00.00, adds the PHS attribution (activation phase) for cyclic handler. This attribution is set in the system configuration file.

```
CRE_CYC (
    cycid  {            // ID
        cycatr,         // Attribute
        exinf,          // Reserved for future use
        cychdr,         // Start address
        cyctim,         // Activation cycle
        cycphs          // Activation phase
    }
)
```

6.11. Improvement of the way to define and write the interrupt handler

- In RI78V4 V1.00.02, the way to define and write the interrupt handler

[C source file]

```
#include “kernel_id.h”
#pragma rtos_interrupt  INTP0 intp0_hdr

void
intp0_hdr(void){
    [The body of interrupt handler]
    return;
}
```

The user has to describe the “#pragma rtos_interrupt” directive and interrupt handler.

- In RI78V4 V2.00.00, the way to define and write the interrupt handler

[System configuration file]

```
DEF_INH (
    INTP0 {
        TA_HLNG,
        intp0_hdr
    }
)
```

[C source file]

```
#include "kernel_id.h"

void
intp0_hdr(void){
    [The body of interrupt handler]
    return;
}
```

By defining the interrupt handler in the system configuration file, the configurator (CF78V4) outputs the description "#pragma rtos_interrupt" directive to the file "kernel_id.h". Therefore the user describes inclusion of "kernel_id.h" and the body of interrupt handler.

If the FAR attribute is specified in the definition of the interrupt handler in the system configuration file and "__far" is specified to the interrupt handler, the interrupt handler is allocated to the FAR area. In this case, the configurator (CF78V4) outputs an interrupt vector and a middle point that located during an interrupt vector to an interrupt handler (the branch code which is from a vector and which is to an interrupt handler) automatically.

[System configuration file]

```
DEF_INH (
    INTP0 {
        TA_HLNG | TA_FAR,
        intp0_hdr
    }
)
```

[C source file]

```
#include "kernel_id.h"

__far void
intp0_hdr(void){
    [The body of interrupt handler]
    return;
}
```

6.12. The way to define the timer handler

The user has to describe the “timer interrupt handler” to call a timer handler in RI78V4 V1.00.02, it is not necessary to describe in V2.00.00. But in V2.00.00, it is necessary to define “Clock timer exception code” in the system configuration file.

- In RI78V4 V1.00.02, the way to define the timer handler

```
#pragma rtos_interrupt  INTTM00 int_timer

void
int_timer(void){

    Timer_Handler();
    return;
}
```

The user has to describe the “#pragma rtos_interrupt” directive and interrupt handler.

- In RI78V4 V2.00.00, the way to define the timer handler

```
[ System configuration file ]

CLK_INTNO (INTTM00)          // When the interrupt factor for time is INTTM00
```

By defining the interrupt handler in the system configuration file, the configurator (CF78V4) outputs the description “#pragma rtos_interrupt” directive to the file “kernel_id.h” and outputs the timer interrupt handler to the interrupt information definition file.

6.13. Related to the configurator (CF78V4)

6.13.1. Changes in the configurator (CF78V4)

(1) [Addition] Adds –cpu option

Specifies the device file name

[e.g.]

```
-cpu R5F10A6A
```

When using CS+, this option is used automatically

(2) [Addition] Adds –devpath option

Specifies the path which is existed the device file.

[e.g.]

```
-devpath="C:\Program Files\Renesas Electronics\CS+\CC\Device\RL78\Devicefile"
```

When using CS+, this option is used automatically

6.13.2. Changes in the description of system configuration file

(1) [Addition] Clock timer exception code

Specifies the timer interrupt for RI78V4.

[e.g.]

```
CLK_INTNO ( INTTM00 );
```

(2) [Addition] Data queues information

Specifies information of the data queues.

(3) [Addition] Interrupt handler information

Specifies information of the interrupt handler.

(4) [Addition] PHS attributes of cyclic handler

The PHS attribute is addition as the attribute of the cyclic handler.

(5) [Changes] The number of characters which can be specified as the ID

The number of characters which can be specified as the “ID” is changed within 255 character from 24 character.

(6) [Changes] The number of characters which can be specified as the symbol name and object name

The number of characters which can be specified as the symbol name and object name is changed within 4095 character from 30 character.

6.14. Realtime OS Task Analyzer Plug-in

In RI78V4 V1.00.02, AZ78V4 is provided for the analysis of Real-Time OS applications. In RI78V4 V2.00.00, AZ850V4 is replaced by a new feature called the Realtime OS Task Analyzer.

Figure 6-1 Image of AZ78V4

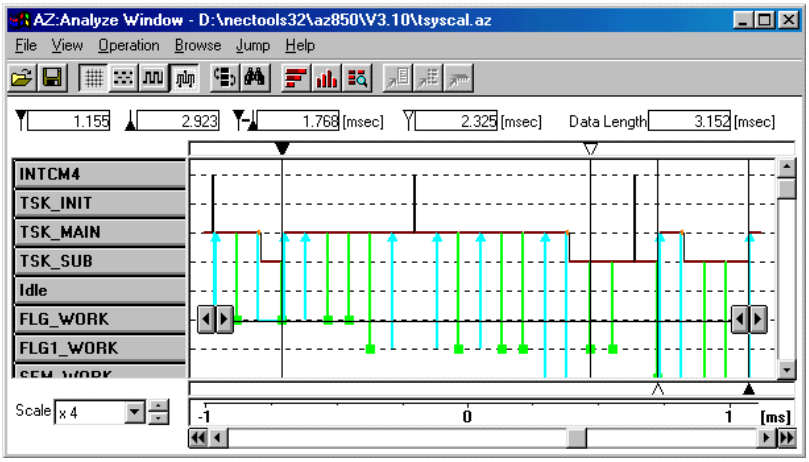
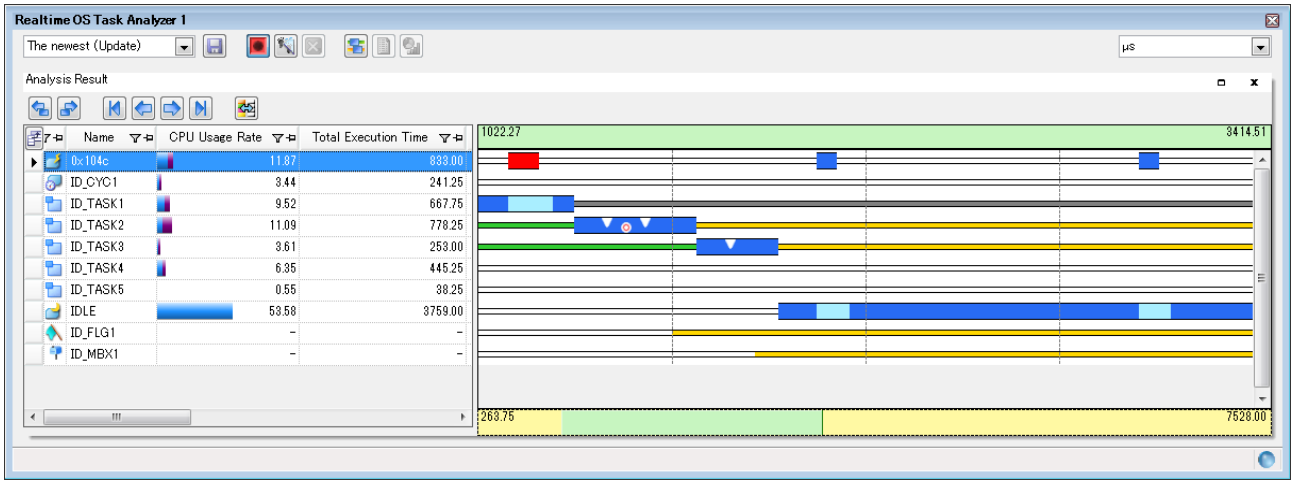
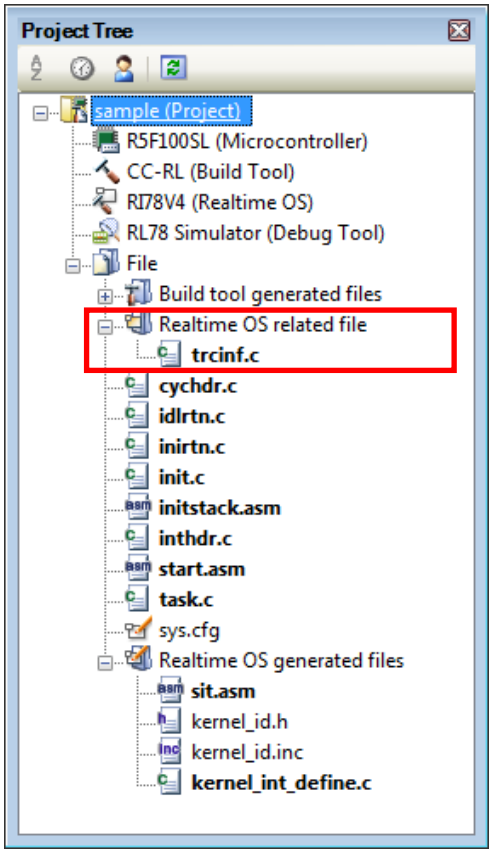


Figure 6-2 Image of Realtime OS Task Analyzer



With the provision of this new feature, we have added a [Realtime OS related file] category to the project tree. The trace information file (trcinf.c) will be registered in the [Realtime OS related file] category. This is a read-only file that is not to be edited by the user but needs to be built together with the application.

Figure 6-3 Project Tree



The header file “usrown.h” is also required as information on system dependencies. This file is included in the sample project (see the following folder).

```
C:\Program Files\Renesas Electronics\CS+\SampleProjects\RL78\RL78_G13_RI78V4_C_Lang\appli\include
```

The address of the counter register to be used by the system timer and time taken to count up by one are defined as macros in the “usrown.h” file. For details, refer to RI78V4 V2.00.00 Real-Time Operating System User’s Manual: Coding.

7. When it is converted from RI78V4 V1.00.02 to RI78V4 V2.00.00

The change point when converting from RI78V4 V1.00.02 application to RI78V4 V2.00.00 application is shown in the following.

When a project of application with RI78V4 V1.00.02 is created by CubeSuite+ (CS+), if the project of application with the RI78V4 V1.00.02 reads in the environment installed RI78V4 V2.00.00, the possible part changes a change automatically.

7.1. The timer interrupt handler

The user has to describe the “timer interrupt handler” in RI78V4 V1.00.02, it is not necessary to describe in V2.00.00. Therefore you delete the timer interrupt handler, and specifies the clock timer exception code (CLK_INTNO) in the system configuration file.

■ V1.00.02 [C source file]	■ V2.00.00 [System configuration file]

`#pragma rtos_interrupt INTP0 timer`
 	
`void`
`timer(void){`
`  TimerHandler();`
`  return;`
`}`
	`CLK_INTNO ( INTTM00 );`

Then CS+ is to read a project of application with RI78V4 V1.00.02, and a system configuration file is converted. But a comment is being output by configurator so that this item may be designated because used timer interrupt cannot be specified for this item,

7.2. Definition of the interrupt handler

In RI78V4 V1.00.02, the interrupt handler is not defined in the system configuration file. In RI78V4 V2.00.00, it is necessary to define the interrupt handler in the system configuration file.

Uses the “interrupt factor and address of interrupt handler” which is specified by “#pragma rtos_interrupt” directive by application with RI78V4 V1.00.02, and defines with “DEF_INH” in the system configuration file.

And deletes the “#pragma rtos_interrupt” directive in RI78V4 V1.00.02,

In the CS+, if a project of application with RI78V4 V1.00.02 reads, this item is not convert automatically.

■ V1.00.02

[C source file]

```
#pragma rtos_interrupt INTP0 intp0_hdr

void
intp0_hdr(void){
       [The body of interrupt handler]
       return;
}
```

■ V2.00.00

[System configuration file]

```
DEF_INH (
    INTP0 {
        TA_HLNG,
        intp0_hdr
    }
)
```

[C source file]

```
#include “kernel_id.h”

void
intp0_hdr(void){
       [The body of interrupt handler]
       return;
}
```

7.3. The item of definition of cyclic handler

In RI78V4 V2.00.00, saves the “activation phase” of the cyclic handler. It is necessary to specify the “activation phase” of cyclic handler which is the reservation item of CRE_CYC. In RI78V4 V1.00.02, it is necessary to specify “0”, but it is error to specify “0” in RI78V4 V2.00.00. Specifies except for “0” (The activation cycle may be input).

In the CS+, if a project of application with RI78V4 V1.00.02 reads, this item is not convert automatically.

■ V1.00.02

[System configuration file]

```
CRE_CYC (
    ID_CYC1 {
        TA_HLNG,
        0,
        cycadr
        0x1000,
        0x0
    }
)
```

■ V2.00.00

[System configuration file]

```
CRE_CYC (
    ID_CYC1 {
        TA_HLNG,
        0,
        cycadr
        0x1000,
        0x1000
    }
)
```

7.4. “__far” keyword for the argument of the service call

In the application with RI78V4 V1.00.02, when the “__far” keyword is added to the argument of the service call, it is necessary to change the “__far” keyword to the “__near” keyword or it is necessary to delete “__far” keyword in the application with RI78V4 V2.00.00.

In the CS+, if a project of application with RI78V4 V1.00.02 reads, this item is not convert automatically.

The change reason of this item depends on “6.4 Changes the supporting memory model”.

7.5. Link directive file

In the C compiler “CA78K0R”, the allocation information is described to “link directive file”, but In the C compiler “CC-RL”, the allocation information is specified by the option of the linker “-start”.
In the CS+, if a project of application with RI78V4 V1.00.02 reads, the allocation information is changed to the option of the linker, but confirms the allocation address. Please refer to the manual “CC-RL Compiler” for setting the option of the linker.

8. Changes about update products

Changes update products from this package are as follows. Update products aren't provided this package, please update by using CS+ update feature.

8.1. Realtime OS Task Analyzer Plug-in

（1） Changes the way to zoom in or out of trace chart (execution transition state of the processing program and Real-Time OS resource usage status)

Changes the way to zoom in or out trace chart only displayed if [Selection of trace mode] is specified [Taking in trace chart by hardware trace mode] or [Taking in trace chart by software trace mode]. For details, refer to RI78V4 V2.00.00 Real-Time Operating System User's Manual: Analysis.

The changes version is as follows.

Tool Name	Version No.
Realtime OS Task Analyzer Plug-in (Panel)	V3.01.00.08

9. Cautions

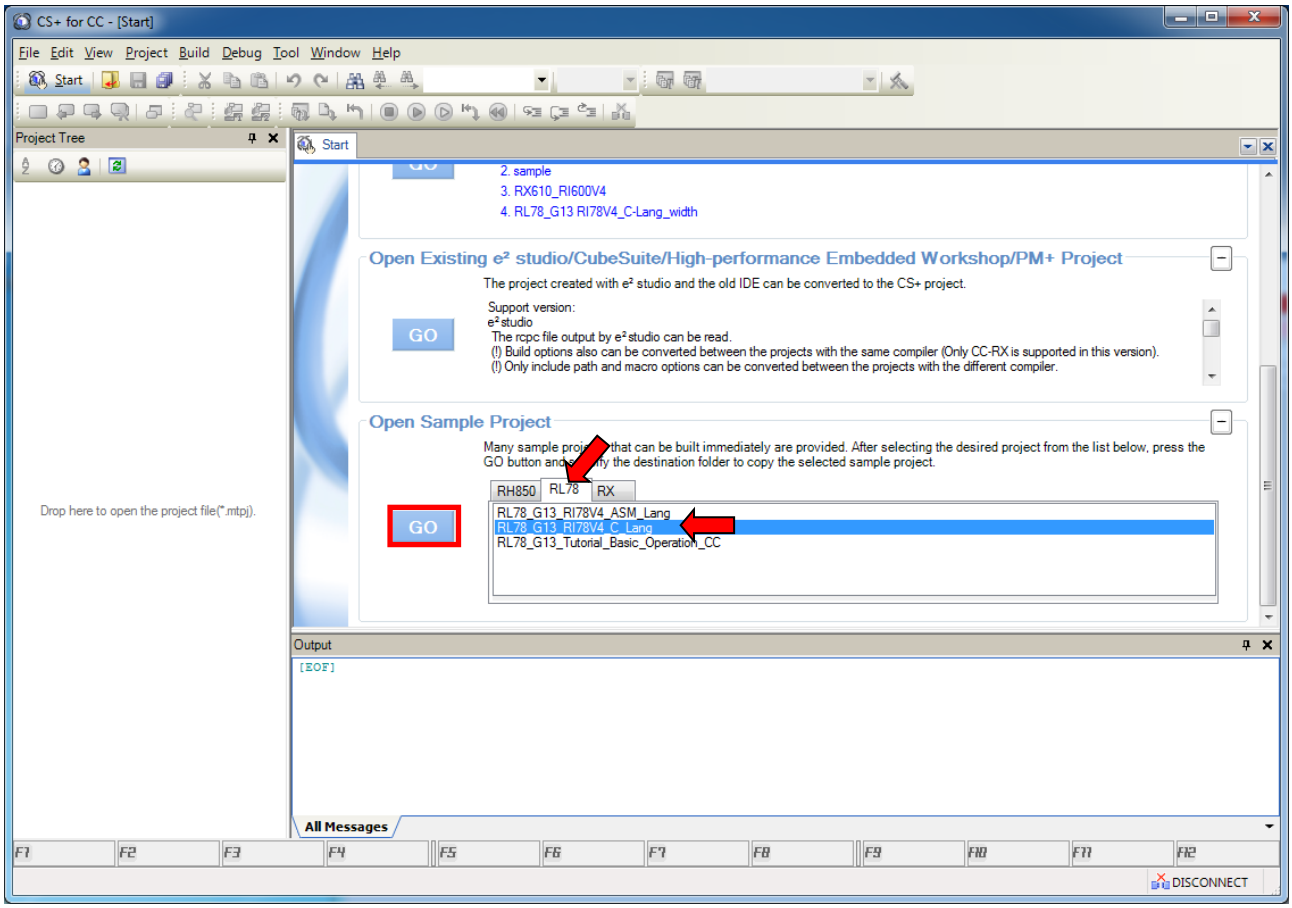
9.1. Create a CS+ Project

In order to create a project which uses this product, there are the following three methods.

- Divert the sample project attached to this product.
- Create a new project.
- Recycle a RI78V4 V1.00.02 project.

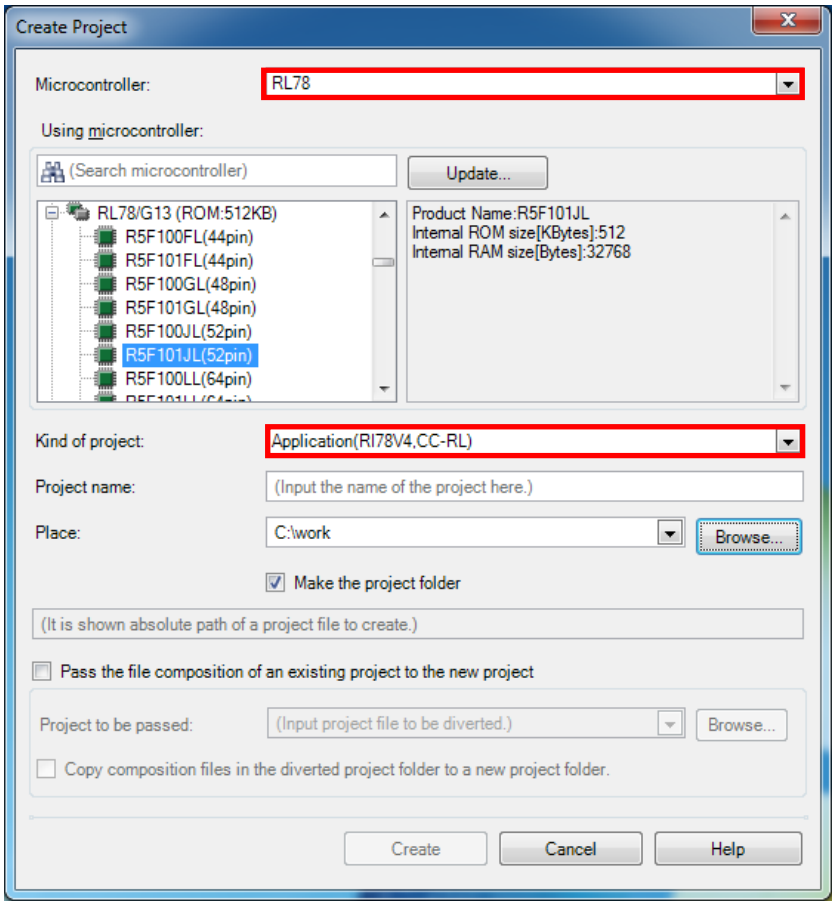
9.1.1. Divert the sample project attached to this product

Select [RL78] tab in [Open Sample Project] area of [Start] panel of the CS+, and choose the project named [RL78_G13_RI78V4_C_Lang] or [RL78_G13_RI78V4_ASM_Lang], and specify the folder in which the sample project will be created.



9.1.2. Create a new project

Press [Go] button in [Create New Project] area of [Start] panel of the CS+, then [Create Project] dialog box will be opened.



- [Microcontroller] : Select [RL78]
- [Kind of project] : Select [Application(RI78V4,CC-RL)]

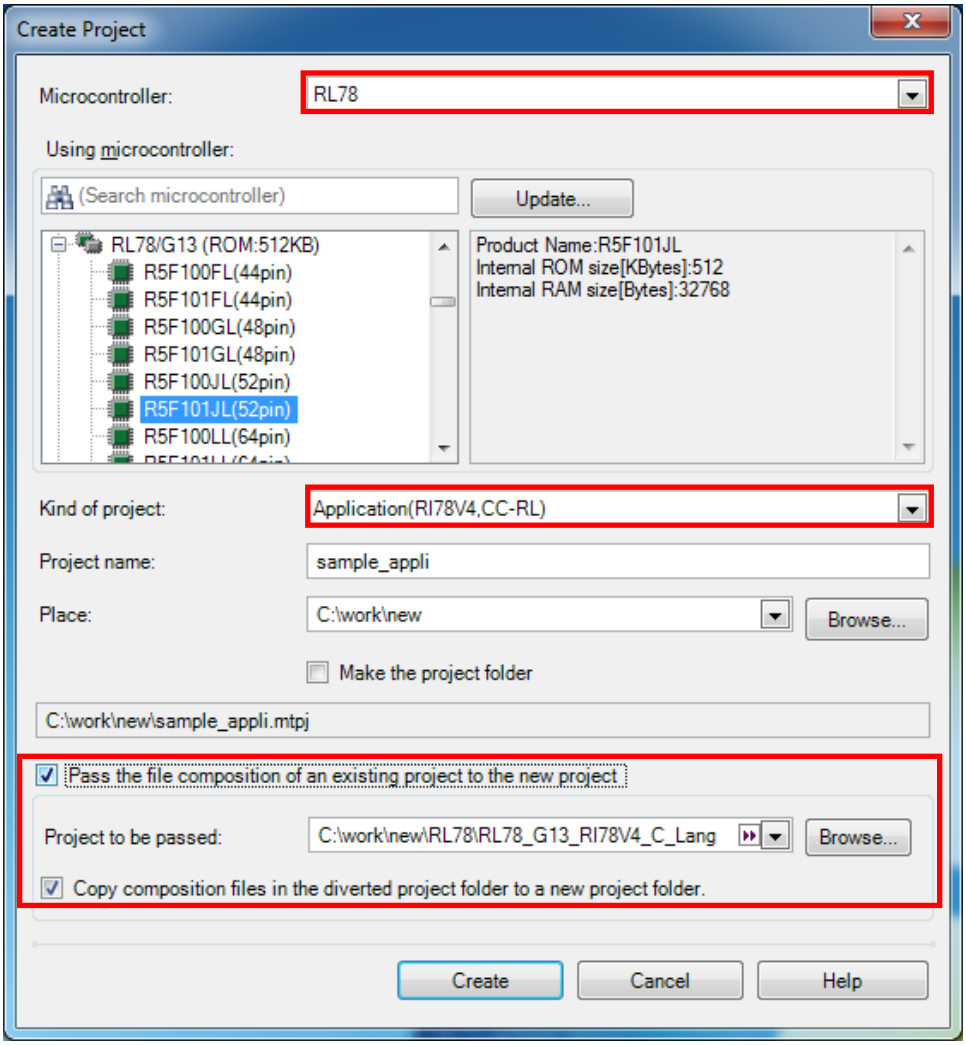
Press [Create] button, then a project will be generated.

Remark In the application project with RI78V4, “iodefine.h” is not registered to project tree automatically. If you need, operates “I/O header file generation option” in the property of a build tool. When the setting registered with a project at the time of a build is performed, it can be used.

9.1.3. Recycle a RI78V4 V1.00.02 project

You can recycle a RI78V4 V1.00.02 project that made CubeSuite+ or CS+ for CACX.

Press [Go] button in [Create New Project] area of [Start] panel of the CS+, then [Create Project] dialog box will be opened.



- [Microcontroller] : Select [RL78]
- [Kind of project] : Select [Application(RI78V4,CC-RL)]
- Select [Pass the file composition of an existing project to the new project] and select the existing project(select project file).
- If you need to have the copies of recycled project files in the new project folder, select [Copy composition files in the diverted project folder to a new project folder].

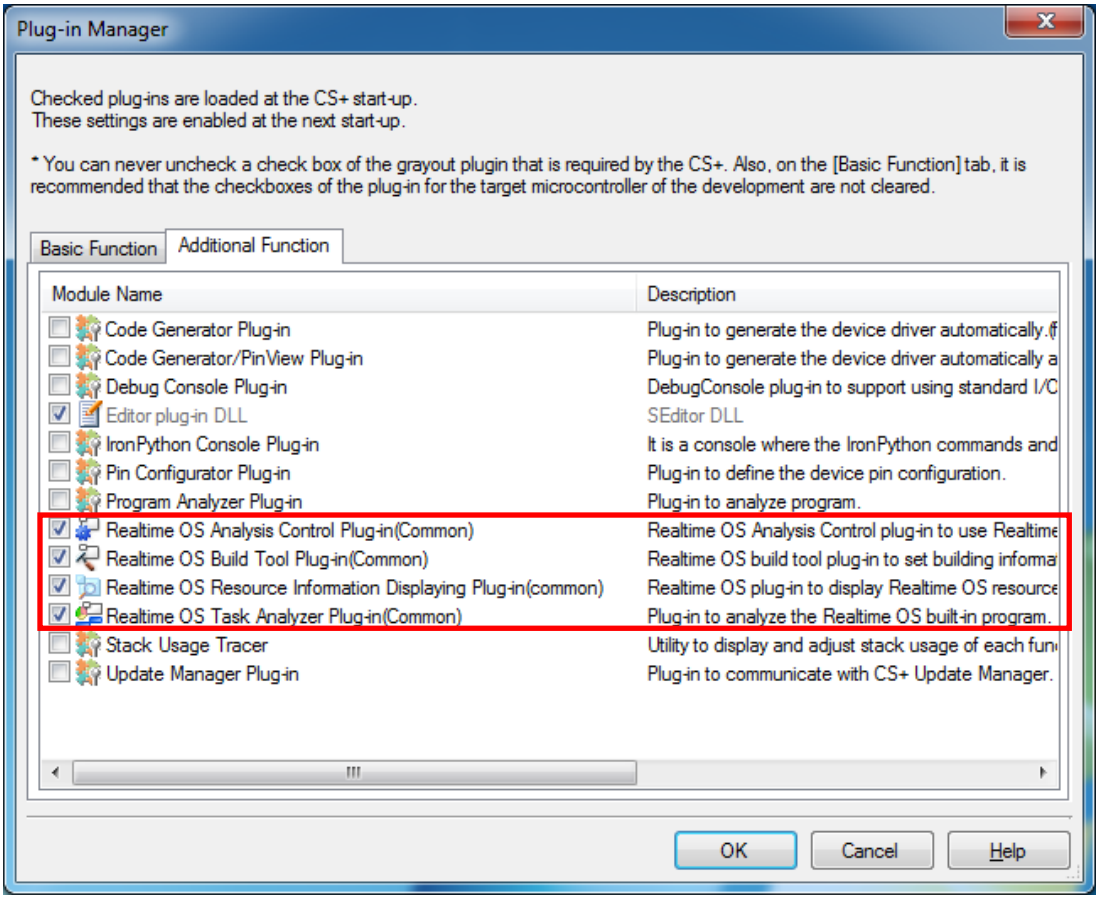
Press [Create] button, then a project will be generated.

9.2. Enable Real-Time OS Plug-ins

Plug-ins of this product may be disabled immediately after installation of this product. If plug-ins are disabled, the problem of being unable to build arises.

Please enable following Plug-ins by [Additional Function] tab in [Plug-in Manager] dialog box of the CS+ for CC.

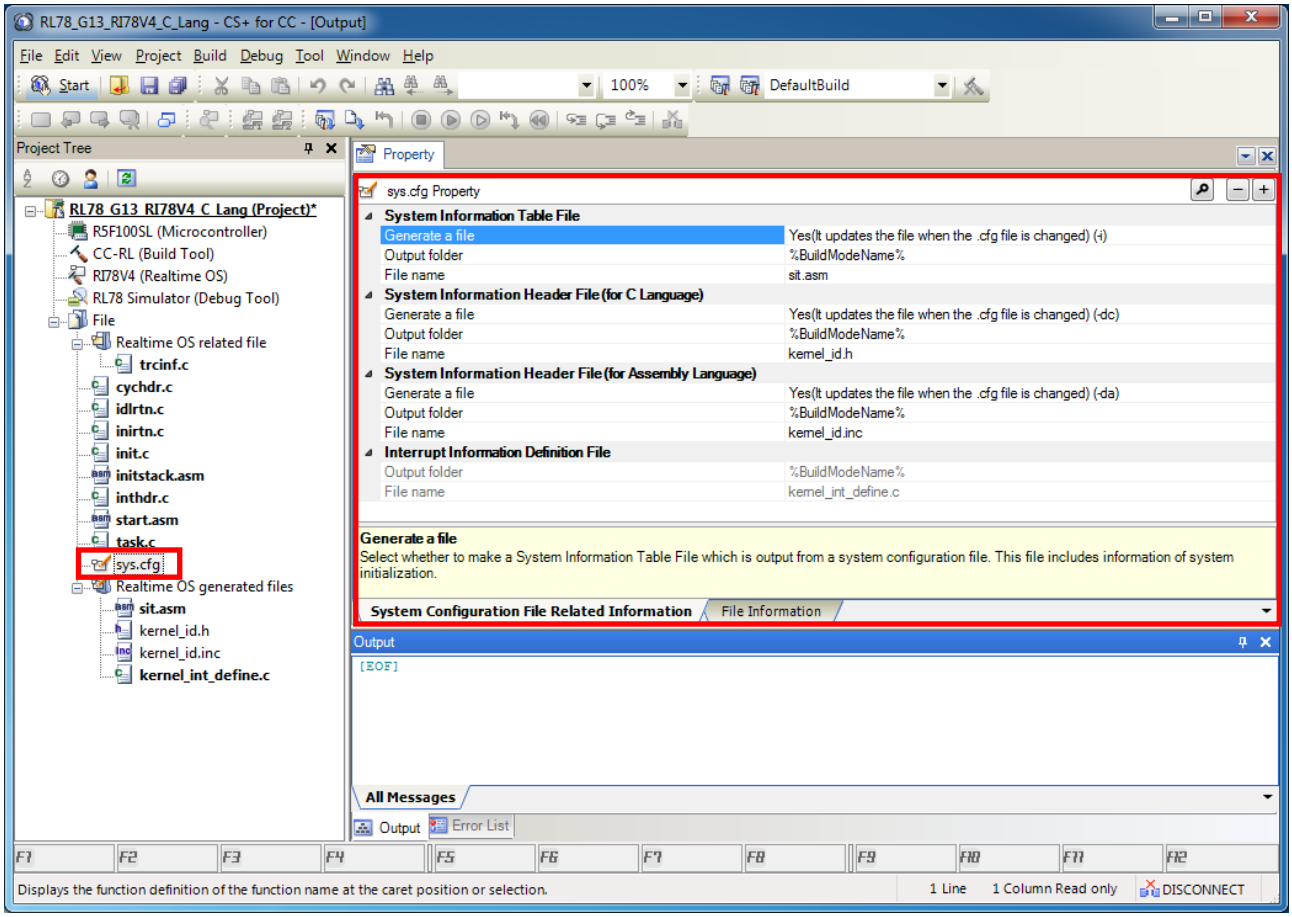
- Realtime OS Build Tool Plug-in(Common)
- Realtime OS Analysis Control Plug-in(Common)
- Realtime OS Resource Information Displaying Plug-in(common)
- Realtime OS Task Analyzer Displaying Plug-in(common)



9.3. Property of Configuration file

Specify the options for configurator with property of configuration file.

Right-click the configuration file in [project tree] and Select [property] to open properties of [System Configuration File Related Information].



Setting items and contents of [System Information Table File] , [System Information Header File(for C Language)] [System Information Header File(for Assembly Language)] are no difference from RI78V4 V1.00.02.

［Interrupt Information Definition File］ is not exist in RI78V4 V1.00.02, this item is not able to change.

9.4. Cautions for Realtime OS Resource Information Panel

9.4.1. View after Real-Time OS initialization

View the Realtime OS Resource Information Panel after the Real-Time OS has been initialized. Before the Real-Time OS has been initialized, the information in the Realtime OS Resource Information Panel is undefined.

9.4.2. Use programs with debug information generated

When using the Realtime OS Resource Information Panel, download a program for which debug information has been generated. Downloading a program without debug information and viewing it in the Realtime OS Resource Information Panel will cause an error.

To generate debug information, under Build Tool, under the Link Options properties, set [Generate debug information] to [Yes].

9.5. Cautions for Realtime OS Task Analyzer Information Panel

9.5.1. To change the trace mode

Select [Selection of trace mode] on the [Task Analyzer] tabbed page of the [Property] panel for RI78V4. The default selection for [Selection of trace mode] is [Not tracing]. If you wish to use the task analyzer, select an option other than [Not tracing]. Building must be re-executed whenever the selection for [Selection of trace mode] is changed. Since the monitor to be used varies with the selected trace mode, re-building leads to incorporation of the monitor for the given mode.

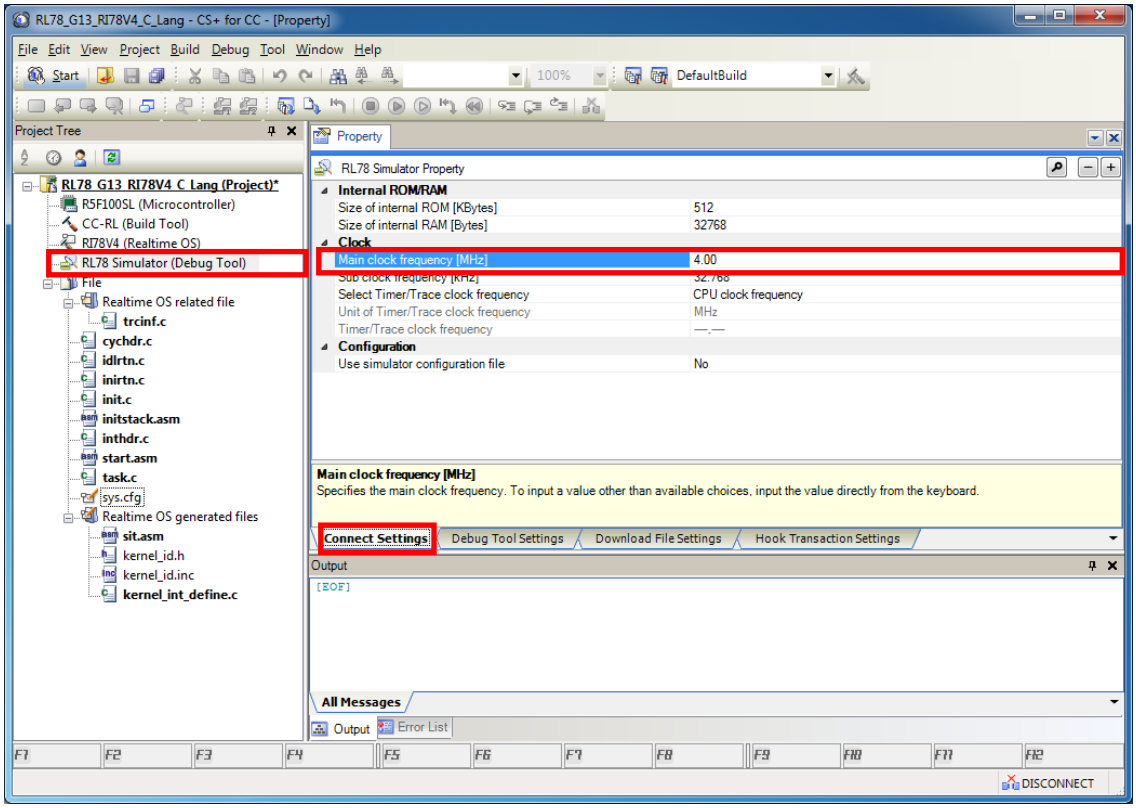
9.5.2. If the [Taking in trace chart by hardware trace mode] is selected with simulator

It is necessary to specify the following settings correctly to obtain correct time on Realtime OS Task Analyzer plug-in.

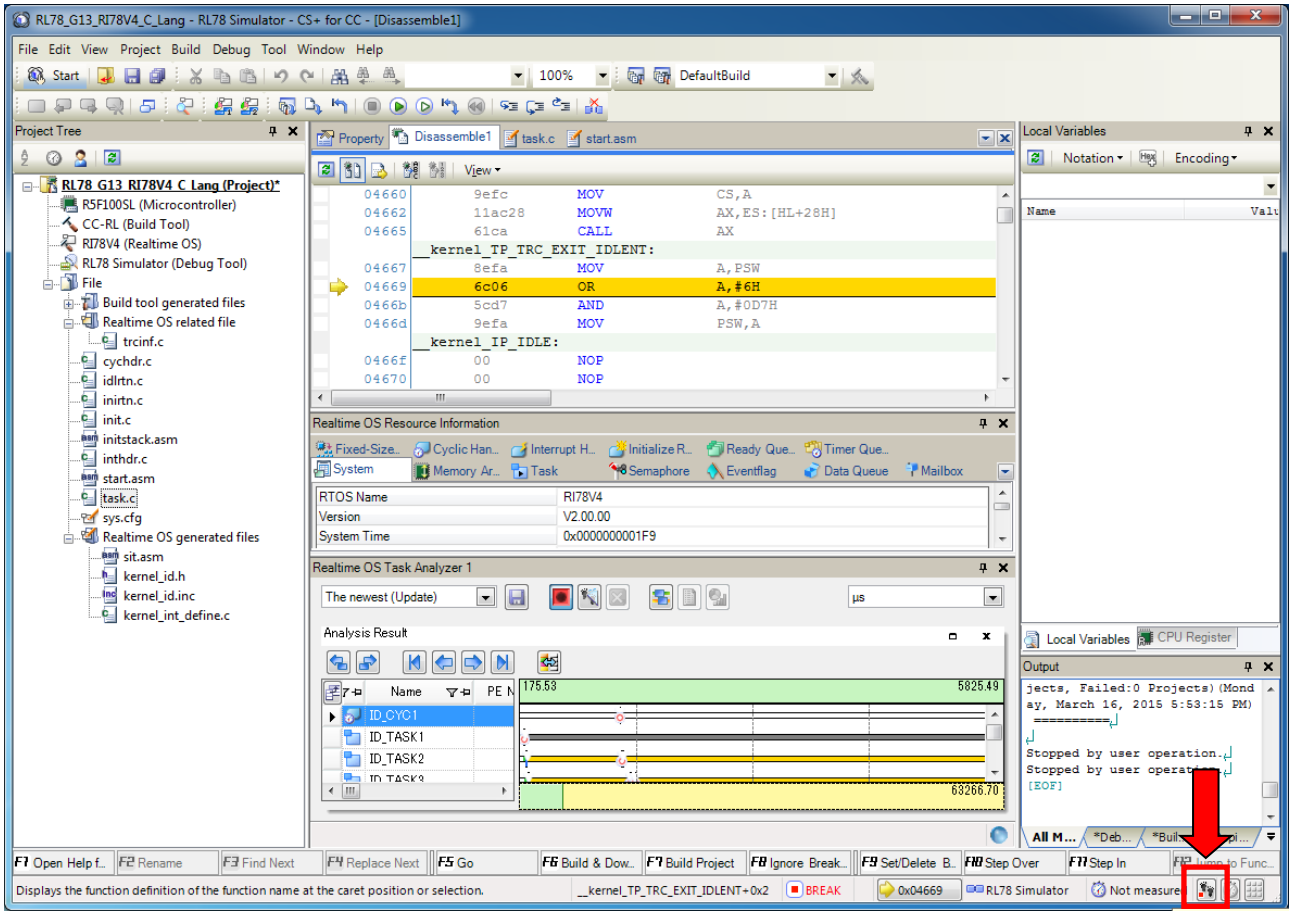
- [Main clock frequency [MHz]]

  (Refers to Figure 9-1) Specify a suitable value to the [Main clock frequency [MHz]] in the [Clock] category in the [RL78 Simulator Settings] tab of the Property Panel for the Debug tool.

Figure 9-1 [Main clock frequency [MHz]] Tab in The property of RL78 Simulator (debug tool)



And turn on the trace function of the debugger(press the icon of a footprint mark).



9.5.3. If the [take in trace chart by hardware trace mode] is selected with E1/E20 emulator

This feature is not support hardware trace mode using E1/E20 emulator. If you wish to use the E1 or E20 emulator, select [Takes in trace chart by software trace mode] or [Takes in long-statistics by software trace mode].

9.5.4. Setting for the debugging tool

Do not assign the following combination to the [Trace] category in the [Debug Tool Settings] tab in the property of the debugging tool.

- [Clear trace memory before running] : No
- [Accumulate trace time] : Yes

9.5.5. Time stamp for the Software trace

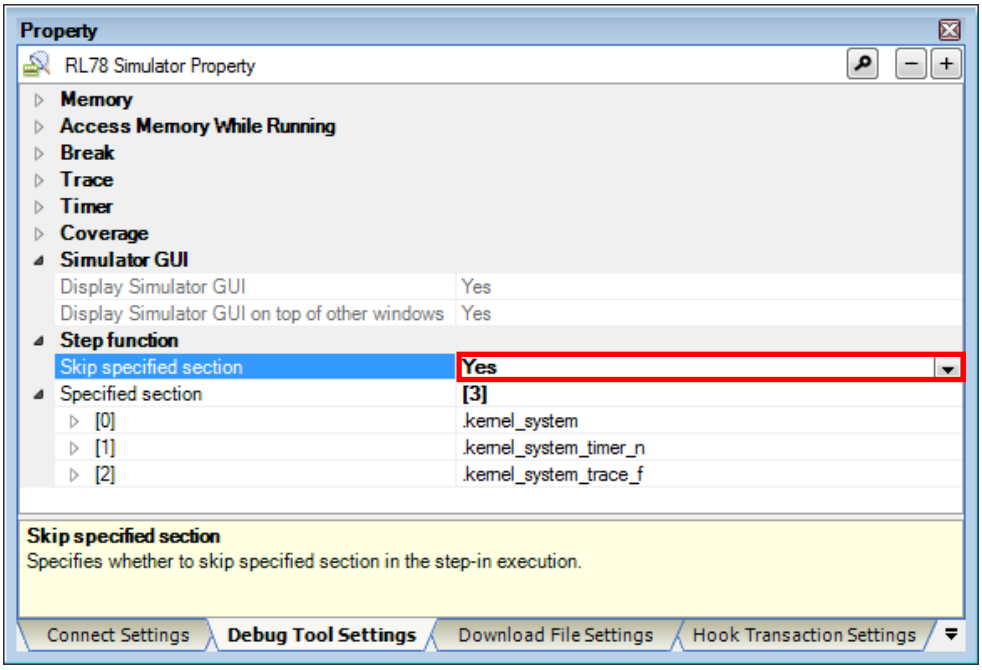
The timer of the kernel is used to acquire timestamps in tracing. Since the timer of the kernel uses OS timer interrupts, handling of timer interrupts is suspended while interrupts are disabled. Correct display of time is not possible while interrupts are disabled (e.g. by a task) for a period of 1 ms or longer (but the occurrence of interrupts or tasks will still be displayed in the correct order).

9.6. The step function for section skip

When doing step execution in the CS+, the function which skips a kernel cord of RI78V4 is added. When carrying out “step-in” after having broken at a service call issuing, the debugger continue to skip a kernel code. And when reaching outside the kernel code, step execution is stopped.

When “step-in” is carried out at a service call issue part when a dispatch occurs by service call issue when this function is used, it is possible to make them break away ty a task after a dispatch.

But this step function is off by default. If it uses this function, terns on the switch “Yes” the “Debug Tool Setting” -> “Step function” -> “Skip specified section”. Further “Skip specified section” is set automatically.



9.7. How to build kernel source code

Since the RI78V4 kernel is provided in the library form, it does not usually need to build the kernel.

If you have “RTRRL7800TR01SRRUU (Mass-production license, unlimited copies, including source code)”, the kernel source code is installed in "<installation folder>\source\kernel". And to build the kernel libraries, makefile is installed in “<installation folder>\source\project\rl78_ccrl\medium”. This makefile presuppose that it runs on Cygwin, therefore, building the kernel, Cygwin installation environment is necessary.

- How to build

  Move to folder that installed makefile on cygwin window. And input

  make [return]

  And then, libraries will be generated in

  <Installation folder> \library\rl78_ccrl\medium

If you do not have the write privilege for the installation folder, copy the installation folder to another folder where you are able to write. After building the kernel source code, the generated libraries must be copied to <name of the installation folder>\library\rl78_ccrl\medium by a user who has the write privilege for the installation folder.

9.8. About the warning when the sample program is built

When the sample program is built, several warnings are output. These warning is not removed because to develop the program based on this sample, and to convert other kind’s device easily. When fixing a program finally, it is recommended to remove all warnings.

10. Restrictions

10.1. RI78V4 V2

10.1.1. Trace function

The [long-statistics by software trace mode] of the trace function provided by RI78V4 V2 and Realtime OS Task Analyzer Plug-in must not be used. When the [long-statistics by software trace mode] is used, an unspecified RAM area may be corrupted.

10.2. Restrictions of CS+ for CC

10.2.1. Realtime OS Build Tool Plug-in

(1) Multiple build modes

Do not use multiple build modes for the following reasons.

- The configurator options are common to all build modes. Even if multiple build modes are used, the same configurator options are applied.
- Every time the build mode is changed, the path to the kernel_id.h file is added to [Additional include paths] of the build tool. Although the build-setting plug-in sets the correct path in [System include paths], the IDE adds the old path prior to the change of the build mode to [Additional include paths]. In the process of building, the build tool refers to the old path set by the IDE. This means that editing the configuration file to change the build mode before editing kernel_id.h, for example, will not be reflected in building.

10.2.2. Realtime OS Resource Information Displaying Plug-in

(1) Effect of resetting the display of waiting tasks (child nodes) on the display of the [Task] tabbed page.

Resetting the display of waiting tasks also resets the display of other tasks in the [Task] tabbed page. However, the information being displayed will be correct.

10.2.3. Realtime OS Task Analyzer Plug-in

(1) The restriction the right result which do filtering operation to a CPU usage rate column cannot get

A filtered result is not indicated by the filtering contents right. For example, when specifying “more than 80% of indication”. Without being filtered, all columns indicate the CPU usage rate.

(2) Restriction of hardware trace mode with E2 emulator / E2 emulator Lite

Realtime OS Task Analyzer Plug-in does not support hardware trace mode using E2 emulator / E2 emulator Lite.

11. Sample Programs

11.1. Sample programs with board support package module

This section describes sample programs with board support package module (BSP).

11.1.1. Overview

The sample programs are composed of the sample programs included in RI78V4 V2.00.00 package and BSP. This sample programs run multi-task process after oscillating the CPU clock using BSP API.

For details on how to use BSP, please refer “RL78 Family Board Support Package Module Using Software Integration System (R01AN5522EJ*xxxx*)”.

11.1.2. Structure for sample programs with BSP

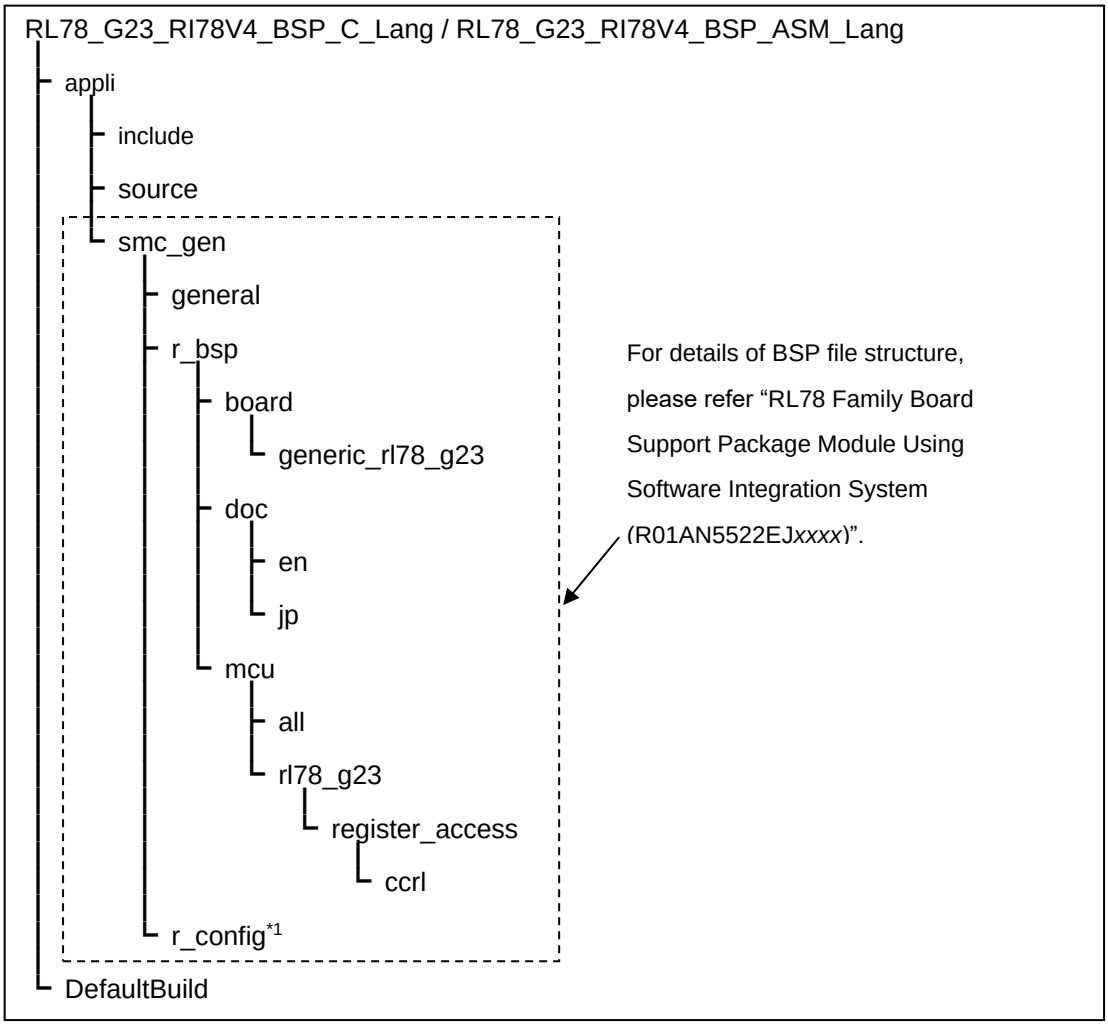
The sample programs are provided as CS+ sample projects. The structure of the sample projects consist of the following.

Sample project name	Contents
RL78_G23_RI78V4_BSP_C_Lang	The source code for objects such as task managed by RI78V4 V2 is coded in C language.
RL78_G23_RI78V4_BSP_ASM_Lang	The source code for objects such as task managed by RI78V4 V2 is coded in assembly language.

Module name	File name	Revision
BSP	r_bsp	v1.10

11.1.3. Folder structure for sample projects with BSP

The folder structure of the sample projects with BSP is shown below.



*1：The file location folder for BSP configuration.

11.1.4. Setting for sample projects with BSP

(1) Macro definition

Set “BSP_RTOS” to the macro definition in the assemble option.

(2) Setting of the BSP configuration file

File name	Macro definition	Value	Contents
r_bsp_config.h	BSP_CFG_RTOS_USED	4	RI78V4 V2 is used.
	BSP_CFG_CONFIGURATOR_SELECT	0	Smart Configurator is not used.
r_bsp_config.inc	BSP_CFG_ASM_RTOS_USED	4	RI78V4 V2 is used.

11.1.5. Cautions for sample project with BSP

(1) The sample projects are not allowed to use of Smart Configurator.

(2) The sample projects are not included in RI78V4 V2.00.00 package. Please download it from the website of Renesas Electronics Corporation.

11.1.6. Changes from the sample projects included in RI78V4 V2.00.00 package

Changes	Sample projects in RI78V4 V2 package	Sample projects with BSP
Microcontroller	RL78/G13	RL78/G23
Debug tool	Simulator	E2 Lite emulator
Startup	Use startup provided with RI78V4 V2 package.	Use startup provided with BSP.
Clock oscillation setting	Set the clock control register within hdwinit function.	Set using R_BSP_StartClock in initialization routine inthdr1.
How to include kernel.h / kernel_id.h within source files which are coded in C language	Directly include kernel.h / kernel_id.h.	Include platform.h. Note that kernel.h and kernel_id.h are included in r_rtos.h.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	25/3/2015	-	New Publication
1.01	31/3/2016	24	Add Update Information of Realtime OS Task Analyzer Plug-in. Changes the way to zoom in or out of trace chart (execution transition state of the processing program and Real-Time OS resource usage status)
1.0.2	28/4/2021	7	Update hardware environment to the latest information.
		7	Update software environment to the latest information.
		7	Update support tool to the latest information.
		36	Add a restriction that prevents the use of [long-statistics by software trace mode].
		36	Add a restriction that Realtime OS Task Analyzer Plug-in is not support hardware trace mode using E2 emulator / E2 emulator Lite.
		37	Newly add section for sample programs with BSP.

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/.