CS+ for CA, CX V4.04.00

Release Note

R20UT4666EJ0100
Rev.1.00
Nov 29, 2019

Contents

Chapter 1. Target Devices

The target devices supported by the CS+ are listed on the Website.
Please see this URL.
CS+ Product Page:

https://www.renesas.com/cs+

Chapter 2. User's Manuals

Please read the following user’s manuals along with this document.

Manual Name	Document Number
CubeSuite+ V2.02.00 Start	R20UT2865EJ0100
CubeSuite+ V1.03.00 78K0 Design	R20UT2138EJ0100
CubeSuite+ V1.03.00 78K0R Design	R20UT2137EJ0100
CubeSuite+ V2.01.00 RL78 Design	R20UT2684EJ0100
CubeSuite+ V1.03.00 V850 Design	R20UT2134EJ0100
CubeSuite+ V1.01.00 78K0 Debug	R20UT0731EJ0100
CubeSuite+ V1.01.00 78K0R Debug	R20UT0732EJ0100
CubeSuite+ V2.02.00 RL78 Debug	R20UT2867EJ0100
CubeSuite+ V2.00.00 V850 Debug	R20UT2446EJ0100
CubeSuite+ V2.02.00 Analysis	R20UT2868EJ0100
CubeSuite+ V2.02.00 Message	R20UT2871EJ0100

Chapter 3. Uninstallation

There are two ways to uninstall this product.

- Use the integrated uninstaller from Renesas (uninstalls all CS+ components)
- Use the Windows uninstaller (only uninstalls this product)

To use the Windows uninstaller, select [CS+ for CA,CX] from [Apps & features] from [Settings] of Windows or [Programs and Features] of the control panel.

Chapter 4. Improvements and changes

This chapter describes changes from V4.03.00 to V4.04.00.

4.1 Improvements and changes to CS+

4.1.1 End of support for Windows 7

Microsoft Corporation will end its free-of-charge support for Windows 7 at January 14, 2020.
In response to this event, Renesas will end the evaluating and guaranteeing the operation of CS+ in the Windows 7 environment and hereafter focus on support for Windows 10 and Windows 8.1.

Chapter 5. Points for Caution

This section describes points for caution regarding CS+.

5.1 Points for caution regarding CS+ (general)

5.1.1 File names

The following rules apply to folder and file names.

- Folder and file names
  Do not use folder or file names that cannot be created from Windows Explorer.
  Network path names cannot be used. Include the drive names to which paths are allocated.
- Source file names, load module file names, and project file names
  File names consist of the characters a-z, A-Z, 0-9, the period (.), the underscore (_), plus sign (+), and minus sign (–).
  File names cannot start or end with a period (.).
  Filenames cannot start with a plus sign (+) or minus sign (–).
  CS+ is not case-sensitive to file names.
  File names may have up to 259 characters, including the path.
  Do not use source files with the same file name. Even if they are on different paths, CS+ cannot classify them.
- File names other than the above
  File names comply with Windows conventions.
  Note that the following characters cannot be used in file names.
  \ / : * ? " < > | ;
  File names cannot start or end with a period (.) or space.
  CS+ is not case-sensitive to file names.
  File names may have up to 259 characters, including the path.
- Folder names
  Folder names comply with Windows conventions.

5.1.2 Panel display

If your hardware environment does not meet the recommended specifications for CS+, the [Property] panel may appear small and have scrambled contents.
If this happens, move the [Property] panel outside the split panel area.
- Enable [Dockable], and make it a docking panel
- Enable [Floating], and make it a floating panel

5.1.3 Changing microcontrollers

Note the following points for caution when changing the microcontroller.

- The microcontroller can only be changed to another within the same family, since this will correspond to the same build tool (V850,-RL78, 78K0R, and 78K0).
- When changing the microcontroller, do so while the debugger is not connected.
- Save the project before changing the microcontroller.
- Information on pin layout (design tool), code generation (design tool), and debugging (except for watch registration information) are not carried over after the microcontroller has been changed.

5.1.4 Plug-in Manager function

We recommend that the checkbox for the plug-in for the microcontroller that is the target for development is not deselected on the [Basic Function] tabbed page of the [Plug-in Manager] dialog box.
Deselect the checkboxes for the build tool and debugging tool plug-ins that are for microcontrollers that are not the target for development. For example, if only the plug-in for the build tool is deselected, the file to be downloaded by the debugging tool will not be found and an error will occur.

5.1.5 [Editor] panel

- The [Page Setup] dialog box is not available.
- Although there is a [Copy] button on the [Print Preview] toolbar, it does not work.

- When a variable or label is selected and the Jump to Function feature is used from the context menu, execution does not jump to the variable or label.
- The Jump to Function feature will not jump to a static function defined in another file.
- The following notes apply to the editor, when source files with the same name but from different folders are registered with a main project and a sub project, and load modules from both the main project and sub project are downloaded.
  - The address of the main project is displayed on the file.
  - At jumping to a source file from disassembled code, the file registered with the main project is opened.
  - If the file is opened from either project, only one file will be opened.
- The smart edit function does not work correctly for a structure that does not have a name.
- If the arguments of a function include a function call, incorrect information will be displayed in the tooltip.
- Names of member variables and functions are not correctly supplemented by the smart edit function in arrays of classes and of pointers to classes.
- Supplementation does not produce appropriate strings even if a part of a member name is entered and 'ctrl + (space)' is also entered.
- Outlining (collapse/expand) only works with files that have been registered with the project. Any files that have not been registered with the project will not be outlined in the [Editor] panel of CS+.
- On Windows 8.1, the display may become unclear due to anti-aliasing.
- In mixed display mode, if a line number is specified for a jump, the disassembled code is inserted and displayed, so the specified line may not be displayed on the screen.
- If a structure is nested, the smart edit function does not work on the third and subsequent stages. In addition, information is not displayed in the tooltip.
- For the first line of code immediately after ‘#ifdef - #endif’, the smart edit function does not show candidate members. In addition, no tooltip is displayed.
- For the first variable in ‘#ifdef - #endif’, the smart edit function does not show candidate members for the code after ‘#endif’. In addition, no tooltip is displayed.
- When a file not included in a project is opened, a bookmark is set, the project is closed, the setting for the bookmark in the file is changed, the project is opened again, and the [Bookmarks] dialog box is opened, the dialog box will display the bookmark with the setting when the project was closed while the source window displays the bookmark with the new setting that was made after the project was closed. That is, different settings are displayed for the bookmark in the dialog box and source window. In such cases, close the file and then open it again. The display of the bookmark in the [Bookmarks] dialog box will then reflect the display in the source window.
- When the rectangular selection tool (obtained by holding the Alt key down and then using the mouse to select an area) is used and the selected area is pasted to add lines in and after the last line, the selected area is pasted from the beginning of the line after last regardless of the location specified for pasting. Insert spaces after pasting as required.
- When saving a file in the [Save As] dialog box, the extension listed at the top of the [Save as type] drop-down list is automatically added unless another extension is specified. Note however, that an extension is not added when a file name is input with an extension that is selected in the [Save as type] drop-down list or with an extension that is registered with Windows. When an automatically added extension is not as expected, modify the name of the file by using, for example, Explorer.

5.1.6 Conversion from PM+ to CS+ projects

CS+ cannot read CA850 projects made by PM+ V6.00/V6.10/V6.11 then the Build Mode has been newly added. Handling by CS+ is as follows.

1) When [Debug Build] or [Release Build] is specified, information on the added Build Mode cannot be read.
2) When the added Build Mode is specified, it leads to an error.

[Workaround]
Use V6.20 or a later version of PM+ to read the project, save it, then read the project into CS+.

5.1.7 Debugging tool settings during project appropriation

When you create a project by appropriate settings from another project, only the settings for the default debugging tool will be imported.

5.1.8 Help

If you close the help while the Search tab is being displayed and you then open the help again and display the [Contents] tab, the Coding and Build sections may disappear.
If this happens, close the help while the [Contents] tab is being displayed, and then open the help again.

5.1.9 Changing the target device during the process of converting a project

When the target device is changed in the [Project Convert Setting] dialog box while it is open for conversion of a project created by using the High-performance Embedded Workshop, PM+, or an earlier version of CubeSuite, the value chosen in [Kind of project] is returned to the default value (top type in the combo box).

For example:
The setting for [Kind of project] changes to the default type (for example, [Application]) when the device selection is changed.

5.1.10 Converting High-performance Embedded Workshop projects

Attempting to load a High-performance Embedded Workshop project into the CS+ under certain conditions may not be possible, or may lead to an error during conversion or building of the project.

(1) Converting a High-performance Embedded Workshop project to make it compatible with the CS+ fails when any of the following conditions is satisfied.

- No toolchain from Renesas Electronics Corp. is selected for the project.
- The project contains no tps file for use in setting up the High-performance Embedded Workshop environment (the tps file is automatically created when the project is opened through the High-performance Embedded Workshop). To avoid this problem, you should open the project through the High-performance Embedded Workshop once before starting conversion.
- The project contains multiple CFG files, each of which is used to set up a realtime OS from Renesas Electronics Corp.

(2) Converting a High-performance Embedded Workshop project to make it compatible with the CS+ succeeds but building of the project leads to an error when any of the following conditions is satisfied.

- Placeholder $(TCINSTALL) is used in the project.

  $(TCINSTALL) remains in the project even after conversion but the CS+ does not recognize $(TCINSTALL). Placeholder $(TCINSTALL) that has been used as a parameter for [Options] in the High-performance Embedded Workshop is simply passed to the CS+ and may cause an unintended result (e.g. an error) upon building of the project. For this reason, you should manually change $(TCINSTALL) after converting the project.

- Placeholder $(WORKSPDIR) is used in the project.

  If you select a HEW project file (with extension hwp) in the CS+, this is automatically converted to “%ProjectDir%\..” (the directory above the project directory). An error may occur during building of the project if the workspace does not exist in the directory indicated by “%ProjectDir%\..”.
  For this reason, you should manually change “%ProjectDir%\..” after converting the project.

- A custom build phase is used in the project.

  Since all custom phases are deleted upon conversion, an error may occur during building of a project that involved a file output created for a custom build phase in the High-performance Embedded Workshop.
  After converting the project, register the custom build-phase command with the CS+ as a command to be executed before or after each phase as required.

- A custom placeholder is used in the project.

  Custom placeholders are not converted because the CS+ does not recognize them. Any custom placeholder that has been used as a parameter for [Options] in the High-performance Embedded Workshop is simply passed to the CS+ and may cause an unintended result (e.g. an error) upon building of the project. For this reason, you should manually change the custom placeholders after converting the project.

(3) Other

(a) $(FILEDIR) is converted to %FileDir%.
Leaving %FileDir% as it is when the pathname is edited in the [Path Edit] dialog box will lead to the following error: The specified path contains a non-existent folder. (W0205012)
Edit the pathnames and replace %FileDir% with another placeholder or directory name.

(b) $(WINDIR) is converted to %WinDir%.

(c) The order in which folders are displayed in CS+ may differ from that in the High-performance Embedded Workshop.

(d) If a High-performance Embedded Workshop project for which downloaded files have been specified is loaded into CS+, CS+ will show these files as the second and subsequent items in the list of downloaded files for each debugging tool.

(e) The compiler option -output=src is converted to –output=obj (default).

(f) If you load a library project that has been linked to the standard library into CS+, the linkage setting will be discarded (this is indicated in the log information that is output as a result of loading the project).

(g) If [Use an existing library file] has been selected for the library generator in the High-performance Embedded Workshop, the setting is changed to [Do not add a library file] in CS+. For this reason, linking with the specified library will not proceed (this is indicated in the log information that is output as a result of loading the project).

(h) Option settings that were made on the [Toolchain Option] tabbed page of the High-performance Embedded Workshop are not converted but discarded (i.e. they are not moved across to CS+).

(i) If a sub-command file has been selected for the linkage editor in the High-performance Embedded Workshop, the [Use external subcommand file] setting is discarded when the project is loaded into CS+. The linkage editor options will have their default settings.

(j) Any files specified with the -library, -input, or -binary option will not be listed in the [Link Order] dialog box. The result is that the order of linkage for these files will not be selectable.

(k) RTOS configuration files will not be displayed under the [Configuration file] category node after the project is loaded into CS+.

(l) RTOS option settings that were made in the High-performance Embedded Workshop are discarded. RTOS options will have their default settings in CS+.

(m) The build mode for RTOS projects will be "DefaultBuild" after the project is loaded into CS+. You will need to change the build mode as required.

(n) The order of linkage of the assembly output files (ritbl.obj) for an RTOS project will differ from that in the High-performance Embedded Workshop.

5.1.11 Tutorials

The Code Generator Plug-in, Pin Configurator Plug-in and Program Analyzer Plug-in are used in tutorials. Enable them through the [Plug-in Manager] dialog box.

5.1.12 Starting multiple instances of CS+

Two or more instances of CS+ can be started on the same host machine, but if you do so, take note of the points listed below.

- When two or more instances of CS+ are started, the most recent information to have been written is saved in the information file for each user’s own PC.
- When two or more instances of CS+ are started, the most recent information to have been written is saved in the information file for the stack analysis tools (including CallWalker).
- When the same project file is used in two or more instances of CS+, the most recent information to have been written is saved.
- When the same project file is used in two or more instances of CS+, do not attempt building from more than one instance at the same time since the names of the output files will be identical.
- When project files for the same build tool are used in two or more instances of CS+, do not attempt building from more than one instance at the same time. This is because the folder in which files are temporarily saved during building (a temporary folder) will be shared. Note, however, that building at the same time becomes possible if options are used to assign different names to the temporary folders. (Applicable build tools: CA78K0, CA78K0R, CA850, and CX)

5.1.13 Options when starting CS+

Although the options below can be specified when starting CS+W.exe, do not specify them. If specified, they will prevent the edit function from operating normally.

- /npall
- /np to specify an editor

5.1.14 Loading projects by using earlier version of CS+

If the version of CS+ being used to load a project is earlier than the CS+ version with which the project was created, some settings may be cleared since the earlier versions of CS+ do not recognize them.

5.1.15 .NET Framework from Microsoft Corporation

CS+ outputs the following message and is closed if the version of the .NET Framework you are using is earlier than 4.5.2.

```
E0200010
  Failed to launch this product.
  Please install the Microsoft .NET Framework 4.5.2 or later on this PC.
```

In such cases, obtain version 4.5.2 or a later version of the .NET Framework from the Web page of Microsoft Corporation and install it before starting up CS+ for CA,CX.

Note that the update manager for CS is not usable for updating the .NET Framework.

5.1.16 Mentions of “R8C” in user’s manuals and online help

“R8C” is mentioned in the user’s manuals and online help, but CS+ does not support the R8C family.

5.2 Points for caution regarding build tools

5.2.1 Build mode

In the [Build Mode Settings] dialog box, the build mode can be duplicated or renamed with different names in terms of upper- and lower-case letters from those of the existing name of the build mode.
Usually, the name of a build mode is that which has been specified for the [Intermediate file output folder] property on the [Common Options] tabbed page of the [Property] panel of the build tool or that which has been specified for the [Output folder] property on the [Link Options] tabbed page or [Hex Output Options] tabbed page. Note that the output destinations of two build modes having names that differ in terms of upper- and lower-case letters will be the same unless the following step is taken. Avoid this problem by specifying different folder names for the [Intermediate file output folder] and [Output folder] properties.

5.3 Points for caution regarding design tools

5.3.1 Changing packages

If you change the package name in the pin layout properties, the data input in the device top view and device pin list will be cleared.

5.3.2 Saving projects

If you save a project that has sub-projects while the [Device Top View] or [Device Pin List] panel is open, then the device top view and device pin list of the last sub-project in the Project Tree will always appear.

5.4 Points for caution regarding debugging tools

The abbreviations listed below collectively denote the corresponding tools in this section.
OCD (serial): MINICUBE2, E1 Emulator (serial), E20 Emulator (serial)
OCD (JTAG): MINICUBE, E1 Emulator (JTAG), E20 Emulator (JTAG)

5.4.1 Adding sub-projects

Applies to: All debugging tools, common to all devices
Disconnect the debugging tool before adding a sub-project that handles a different device from the main project.

5.4.2 Executing a boot swap

Applies to: Simulator, OCD (JTAG), and OCD (serial) for V850, 78K0, 78K0R, and RL78
If a software break is set in a boot-swap area, then the break instruction will be written to the Flash ROM. Accordingly, the break instruction will remain after the boot swap.

- OCD (JTAG) and OCD (serial): Use a hardware break if you wish to set a breakpoint.
- Simulator: Do not use breakpoints in this area.

5.4.3 Standby mode

Applies to: All debugging tools for V850, 78K0, 78K0R, and RL78
If a forced break is applied while on standby (e.g. in STOP mode or HALT mode), or an instruction for a transition to standby mode is issued during stepped execution, the behavior of the simulator will differ from that of the emulators (IECUBE, OCD (JTAG), and OCD (serial)).

- Emulator: The forced break will lead to release from standby mode. In stepped execution, the target device will not go into standby mode.
- Simulator: The forced break will not lead to release from standby mode. In stepped execution, the target device will go into standby mode.

In either case, the program counter (PC) line upon a forced break will break at the next instruction after the standby mode instruction (e.g. HALT). Thus in the case of the simulator, it will appear that the device has been released from standby mode. Check the status bar to see if the device has been released. If the simulator is in standby mode, "Halt" or "Standby" will appear on the status bar.

5.4.4 Multipliers/dividers

Applies to: Simulator for 78K0

The simulation of 78K0 instructions does not support those for the multiplier and divider. For this reason, to perform multiplication or division within the program, open the [Property] panel from the build tool, and select [No] from the [Use multiplier/divider] drop-down list on the [Compiler Options] tabbed page.

5.4.5 Memory banks

Applies to: Simulator for 78K0

The simulation of 78K0 instructions does not support the memory bank function.

5.4.6 CPU operating clock

Applies to: Simulator for 78K0R and RL78

- In the simulation of 78K0R instructions, the frequency of the on-chip oscillator is fixed to 8 MHz.
- In the simulation of RL78 instructions, the CPU operating clock operates in accord with the RL78/G13 specification.

5.4.7 Multiplier and divider/multiplier-accumulator

Applies to: Simulator for 78K0R and RL78

In the simulation of 78K0R and RL78 instructions, the following points for caution apply to instructions for the multiplier and divider and multiplier-accumulator.

(1) When the multiplier and divider is used for division, the division processing will be finished in 1 clock cycle.

(2) When the multiplier and divider is used for division, it does not generate an INTMD interrupt to indicate the completion of division. However, the value of the DIVST bit in the multiplication/division control register (MDUC) does change (the DIVST bit indicates the state of division operations).

5.4.8 Traces over desired intervals

Applies to: Simulator for all devices

If you perform a trace from a trace start event until a trace end event, the simulator will not include the trace end event in the results of the trace. For this reason, if you are using a simulator, set the trace end event one line below the range for which you require display of the trace data.

5.4.9 Measuring execution times over desired intervals

Applies to: Simulator for V850, 78K0, 78K0R, and RL78

If you measure execution time from a timer start event until a timer end event, the simulator will not include the time taken by the timer end event in the measured results. For this reason, if you are using a simulator, set the timer end event to one line below the range for which you require measurement of the execution time.

5.4.10 Displaying maximum address space in the [Memory] panel

Applies to: OCD (serial) and IECUBE for 78K0

To access the device maximum internal ROM, internal fast RAM, or internal extended RAM sizes from the [Memory] panel and so on, use a hook process to set the memory size switching register (IMS) and the internal expansion RAM size switching register (IXS).

5.4.11 Executing return from interrupt handling and displaying the call stack

Applies to: All debugging tools for 78K0R and RL78

When stepped execution is being controlled from the [Editor] panel (in source mode), the debugging tool uses the NP, EP, and ID flags in the POWER switch register determines whether an interrupt is being processed. For this reason, if the above flags or register are changed (e.g. when using multiple interrupts), then the execution of return from the interrupt and display of the call stack may be incorrect.

5.4.12 Adding sub-projects

Applies to: Common to all debugging tools and devices

If you add a sub-project while a debugging tool is connected, downloading and so on may fail. Only add sub-projects while the debugging tool is disconnected.

5.4.13 Configuring flash options

Applies to: OCD (JTAG) for V850E2M

The bits listed below indicating the corresponding flash options are fixed to 1. Use a flash programmer if you wish to write 0 to the bits.

- Bit 95 of on-chip debugging security ID (security lock signal release)
- Bit 31 of option byte 0 (debug interface connection disabled bit)

5.4.14 Stack-trace display

Applies to: All debugging tools for 78K0

The stack-trace display may fail to correctly display information up to the **main** function if a function that does not push the frame pointer (HL) onto the stack is called (e.g. the **noauto** and **norec** functions), or if the memory banks are used.

Additionally, the program may keep running following return from a function that does not push the frame pointer (HL) onto the stack (e.g. **noauto** or **norec** function) or the use of memory banks.

5.4.15 Using step-in across memory banks

Applies to: All debugging tools for 78K0

Stepping in at the source level across memory banks to a user-defined library function or function that does not have debugging information leads to a break in execution within the bank-switching library.

5.4.16 Display of local variables

Applies to: All debugging tools for 78K0

Local variables other than those in the same scope as the line corresponding to the current PC value are not displayed correctly in the [Stack Trace] panel.

5.4.17 [Disassembly] window

Applies to: All debugging tools for 78K0

When displaying instructions in the common area in the [Disassembly] window, if the displayed instruction uses a symbol in a memory bank area, a symbol from a different bank may be displayed.

5.4.18 Breakpoints and other settings becoming invalid

Applies to: Common to all debugging tools and devices

If you use leading underscores to differentiate function or variable names, the debugger may recognize them incorrectly and change the symbols or invalidate breakpoint settings.

This applies in cases such as when you have two functions, one named _reset and the other named __reset.

5.4.19 Clock generating circuit

Applies to: Simulators for 78K0

The simulation of the clock generating circuit is in accord with the specification of the 78K0/Kx2.

5.4.20 Data Flash

Applies to: Simulators for RL78 and 78K0R

The data flash memory is not supported.

The data flash area is handled in the same way as normal RAM.

5.4.21 Code Flash

Applies to: Simulators for RL78 and 78K0R

Fetching from the code flash area and the RAM area is possible except for the last four bytes, for which fetching is not possible.

Example: When the code flash area is from 0x0 to 0x1ffff, this restriction applies to 0x1fffc to 0x1ffff.

5.4.22 Pipelining

Applies to: Simulators for RL78 and 78K0R

Pipelining is not supported.

5.4.23 Voltage for target system during breaks

Applies to: OCD (JTAG) and OCD (serial) for 78K, RL78, V850, and V850E2M

Do not decrease the voltage for the target system during a break.

Generation of a reset by the low-voltage detector (LVI) or by power-on-clear (POC) during a break leads to incorrect operation of the debugger or communication errors.
This also applies to switching power to the target system off in a break during emulation; this leads to errors in communications.

5.4.24 Debugger's [Property] panel

Applies to: OCD (serial) for V850
The Open-Break function cannot be enabled or disabled on the [Debug Tool Settings] tabbed page of the [Property] panel.
Please use the Python Console when you need to enable or disable the Open-Break function.

5.4.25 Load modules generated by the GHS compiler

Applies to: IECUBE, OCD (JTAG), and OCD (serial) for V850
Specifying the scope of an undefined function for a defined valuable does not always lead to an error. Do not use scope specifications with undefined functions.

5.4.26 Writing new values to option bytes

Applies to: OCD (Serial), RL78
Reset the CPU after writing a new value to an option byte by using, for example, the [Memory] panel. The debugger might not operate normally if this is not done.

5.4.27 DMA display of trace panel

Applies to: IECUBE for RL78 and 78K0R
When there is access to SFR/ memory by DMA, character string of "DMA" is not displayed in the trace panel. ("Address" and "Data" of the access to SFR/memory by DMA are displayed correctly.)

5.4.28 Break causes display of trace panel

Applies to: OCD (Serial) for RL78 and 78K0R
When a program was halted, the break causes displayed on the trace panel may not be correct. Please confirm the break causes by the status message of the main window.

5.4.29 Simulation of peripheral modules

Applies to: Simulator for V850ES/Sx2, V850ES/Jx2, V850ES/Fx3, 78K0R/Lx3, 78K0R/Kx3, 78K0R/Ix3, 78K0/Kx2, RL78/G10
Operation of CS+ may be incorrect if the user program or a debugging operation makes settings of the IOR and SFR which are prohibited in the user's manual for the target device.

5.4.30 Release from the HALT mode in the peripheral simulator

Applies to: Simulator for 78K0R/Lx3, 78K0R/Kx3, 78K0R/Ix3, RL78/G10
In the simulator, the user program is released from the HALT mode in response to any of the following debugger operations.

- Changing the frequency of the main clock or sub-clock on the [Connect Settings] tabbed page of the Property panel while the user program is stopped
- Writing to the registers controlling clock generator
  However, in the case of RL78/G10, it is not applicable when the same value is written to the high-speed on-chip oscillator frequency selection register (HOCODIV) in the clock generator

5.5 Points for caution regarding analysis tools

5.5.1 [Analysis Chart] panel

- When a simulator is specified as the debugging tool in a graph of transitions in values, realtime sampling of IORs and SFRs is not supported.

5.6 Points for caution regarding the Python Console

5.6.1 Japanese input

The Japanese input facility cannot be activated from the Python Console. To enter Japanese text, write it in an external editor etc., copy it, and paste it into the console.

5.6.2 Display of the prompt

The Python Console prompt “>>>” may be displayed more than once, as in “>>>>>>”, results may be displayed after the “>>>”, or the caret may appear without a preceding “>>>” prompt. Entering functions is still possible in these situations.

5.6.3 Paths to folders and files

IronPython recognizes the backslash character (\) as a control character. For example, if a folder or file name starts with a “t”, then the sequence “\t” will be recognized as a tab character. Please use r + "path_name" to avoid this.

Example: r"c:\test\test.py"

A forward slash (/) can be used instead of a backslash (\).

5.6.4 Executing scripts for projects that do not have load modules

If a script is specified in the startup options for use with a project that does not have a load module file, or if project_file.py is placed in the same folder as the project file, then although the script would have been executed automatically after normal loading of the project, it will not be executed if there is no load module file.

5.6.5 Forced termination

The following actions while a script such as an infinite loop is running may lead to the results of function execution being in error because the actions forcibly terminate the execution of functions.

1. Forcible termination by selecting “Forcibly terminate” from the context menu or pressing Ctrl+D in the Python Console
2. Changing the active project in a project with multiple projects

5.6.6 Forced stopping

Executing “Abort” from the context menu will forcibly stop an executing script or function, but hook and callback functions that had not started at the time the “Abort” was executed will still be executed in order after that.

5.6.7 Executing Python commands during building

Do not issue Python commands while building is in progress.

Chapter 6. Restrictions

This section describes restrictions on CS+.

6.1 Restrictions imposed by debugging tools

The abbreviations listed below collectively denote the corresponding tools in this section.
OCD (serial): MINICUBE2, E1 Emulator (serial), E20 Emulator (serial)
OCD (JTAG): MINICUBE, E1 Emulator (JTAG), E20 Emulator (JTAG)
OCD (LPD): E1 Emulator (LPD), E20 Emulator (LPD)

6.1.1 List of restrictions imposed by debugging tools

No.	Target tool	Target device	Description	Remarks
1	OCD (serial) OCD (JTAG) OCD (LPD)	V850E2M V850E2S	Restriction on flash options	
2	OCD (serial)	RL78	Restriction on stopping the high-speed on-chip oscillator during subsystem clock operation when the target is the RL78/G14.	This restriction has been removed in CS+ V2.02.00.
3	OCD (serial)	RL78 (except the RL78/G10)	Restriction on writing new values to data flash memory.	This restriction has been removed in CS+ V3.00.00.

6.1.2 Details of restrictions imposed by debugging tools

No.1 Flash options

Applies to: OCD (serial) , OCD (JTAG) and OCD (LPD) for V850E2M and V850E2M

Description: All security settings and boot-block cluster settings of the [Flash Option Settings] property are invalid.

Workaround: There is no workaround.

Chapter 7. Changes to the User's Manuals

This section describes changes to the user's manuals for CS+.

7.1 Changes related to the editor

This section describes additional functions of the editor. The editor is described in the Coding and Debugging editions.

7.1.1 Added description of toolbar

[Before addition]

None

[After addition]

	This toggles between normal and mixed display modes when the debugging tool is connected. In normal display mode, the source program is displayed. In mixed display mode, the source program and assembly code are displayed.
	This toggles between stepped execution at the source and assembly levels when the debugging tool is connected and the display is in mixed mode. When stepped execution is at the source level, the program counter indicates the location of the current line in the source program. When stepped execution is at the assembly level, the program counter indicates the line in the assembly code.
⇨	This displays the current position of the program counter (PC) when the debugging tool is connected and the display is in mixed mode.
↷	This jumps to the position before execution of the **Back to Last Cursor Position** command.
↶	This returns to the position before execution of the **Jump to Function** command.
Columns ▾	This is used for toggling between showing and hiding columns. Clicking on this indicates the columns for which display can be toggled.

7.1.2 Added description of column headers

[Addition]

The following information items are displayed in the respective columns.

Display on the column header	Description of the items
Line	Line numbers for the files being displayed
(No display)	The display is colored to reflect the state in terms of saving of the state of editing. Yellow: The line has been newly created or changed but the file has not yet been saved. Green: The line has been newly created or modified and the file has been saved. This column is displayed except in the mixed display mode.
(No display)	The display is colored to reflect cases where a source file has been updated more recently than the corresponding load module file. The display in this area is cleared when the load module file is downloaded again after executing a build. This column is displayed if the display is in normal mode and a debugging tool is connected.
▦	When the coverage function is enabled, the display is colored to reflect the results of code coverage measurement. This column is displayed while a debugging tool is connected.
Address	Addresses in the microcontroller This column is displayed when a debugging tool is connected.
Op code	Instruction codes This column is displayed if the display is in mixed mode while a debugging tool is connected.
Label	Labels This column is displayed if the display is in mixed mode while a debugging tool is connected.
▣	Items in this column indicate the settings of events such as timer and trace events by the context menu on lines having an address display. When an event has been set, an event mark indicating its type is displayed here. Trace event enabled Trace event disabled Action event enabled Action event disabled This column is displayed while a debugging tool is connected.

☞	When a debugging tool is not connected: Items in this column are bookmarks or marks which indicate error and warning messages. Marks indicating error and warning messages are only displayed for source files registered in a project. The marks are not displayed immediately after a project is opened but are displayed after building. When cleaning is performed, all marks indicating error or warning messages are deleted. When building is repeated, a warning message which was displayed for the first build might not be displayed as a result of building the second and subsequent times. In such cases, the marks are not displayed in the [Editor] panel, either. Marks indicating error or warning messages do not reflect editing of the source file (changes in line numbers). When the mouse cursor is placed over a mark indicating an error or warning message, the message output by the compiler or assembler is displayed in a pop-up box. ⊗ The line led to an error message. ⓘ The line led to a warning message. ▭ Bookmark When the debugging tool is connected: This column displays the breakpoint enabled/disabled state, current PC location, bookmarks, and address marks. On lines where an address mark is shown, this column is used to set breakpoints. This column is manipulated from the context menu in most cases, but to set or clear a breakpoint, click on this column with the left mouse button. If a breakpoint has been set, an event mark is displayed here. A mark indicating the current PC location is also displayed if the current PC location (value of the PC register) is within the active panel (only in the state where the program is halted). ● Breakpoint enabled ☒ Breakpoint disabled ➡ Current PC location ▭ Bookmark ▌ Address mark

7.1.3 Added description of dragging and dropping

[Before addition]

None

[After addition]

You can add symbols to the [Watch] panel or [Analysis Chart] panel by dragging and dropping them.

7.1.4 Added description of highlighting

[Addition]

- Highlighting the current line

  The current line (where the caret is located) can be displayed within a rectangle.
  This feature can be enabled on the [Text editor] page of the [Option] dialog box.

- Emphasizing brackets

The editor is capable of emphasizing the bracket that corresponds to a bracket at the caret position. Supported types of brackets vary with the file type.

File Type	Types of Brackets
C or C++	(and), { and }, [and]
Python	(and), { and }, [and]
HTML or XML	< and >

When CS+ searches for the corresponding bracket, it does not consider those within comments, character constants, character strings, or string constants. For this reason, if the bracket at the position of the caret is within a comment, character constant, character string, or string constant, CS+ may emphasize a bracket that is not actually the corresponding bracket.

7.1.5 Added description of zooming in and out

[Addition]

The display in the editor is in accord with the zoom ratio selected from the toolbar of CS+. When you select a zoom ratio, it is applied to the panel that is currently in focus. If you close the panel and open the same file again, the zoom ratio will return to 100%. Separate zoom ratios can be selected for the normal mode and the mixed display mode (with the debugging tool connected).

7.1.6 Added description of smart edit functions

[Addition]

The smart edit function is used to supplement the names of functions, variables and the parameters of functions during their entry and offer candidates for the names.

This function is only supported if the project is for the V850E2. [V850E2]

The function operates with the items listed below.

- Global functions in the C language and C++ language
- Global variables in the C language and C++ language
- Class member functions in the C++ language
- Class member variables in the C++ language
- Multiple overloaded functions in the C++ language

To display candidates for functions and variables for the smart edit function, operate the program as described below.

(1) When the automatic display of candidate names is selected

- In the C language and C++ language, if there are relevant members for the name on the left side when '.' is input, candidate names are automatically shown at the location of the text cursor.
- In the C language and C++ language, if there are relevant members for the name on the left side when '->' is input, candidate names are automatically shown at the location of the text cursor.
- In the C++ language, if there are relevant members for the name on the left side when '::' is input, candidate names are automatically shown at the location of the text cursor.
- If there are candidate parameter names for the method (function) on the left side of '(' when '(' is input, they are automatically shown.

(2) When candidate names are shown in response to keyboard and mouse operations

- When Ctrl + Space is pressed on the keyboard, all candidate names are displayed at the location of the text cursor. If there is only one candidate, the relevant character string is pasted without displaying the candidate.
- When Ctrl + Shift + Space is pressed on the keyboard while the text cursor is at the location of a parameter for a method (function), a list of candidate parameter names will be shown.
- Information to supplement the name of a method (function) and variable is displayed in a tooltip when a mouse cursor is placed over a partial method (function) or variable name (only when a debugging tool is not connected).

Any of the actions listed below stops the display of candidate names by the smart edit function.

- Pressing the Esc key

  Pressing the Esc key while the smart edit function is displaying candidate names for functions and variables halts the display of candidate names.
- Input of a character other than an alphanumeric character while nothing is selected from the candidate list

  When no candidate for a function or variable is selected, the input of a character other than an alphanumeric character halts the display of candidate names.
- Input of a character other than an alphanumeric character while an item in the list of candidate names is selected

  When no candidate for a function or variable is selected, the input of a character other than an alphanumeric character leads to pasting of the selected candidate and thus stops the display of candidates.

Follow the procedure below to enable the smart edit function.

- In the [Text editor] page on the [Option] dialog box, select [Smart edit].
- Set the build tool property so that cross-reference information is output.
- Candidates are displayed by using cross-reference information. Execute and complete a build.

  If an error in building occurs, cross-reference information before the error occurred is used if any exists (when the output of cross-reference information by the build tool is disabled, the smart edit function cannot be used since the output will not include cross-reference information).

Note the following items regarding the smart edit function.

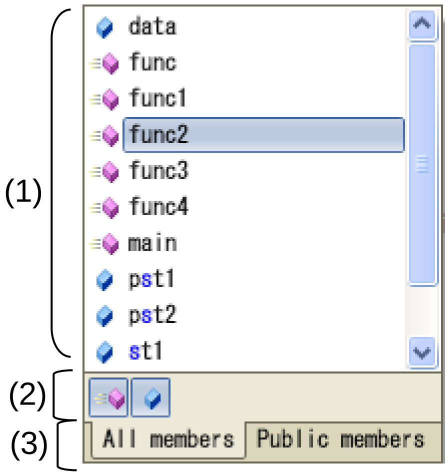
- Cross-reference information is necessary.
- The smart edit function cannot be used while the mixed display mode is in use and a debugging tool is connected.
- The function does not support macro definitions.
- The function does not support local variables.
- The function does not support the display of candidate members of types defined by using typedef. [V850E2]
- The function does not support namespaces and subclasses (classes declared within classes) of C++.
- When a structure, union or (in C++) class is declared within a function, candidates are not displayed within the function after its own declaration.
- The const and mutable attributes are not displayed.
- Candidates for the names of functions and variables are not displayed in C++ in response to the input of “(*class)” or “(*this)”.
- Candidates for the names of functions and variables are not displayed for an array declared within a class in C++.
- In C++, when candidates for the names of functions and variables are displayed by using the Ctrl + Space key while a class name is specified to the left and input of a method name is incomplete, the candidates displayed are the names of global functions and variables instead of candidates for the names of functions and variables in the class.
- In some cases when a compiler option which affects the size of variables is set, the type of variables for which names are displayed differs from the type which was actually declared.
- Tooltip information is not displayed while the debugging tool is connected.
- The tooltip display does not support the friend attribute of C++.
- The tooltip display does not indicate the const, static, volatile and virtual attributes.
- The tooltip display does not show the names of member functions of structures, unions, and (in C++) classes that were declared header files.
- When the build tool is CX and a union is declared, a tooltip is not displayed even if the mouse cursor is placed over the tag name of the union.

7.1.7 Added description of the smart editing display

[Addition]

Display of candidates for functions and variables

The smart edit function is used to display candidate names for functions and variables.



The following items are described.

- [How to open the list of candidate names]
- [Descriptions of the individual areas]

[How to open the list]

(1) When the automatic display of candidate names is selected

- In the C language and C++ language, if there are relevant members for the name on the left side when '.' is input, candidate names are automatically shown at the location of the text cursor.
- In the C language and C++ language, if there are relevant members for the name on the left side when '->' is input, candidate names are automatically shown at the location of the text cursor.
- In the C++ language, if there are relevant members for the name on the left side when '::' is input, candidate names are automatically shown at the location of the text cursor.

(2) When candidate names are shown in response to keyboard and mouse operations

- When Ctrl + Space is pressed on the keyboard, all candidate names are displayed at the location of the text cursor. If there is only one candidate name, the relevant character string is pasted without displaying the candidate name.

[Descriptions of the individual areas]

(1) Area for displaying candidate names

Candidate names for functions and variables are displayed in alphabetical order.

If a key is pressed while the smart editing display is active, the corresponding character strings are highlighted.

The following icons are displayed as labels for the list of candidate names.

Icon	Description
📄	Shows that the candidate name is for a typedef.
≡◆	Shows that the candidate name is for a function.
◆	Shows that the candidate name is for a variable.
◆	Shows that the candidate name is for a class.

◆	Shows that the candidate name is for a structure.
◇	Shows that the candidate name is for a union.
{}	Shows that the candidate name is for a namespace.
◆	Shows that the candidate name is for a protected member.
◆	Shows that the candidate name is for a private member.

A candidate for a string is inserted at the location of the text cursor by selecting a candidate name from the list and pressing the Enter key or the Tab key.

(2) Toolbar area

This area is used to switch the display of candidate names for functions and variables on or off.

Icon	Description
◆	Selected: Candidate names for methods (functions) are displayed. Deselected: Candidate names for methods (functions) are not displayed.
◆	Selected: Candidate names for variables are displayed. Deselected: Candidate names for variables are not displayed.

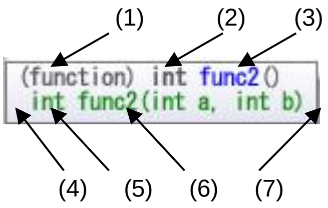
(3) Tab area

This area is used to switch the members to be displayed.

Tab name	Description
All members	All candidate names are displayed.
Public members	Only candidate names declared with the public attribute are displayed.

Detailed display of candidates

The smart edit function is used to display details of candidate names for functions and variables.



```
(function) int func2()
 int func2(int a, int b)
```

The following items are described.

- [How to open the list of candidate names]
- [Descriptions of the individual areas]

[How to open the list]

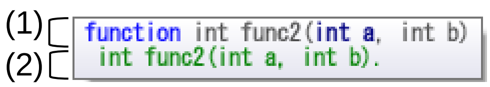
A candidate name for a method (function) or variable is selected from the display of candidate names for functions and variables.

[Descriptions of the individual areas]

No.	Description	
(1)	Kind	Shows whether the selected item is a method (function) or a variable. (function): method (function) (variable): variable
(2)	Type	Shows the type of the function or variable.
(3)	Name	Shows the name of the function or variable.
(4)	Attribute	Shows the attribute (public, protected, or private). If none of these attributes is defined, this item is not shown.
(5)	Type	Shows the type of the function or a variable.
(6)	Name and parameter	Shows the name of the function or a variable. When the name is that of a function, the parameters are also shown.
(7)	Overloading information	Shows the number of times the name has been overloaded. (Example): (+1 overloads)

Display of candidates for parameters

The smart edit function is used to display details of candidate names for parameters.



```
function int func2(int a, int b)
  int func2(int a, int b).
```

The following items are described.
- [How to open the list of candidate names]
- [Descriptions of the individual areas]

[How to open the list]

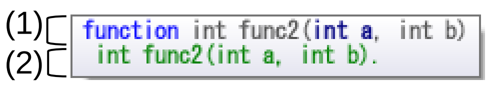
(1) When the automatic display of candidate names is selected

・ If there are relevant parameter names for the method (function) on the left side of '(' when '(' is input, the candidate parameter names are automatically shown.

(2) When candidate names are shown in response to keyboard and mouse operations

・ When Ctrl + Shift + Space is pressed on the keyboard while the text cursor is at the location of a parameter for a method (function), a list of candidate parameter names will be shown.

[Descriptions of the individual areas]



```
function int func2(int a, int b)
  int func2(int a, int b).
```

No.	Description	
(1)	Candidate	Shows the name of the candidate and its parameters.
(2)	Attribute of candidate	Shows the attribute of a candidate.

When a candidate name for a parameter is shown, the parameter at the location of the text cursor is highlighted (in the figure above, the first parameter is highlighted).

Display of tooltip information

The smart edit function is used to display information on functions and variables.

The following items are described.
- [How to open the list of candidate names]
- [Descriptions of the individual areas]

[How to open the list]

Place the mouse cursor on a function or variable.

[Descriptions of the individual areas]

The following icons are displayed according to the type of the target on which the mouse cursor is placed.

Icon	Description
📄	Indicates that the target is a typedef.
◆	Indicates that the target is a function.
◆	Indicates that the target is a variable.
◆	Indicates that the target is a class.
◆	Indicates that the target is a structure.
◇	Indicates that the target is a union.
{ }	Indicates that the target is a namespace.
▤	Indicates that the target is of enumeration type.

When the mouse cursor is placed on a variable of class, structure, union, or enumeration type, its members are displayed as tooltip information.

- If the target is a class-, structure-, or union-type variable, the types and names of its members are displayed.
- If the target is a class-type variable that includes methods (functions) among its members, the types of the return values and names of the methods (functions) are displayed. Also, ‘(‘ ‘)’ is appended to the end of each method name.
- If the target is an enumeration-type variable, only the names of the members are displayed.
- Up to 20 members can be displayed, and each is placed on a single line. If the number of members exceeds 20, the 21st and subsequent members are indicated by “…”.
- Members are displayed in the same order as they are defined in the source file.

7.1.8 Added description of mixed display

[Before addition]
None
[After addition]
The mixed display feature displays the source program and assembly code together; in normal display mode, only the source program is displayed.
You can set mixed display from the toolbar.

When the same file is opened again after having been open in the mixed display mode in the [Editor] panel and then closed, it will be displayed in the normal display mode rather than the mixed display mode. If the debugging tool is disconnected and reconnected, the file will also be displayed in the normal display mode.

Note the following when using mixed display mode.
- Editing is not available.
- Features that change file content, such as cutting and pasting, deletion, redo, replacement, outlining, and indenting, are not available.
- The select all feature is not available.
- The display on the panel cannot be split.

You can save mixed code into a text file or CSV file in either of the following ways.
- Activate an [Editor] panel in mixed display mode and select [File] – [Save Source Mixed Data As...] from the CS+ menu.
- Select [Save Source Mixed Data As...] from the context menu on an [Editor] panel in mixed display mode

7.1.9 Added description of recycle mode

[Addition]

The recycle mode is used to display the sequence of multiple source files in the [Editor] panel during debugging while the program counter (PC) enters code generated by multiple source files during stepped execution, with the multiple files of source code successively displayed in the same [Editor] panel.

The recycle mode is set from [Tools] – [Options] – [Text Editor].

If an [Editor] panel for the corresponding source file has been already opened, the file will not be displayed in the panel running in recycle mode; instead, it is only displayed in the [Editor] panel where it was already open.

Editing in the [Editor] panel is still possible in recycle mode. When the program counter (PC) is in a given file so that file is open in the [Editor] panel in recycle mode while a program is being executed and the contents of the panel are edited, that panel leaves recycle mode and a new [Editor] panel is opened in recycle mode.

When an [Editor] panel operating in the recycle mode is closed, the next [Editor] panel to be opened will initially operate in recycle mode.

7.1.10 Change to description of split bar

[Before change]

You can split the [Editor] panel display by using horizontal and vertical split bars within the view. This panel can be split up to **four times**.

[After change]

You can split the [Editor] panel display by using horizontal and vertical split bars within the view. This panel can be split into **two sections vertically, and two sections horizontally**.

7.1.11 Additional description of jump to functions

[Addition]

The Jump to Function feature depends on the build tool in use and is only available when the following conditions are satisfied.

(a) When the build tool is CA78K0R, CA850, or CA78K0

- The target is a function in the active project.
- The type of project specified as the active project is “application”.
- The file specified in [Download files] includes symbol information.

However, the file is specified as the first file in the [Download files] property if the debugging tool is disconnected. (Downloading of symbol information must also be enabled if the file is in the hexadecimal format.)

Note: When the debugging tool is disconnected, jumping to static functions is not possible in a file that includes symbol information.

(b) When the build tool is CX

- When the debugging tool is disconnected
  - The type of project specified as the active project is "application".
  - The file specified as the first file in the [Download files] property includes symbol information.
  - The above file includes information on the target function.
  - The target function is a global function.
- When the debugging tool is connected to download the load module
  - The downloaded load module file includes the symbol information for the function.
  - Calling the target function from the file corresponding to the address of the program counter (PC).

    * For example, jumping to a static function defined other than in the file corresponding to the address of the program counter (PC) is not possible.
- Furthermore, in the case of jumping to functions in C++ programs, the following point for caution also applies to specifying the functions.
  - When a given character string is specified as the chosen function name, there is a possibility that the jump will actually be to a different function with the same name.

(1) Member functions of classes

It is necessary to include the name of the class of which the target function is a member.

When other functions have the same name as the target function but the arguments are different, please also include the argument type.

Example: "memfunc": Jumping is not possible.
"Class::memfunc(short)": Jumping is possible.

(2) Functions defined in namespaces

It is necessary to include the full name of the namespace to which the target function target belongs.

When other functions have the same name as the target function but the arguments are different, please also include the argument type.

Example: "func": Jumping is not possible.
"Namespace1::Namespace2::func(int)": Jumping is possible.

(3) Template functions

It is necessary to include the type of the arguments in the function generated by the compiler.

Example: "template": Jumping is not possible.
"template(int, short)": Jumping is possible.

(c) External build tools

- The target is a function in the active project.
- The file specified in [Download files] includes symbol information.

  However, if the debugging tool is disconnected, the file is specified as the first file in the [Download files] property. (Downloading of symbol information must also be enabled if the file is in the hexadecimal format.)

  Note: When the debugging tool is disconnected, jumping to static functions is not possible in a file that includes symbol information.

7.1.12 Added description of the context menu

[Addition]

ProgramAnalyzerAddVariableToChart	Registers a variable in the [Variable Value Changing Chart] tab of the [Analysis Chart] panel.

7.1.13 Added description of the [Print Preview] dialog box

[Before addition]

(1) Preview area

The image to be printed is displayed as a preview.

[After addition]

(1) Preview area

The image to be printed is displayed as a preview. The display differs according to whether the debugging tool is or is not connected, and when it is connected, to whether the display is in normal mode or mixed mode.

Even collapsed lines are displayed in their expanded state in a print preview.

(a) When the debugging tool is not connected

The display is as follows.

Left of page: line number

However, this is not displayed when line numbers are hidden on the [Editor] panel.

Page header: file name (fully qualified path)

Page footer: page number

(b) When the debugging tool is connected (in normal display mode)

The display is as follows.

Left of page: line number and address

However, these are not displayed if line numbers and addresses are hidden on the [Editor] panel.

Page header: file name (fully qualified path)

Page footer: page number

(c) When the debugging tool is connected (in mixed display mode)

The display is as follows.

Left of page: line number, address, instruction code

However, any of these items that are hidden on the [Editor] panel are not displayed.

Page header: file name (fully qualified path)

Page footer: page number

7.1.14 Added description on the display of large files

[Added description]

When a 24-Mbyte or larger file (referred to as a huge file below) is opened, the message Q2000007 is displayed. After seeing the message, you can select whether the functions in the [Editor] panel listed below should be disabled.

- Syntax coloring
- Smart edit function
- Code outlining

The selected state is only effective while the project is open. This confirmation message is only opened for the first huge file of a project to be opened. If another huge file is subsequently opened, it is opened in the [Editor] panel with the settings selected for the first huge file to have been opened.

7.2 Changes to the description of hook processes

This section describes changes to the description of hook processes.
Hook processes are described in the Debugging editions.

You can configure a hook transaction in the [Hook Transaction Settings] tab on the [Property] panel. Setting up a hook transaction provides a way of changing the values of I/O registers and CPU registers by executing a Python script before downloading a load module, or before running or after resetting the CPU. This makes the following points realizable.

(1) In program development, even if the setting program for the I/O registers is incomplete, debugging can proceed by setting the I/O registers before running the program.
(2) You can increase the download speed by configuring the I/O registers before downloading a program.
(3) Downloading to external RAM can be easily handled by setting up I/O registers before downloading.

You can use the following commands in a Python script to be executed as a hook process of the debugger.

```
debugger.Register.GetValue
debugger.Register.SetValue
debugger.Memory.GetValue
debugger.Memory.SetValue
```

If you want to use other Python commands, please use the Hook command in the Python console.

7.3 Changes to descriptions of building

This section describes changes to the CS+ V1.01.00 User's Manual: Build.
(Document number R20UT0783EJ0100, R20UT2143EJ0100)

7.3.1 Change to description of the [Link Options] tab (only in R20UT2143EJ0100)

[Location] Page 210

[Before change] (5) [Device]
Use on-chip debug

[After change] (5) [Device]
Use a linkage option to enable or disable on-chip debugging

7.3.2 Added description of the stack usage tracer

[Location] Page 315 → Functions analyzed

[After addition] Consequently, assembly files written by the user and functions in library files created by the user are not analyzed. For this reason, the information for these files must be set by using the [Adjust Stack Size] dialog box.

Since interrupt handlers are also outside the scope of analysis management, the applicable information for these files must also be set up by using the [Adjust Stack Size] dialog box.

[Location] Page 324 → Functions analyzed

[After addition] Consequently, assembly files written by the user and functions in library files created by the user are not analyzed. For this reason, the information for these files must be set by using the [Adjust Stack Size] dialog box.

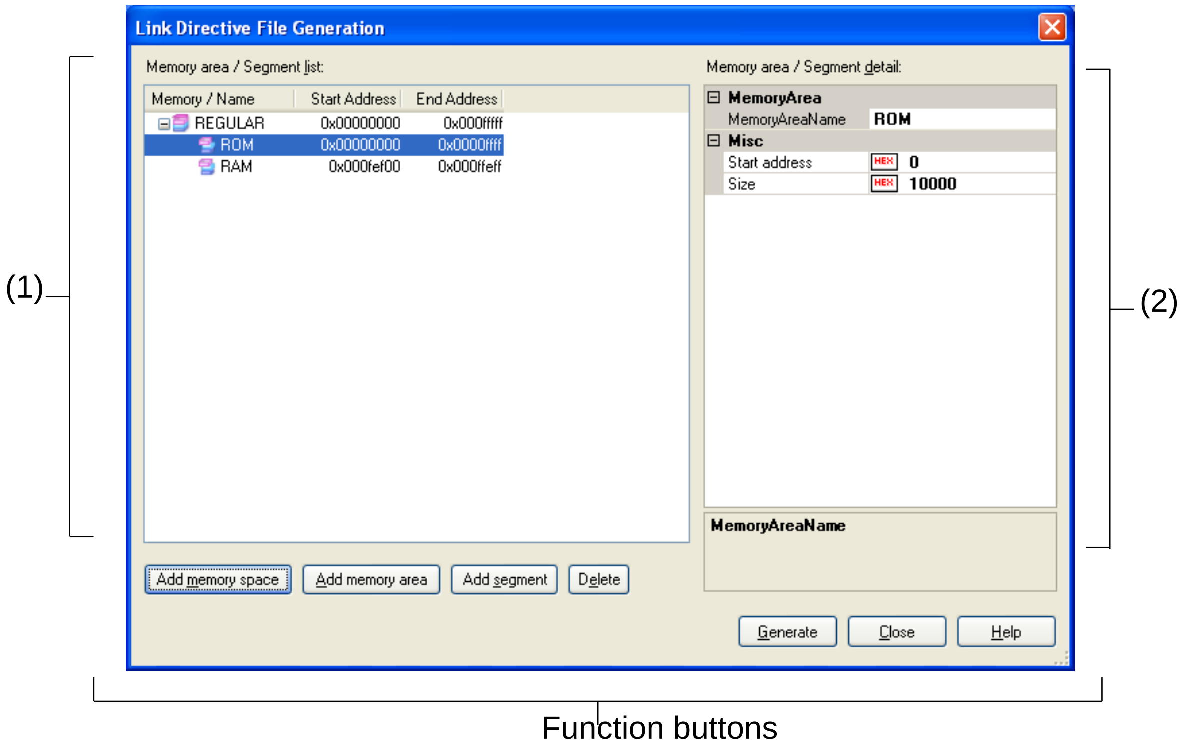
Since interrupt handlers are also outside the scope of analysis management, the applicable information for these files must also be set up by using the [Adjust Stack Size] dialog box.

7.3.3 Added description of the [Link Directive File Generation] dialog box

[Addition]

Link Directive File Generation dialog box

This dialog box is used to generate a link directive file based on the specified memory and segment allocation information.



Memory / Name	Start Address	End Address
REGULAR	0x00000000	0x000fffff
ROM	0x00000000	0x0000ffff
RAM	0x000fef00	0x000ffeff

MemoryArea	
MemoryAreaName	ROM
Misc	
Start address	HEX 0
Size	HEX 10000

The following items are explained here.

- [How to open the dialog box]
- [Descriptions of the individual areas]
- [Function buttons]

[How to open the dialog box]

- On the [Project Tree] panel, select the Build tool node, and then select [Create Link Directive File...] from the context menu.
- On the [Property] panel of the build tool, select the [Link Options] tabbed page, and then click on the [...] button for the [Input File] — [Generate link directive file] property.

[Descriptions of the individual areas]

(1) [Memory area/Segment list:] area

This area displays a list of the memory spaces, memory areas, and segments to be generated in the link directive file.

(a) [Memory/Name]

This column displays the names of memory spaces, memory areas, and segments.

For memory spaces, the relevant names from among the following memory space names are displayed.

- REGULAR
- EX1, EX2, EX3, EX4, EX5, EX6, EX7, EX8, EX9, EX10
- EX11, EX12, EX13, EX14, EX15

For memory areas and segments, the contents of this column can be directly edited. If a memory area name or a segment name is changed, the [Name] in the [Memory area/Segment detail:] area is also changed.

(b) [Start Address]

This column displays the addresses where memory spaces, memory areas, and segments start.

For memory areas and segments, the contents of this column can be directly edited. If a start address is changed, the [Start Address] in the [Memory area/Segment detail:] area is also changed.

(c) [End Address]

This column displays the addresses where memory spaces and memory areas end.

In rows for segments, "—" is shown in this column.

(d) Buttons

Add memory space	Adds a new memory space below the last row of the list. The new memory space is named EX*n* with the smallest value that can be registered when the space is created (n: a decimal value from 0 to 15). Note that this button is deactivated after memory spaces EX1 to EX15 (15 spaces in total) have been registered.
Add memory area	Adds a new memory area below the row selected in the list. The default name for the new memory area is "NewMemoryArea_XXX" (XXX: a decimal value from 0 to 255). Make detailed settings for the memory area in the [Memory area/Segment detail:] area. Note that this button is deactivated when a row for a segment is selected or when 256 memory areas have been registered in the list.
Add segment	Adds a new segment below the row selected in the list. The default name for the new segment is "Seg_XXX" (XXX: a decimal value from 0 to 255). Make detailed settings for the segment in the [Memory area/Segment detail:] area. Note that this button is deactivated when a row for a memory space is selected or when 256 segments have been registered in the list.
Delete	Deletes the memory space (except for REGULAR), memory area, or segment selected in the list. When a memory area is deleted, the segments included in the memory area are also deleted. Memory space REGULAR cannot be deleted.

This area has the following functions.

- Expanding or collapsing the display of a row

  Double-click on a row or click the + or – mark at the beginning of a row to expand or collapse the display of the row.

- Moving a memory area or segment row

  You can move a memory area row or a segment row by dragging and dropping it.

  Remark If a memory area row is moved, the segments it includes are also moved.

- Copying a memory area or a segment

  Select a memory area or a segment and press the [Ctrl] + [C] keys to copy it. To paste it, press the [Ctrl] + [V] keys.

  The memory area or segment is pasted under the selected row when the [Ctrl] + [V] keys are pressed.

  "C_" is added at the beginning of the name of the pasted memory area or segment, and if the number of characters in the name exceeds the maximum allowed number, characters at the end of the name are truncated as required.

  Remarks 1. When a memory area is copied, the segments in the memory area are not copied.

2. The start address column for the pasted memory area or segment is blank.
3. If a memory area or a segment cannot be pasted due to an attribute of the destination memory area, an error will occur.

(2) [Memory area/Segment detail:] area

This area displays detailed information on the memory area or segment selected in the [Memory area/Segment list:] area. You can also edit the information in this area.

(a) Detailed information on memory areas

MemoryArea Name	Specify the memory area name. Specifiable characters are numeric characters (0 to 9), uppercase and lowercase letters of the alphabet (A to Z and a to z), the underscore (_), the question mark (?), and the "at" sign (@).	
	Default	NewMemoryArea_*XXX* (*XXX*: a decimal value from 0 to 255)
	Changing a name	Directly enter a value in the text box, or click on the [...] button and edit the contents in the opened [Character String Input] dialog box.
	Specifiable values	Strings of up to 31 characters
Start address	Specify the address where the memory area starts. This field can be left blank or a hexadecimal value (only) can be specified.	
	Default	Blank
	Changing the address	Directly enter a value in the text box.
	Specifiable values	0x0 to 0xFFFF (hexadecimal): 78K0, except products with memory banks 0x0 to 0xFFFFF (hexadecimal): RL78, 78K0R, and 78K0 products with memory banks The specifiable values also depend on the device.
Size	Specify the size of the memory area. This field can be left blank or a hexadecimal value (only) can be specified. Note that an error will occur in linkage if this field is blank.	
	Default	Blank
	Changing the size	Directly enter a value in the text box.
	Specifiable values	0x1 to 0xFFFF (hexadecimal): 78K0, except products with memory banks 0x1 to 0xFFFFF (hexadecimal): RL78, 78K0R, and 78K0 products with memory banks The specifiable values also depend on the device.

(b) Detailed information on segments

Name	Specify the segment name. Specifiable characters are numeric characters (0 to 9), uppercase and lowercase letters of the alphabet (A to Z and a to z), the underscore (_), the question mark (?), and the "at" sign (@).	
	Default	NewMemoryArea_*XXX* (*XXX*: a decimal value from 0 to 255)
	Changing a name	Directly enter a value in the text box, or click on the [...] button and edit the contents in the opened [Character String Input] dialog box.
	Specifiable values	Strings of up to eight characters

Specify the start address	Select whether to specify the address where the segment starts.		
	Default	No	
	Changing a name	Select the name from the drop-down list.	
	Specifiable values	Yes	Specifies the segment start address.
		No	Does not specify the segment start address.
Start address	Specify the start address of the segment. This field can be left blank or only a hexadecimal value can be specified. This property is displayed only when [Yes] is selected in the [Specify the start address] property.		
	Default	Blank	
	Changing the address	Directly enter a value in the text box.	
	Specifiable values	0x0 to 0xFFFF (hexadecimal): 78K0, except products with memory banks 0x0 to 0xFFFFF (hexadecimal): RL78, 78K0R, and 78K0 products with memory banks The specifiable values also depend on the device.	
Merge attribute	Specify the merge attribute of the segment. This property specifies how to merge multiple segments with the same name.		
	Default	Auto (None)	
	Changing the attribute	Select it from the drop-down list.	
	Restriction	Auto(None)	Executes the same processing as sequential merge (SEQUENT). However, if relocation attribute AT is specified at assembly for any of the input segments having the same name, error (COMPLETE) processing will proceed.
		Merge sequentially (SEQUENT)	Multiple segments having the same name are merged sequentially in order of their appearance so that there is no vacant space between them. BSEG-attribute segments are merged in bit units in the order of appearance.
		Error (COMPLETE)	Multiple segments having the same name will lead to an error.

[Function buttons]

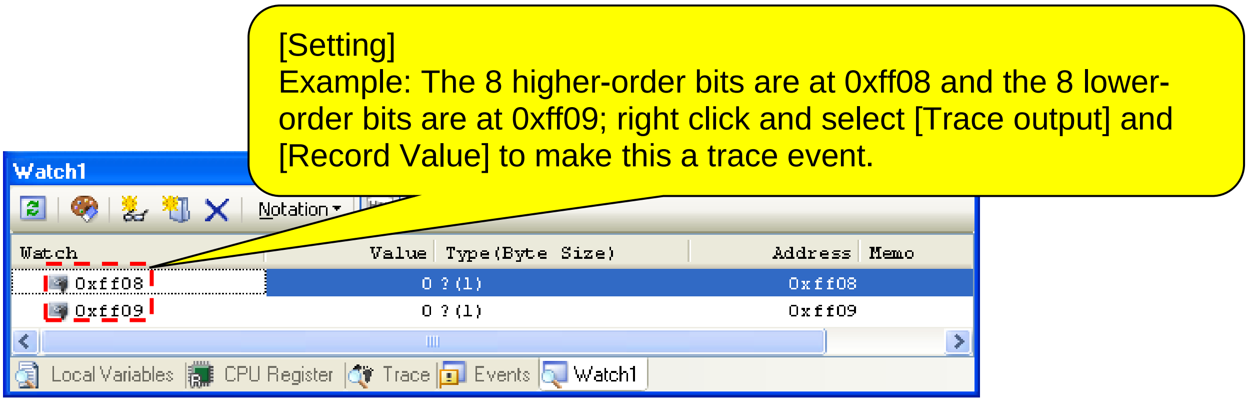
Button	Function
Generate	Generates a link directive file (named project-name.dir) based on the specified memory and segments allocation information, and then adds the file to the project. The link directive file is generated in the project folder. The link directive file that has been generated is also shown on the project tree, under the File node. The generated link directive file will be a build target. If a link directive file has already been registered with the project, then that file will be removed from set of build targets.
Close	Closes the dialog box.
Help	Displays the help for this dialog box. This is not supported in the current version.

7.4 Change to the 78K0 debug user's manual

This section describes a change to the CS+ V1.01.00 User's Manual: 78K0 Debug (document number R20UT0731EJ0100).

7.4.1 Addition of point trace description for special function registers (SFRs) and two-byte variables

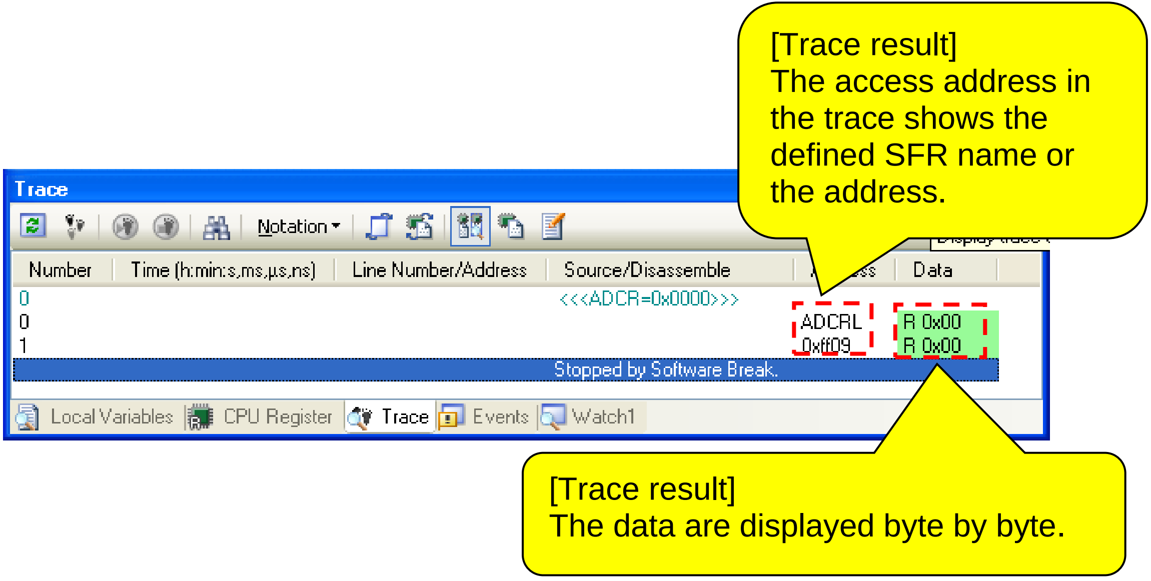
[Location] page 118, before "2.11.5 Display the collected execution history"
[Addition]
2.11.4 (2) Access to SFRs and two-byte variables [IECUBE]
For point tracing of SFRs and two-byte variables, separately register the addresses of the 8 higher- and 8 lower-order bits in the [Watch] panel and make a trace event (see the figure below).



[Setting]
Example: The 8 higher-order bits are at 0xff08 and the 8 lower-order bits are at 0xff09; right click and select [Trace output] and [Record Value] to make this a trace event.

Watch	Value	Type (Byte Size)	Address	Memo
0xff08	0 ? (1)		0xff08	
0xff09	0 ? (1)		0xff09	

A trace result of this setting will be as follows.



[Trace result]
The access address in the trace shows the defined SFR name or the address.

Number	Time (h:min:s,ms,μs,ns)	Line Number/Address	Source/Disassemble	Address	Data
0			<<<ADCR=0x0000>>>		
0				ADCRL	R 0x00
1				0xff09	R 0x00

Stopped by Software Break.

[Trace result]
The data are displayed byte by byte.

7.5 Change to the V850 debug user's manual

This section describes a change to the CS+ V2.00.00 User's Manual: V850 Debug (document number R20UT2446EJ0100).

7.5.1 Change to a description in "Notes on using the GHS compiler"

[Location] page 95 **Notes 1.** Notes on using GHS compiler (Green Hills Software, LLC, USA)
[Before change]
Supported options
- Other than the above: -prepare_dispose, -call

[After change]
Supported options

- Other than the above: -prepare_dispose, -callt

7.6 Change to CubeSuite+ V2.02.00 Analysis

This section describes a change to the CubeSuite+ V2.02.00 Analysis:(document number R20UT2868EJ0100).

7.6.1 The setting of Dynamic analysis

(1) Change part 1

[Location] page 10 1.1.2 Types of analysis information (2) Dynamic analysis information (a) Trace function after Remarks2.

[Before change]

non

[After change]

In order to use dynamic analysis using the trace function, it is necessary to set up the property of the debug tool.

Below, the list of each debug tool and properties to be set up is shown.

Debug tool	**Properties**	**Description**
[RL78] [Simulator] [V850E2] [Simulator] [78K0] [Simulator] [78K0R] [Simulator]	[Debug Tool Settings]tab -> [Trace]category -> [Use trace function] property(*)	Select [Yes]
	[Debug Tool Settings]tab -> [Trace]category -> [Trace memory size[frames]] property	Select trace size from the drop-down list
[RL78] [IECUBE] [78K0] [IECUBE] [78K0R] [IECUBE]	-	It is not necessary to set up.
[V850E2] [MINICUBE]/ [MINICUBE2]/[E1]/[E20] [78K0R] [MINICUBE2]/ [E1]/[E20]/[EZ Emulator] [78K0] [MINICUBE2]/ [E1]/[E20]/[EZ Emulator]	-	The trace function is not supported.
[RL78] [E1]/[E20]/[EZ Emulator]	[Debug Tool Settings]tab -> [Trace]category -> [Use trace function] property(*)	Select [Yes] The device which is not supporting the trace function does not display this property.
	[Debug Tool Settings]tab ->	It is not necessary to set up.

	[Trace]category -> [Trace memory size[frames]] property	
[V850E2] [IECUBE2]	[Debug Tool Settings]tab -> [Trace]category -> [Use for trace data] property (*)	Select [Trace]
	[Debug Tool Settings]tab -> [Trace]category -> [Select trace data] property	Select [Branch PC] when acquiring only the function information. Select [Access PC + Access Data] when acquiring only the variable information. Select [Branch PC + Access PC + Access Data] when acquiring the function information and the variable information. The setup which can be selected is dependent on device.
	[Debug Tool Settings]tab -> [Trace]category -> [Trace memory size[frames]] property	Select trace size from the drop-down list.

(*)If [Settings] tab-> [General] category-> [Enable dynamic analysis information] property in Program Analyzer plugin is set as [Yes], the setting of the debug tool will be automatically changed at the time of connection

(2) Change part 2
[Location] page 10 1.1.2 Types of analysis information (2) Dynamic analysis information (b) RRM function/RAM monitor (pseudo RRM) function after caution.
[Before change]
non
[After change]
In order to use dynamic analysis using the RRM function/RAM monitor (pseudo RRM) function, it is necessary to set up the property of the debug tool.
Below, the list of each debug tool and properties to be set up is shown.

Debug tool	Properties	Description
[RL78] [Simulator] [V850E2] [Simulator] [78K0] [Simulator] [78K0R] [Simulator]	[Debug Tool Settings]tab -> [Access Memory While Running] category-> [Access during the execution] property(*)	Select [Yes]
	[Debug Tool Settings]tab -> [Access Memory While Running] category-> [Display update interval[ms]] property	Specify the interval between 100 and 65500
[RL78] [IECUBE]/ [E1]/ [E20]/[EZ Emulator] [V850E2] [IECUBE2]/[MINICUBE]/ [MINICUBE2]/[E1]/[E20] [78K0R] [IECUBE]/[MINICUBE2]/ [E1]/[E20]/[EZ Emulator] [78K0] [IECUBE]/ [MINICUBE2]/[E1]/[E20]/ [EZ Emulator]	[Debug Tool Settings]tab -> [Access Memory While Running] category-> [Access during the execution] property(*)	Select [Yes]
	[Debug Tool Settings]tab -> [Access Memory While Running] category-> [Update display during the execution] property(*)	Select [Yes]
	[Debug Tool Settings]tab -> [Access Memory While Running] category-> [Display update interval[ms]] property	Specify the interval between 100 and 65500

(*)If [Settings] tab-> [General] category-> [Enable dynamic analysis information] property in Program Analyzer plugin is set as [Yes], the setting of the debug tool will be automatically changed at the time of connection

(3) Change part 3
[Location] page 11 1.1.2 Types of analysis information (2) Dynamic analysis information (c) Coverage function after caution 3.
[Before change]
non
[After change]
In order to use dynamic analysis using the Coverage function, it is necessary to set up the property of the debug tool.
Below, the list of each debug tool and properties to be set up is shown.

Debug tool	Properties	Description
[RL78] [Simulator] [V850E2] [Simulator] [78K0] [Simulator] [78K0R] [Simulator]	[Debug Tool Settings]tab -> [Coverage] category-> [Use coverage function] property(*)	Select [Yes]
[V850E2] [Simulator]	[Debug Tool Settings]tab -> [Coverage] category-> [Use coverage function] property(*)	Select [Yes]
	[Debug Tool Settings]tab -> [Coverage] category-> [Coverage area of measurement(1MBytes)] property	Specify the start address that performs coverage measurement
[V850E2] [IECUBE2]	[Debug Tool Settings]tab -> [Trace]category -> [Use for trace data] property	Select [Coverage]
[RL78] [IECUBE] [78K0] [IECUBE] [78K0R] [IECUBE]	-	It is not necessary to set up.
[RL78] [E1]/[E20]/[EZ Emulator] [V850E2] [MINICUBE]/ [MINICUBE2]/ [E1]/[E20] [78K0R] [MINICUBE2]/ [E1]/[E20]/[EZ Emulator] [78K0] [MINICUBE2]/[E1]/[E20]/ [EZ Emulator]	-	The coverage function is not supported.

(*)If [Settings] tab-> [General] category-> [Enable dynamic analysis information] property in Program Analyzer plugin is set as [Yes], the setting of the debug tool will be automatically changed at the time of connection

7.7 Change to the Start user’s manual

This section describes a change to the CS+ V2.02.00 User’s Manual: Start (document number R20UT2865EJ0100).

7.7.1 Deletion a description of “acceleration utility”

[Location] page 96 2.10.2 Use acceleration utility

7.7.2 Operating Environment

[Location] page 11 1.4 Operating Environment (2) Software environment

[Before change]
- Microsoft Visual C++ 2010 SP1 runtime libraries

[After change]
- Microsoft Visual C++ 2015 Update 3 runtime libraries (x86)

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 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. This document shall not be reprinted, 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.

(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.4.0-1 November 2017)

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/.

© 2019 Renesas Electronics Corporation. All rights reserved.