RH850 Family Real-Time OS RI850V4 V2.01.01 Release Note

R20UT3679EJ0101
Rev.1.01
2016.03.31

Contents

1. Tools in the RTOS Packages

Available types of agreement and their contents differ with the product.

Product Name	Agreement Type	Contents
RTRRH8500TR01ERR	Evaluation license, limited to 1 host	A
RTRRH8500TR01ERRLU	Evaluation license, unlimited hosts	A
RTRRH8500TR01RRRUL	Mass-production license, 3000 copies	A
RTRRH8500TR01RRRUU	Mass-production license, unlimited copies	A
RTRRH8500TR01SRRUU	Mass-production license, unlimited copies, including source code	B

The following tools are provided.

Contents		Tool Name	Version No.
B	A	RI850V4 Real-Time OS Kernel Object Code	2.01.00
		CF850V4 Configurator for RI850V4 Real-Time OS	2.01.01.01
		CS+ Plug-ins	
		Realtime OS Build Tool Plug-in (Common)	3.02.01.01
		Realtime OS Build Tool Plug-in (RI850V4RH)	2.01.01.01
		Realtime OS Analysis Control Plug-in (Common)	3.00.00.03
		Realtime OS Analysis Control Plug-in (uITRON4)	3.02.00.01
		Realtime OS Analysis Control Plug-in (RI850V4RH)	2.01.00.03
		Realtime OS Resource Information Display Plug-in (Common)	3.01.00.01
		Realtime OS Resource Information Display Plug-in (uITRON4)	3.01.01.01
		Realtime OS Task Analyzer Plug-in (Common)	3.00.02.01
		Realtime OS Task Analyzer Plug-in (Panel)	3.00.00.03
		Realtime OS Task Analyzer Plug-in (RI850V4RH)	2.01.00.03
		Real-Time OS RI850V4 Kernel Source Code	2.01.00

2. User's Manuals

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	R20UT0751EJ0106
RI850V4 V2 Real-Time Operating System User's Manual: Coding	R20UT2889EJ0102
RI850V4 V2 Real-Time Operating System User's Manual: Debug	R20UT2890EJ0102
RI850V4 V2 Real-Time Operating System User's Manual: Analysis	R20UT2891EJ0102
RI Series Real-Time Operating System User's Manual: Message	R20UT0756EJ0105

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.

- RH850 (G3K core, G3M core, G3KH core, G3MH core) family

4. Operating Environment

The operating environment for this product is described below.

4.1. Hardware environment

- Processor: At least 1GHz (supported for hyper threading/multicore CPU)
- Memory capacity: 2 GB or more recommended. Minimum requirement is 1 GB or more (64-bit Windows® requires 2 GB or more)
- Display: Resolution at least 1024 x 768; at least 65,536 colors

4.2. Software environment

- Windows 7, Windows 8.1, Windows10(32- or 64-bit version)
- .NET Framework 4.5.2
- Runtime library of Microsoft Visual C++ 2010SP1

Remark For any of these, we recommend that you also have the latest service pack installed.

4.3. Support Tools

Tool Name	Manufacturer	Version
CS+ integrated development environment	Renesas Electronics	V3.02.00 or later
CC-RH C Compiler	Renesas Electronics	V1.03.00 or later
MULTI integrated development environment	Green Hills Software.	V6.1.6 Rel9.4.0 or later However, if you use RH850 G3KH or G3MH, need to V6.1.6 Rel9.6.2 or later for that.

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 Realtime OS Plug-in“

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 RI850V4RH Plugins
- CS+ Realtime OS RI850V4RH Object Release, or CS+ Realtime OS RI850V4RH Source Release

6. Changes from Previous Version

6.1. Kernel

1.1 Differences in Kernel

(1) Supports the RH850 G3KH and G3MH core

This version newly supports the RH850 G3KH and G3MH core(kernel is not changed this version).

6.2. Configurator

(1) Changed: "G3KH" and "G3MH" can be specified for CPU type *chip_type.*

“G3KH" and “G3MH” can be specified for *chip_type* because this version newly supports the RH850 G3KH and G3MH core.

(2) Changed: The device file for the G3KH and G3MH core can be specified for activation option -cpu ∆<*name*>.

The device files for the G3KH and G3MH core can be specified for the activation option -cpu ∆ <*name*> because RI850V4 V2.01.00 newly supports the RH850 G3KH and G3MH core.

6.3. Realtime OS Build Tool Plug-in

There is no difference in the Realtime OS Build Tool Plug-in.

6.4. Realtime OS Resource Information Displaying Plug-in

There is no difference in the Realtime OS Resource Information Displaying Plug-in.

6.5. Realtime OS Task Analyzer Plug-in

There is no difference in the Task Analyzer Plug-in.

7. Points of difference with the previous RI850V4 (V1.00)

7.1. Support for a further RH850 device

RI850V4 (V2.00.00) supports the RH850.

7.2. Changed points of kernel

（1） Change in interrupt exclusion method during kernel processing

In V1.00 of RI850V4, exclusive control of interrupts was achieved by setting the PSW.ID bit to 1 during kernel processing, which disables the acceptance of any maskable interrupt. In V2.00, on the other hand, the interrupt priority mask register (PMR) is used to disable the acceptance of interrupts for kernel service that have a specified priority level or a lower level. The value set in the PMR is called the maximum kernel interrupt priority level (maxintpri). Fast response to interrupts can be expected since all interrupts for kernel service that have higher priority levels than maxintpri are accepted during kernel processing.

Other interrupts with higher priority levels than maxintpri (i.e. non-kernel interrupts) cannot be used to issue kernel service calls. Since the kernel is not involved with the handling of such interrupts, processing at the branch destinations of the vectors for those interrupt handlers must be written separately.

The kernel does not set priority levels for any interrupts including those with priority levels below or equal to maxintpri (i.e. kernel interrupts). For this reason, you should directly manipulate the ICxxx register in the interrupt controller to set the priority levels of such interrupts. All interrupt handlers to be defined in the configuration file must be given priority levels that are appropriate for kernel interrupts. This is also required for base clock timer interrupts.

The “maxintpri” can be set in the system configuration file. For details, refer to section,”（1）“.

（2） loc_cpu/iloc_cpu

Whether to accept maskable interrupts and which to accept is now controlled by adjusting the setting of the PMR. For this reason, if loc_cpu/iloc_cpu is issued, interrupts that should be managed by the kernel are not accepted.

In this state, the kernel does not manipulate the IMRm and ICxxx registers.

（3） Changing the addresses of exception handlers

In RH850-family MCUs, the addresses of exception handlers can be changed as desired. The reset and exception vector addresses are determined by direct vector selection (in which bit PSW.EBV and registers RBASE and EBASE are used to change the base address of the vector table for the exception handlers). For interrupts, on the other hand, either direct vector selection or table reference can be selected for individual interrupt channels. For details, refer to the hardware manual.

RI850V4 outputs a file of entry points for interrupt handling routines in a format that corresponds to direct vector selection or table reference depending on the property setting of the configuration file. For details, refer to section “9.3 Property of Configuration file“. When direct vector selection is selected, the exception handler vector is output in the file of entry points. When table reference is selected, the interrupt handler address table is

output in the file of entry points.

Since the file of entry points is in either of the two formats, we recommend selecting table reference for all of the interrupt channels to be used by the application if you wish to use table reference for any of them. Since the kernel does not select the interrupt vector formats for individual channels, directly manipulate the ICxxx registers in the interrupt controller.

（4） Service call of the Interrupt Management In V1.00 of RI850V4

In RI850V4 V2.00, the following service calls are not supported.

dis_int/ena_int/chg_ims/ichg_ims/get_ims/iget_ims

（5） CPU exception handlers

In RI850V4 V2, static creation of CPU exception handler is not supported.

（6） Task exception handling functions

In RI850V4 V2, the task exception routines are not supported. And the following service calls are not supported.

ras_tex/iras_tex/dis_tex/ena_tex/sns_tex/ref_tex/iref_tex

（7） Register mode

In V1.00 of RI850V4, the supported register mode are 22, 26 and 32, In V2.00 of RI850V4, the supported register mode is only 32.

7.3. Changed points of configurator

（1） [Additional Basic Information] Maximum kernel interrupt priority “maxintpri”

Specify the maxium kernel interrupt priority. This definition must precede the static API information.

Format:

MAX_INTPRI (maxintpri);

Values specifiable for maxintpri as follows.

In G3K core: INTPRI0 (highest), INTPRI1, INTPRI2, INTPRI3, INTPRI4, INTPRI5, INTPRI6, and INTPRI7 (lowest).

In G3M or G3KH or G3MH core: INTPRI0 (highest), INTPRI1, INTPRI2, INTPRI3, INTPRI4, INTPRI5, INTPRI6, INTPRI7, INTPRI8, INTPRI9, INTPRI10, INTPRI11, INTPRI12, INTPRI13, INTPRI14, and INTPRI15 (lowest)

When this definition is omitted, the value is set as INTPRI0 by default.

Description example：

MAX_INTPRI (INTPRI3);

（2） [Changed] RI Series information “RI_SERIES”

In RI_SERIES, values specifiable for rtos_ver is the “V2xy”(as version number).

If the version number of RI850V4 is "V2.01.23", rtos_ver should be set to "V201".

(3) [Changed] CPU type “CPU_TYPE”

In CPU_TYPE, values specifiable for chip_type is the “G3K”, “G3M”, “G3KH”, “G3MH” only.

(4) [Changed] Exception code of base clock timer interrupt “tim_intno”

In CLK_INTNO, values specifiable for tim_intno are the names or codes (0x1000 to 0x11ff) of interrupt sources defined in the device file. Note, however, that specifying the code of an interrupt source which does not exist will lead to an error.

(5) [Changed] Maximum number of interrupt handlers “maxint”, and maximum value of exception code “maxintno”

In MAX_INT, values specifiable for maxint are 0x0 to 0x200 and those for maxintno are 0x1000 to 0x11ff. Note, however, that these are simply the maximum values in the RH850 architecture. The actual maximum values may vary with the number of interrupt sources defined in the device file for the MCU in use.

(6) [Changed] Interrupt handler information “DEF_INH”

In DEF_INH, values specifiable for inhno are 0x1000 to maxintno. Note, however, that specifying the code of an interrupt source which does not exist will lead to an error.

(7) [Deleted] System information / Static API information

The following keywords are not supported.

REGMODE/DEF_FPSR/DEF_TEX/DEF_EXC

7.4. Realtime OS Task Analyzer Plug-in

In V1.00 of RI850V4, AZ850V4 was provided for the analysis of Realtime OS applications. From RI850V4RH, AZ850V4 is replaced by a new feature called the Realtime OS task analyzer.

The Realtime OS task analyzer is provided as a CS+ plugin. For the usage of this feature, refer to the RI-Series Real-Time Operating System User’s Manual: Analysis.

figure 7-1 Image of AZ850V4

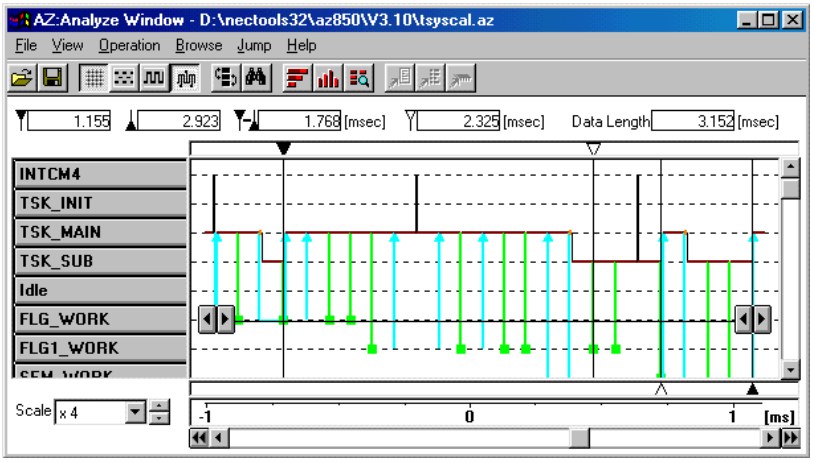
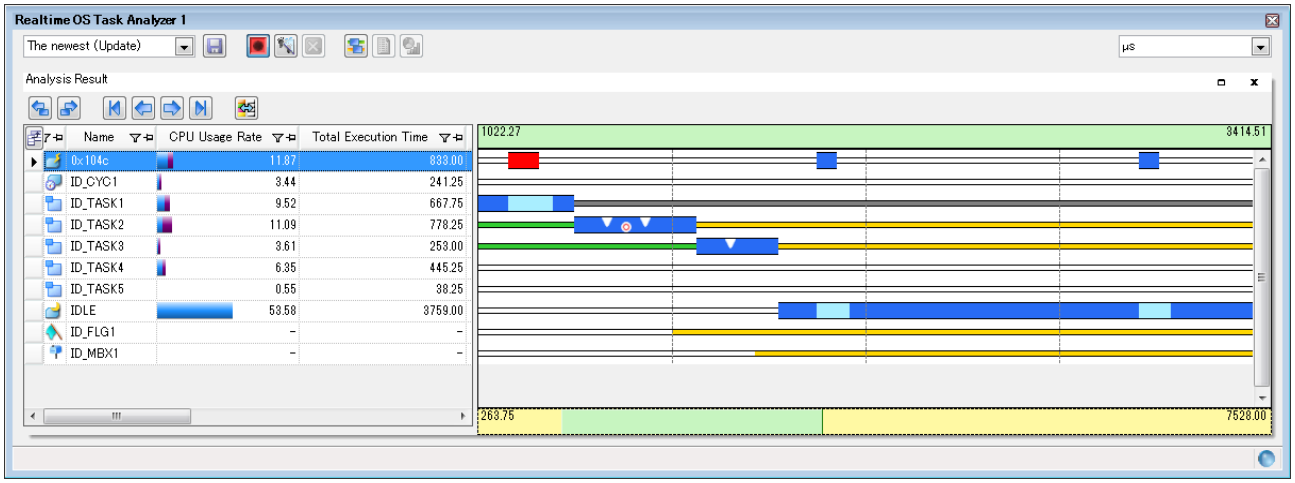
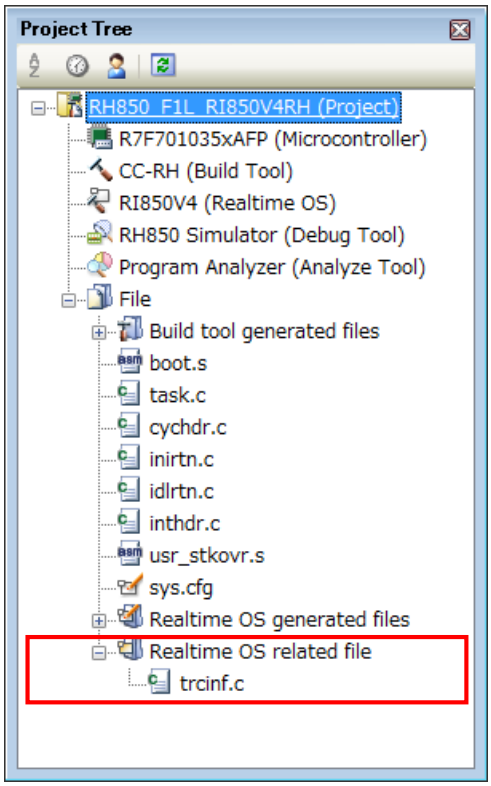


Figure 7-2 Image of Realtime OS Task Analyzer



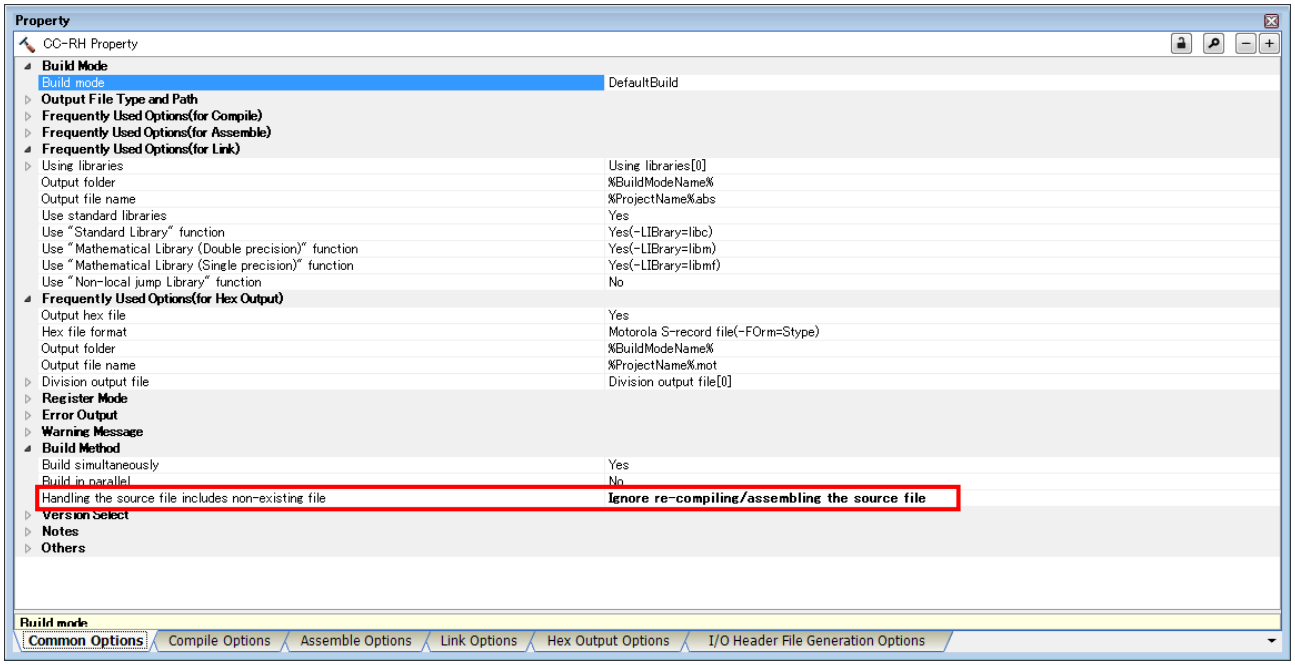
With the provision of this new feature, we have added a [Realtime OS related file] category to the project tree. The trace information file (trcinf.c) will be registered in the [Realtime OS related file] category. This is a read-only file that is not to be edited by the user but needs to be built together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

figure 7-3 Project Tree



For the trace information file, select [Ignore re-compiling/assembling the source file] for [Handling the source file includes non-existing file] under [Build Method] on the [Property] panel of the CC-RH build tool.

figure 7-4 Property of CC-RH



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+\CC\SampleProjects\RH850\RH850_F1L_RI850V4RH\appli\include

The address of the counter register to be used by the base clock timer and time taken to count up by one are defined as macros in the “usrown.h” file. For details, refer to RI850V4 V2.00.00 Real-Time Operating System User’s Manual: Coding.

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 RI850V4 V2 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)	3.01.00.08

9. Cautions

9.1. Create a CS+ Project

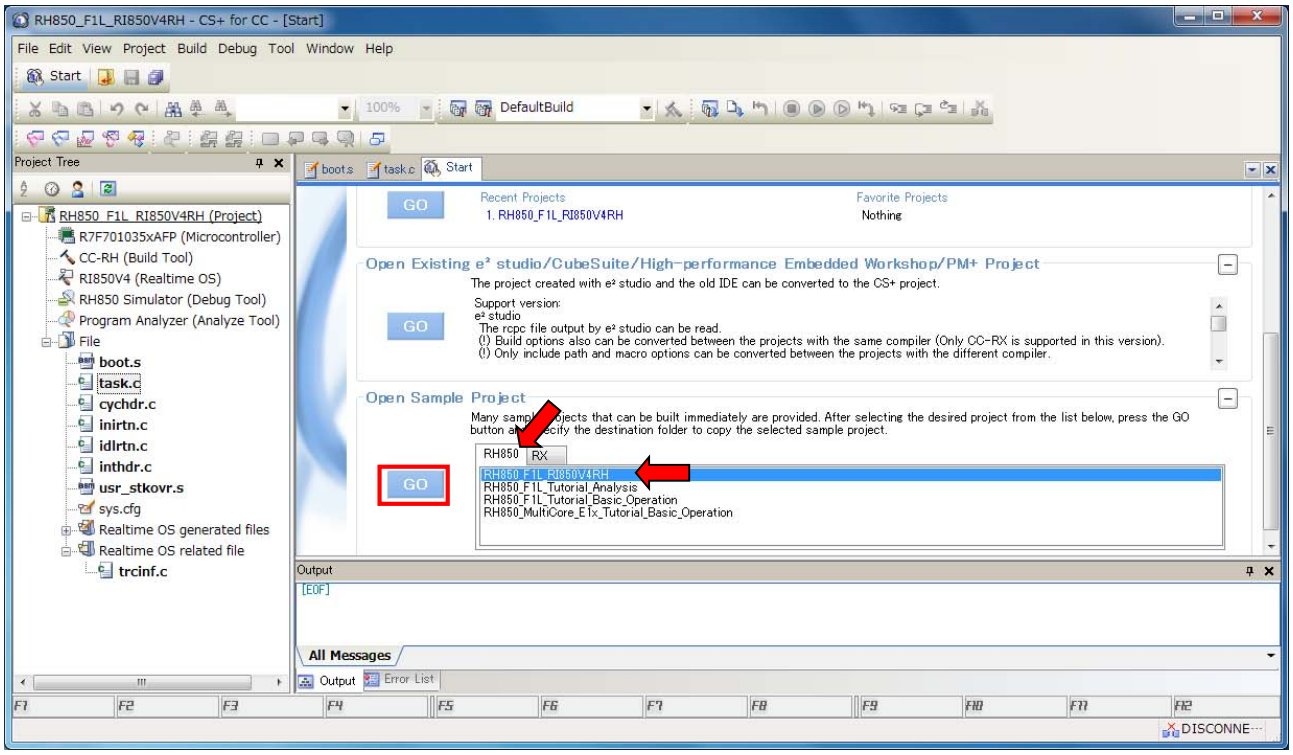
In order to create a project which uses this product, there are the following three methods.

- Divert the sample project attached to this product.
- Create a new project.
- Recycle a RI850V4 V1.00 project.

9.1.1. Divert the sample project attached to this product

Select [RH850] tab in [Open Sample Project] area of [Start] panel of the CS+, and choose the project named [RH850_F1L_RI850V4], and specify the folder in which the sample project will be created.

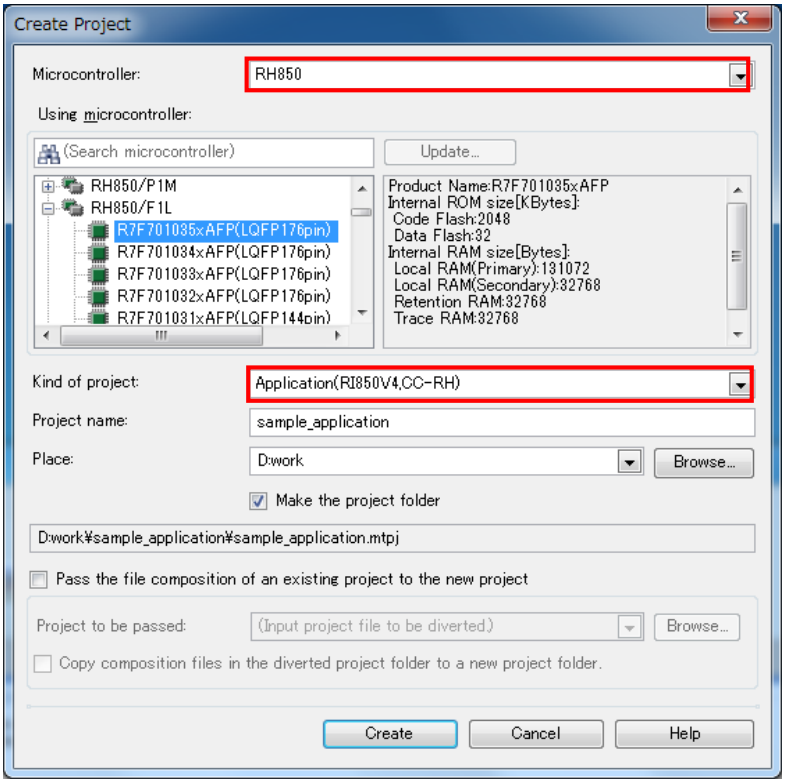
figure 9-1 Open Sample Project



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.

figure 9-2 Create project (Create New Project)



- [Microcontroller] : Select [RH850]
- [Kind of project] : Select [Application(RI850V4,CC-RH)]

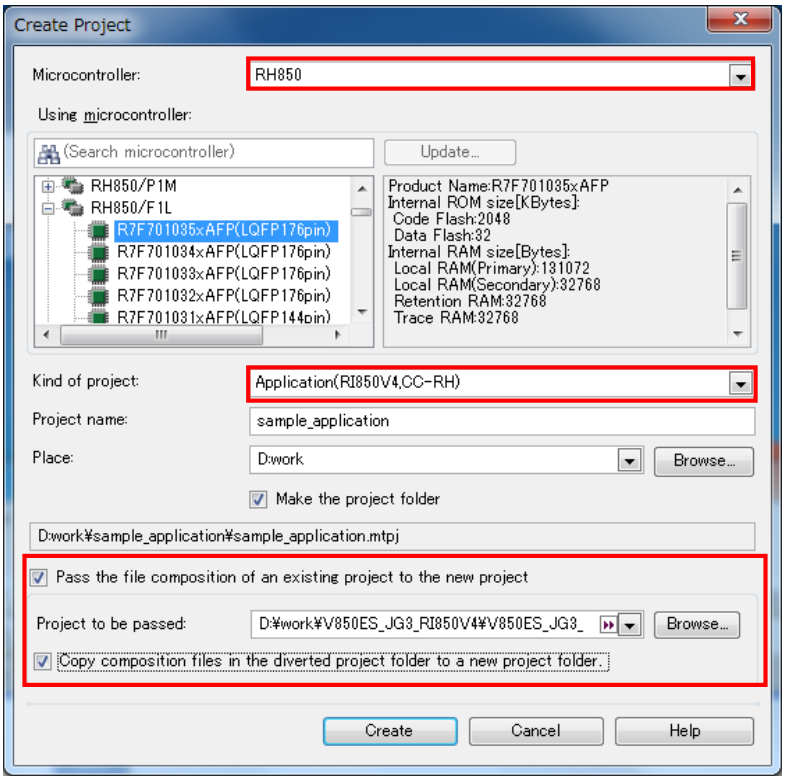
Press [Create] button, then a project will be generated.

9.1.3. Recycle a RI850V4 V1 project

You can recycle a RI850V4 (V1) 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.

figure 9-3 Create Project (Create New Project from Existing Project)



- [Microcontroller] : Select [RH850]
- [Kind of project] : Select [Application(RI850V4,CC-RH)]
- 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.

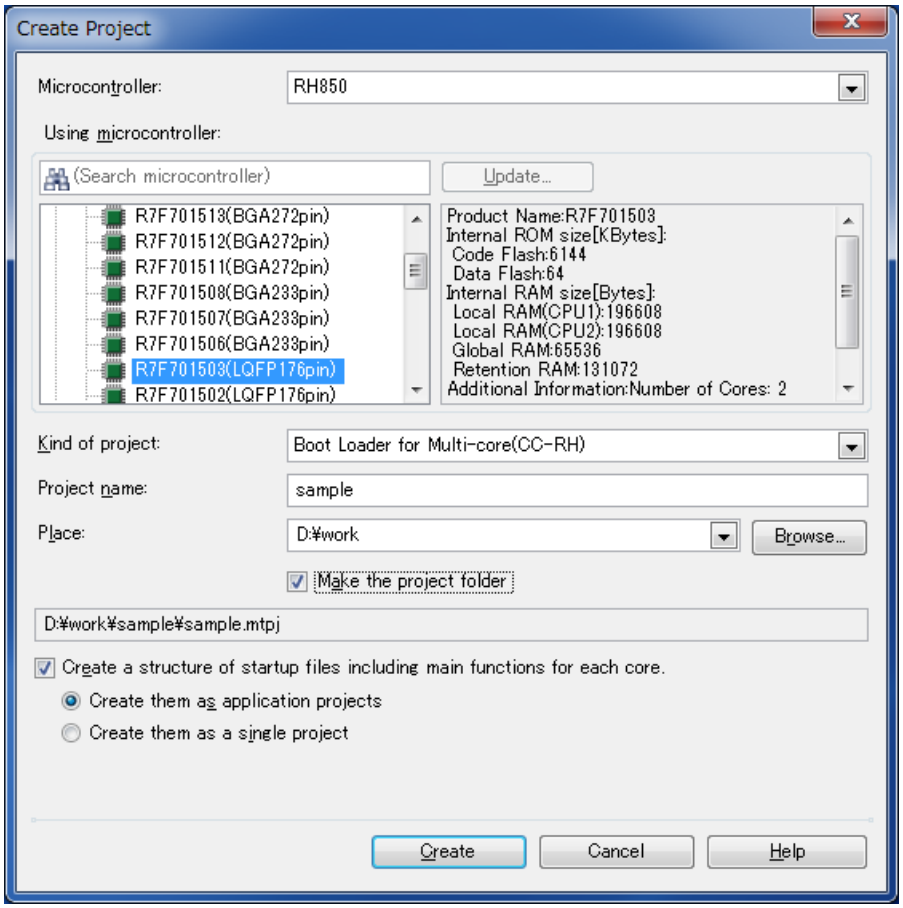
9.1.4. Creating an RI850V4 project for a multi-core device

This section describes the steps for configuring an RI850V4 project for an RH850 multi-core device through CS+. When you make projects for multi-core in CS+, you make a project of each core as a sub-project.

1. Creating a new project for a multi-core device

Click the [GO] button in the [Create New Multi-core Project] area to open the [Create Project] dialog box.

figure 9-4 Create Project (Creating a new project for a multi-core device)



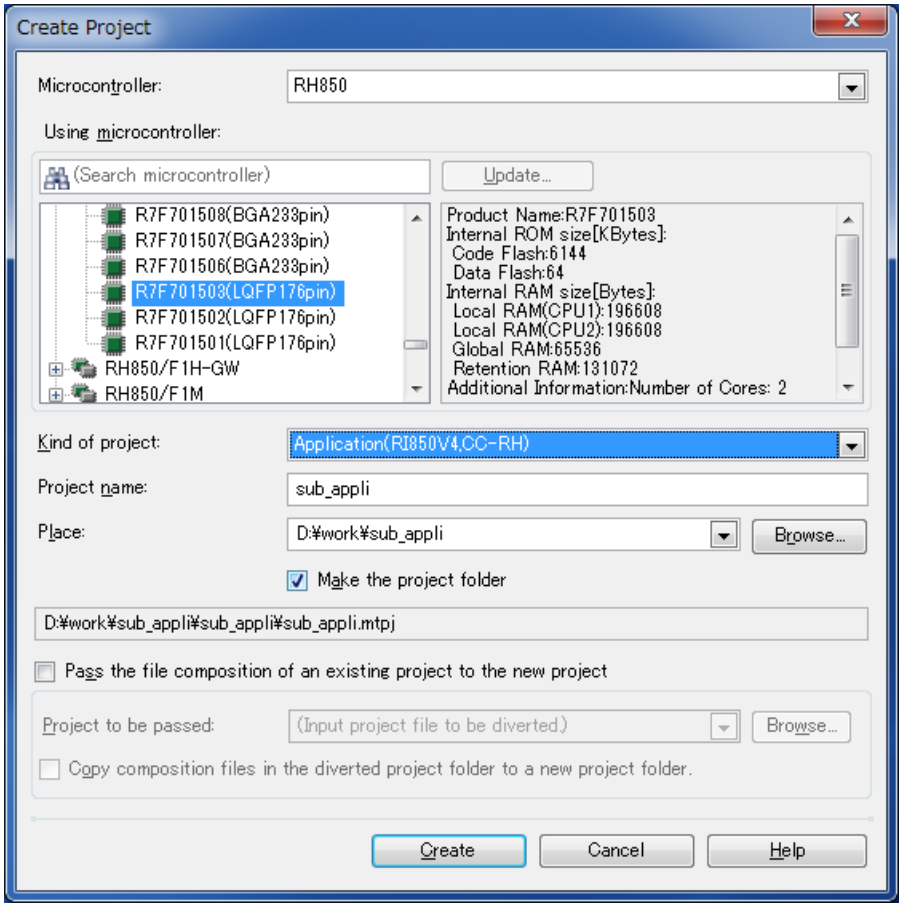
- [Microcontroller]: Select [RH850].
- [Kind of project]: Select [Boot Loader for Multi-core(CC-RH)].
- Deselect the [Create a structure of startup files including main functions for each core.] check box.

Click the [Create] button to create a project.

2. Adding a subproject

Open the [Project Tree] panel for the project. Place the cursor over the project folder in the [Project Tree] panel and right-click it. Select [Add] → [Add New Subproject] to open a dialog box for creating a subproject.

figure 9-5 Create Project (Adding a subproject)



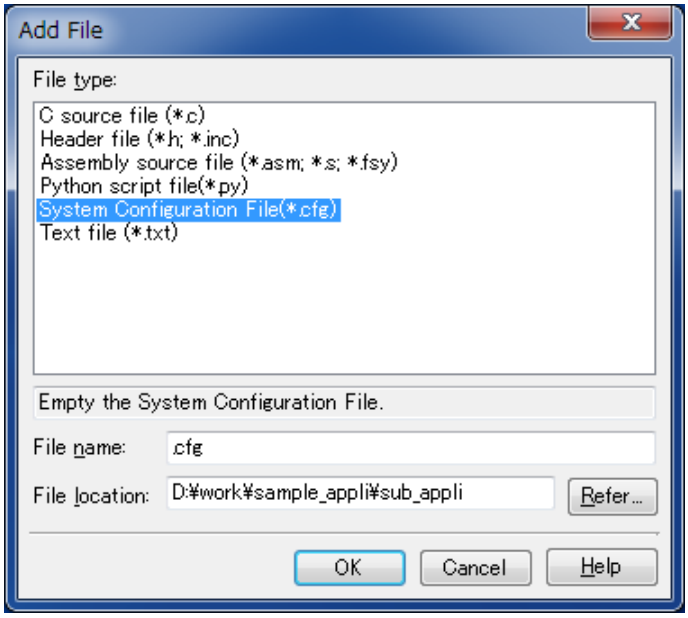
- [Microcontroller]: Select [RH850].
- [Kind of project]: Select [Application(RI850V4,CC-RH)].

A subproject can also be created based on an existing RI850V4 project. Refer to the steps described in "9.1.3 Recycle a RI850V4 V1 project".

3. Registering the system configuration file

Open the [Project Tree] panel for the project. Place the cursor over the subproject folder in the [Project Tree] panel and right-click it. Select [Add] → [Add New File] to open the [Add File] dialog box.

figure 9-6 [Add File] dialog box



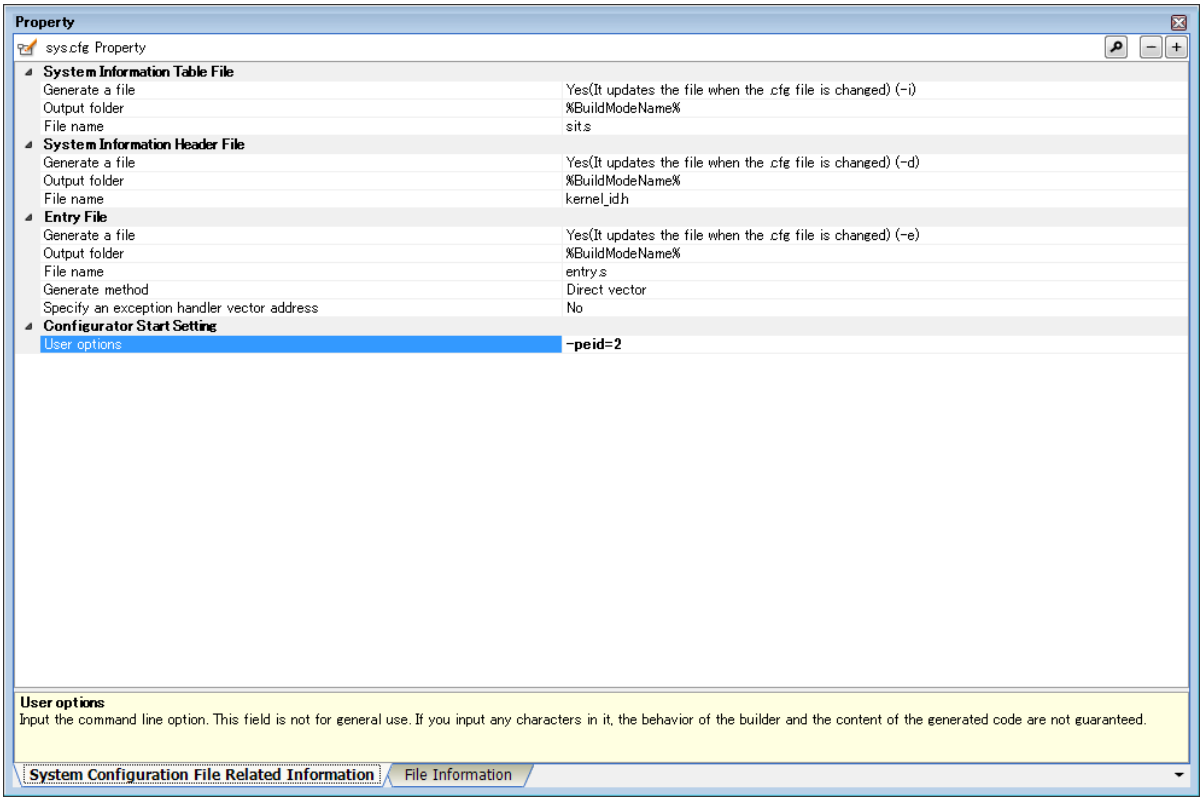
- [File type]: Select [System Configuration File(*cfg)].

Click the [OK] button to create a file.

4. Specifying the target PE number for the subproject

Open the [Project Tree] panel for the project. Place the cursor over the system configuration file registered for the subproject in the [Project Tree] panel and right-click it. Open the [Property] dialog box for the system configuration file.

figure 9-7 System Configuration File Related Information



- [User options]: Select -peid=<*id*>. Specify the target PE number for this project for *id*.

When using the RI850V4 for multiple PEs, repeat steps 2 to 4 described above.

When creating a project for a multi-core device, note the following.

A) In step 1, the boot processing (boot.asm) for a multi-core device is generated in addition to the boot loader for a multi-core device. This "boot.asm" that is generated automatically includes the definition of interrupt vectors. On the other hand, the information file that is generated by CF850V4 include the definition of interrupt vectors. Depending on select option, these are conflicted. For details, in the [Property] dialog box opened for the system configuration file, if [Direct vector] is selected for the [Generate method] item under [Entry File] and [No] is selected for the [Specify an exception handler vector address] item, the interrupt entry processing output by the configurator for the RI850V4 will be redundant because of the interrupt vector defined in the boot.asm processing. In this case, delete the interrupt vector processing in boot.asm. If a code is executed with these redundant settings left unchanged, the RI850V4 interrupt management facility will not operate correctly. For the necessary processes to be done in the boot processing, see "RI850V4 V2 Real-Time Operating System User's Manual: Coding".

B) In step1, When [Create a structure of startup files including main functions for each core.] is selected, a project that does not use the RI850V4 is generated. Do not select this option when using the

RI850V4. If a subproject is generated with this option selected, delete the subproject and follow the procedure from step 2 again to register a new subproject for the RI850V4.

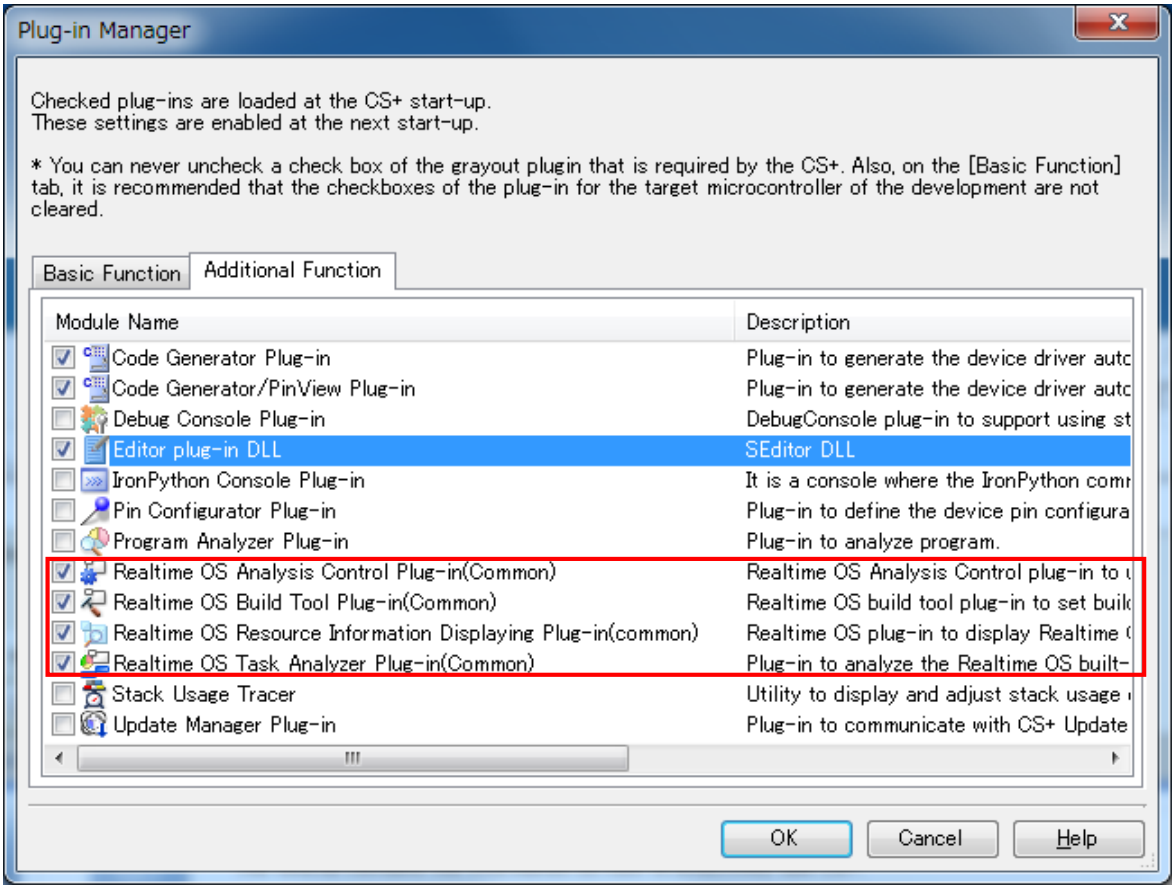
9.2. Enable Realtime OS Plug-ins

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)

figure 9-8 Plug-in Manager

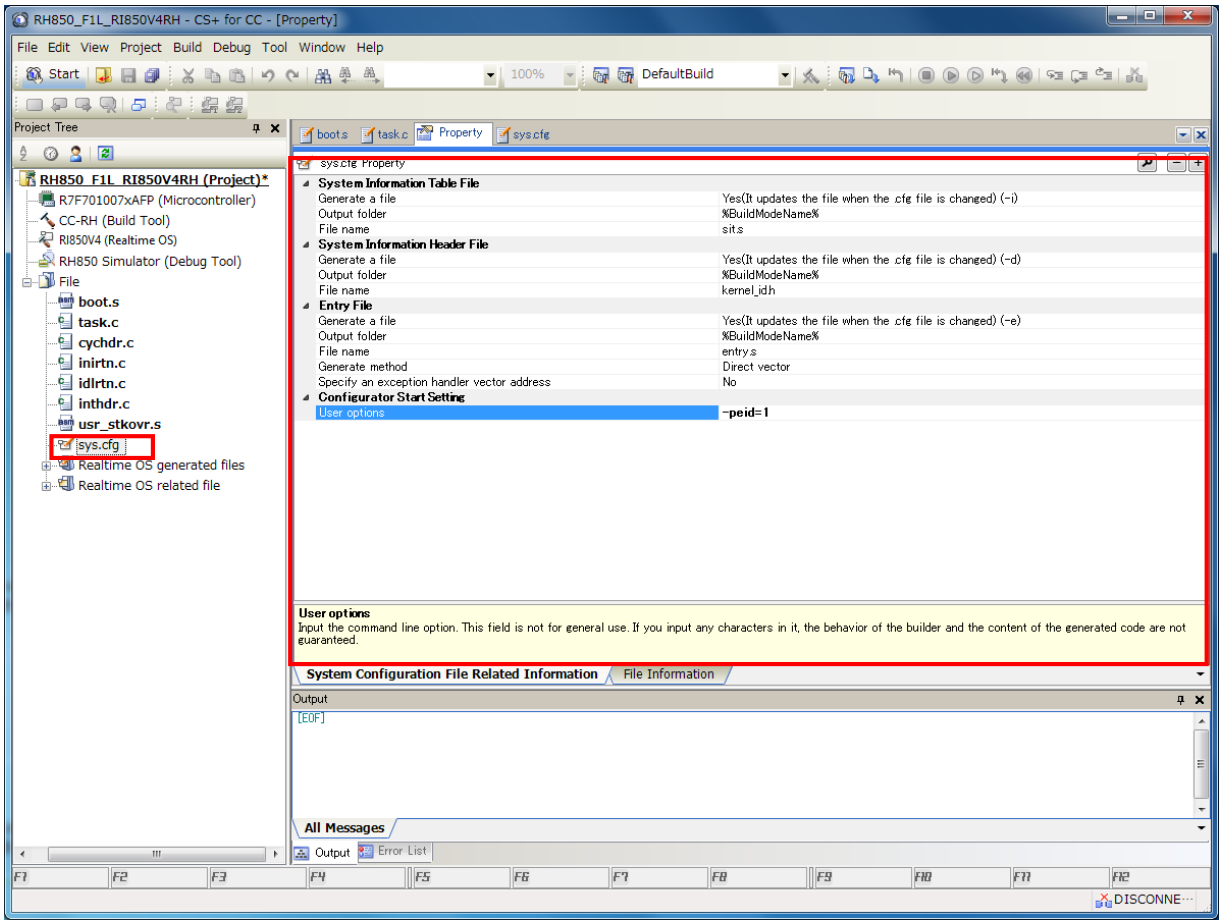


9.3. Property of Configuration file

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].

figure 9-9 Property of Configuration file



Setting items and contents of [System Information Table File] and [System Information Header File] are no difference from RI850V4 V1.00.

The following .setting items and contents of [Entry File] are no difference from RI850V4 V1.00.

- [Generate a file]
- [Output folder]
- [File name]

However, note that following setting items are added in RI850V4 for RH850.

- [Generate method]
- [Base address of the interrupt handler table]
- [Specify an exception handler vector address]
- [Exception handler vector address]

In [Generate method], there are the following two methods.

- [Direct vector]
- [Table reference]

If you select [Table reference], select [Base address of the interrupt handler table] in addition.

In [Specify an exception handler vector address], there are the following two items.

- [yes]
- [no]

If you select [yes], specify [Exception handler vector address].

In [Configurator Start Setting], there are the following methods.

- [user option]

In [user option], If you specify the option "-peid=<id>", you can select the PE number for this project.

9.4. Cautions for using simulator

9.4.1. About simulator setting file

Running an application on the simulator of CS+ requires a “rh850_simulator.setini” file for setting up the simulator. Place this file in the same directory as the CS+ project file (a file with the filename extension mtpj). This file is automatically placed in the correct location when you are using the sample project. If you have manually created a new project, copy the file from either of the following locations.

- If Windows® is 32bit, and system drive is C:
  C:\Program Files\Renesas Electronics\CS+\CC\SampleProjects\RH850\RH850_F1L_RI850V4RH
- If Windows® is 64bit, and system drive is C:
  C:\Program Files (x86)\Renesas Electronics\CS+\CC\SampleProjects\RH850\RH850_F1L_RI850V4RH

9.4.2. About OS timer(using simulator)

If you are using simulator, OS timer is limited by simulator restriction. For details, refer to simulator user’s manuals.

9.5. Cautions for Realtime OS Resource Information Panel

9.5.1. View after Realtime OS is initialized

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

9.5.2. 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.

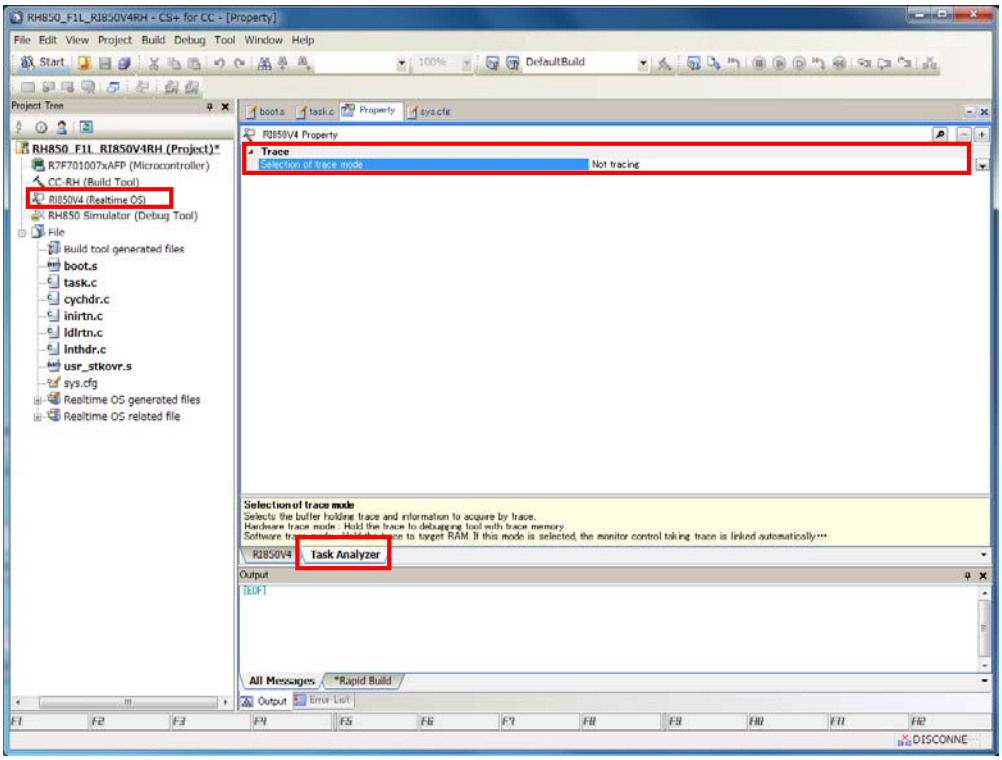
To generate debug information, under Build Tool, under the Link Options properties, set [Generate debug information] to [Yes].

9.6. Cautions for Realtime OS Task Analyzer Information Panel

9.6.1. To change the trace mode

Select [Selection of trace mode] on the [Task Analyzer] tabbed page of the [Property] panel for RI850V4. 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.

figure 9-10 Select Trace Mode on the Task Analyzer



If you select [Taking in trace chart by hardware trace mode] and you use simulator,

You need to specify the following settings correctly to obtain correct time on Realtime OS Task Analyzer plug-ins.

- [Main clock frequency [MHz]]

  Specify a suitable value to the [Main clock frequency [MHz]] in the [Clock] category in the [RH850 Simulator Settings] tab of the Property Panel for the Debug tool.
- [Rate of frequency division of trace time tag]

  Sets a frequency dividing rate to the [Rate of frequency division of trace time tag] in the [trace] category in the [RH850 Simulator Settings] tab of the Property Panel for the Debug tool.

figure 9-11 [Main clock frequency [MHz]] Tab in The property of RH850 Simulator（debug tool）

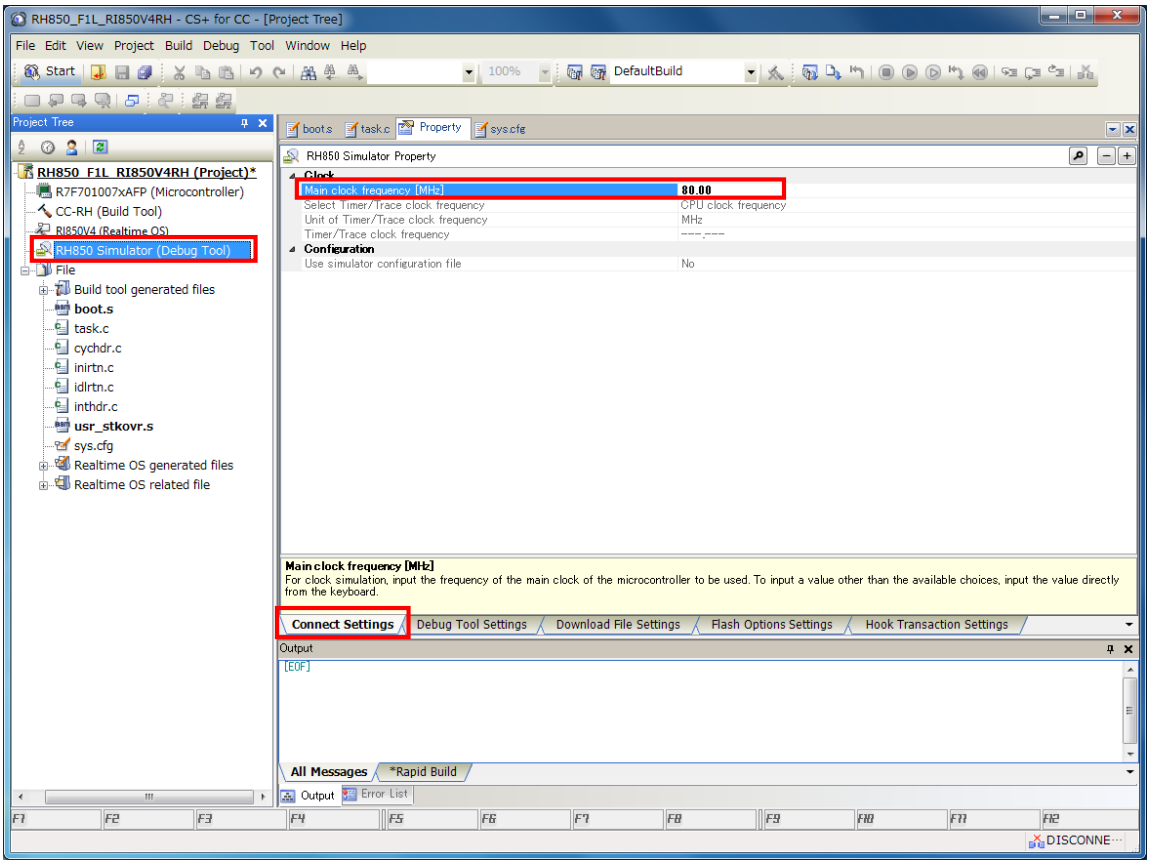
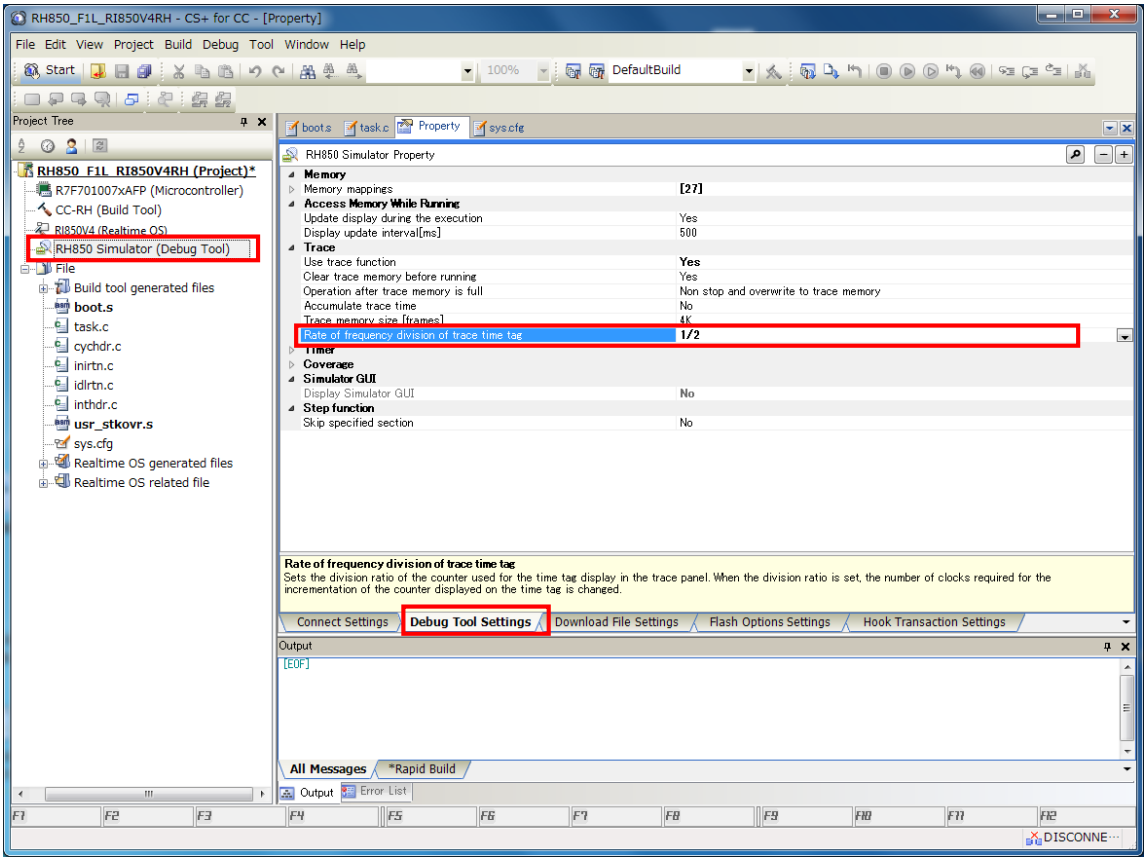
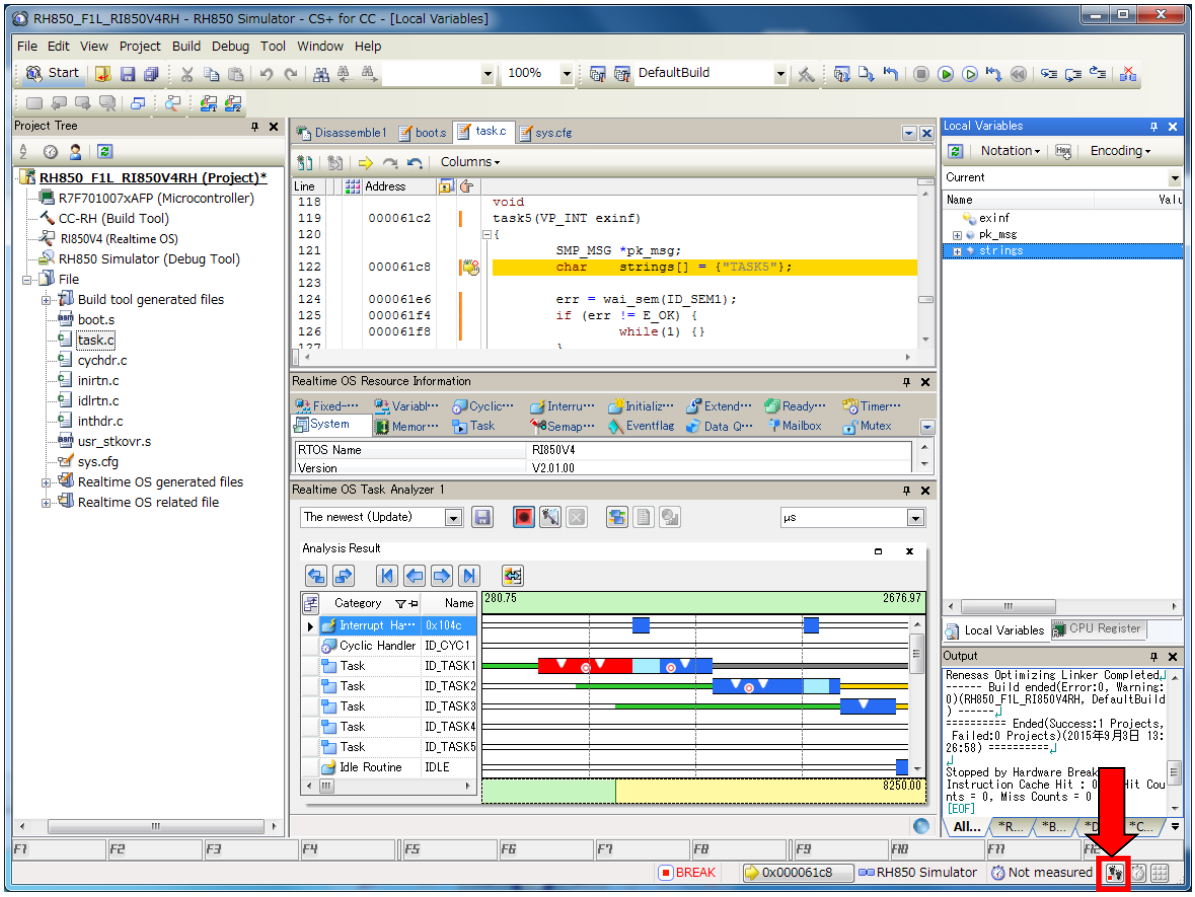


figure 9-12 [debug Tool] Tab in The property of RH850 Simulator(debug tool RH850）



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

figure 9-13 Enable The Trace Function of The Debugger



9.6.2. 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

9.6.3. 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).

9.7. How to build kernel source code

Since the RI850V4 kernel is provided in the library form, it does not usually need to build the kernel.

If you have “RTRRH8500TR01SRRUU(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\rh850_ccrh\r32”. This makefile presuppose that it runs on Cygwin, therefore, building the kernel, Cygwin installation environment is necessary.

- If your environment is Windows® 32bit

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

  ```
  make [return]
  ```

  And then, libraries will be generated in

  “< installation folder >\library\rh850_ccrh\r32”.

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\rh850_ccrh\r32 by a user who has the write privilege for the installation folder.

Please not that, in RI850V4 V2.00.01, kernel source code is not include product. This caution is not applicable.

9.8. Cautions for Using Variable Reset Vector Function

Some RH850 product supports “the Variable Reset Vector Function”. Therefore, RH850 can change the Reset Vector Base Address(RBASE). (For details, refer to Hardware user’s Manual.) However, RI850V4 does not support the Variable Reset Vector Function. Therefore, if RBASE was changed by the Variable Reset Vector Function, you can’t use the Interrupt Management Functions with RBASE.

If you use the Variable Reset Vector Function, specify “Exception Handler Vector Address(EBASE)” or “Base Address of The Interrupt Handler Table(INTBP)”(for details, refer to the section “9.3 Property of Configuration file“).

If you use both Variable Reset Vector Function and Interrupt Management Functions with RBASE, you need to modify the entry file(for details about this modification, refer to the section “9.8.1 Using Variable Reset Vector Function and Interrupt Management Functions with RBASE“).

9.8.1. Using Variable Reset Vector Function and Interrupt Management Functions with RBASE

As above, when you change RBASE by the Variable Reset Vector Function, the configurator can’t output the entry file on the basis of new RBASE. You need to modify the entry file manually.

The following is basic form of the entry file. Configurator output to the entry file in the following format. You need to modify *base_address*(the RBASE value into the new RBASE value).

```
.org    base_address + offset          -- setting branch address

jr      __kernel_int_entry             -- jump to interrupt process
```

Example：If you changed RBASE from 0x00000000 to 0x00010000, you need to modify entry file as follows.

- before

```
        ; 0x0140, call "priority 4"

        .org    0x00000000 + 0x140

        jr    __kernel_int_entry
```

- after

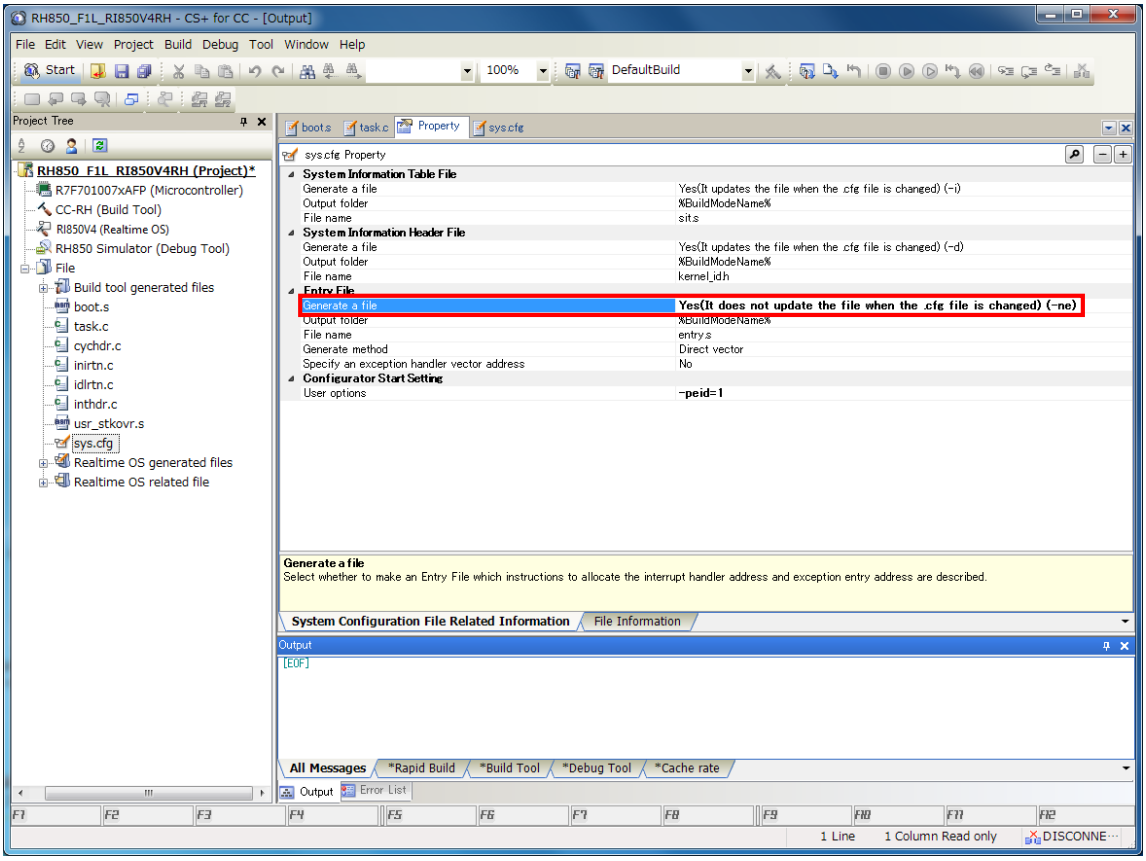
```
        ; 0x0140, call "priority 4"

        .org    0x00010000  + 0x140

        jr    __kernel_int_entry
```

The entry file is overwritten when you re-execute the configurator. Therefore, if you modify the entry file, you need to set the option about entry file.

figure 9-14 Set the option about entry file



- [Generate a file]: Select [Yes(It does not update the file when the .cfg file is changed) (-ne)]

9.9. Cautions of using E1/E20 emulator and select [take in trace chart by hardware trace mode]

If you use E1/E20 emulator and select [take in trace chart by hardware trace mode], the capacity for trace data is too little. For details, refer to user's manual for E1/E20 emulator.

In RI850V4 V2, it is recommend to use [take in trace chart by software trace model].

9.10. Cautions of using I/O header file generator

If you use I/O header file that is generated by the "I/O header file generator" and header file that are provided by RI850V4 V2 simultaneously, it may be occurred the multiple define. In this case, it is necessary to correct the macro in the I/O header file.

9.11. Cautions of setting frequency of the CPU clock

RH850-family have clock controllers to set the CPU clock frequency. (For details, refer to user's manual for hardware.) In the sample program for RI850V4 V2, the clock controller is not set, so the sample program runs on the default CPU clock frequency (8MHz).

When you change the CPU clock frequency, you have to set the value of the macro "KERNEL_USR_BASETIME" that is referred from the trace function. As the value of the macro "KERNEL_USR_BSETIME", specifies the time per count of OS timer. Time per count of OS timer can be calculated from the frequency of OS timer. The frequency of OS timer is half of the CPU clock frequency, so the frequency of OS timer is 4MHz, by default.

10. Restrictions

10.1. Restrictions of CS+ for CC

10.1.1. Realtime OS Build Tool Plug-in

（1） Display of temporary file names in the output pane of the IDE when the preprocessor is started up.

The configuration file registered with a project is input to the preprocessor of the compiler. If any macro definition in this configuration file has a syntax error, an error message is output with the name of a temporary file (cf850***.c) created by the configurator (instead of the name of the configuration file). Since the temporary file is deleted immediately after use, jumping from the output pane to the line that has the syntax error is not possible.

（2） 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.

（3） Utilizing existing projects

If you choose to recycle as the basis of a new project an existing project that does not contain any files such as sit.s which are generated by the configurator, and you select copy processing for the files you will be reusing, the missing files such as sit.s that are supposed to be grayed out in the project tree will be deleted from the project tree.

10.1.2. 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.

（2） Opening the [Realtime OS Resource Information] panel by selecting this from the [View] menu or clicking on the corresponding toolbar button.
The panel will not have focus when it is opened. Click on the panel to give it focus.

10.1.3. Realtime OS Task Analyzer Plug-in

（1） Swapping objects (by drag and drop) in the object information area.
In some cases, dragging and dropping of objects such as the names of tasks and interrupt handlers does not work. This problem can be solved by sorting the objects (in any way), after which dragging and dropping will work again.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	29/1/2016	-	New Publication
1.01	31/3/2016	15	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)

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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of 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. 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 should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended 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; and industrial robots etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may 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 should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 3.0