

# CubeSuite+ V2.02.00

Integrated Development Environment

User's Manual: Analysis

Target Device

78K0 Microcontroller

RL78 Family

78K0R Microcontroller

V850 Family

RX Family

RH850 Family

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (<http://www.renesas.com>).

## 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.

# How to Use This Manual

This manual describes the role of the CubeSuite+ integrated development environment for developing applications and systems for RH850 family, RX family, V850 family, RL78 family, 78K0R microcontrollers, 78K0 microcontrollers, and provides an outline of its features.

CubeSuite+ is an integrated development environment (IDE) for RH850 family, RX family, V850 family, RL78 family, 78K0R microcontrollers, 78K0 microcontrollers, integrating the necessary tools for the development phase of software (e.g. design, implementation, and debugging) into a single platform.

By providing an integrated environment, it is possible to perform all development using just this product, without the need to use many different tools separately.

**Readers** This manual is intended for users who wish to understand the functions of the CubeSuite+ and design software and hardware application systems.

**Purpose** This manual is intended to give users an understanding of the functions of the CubeSuite+ to use for reference in developing the hardware or software of systems using these devices.

**Organization** This manual can be broadly divided into the following units.

**CHAPTER 1 GENERAL**  
**CHAPTER 2 FUNCTIONS**  
**CHAPTER 3 CAUTIONS**  
**APPENDIX A WINDOW REFERENCE**  
**APPENDIX B INDEX**

**How to Read This Manual** It is assumed that the readers of this manual have general knowledge of electricity, logic circuits, and microcontrollers.

**Conventions**

Data significance:	Higher digits on the left and lower digits on the right
Active low representation:	XXX (overscore over pin or signal name)
Note:	Footnote for item marked with Note in the text
Caution:	Information requiring particular attention
Remark:	Supplementary information
Numeric representation:	Decimal ... XXXX
	Hexadecimal ... 0xXXXX

**Related Documents**

The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such.

Document Name		Document No.
CubeSuite+ Integrated Development Environment User's Manual	Start	R20UT2865E
	RX Design	R20UT2862E
	V850 Design	R20UT2134E
	RL78 Design	R20UT2684E
	78K0R Design	R20UT2137E
	78K0 Design	R20UT2138E
	RH850 Coding	R20UT2584E
	RX Coding	R20UT2470E
	V850 Coding	R20UT0553E
	Coding for CX Compiler	R20UT2659E
	RL78,78K0R Coding	R20UT2774E
	78K0 Coding	R20UT2141E
	RH850 Build	R20UT2585E
	RX Build	R20UT2472E
	V850 Build	R20UT0557E
	Build for CX Compiler	R20UT2142E
	RL78,78K0R Build	R20UT2623E
	78K0 Build	R20UT0783E
	RH850 Debug	R20UT2866E
	RX Debug	R20UT2875E
	V850 Debug	R20UT2446E
RL78 Debug	R20UT2867E	
78K0R Debug	R20UT0732E	
78K0 Debug	R20UT0731E	
Analysis	This manual	
Message	R20UT2871E	

**Caution** The related documents listed above are subject to change without notice. Be sure to use the latest edition of each document when designing.

All trademarks or registered trademarks in this document are the property of their respective owners.

# TABLE OF CONTENTS

## CHAPTER 1 GENERAL ... 7

- 1.1 Summary ... 7
  - 1.1.1 What is analyzed ... 7
  - 1.1.2 Types of analysis information ... 8
- 1.2 Features ... 12

## CHAPTER 2 FUNCTIONS ... 16

- 2.1 Overview ... 16
- 2.2 Display Function Information ... 21
- 2.3 Display Variable Information ... 23
- 2.4 Display Calling Relationships between Functions (Call Graph) ... 25
- 2.5 Display Class/Function/Variable Information ... 29
- 2.6 Customize Display Method ... 31
  - 2.6.1 Set the columns to display ... 31
  - 2.6.2 Change the order of the columns to display ... 33
  - 2.6.3 Lock the specific column to display ... 34
  - 2.6.4 Sort the analysis information ... 34
  - 2.6.5 Filter the analysis information ... 34
- 2.7 Jump to Defined Location ... 37
- 2.8 Jump to Declared Location ... 39
- 2.9 Set Break Events ... 40
  - 2.9.1 Set a breakpoint to a function ... 40
  - 2.9.2 Set a break event to a variable ... 40
- 2.10 Register Watch-Expressions ... 42
- 2.11 Display List of Referencing Location ... 43
- 2.12 Import or Export Information File ... 44
- 2.13 Display Analysis Information in Chart ... 48
  - 2.13.1 Graph transitions in values ... 48
  - 2.13.2 Chart the function execution time ratios ... 62
- 2.14 Save Analysis Information ... 64

## CHAPTER 3 CAUTIONS ... 66

- 3.1 Changing Active Project ... 66
- 3.2 Coverage Results ... 66
- 3.3 Real-time Sampling Analysis ... 66
- 3.4 Trace Data Analysis ... 67
- 3.5 When the Panel Is Opened during Execution of Program ... 67
- 3.6 When CC-RX (C++ Source Files) Is Used ... 68

**APPENDIX A WINDOW REFERENCE ... 70**

**A.1 Description ... 70**

**APPENDIX B INDEX ... 183**

## CHAPTER 1 GENERAL

This chapter describes an overview of the analyze tool (Program Analyzer).

## 1.1 Summary

The analyze tool is a component provided by CubeSuite+. It supports the analysis of the behavior of programs developed for RH850 family, RX family, V850 family, R8C family ([Localised support](#)), RL78 family, 78K0R microcontrollers, 78K0 microcontrollers.

**Remark** Localised support

"Localised support " refers to specific regions support only.

CubeSuite+ for R8C (including NC30) is shipped and supported to the following regions only.

- Renesas Electronics Hong Kong Limited
- Renesas Electronics (China) Co., Ltd.
- Renesas Electronics (Shanghai) Co., Ltd.

### 1.1.1 What is analyzed

The analyze tool only analyzes C/C++ source files (including header files) in a project being set to the active project in the [Project Tree panel](#).

Consequently, assembler source files are not analyzed. Note, however, that symbols defined in assembler source files that are referenced from C/C++ source files are analyzed.

**Cautions 1. C++ source files are only supported by CC-RX.**

**2. If the target project to be analyzed satisfies the conditions shown below, the analyze tool cannot be used.**

- A debug-dedicated project
- A boot loader for multi-core project [RH850]
- A project in which a non-supported build tool is registered
- A project in which no build tool is registered

**3. [CA850]**

**Files in K&R format are not analyzed.**

**Remark** It is possible to forcibly display the function/variable information for C/C++ source files and header files in a project other than the active project by importing information files (function list file (\*.mtfl)/variable list file (\*.mtvl)) (see "[2.12 Import or Export Information File](#)" for details).

Among the above files that can be analyzed, only desired files can be selected as the files to be analyzed or files not to be analyzed to improve the efficiency of analysis.

The function information and variable information in the files not to be analyzed is not displayed in the panels (except for the [Analysis Chart panel](#)) provided by the analyze tool.

Only the function information and variable information in the files to be analyzed is displayed in the panels (except for the [Analysis Chart panel](#)) provided by the analyze tool.

Use any of the following methods to specify files to be analyzed and files not to be analyzed (use a convenient method for the analysis purpose).

**(1) When the files not to be analyzed are specified (default)**

- On the [\[Settings\]](#) tab in the [Property panel](#), specify [\[Files not to analyze\]](#) with the [\[Select specification method of files to analyze\]](#) property in the [\[Analysis Object\]](#) category.

- In the same category above, click the [...] button that appears at right edge in the column when [Files not to analyze] property is selected.
- On the [Select Files not To Analyze dialog box](#) opened, select the files not to be analyzed.

## (2) When the files to be analyzed are specified

- On the [\[Settings\] tab](#) in the [Property panel](#), specify [Files to analyze] with the [Select specification method of files to analyze] property in the [Analysis Object] category.
- In the same category above, click the [...] button that appears at right edge in the column when [Files to analyze] property is selected.
- On the [Select Files To Analyze dialog box](#) opened, select the files to be analyzed.

### 1.1.2 Types of analysis information

The analyze tool acquires, analyzes, and displays the following two types of information.

#### (1) Static analysis information

This is the function information/variable information that can be displayed when a build has completed without any compiler or assembler errors. This is the result of analyzing the load module and cross reference information generated by the build tool.

For this reason, in order to acquire static analysis information, you should configure your build tool to generate cross reference information when performing the build. The analyze tool does, however, provide the following property that forces the build tool to generate cross reference information:

- [Property panel](#) >> [\[Settings\] tab](#) >> [General] category >> [Enable static analysis information] property

By setting this property to [Yes], it is possible to generate cross reference information regardless of the build tool's configuration. Note that if this property is set to [No] (default), the current setting in the build tool will take precedence.

#### (2) Dynamic analysis information

This is the function information or variable information provided as a result of analyzing trace data, real-time RAM monitoring, or coverage data acquired by the debug tool.

For this reason, you should enable [Trace function](#), [RRM function/RAM monitor \(pseudo RRM\) function](#), or [Coverage function \[IECUBE\]\[IECUBE2\]\[E20\[RX\]\]\[Simulator\]](#) to acquire dynamic analysis information. The analyze tool does, however, provide the following property that automatically enables all of the above functions of the debug tool:

- [Property panel](#) >> [\[Settings\] tab](#) >> [General] category >> [Enable dynamic analysis information] property

By setting this property to [Yes], it is possible to use the above debug tool's functions regardless of the debug tool's configuration. Note that if this property is set to [No] (default), the current setting in the debug tool will take precedence.

- Cautions 1.** You must connect to the debug tool after the completion of the setting above.  
It is necessary to re-connect to the debug tool if you change this setting during debugging.
- 2.** Functions that the debug tool does not support cannot be enabled.  
For some debug tools, the following functions are mutually exclusive. In that case, the priority is as follows:  
Trace function > RRM function/RAM monitor (pseudo RRM) function > Coverage function
- 3.** The dynamic analysis information cannot be acquired when the program analyzer tool or task analyzer tool provided by "RI series" is used.

The dynamic analysis information that each debug tool's function provides to the analyze tool is as follows:

**(a) Trace function**

This function is used to collect an execution history of the program as a trace data.

The trace function provides the following [Dynamic analysis information](#):

- Execution time (including a proportion of total execution time and average execution time)
- Execution count (including access type (read count, write count, and so on))
- Variable values (including maximum values and minimum values)
- Graph data (when the method for acquiring the graph data is [Trace data analysis](#))

**Cautions 1. [RH850]**

- [Full-spec emulator][E1/E20]

The contents of the dynamic analysis information differ depending on the [Trace target setting] property in the [Trace] category on the [Debug Tool Settings] tab of the debug tool's Property panel.

- When [Debug core only] is selected

The debug tool collects trace data regarding PE $n$  currently selected in the Debug Manager panel. Therefore, the analyze tool also displays the dynamic analysis information regarding PE $n$ .

- When [All core] is selected

The debug tool collects trace data in all PEs. Therefore, after collecting trace data, the analyze tool displays the corresponding dynamic analysis information by switching to another PE $n$  in the Debug Manager panel.

- [Simulator]

The debug tool collects trace data regarding PE $n$  currently selected in the Debug Manager panel. Therefore, the analyze tool also displays the dynamic analysis information regarding PE $n$ .

2. [Full-spec emulator][IECUBE][IECUBE2][Simulator]

Perform the following setting on the debug tool's Property panel to acquire the execution time correctly:

- [Debug Tool Settings] tab >> [Trace] category >> [Clear trace memory before running] property >> [Yes] (default)

3. [IECUBE[V850E1][V850ES]][E20[RX]]

If [Start Trace]/[Stop Trace] from the context menu on the Trace panel is selected during execution of a program, then an invalid value will be displayed as the execution time.

4. [IECUBE[78K0]][E1/E20[RH850]]

Since the trace time tag feature is not supported, execution time of functions and graphs through [Trace data analysis](#) cannot be displayed.

5. [E1/E20[RX]][EZ Emulator[RX]]

Since the trace time tag feature is not supported, execution time of functions cannot be displayed. The time information displayed in graphs through [Trace data analysis](#) will not be correct.

6. [E20[RX]]

In the debug tool's Property panel, when the [Trace data type] property in the [Trace] category on the [Debug Tool Settings] tab is set to [Branch], the acquisition time of the dynamic analysis information by trace data will increase because complement of instructions between branch instructions is performed. If you focus only on variables, we recommend selecting [Data access] in this property.

## 7. [E1/E20[RL78]][EZ Emulator[RL78]]

Since branch origin addresses can only be acquired from the trace data, only the [Execution Count] item in the [Function List panel/Call Graph panel](#) is supported.

## Remarks 1. [IECUBE[V850E1]][V850ES]]

Some of the trace functions, RRM functions and coverage functions are mutually exclusive. For this reason, it is necessary to specify the function to take precedence in the [Use for trace data] property of the [Trace] category. When doing so, if this property is set to other than [Trace] ([RRM] or [Coverage]), the trace functions necessary to acquire dynamic analysis information will still be enabled.

## 2. [IECUBE2]

Some of the trace functions, timer functions and coverage functions are mutually exclusive. For this reason, it is necessary to specify the function to take precedence in the [Use for trace data] property of the [Trace] category. When doing so, if this property is set to other than [Trace] ([Timer] or [Coverage]), the trace functions necessary to acquire dynamic analysis information will still be enabled.

## (b) RRM function/RAM monitor (pseudo RRM) function

This function permits you to read the contents of memory (variables/registers/addresses) in real-time while executing the program.

The RRM function/RAM monitor (pseudo RRM) function provides the following [Dynamic analysis information](#):

- Graph data (when the method for acquiring the graph data is [Real-time sampling analysis](#)<sup>Note</sup>)

## Note [E1/E20[RL78]]

If the selected microcontroller incorporates a Smart Analog IC and the debug tool is in the [Data collection mode](#), graph data is not acquired through real-time sampling analysis. Instead a sampling method specific to Smart Analog is used to acquire data and display graphs.

## Caution [RH850]

## - Variable values and addresses

The address and value are determined based on the PC value in PEn currently selected in the Debug Manager panel.

## - Register values

The value in PEn currently selected in the Debug Manager panel is acquired.

## - The area that can be read by this function

The contents of memory can be read for access in all PEs. In the Local RAM self area, note that they can be read only for the access in PEn currently selected in the Debug Manager panel.

**Remark** The target area from which you can read using the RRM function/RAM monitor (pseudo RRM) function varies depending on the microcontroller being selected in the project and the debug tool used.

For details on the relationship between the RRM function/RAM monitor (pseudo RRM) function and their target area, see "CubeSuite+ Integrated Development Environment User's Manual: Debug".

## (c) Coverage function [IECUBE][IECUBE2][E20[RX]][Simulator]

This function is used to perform coverage measurements.

The coverage function provides the following [Dynamic analysis information](#):

- Code coverage rates
- Data coverage rates

- Cautions
1. [IECUBE[V850]][IECUBE2]  
A measurement of the data coverage rates is not supported.
  2. [E20[RX]]  
The code coverage rates are measured only when the selected microcontroller supports the coverage function (a measurement of the data coverage is not supported).
  3. [Simulator[RH850]]  
The coverage is measured for access in all PEs. In the Local RAM self area, note that the measurement results will be displayed only for the access in PE $n$  currently selected in the Debug Manager panel.

## 1.2 Features

The features of the analyze tool are shown below.

### - Displaying function information

The following information about functions is displayed:

[Static analysis information](#) (e.g. function name, file name, attribute, return type, reference count, etc.) and [Dynamic analysis information](#) (e.g. execution count, execution time, code coverage, etc.).

Function Name	File Name	Attribute	Return Type	Arguments
AD_Init	CG_ad.c	-	void	void
AD_Read	CG_ad.c	-	unsigned short	unsigned short * buffer
AD_Start	CG_ad.c	-	void	void
AD_Stop	CG_ad.c	-	void	void
MD_INTAD	CG_ad_user.c	-	void	void
TMP0_Start	CG_timer.c	-	void	void
TMP1_Start	CG_timer.c	-	void	void
ad_receive	(No Definition)	-	-	-
func1	CG_main.c	-	void	void
func1a	CG_main.c	-	void	void
func2	CG_main.c	-	void	void
func2a	CG_main.c	-	void	void

### - Displaying variable information

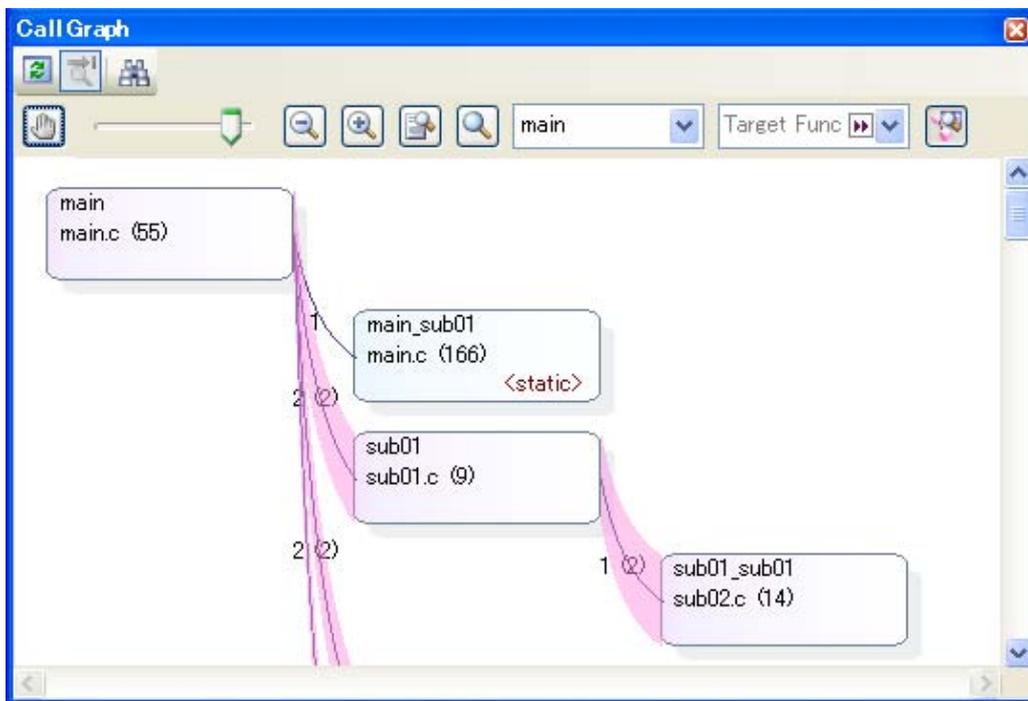
The following information about variables is displayed:

[Static analysis information](#) (e.g. variable name, file name, attribute, type, reference count, etc.) and [Dynamic analysis information](#) (e.g. read count, write count, data coverage, etc.).

Variable Name	File Name	Attribute	Type	Members	Address
g_ad_data	CG_main.c	-	unsigned short	-	0x03ffb116
g_ad_finish	CG_main.c	-	unsigned char	-	0x03ffb114
g_count_10ms	CG_main.c	-	unsigned int	-	0x03ffb110
g_count_1ms	CG_main.c	-	unsigned int	-	0x03ffb10c
g_flag_detect	CG_main.c	-	unsigned short	-	0x03ffb118
_S_romp	CG_systeminit.c	-	-	-	-
*Total*	CG_main.c	-	-	-	-
*Total*	CG_systeminit.c	-	-	-	-

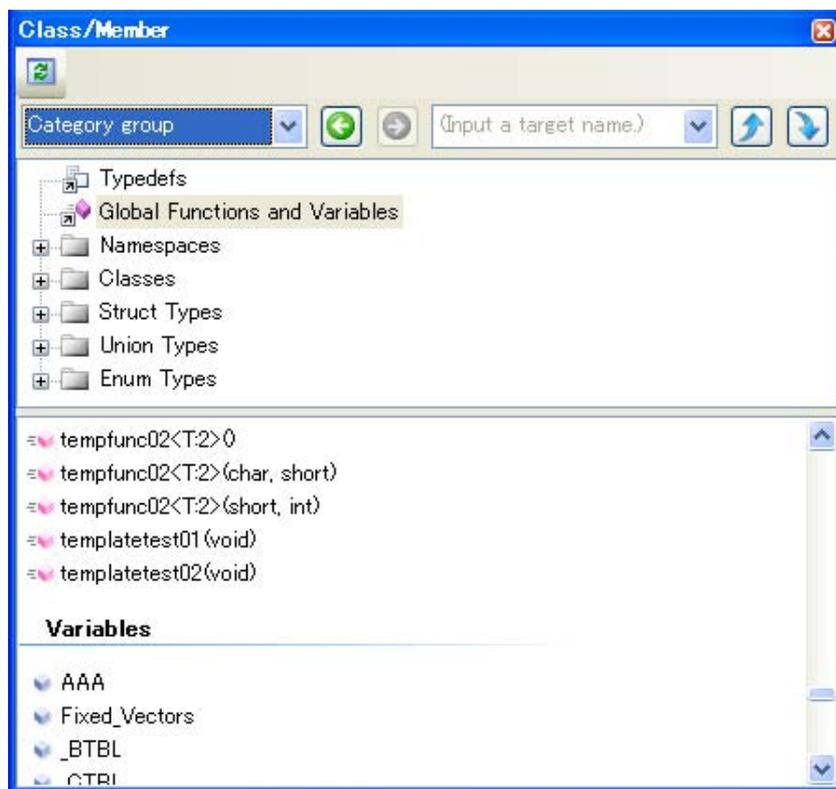
- Displaying the call graph

A tree-structured chart (call graph) indicating calling relationships between functions in the program is displayed.



- Displaying class information/function information/variable information

The class information [CC-RX], function information and variable information in the program are displayed in a tree structure.



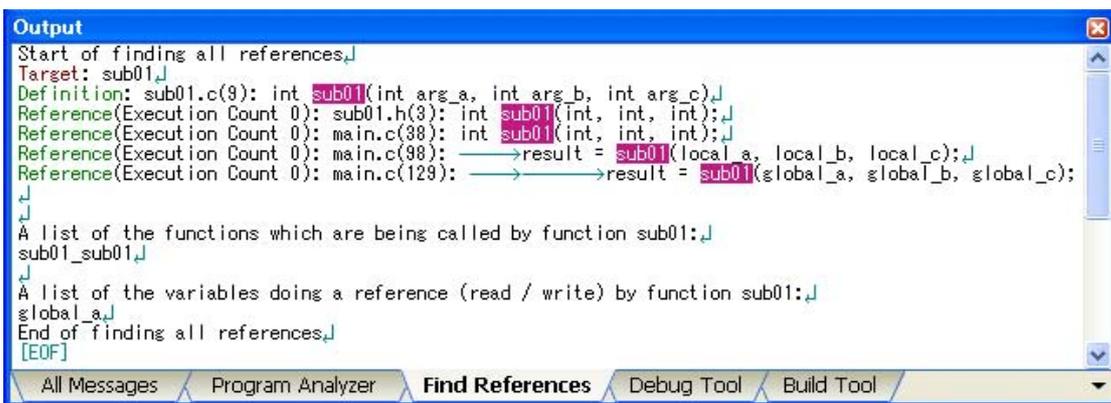
**- Linked operation with debug tool**

The following operations can be performed on the debug tool from the various panels of the analyze tool.

- Jump to the location where the specified function or variable is defined:  
Function List panel/Variable List panel/Call Graph panel/Class/Member panel
- Set a break event at the specified function or variable:  
Function List panel/Variable List panel
- Register a watch expression for the specified variable:  
Variable List panel

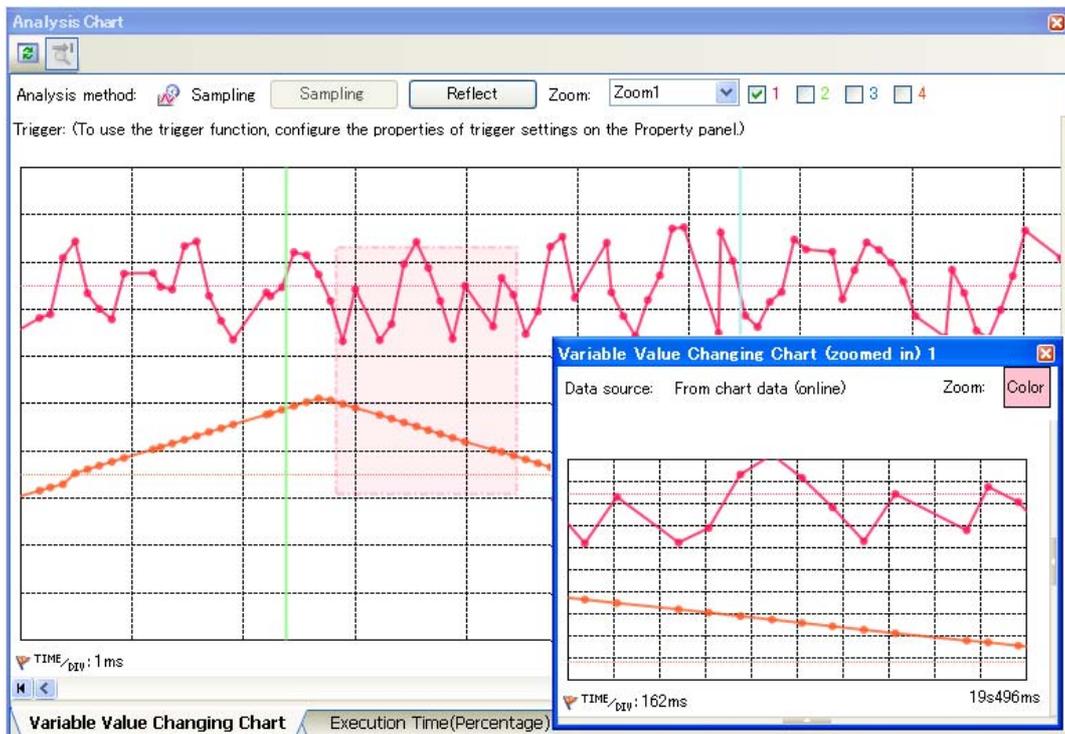
**- View list of locations referencing function/variable**

You can find and display a list of locations referencing a function/variable.

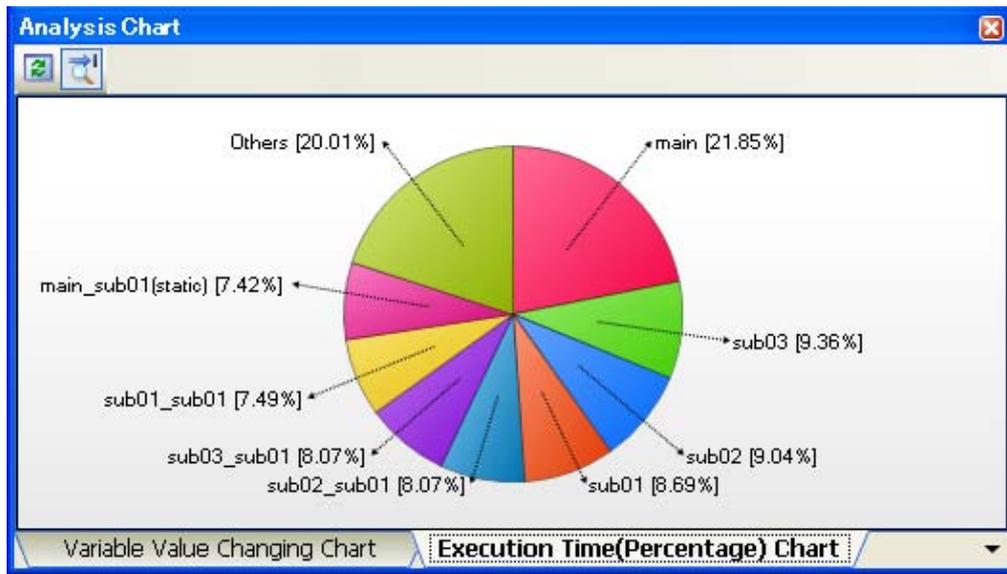


**- Graph analysis information**

- A line chart indicating changes in values of variables or of data collected for Smart Analog [E1/E20[RL78]] is displayed.



- A pie chart indicating function execution time ratios is displayed.



CHAPTER 2 FUNCTIONS

This chapter describes the main features and operating procedures of the analyze tool.

2.1 Overview

The analysis information acquired by the analyze tool is displayed on the following panels.

Table 2-1. Panels Displaying Analysis Information

Panel	Description	
Function List panel	Displays the function information.	
Variable List panel	Displays the variable information.	
Analysis Chart panel	Displays charts of the function information and the variable information.	
	[Variable Value Changing Chart] tab	Displays changes in values of variables or data collected for Smart Analog [E1/E20[RL78]] <sup>Note 1</sup> (line chart).
	[Execution Time(Percentage) Chart] tab	Displays function execution time ratios (pie chart).
Call Graph panel	Displays calling relationships between functions (call graph).	
Class/Member panel	Displays the class information [CC-RX] <sup>Note 2</sup> , function information and variable information in a tree structure.	
Variable Value Changing Chart (zoomed in) panel	Zooms in on the chart displayed in the [Variable Value Changing Chart] tab.	

Notes 1. [E1/E20[RL78]]

Data collection for Smart Analog is supported only when the selected microcontroller incorporates a Smart Analog IC.

2. [CC-RX]

The class information is provided only when C++ source files are subject to analysis.

In the each panel, the following types of function/variable are subject to analysis.

Table 2-2. Types of Functions/Variables That Are Subject to Analysis

Types	Panel				
	Function List	Variable List	Analysis Chart Variable Value Changing Chart (zoomed in)	Call Graph	Class/Member
Global function	✓	-	-	✓	✓
Static function	✓	-	-	✓	✓
Member function [CC-RX] <sup>Note 1</sup>	✓	-	-	✓	✓
Global variable	-	✓	✓	✓	✓
Static variable inside a file	-	✓	✓	✓	✓
Static variable inside a function	-	✓	✓	✓	-
Local variable	-	-	-	-	-

Types	Panel				
	Function List	Variable List	Analysis Chart Variable Value Changing Chart (zoomed in)	Call Graph	Class/Member
IOR/SFR <sup>Note 2</sup>	-	✓	✓	✓	-
Class variable [CC-RX] <sup>Note 1</sup>	-	✓	✓	✓	✓
Instance variable [CC-RX] <sup>Note 1</sup>	-	-	✓	-	✓

**Notes 1. [CC-RX]**

This is a function/variable that exists only in C++ source files.

**2. [RH850][RX][V850]: IOR**

**[R8C(Localised support)][RL78][78K0R][78K0]: SFR**

By validating this analysis information, it is possible to search for unused functions and variables, and processing that is creating bottlenecks, allocate memory effectively to reduce code size, and the like.

The basic operation sequence of the analyze tool is as follows:

**(1) Start CubeSuite+**

Launch CubeSuite+ from the [Start] menu of Windows.

**Remark** For details on "Start CubeSuite+", see "CubeSuite+ Integrated Development Environment User's Manual: Start".

**(2) Set a project**

Create a new project, or load an existing one.

**Remark** For details on "Set a project", see "CubeSuite+ Integrated Development Environment User's Manual: Start".

**(3) Create a load module**

Create a load module by running a build after setting of the active project and the build tool to be used.

**Caution** Run a build after setting the [Enable static analysis information] property in the [General] category on the [Settings] tab of the Property panel to [Yes], in order to generate the cross reference information required for the analyze tool without depending on the configuration of the build-tool being used (see "(1) Static analysis information").

**Remarks 1.** For details on "Create a load module", see "CubeSuite+ Integrated Development Environment User's Manual: Build" of the compiler to use.

2. If the build completes without compiler or assembler errors, then at this point the [Static analysis information](#) can be displayed in the [Function List panel](#)/[Variable List panel](#)/[Call Graph panel](#)/[Class/Member panel](#).

**(4) Execute downloading**

Connect CubeSuite+ to the debug tool after configuring the operational environment of the debug tool, and then download the load module created in steps (3) to the debug tool.

**Caution** Connect to the debug tool after setting the [Enable dynamic analysis information] property in the [General] category on the [Settings] tab of the **Property panel** to [Yes], in order to acquire the dynamic information necessary for the analyze tool without depending on the configuration of the debug tool being used (see "(2) [Dynamic analysis information](#)").

**Remark** For details on "Execute downloading", see "CubeSuite+ Integrated Development Environment User's Manual: Debug" of the microcontroller used.

#### (5) **Execute programs**

Execute the program on the debug tool.

**Remark** For details on "Execute programs", see "CubeSuite+ Integrated Development Environment User's Manual: Debug" of the microcontroller used.

#### (6) **Specify the files to be analyzed**

If necessary, specify the files to be analyzed by the analyze tool (see "1.1.1 [What is analyzed](#)").

#### (7) **Display Function Information**

Display the **Function List panel** in order to validate the acquired function information.

The **Function List panel** can be operated as follows, in accordance with the purpose.

Note that if the program is modified during debugging, the operation is repeated via (3), above.

##### (a) **Customize Display Method**

Change the display method (e.g. set which columns to display and the sorting order) on the panels.

##### (b) **Jump to Defined Location**

Jump to the location where the function on the panel has been defined.

##### (c) **Set a breakpoint to a function**

Set a breakpoint to the function on the panel.

##### (d) **Display List of Referencing Location**

Find and display a list of locations referencing the function on the panel.

**Remark** It is also possible to display the function information defined in other than the active project.

For detailed instructions on displaying function information, see "2.12 [Import or Export Information File](#)".

#### (8) **Display Variable Information**

Display the **Variable List panel** in order to validate the acquired variable information.

The **Variable List panel** can be operated as follows, in accordance with the purpose.

Note that if the program is modified during debugging, the operation is repeated via (3), above.

##### (a) **Customize Display Method**

Change the display method (e.g. set which columns to display and the sorting order) on the panels.

##### (b) **Jump to Defined Location**

Jump to the location where the function on the panel has been defined.

**(c) Set a break event to a variable**

Set a break event with access condition to the variable on the panel.

**(d) Register Watch-Expressions**

Register variables to the Watch panel (Watch1) as watch-expressions.

**(e) Display List of Referencing Location**

Find and display a list of locations referencing the variable on the panel.

**Remark** It is also possible to display the function information defined in other than the active project.

For detailed instructions on displaying function information, see "2.12 Import or Export Information File".

**(9) Display Calling Relationships between Functions (Call Graph)**

Display the [Call Graph panel](#) in order to validate the relationship between functions.

The [Call Graph panel](#) can be operated as follows, in accordance with the purpose.

**(a) Jump to Defined Location**

Jump to the location where the function/variable on the call graph has been defined.

**(b) Search for a function/variable**

Search for an arbitrary function/variable existing in the call graph.

**(10) Display Class/Function/Variable Information**

Display the [Class/Member panel](#) in order to validate the class information **[CC-RX]**, function information and variable information.

The [Class/Member panel](#) can be operated as follows, in accordance with the purpose.

**(a) Jump to Defined Location**

Jump to the location where the class/function/variable on the tree has been defined.

**(b) Jump to Declared Location**

Jump to the location where the class/function/variable on the tree has been declared.

**Remark** **[CC-RX]**

The class information is provided only when C++ source files are subject to analysis.

**(11) Display Analysis Information in Chart**

Display acquired function information/variable information in a chart.

Note that if the program is modified during debugging, the operation is repeated via (3), above.

**(a) Graph transitions in values**

Display the relationship between values of variable/register/address or of data collected for Smart Analog<sup>Note</sup> and time in a line chart.

**Note** **[E1/E20[RL78]]**

This feature is supported only when the selected microcontroller incorporates a Smart Analog IC.

**(b) Chart the function execution time ratios**

Display a pie chart of the function execution time ratios.

**(12) Save Analysis Information**

Save the acquired analysis information to a file.

**(13) Save the project file**

Save the setting information of the project to the project file.

**Remark** For details on "Save the project file", see "CubeSuite+ Integrated Development Environment User's Manual: Start".

## 2.2 Display Function Information

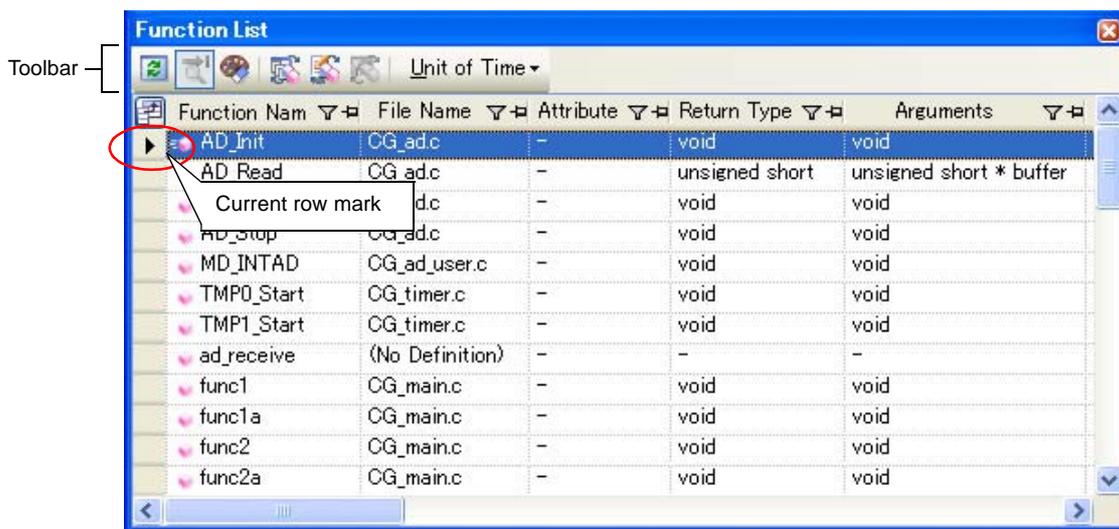
Display the function information (function name, file name, attribute, reference count, execution count, code coverage, etc.) in the program.

The latest function information that can currently be displayed (see "1.1.2 Types of analysis information") is displayed on the following **Function List panel**, which is opened by clicking the  button on the **Main window's toolbar**.

Note, however, that the function information in the files not to be analyzed is not displayed in this panel (see "1.1.1 What is analyzed").

See the **Function List panel** for details on each of the function information items displayed.

Figure 2-1. Display Function Information (Function List Panel)



- Cautions 1.** [E20[RX]]  
 In the debug tool's Property panel, when the [Trace data type] property in the [Trace] category on the [Debug Tool Settings] tab is set to [Branch], the acquisition time of the **Dynamic analysis information** by trace data will increase because complement of instructions between branch instructions is performed (the processing time is proportional to the set value of the [Trace memory size[MByte]] property in the same category above).
2. [CC-RX]  
 When C++ source files are subject to analysis, see "3.6 When CC-RX (C++ Source Files) Is Used".
  3. [CC-RX][CX][NC30(Localised support)]  
 Unused static functions deleted via compiler optimization cannot be displayed in the panel.
  4. [CA850][CA78K0R][CA78K0]  
 When a clean is executed in the build tool, the contents currently being displayed in this panel will be cleared.
  5. [CA78K0R][CA78K0]  
 If a source file with the same name exists in a project, the build tool rewrites the cross reference information of them. Therefore, the information of other than the source file that has been compiled last in files with the same name cannot be acquired.

**Remarks 1.** The display contents are updated each time program execution is stopped.  
 Note, however, that if the [Refresh at program stop] property in the [General] category on the [Settings] tab of the **Property panel** is set to other than [Yes] (default), the information will be updated in accordance with the setting on the **Property panel**.

2. The value of the information that has been changed because of the execution of a program is shown highlighted (the color depends on the configuration in the [General - Font and Color] category of the Option dialog box). To reset the highlighting, click the  button on the toolbar.
3. The current row mark ( ▶ ) indicates the current row.  
The following operations can be performed on the function with the current row mark.
  - [Jump to Defined Location](#)
  - [Set Break Events](#)
  - [Display List of Referencing Location](#)
4. Information for functions defined outside the active project, and which are never referenced from the active project, is ordinary not displayed in the [Function List panel](#).  
For detailed instructions on displaying function information in this case, see "[2.12 Import or Export Information File](#)".

### 2.3 Display Variable Information

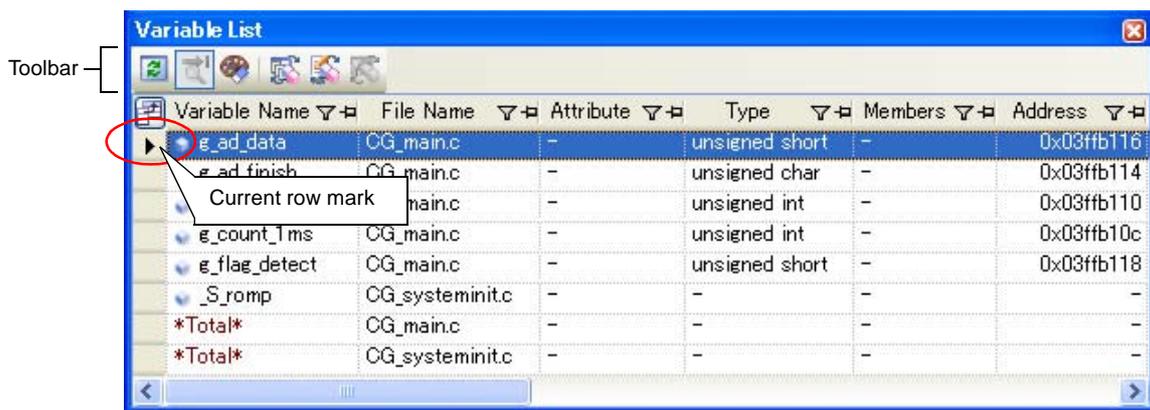
Display the variable information (variable name, file name, attribute, read count, write count, data coverage, etc.) in the program.

The latest variable information that can currently be displayed (see "1.1.2 Types of analysis information") is displayed on the following [Variable List panel](#), which is opened by clicking the  button on the [Main window's toolbar](#).

Note, however, that the variable information in the files not to be analyzed is not displayed in this panel (see "1.1.1 What is analyzed").

See the [Variable List panel](#) for details on each of the variable information items displayed.

Figure 2-2. Display Variable Information (Variable List Panel)



- Cautions 1.** [E20[RX]]  
 In the debug tool's Property panel, when the [Trace data type] property in the [Trace] category on the [Debug Tool Settings] tab is set to [Branch], the acquisition time of the [Dynamic analysis information](#) by trace data will increase because complement of instructions between branch instructions is performed. If you focus only on variables, we recommend selecting [Data access] in this property.
2. [CC-RX]  
 When C++ source files are subject to analysis, see "3.6 [When CC-RX \(C++ Source Files\) Is Used](#)".
3. [CC-RX][CX][NC30(Localised support)]  
 Unused variables deleted via compiler optimization are not displayed.
4. [CA850]  
 If there are assembler instructions coded between "#pragma asm" and "#pragma endasm", then the registers and instructions coded in that location will be displayed as variables.
5. [CA850][CA78K0R][CA78K0]  
 When a clean is executed in the build tool, the contents currently being displayed in this panel will be cleared.
6. [CA78K0R][CA78K0]  
 If a source file with the same name exists in a project, the build tool rewrites the cross reference information of them. Therefore, the information of other than the source file that has been compiled last in files with the same name cannot be acquired.

- Remarks 1.** The display contents are updated each time program execution is stopped.  
 Note, however, that if the [Refresh at program stop] property in the [General] category on the [Settings tab](#) of the [Property panel](#) is set to other than [Yes] (default), the information will be updated in accordance with the setting on the [Property panel](#).

2. The value of the information that has been changed because of the execution of a program is shown highlighted (the color depends on the configuration in the [General - Font and Color] category of the Option dialog box). To reset the highlighting, click the  button on the toolbar.
3. The current row mark ( ▶ ) indicates the current row.  
The following operations can be performed on the function with the current row mark.
  - [Jump to Defined Location](#)
  - [Set Break Events](#)
  - [Display List of Referencing Location](#)
  - [Graph transitions in values](#)
4. Information for functions defined outside the active project, and which are never referenced from the active project, is ordinary not displayed in the [Variable List panel](#).  
For detailed instructions on displaying function information in this case, see "[2.12 Import or Export Information File](#)".

### 2.4 Display Calling Relationships between Functions (Call Graph)

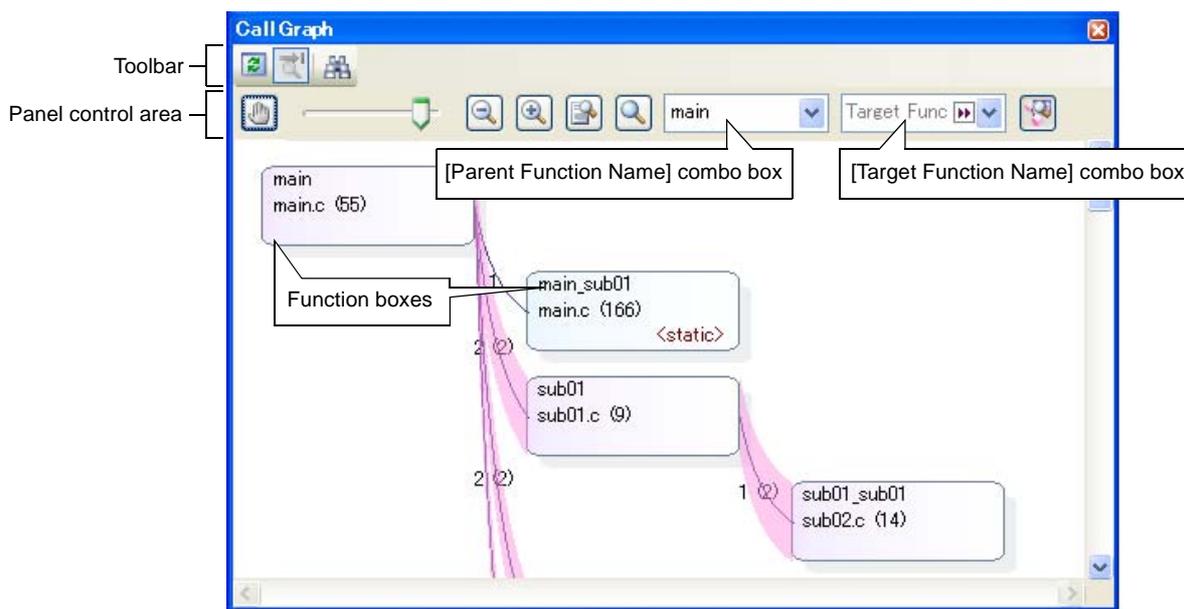
You can display a tree-structured chart (call graph) that represents calling relationships between functions in the program based on the acquired function information.

The latest call graph that can currently be displayed (see "1.1.2 Types of analysis information") is displayed on the following Call Graph panel, which is opened by clicking the  button on the Main window's toolbar.

Note, however, that the function information and variable information in the files not to be analyzed is not displayed in this panel (see "1.1.1 What is analyzed").

See the Call Graph panel for details on the call graph displayed.

Figure 2-3. Display Calling Relationships between Functions (Call Graph Panel: Whole View)



- Cautions 1.** If the debug tool to use does not support the trace function, or if the debug tool's trace function is not enabled, then the **Dynamic analysis information** (e.g. the execution count/read count/write count) in the call graph cannot be displayed.  
For the dynamic analysis information, see the caution of "(a) Trace function".
2. [CC-RX]  
When C++ source files are subject to analysis, see "3.6 When CC-RX (C++ Source Files) Is Used".
  3. [CA850][CA78K0R][CA78K0]  
When a clean is executed in the build tool, the contents currently being displayed in this panel will be cleared.

The following operations can be performed on the call graph displayed.

**(1) Change a parent function**

By default, the first function to appear named "main" [RH850][V850][RL78][78K0R][78K0]/"reset "[RX]/"start" [R8C(Localised support)] or containing such word is considered to be the parent function, and that function is placed on the left edge of the call graph in a box format (function box).

Select a function from the drop-down list of the [Parent Function Name] combo box to change the parent function to be displayed in call graph.

**Remark [RH850][V850E2]**

If the selected microcontroller version supports multi-core, you can select "PMn"[RH850]/"PEn"[V850E2] in the [Parent Function Name] combo box. In this case, the call graph will only be displayed for the functions executed in the selected PMn/PEn.

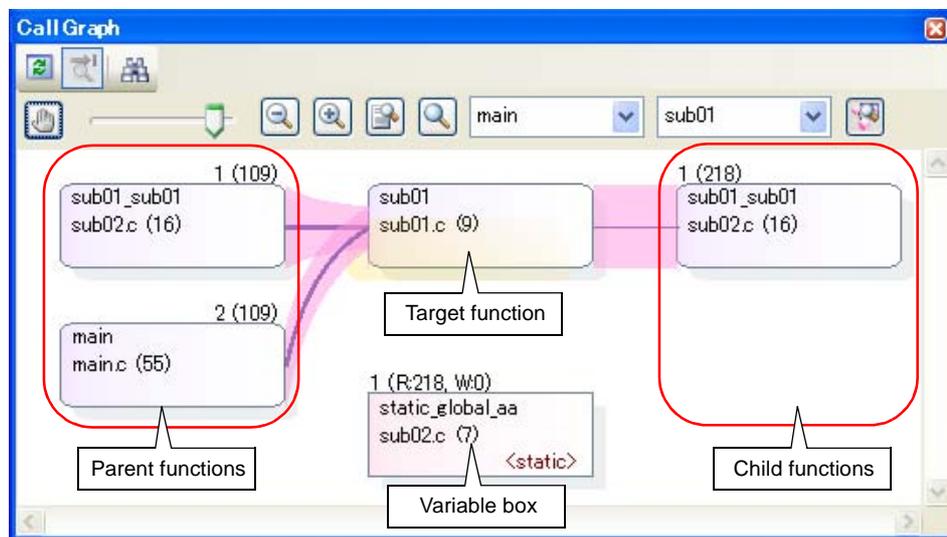
**(2) Display the parent and child functions of the specified function**

By clicking the  button on the panel control area, the call graph is switched to **Details view** that displays the parent functions and child functions of the function currently being specified with the [Target Function Name] combo box.

In the details view, when global variables, static variables inside a file or static variables inside a function accessed from the target function exist, these variables are also displayed as variable boxes.

To switch to **Whole view (default)**, click the  button again.

**Figure 2-4. Display Parent and Child Functions (Call Graph Panel: Details View)**

**(3) Jump to defined location**

By double-clicking on a function box/variable box in the call graph, you can jump to the location of the source text where that function/variable is defined (see "2.7 Jump to Defined Location").

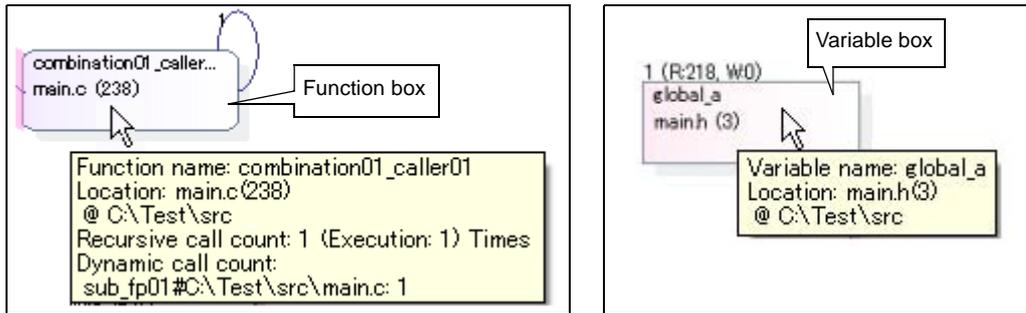
**(4) Pop-up display of the function/variable information**

When the mouse cursor is hovered over a function box/variable box in the call graph, information about that function/variable appears in a pop-up.

**Caution** This function cannot be used while scrolling via the mouse dragging is being enabled by the

 button on the panel control area.

Figure 2-5. Pop-up Display of Function/Variable Information

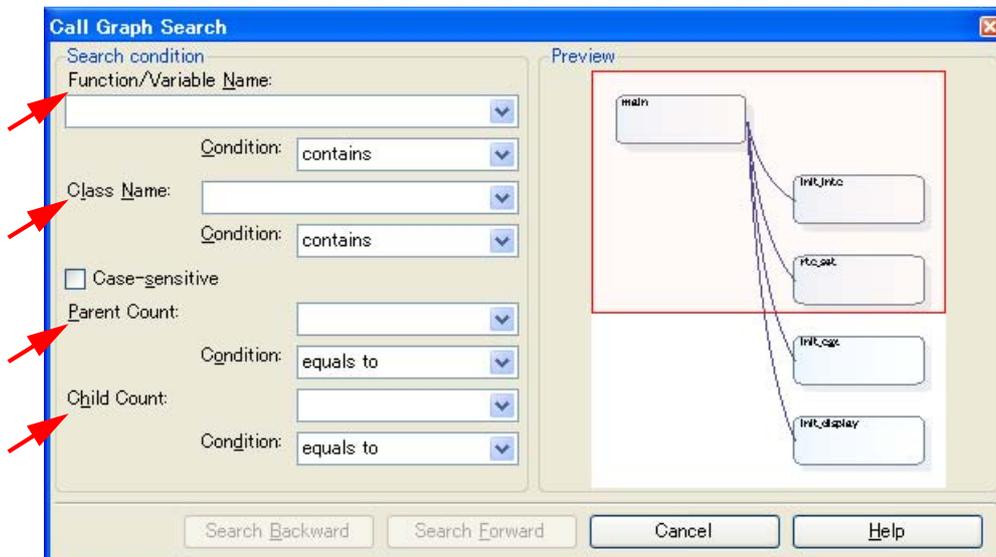


**(5) Search for a function/variable**

You can search for an arbitrary function/variable existing in the call graph.

To do this, click the  button on the Call Graph panel's toolbar to open the Call Graph Search dialog box. In this dialog box, follow the steps below.

Figure 2-6. Search for Functions (Call Graph Dialog Box)



**(a) Specify [Function/Variable Name]**

Specify a function name/variable name to search.

You can either type character strings directly into the text box (up to 2046 characters), or select one from the input history via the drop-down list (up to 10 items).

Next, to specify conditions for the specified character strings, select the desired conditions in the [Condition] combo box ([contains] is selected by default).

Searches are case-insensitive if the [Case-sensitive] check box is selected.

**Remark** You can also specify the function name/variable name with any one of the following methods.

- Drag and drop an arbitrary row of the [Function List panel](#)/[Variable List panel](#) to this area.
- Drag and drop an arbitrary character strings to this area.

**(b) Specify [Class Name] [CC-RX]**

If required as a search parameter for a function/variable, you can specify the class name to which a member function/member variable to search belongs.

You can either type character strings directly into the text box (up to 2046 characters), or select one from the input history via the drop-down list (up to 10 items).

Next, to specify conditions for the specified character strings, select the desired conditions in the [Condition] combo box ([contains] is selected by default).

Searches are case-insensitive if the [Case-sensitive] check box is selected.

**Remark** You can also specify the class name with any one of the following methods.

- Drag and drop an arbitrary character strings to this area.

**(c) Specify [Parent Count]**

If required as a search parameter for a function, you can specify the number of parent functions to search.

You can either type a value directly into the text box (settable range: 0 to 65535), or select one from the input history via the drop-down list (up to 10 items).

Next, to specify conditions for the specified value, select the desired conditions in the [Condition] combo box ([equals] is selected by default).

**(d) Specify [Child Count]**

If required as a search parameter for a function, you can specify the number of child functions to search.

You can either type a value directly into the text box (settable range: 0 to 65535), or select one from the input history via the drop-down list (up to 10 items).

Next, to specify conditions for the specified value, select the desired conditions in the [Condition] combo box ([equals] is selected by default).

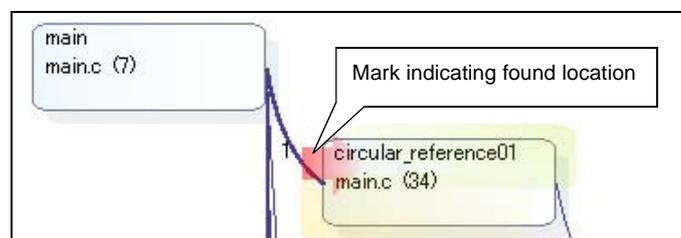
**(e) Click the [Search Backward]/[Search Forward] button**

When the [Search Backward] button is clicked, the search uses the specified conditions, starting from the function name/variable name at the bottom of the call graph, and proceeding upward. The corresponding function box/variable box is highlighted.

When the [Search Forward] button is clicked, the search uses the specified conditions, starting from the function name/variable name at the top of the call graph, and proceeding downward. The corresponding function box/variable box is highlighted.

If a function box/variable box has been selected on the [Call Graph panel](#), then the search starts from the corresponding function, moving upward/downward.

**Figure 2-7. Location Where Function Has Been Found**



- Remarks 1.** The display contents are updated each time program execution is stopped. Note, however, that if the [Refresh at program stop] property in the [General] category on the [\[Settings\] tab](#) of the [Property panel](#) is set to other than [Yes] (default), the information will be updated in accordance with the setting on the [Property panel](#).
- 2.** A function/variable whose source file does not exist in the project can be displayed in the call graph if the [Display the function/variable without definition at Call Graph panel] property in the [General] category on the [\[Settings\] tab](#) of the [Property panel](#) is set to [Yes].

## 2.5 Display Class/Function/Variable Information

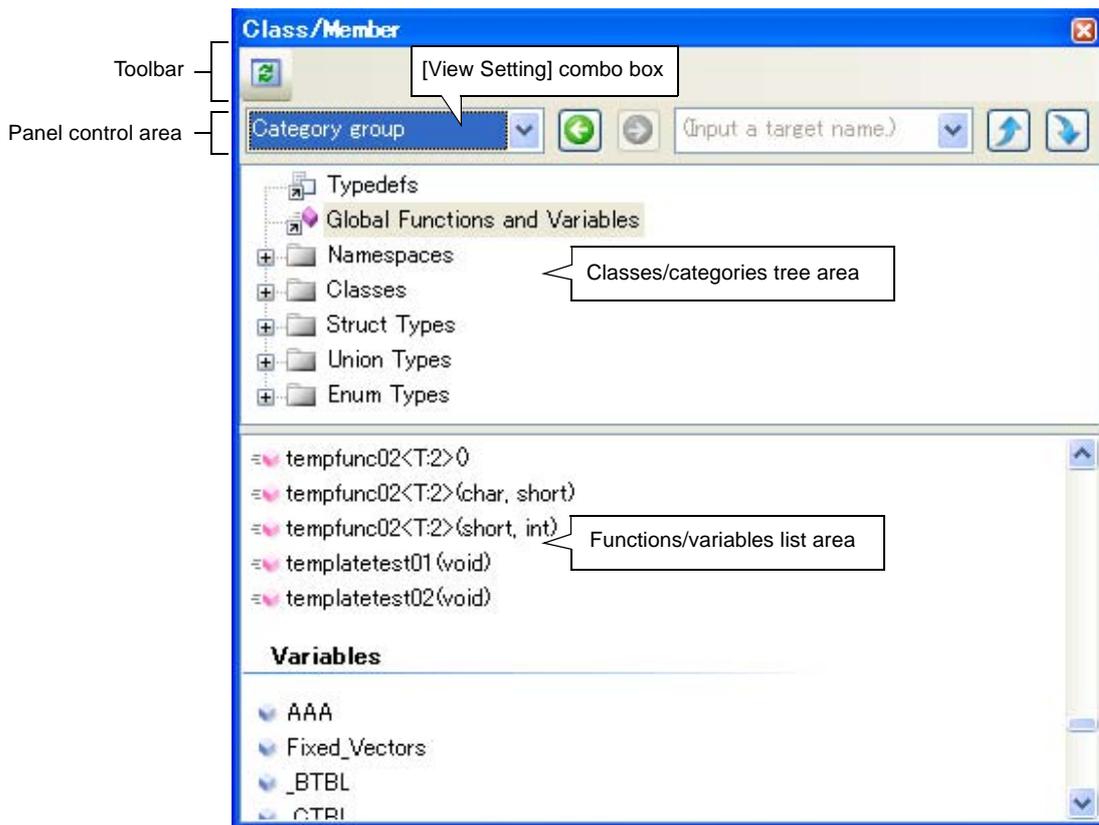
You can display the class information [CC-RX], function information and variable information in the program by using a tree structure.

These information is displayed on the following [Class/Member panel](#), which is opened by clicking the  button on the [Main window's](#) toolbar.

Note, however, that the class information [CC-RX], function information and variable information in the files not to be analyzed is not displayed in this panel (see "1.1.1 What is analyzed").

See the [Class/Member panel](#) for details on the tree displayed.

**Figure 2-8. Display Class Information (Class/Member Panel)**



### Cautions 1. [CC-RX]

The class information is provided only when C++ source files are subject to analysis. When C++ source files are subject to analysis, see "3.6 When CC-RX (C++ Source Files) Is Used".

### 2. [CA850][CA78K0R][CA78K0]

If a clean is executed in the build tool, the contents of this panel being displayed will be erased.

The acquired information is displayed in the classes/categories tree area (top) and the functions/variables list area (bottom). When a node on the tree displayed in the classes/categories tree area is selected, the information of the node (e.g. function names/variable names defined in the selected node) will be displayed in the functions/variables list area.

You can use the [\[View Setting\] combo box](#) in the panel control area to classify the items in the classes/categories tree into the following five groups.

Table 2-3. Classification Methods for Tree on Class/Member Panel

Items of List	Description
<a href="#">Category group (default)</a>	Classifies and displays classes, interfaces, and other types.
<a href="#">Access group</a>	Classifies and displays nodes by their access specifiers specified in their classes, etc..
<a href="#">Namespace group</a>	Classifies and displays nodes by namespaces in which the class and other elements are defined.
<a href="#">File group</a>	Classifies and displays nodes by the files in which the class and other elements are defined.
<a href="#">Alphabet group</a>	Displays nodes in alphabetical order.

**Remarks 1.** The following operations can be performed on the node currently being selected.

- [Jump to Defined Location](#)
- [Jump to Declared Location](#)

**2. [CC-RX]**

When the mouse cursor is hovered over a node (except for a category node), "*Namespace name*" to which the node belongs appears in a pop-up.

**2.6 Customize Display Method**

The following operations will enable you to customize the displaying of items (columns) and the analysis information on the [Function List panel/Variable List panel](#).

All operations refer to the header area of each panel (the area where the column labels are displayed).

- [Set the columns to display](#)
- [Change the order of the columns to display](#)
- [Lock the specific column to display](#)
- [Sort the analysis information](#)
- [Filter the analysis information](#)

**Remark** In the [Column Chooser dialog box](#), click the [Default] button to return all the customizations you have performed via the operations above to the default values.

**2.6.1 Set the columns to display**

You can show or hide any of the items (columns) displayed on the [Function List panel/Variable List panel](#).

- (1) [Restricting the columns displayed](#)
- (2) [Adding columns to display](#)

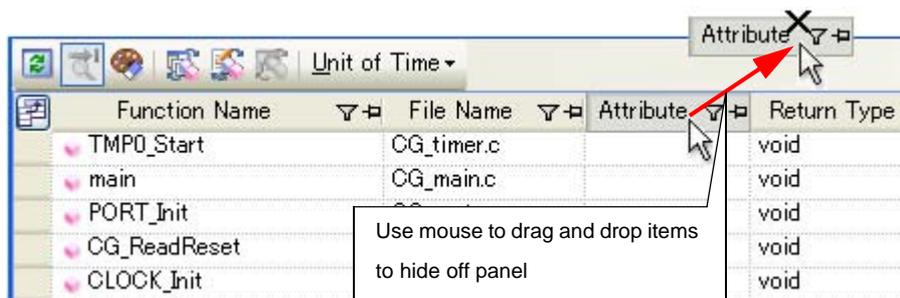
**(1) Restricting the columns displayed**

There are two ways to perform this operation, as shown below.

**(a) Performing the operation from the panel**

Use the mouse to drag and drop the items to hide off of the panel.

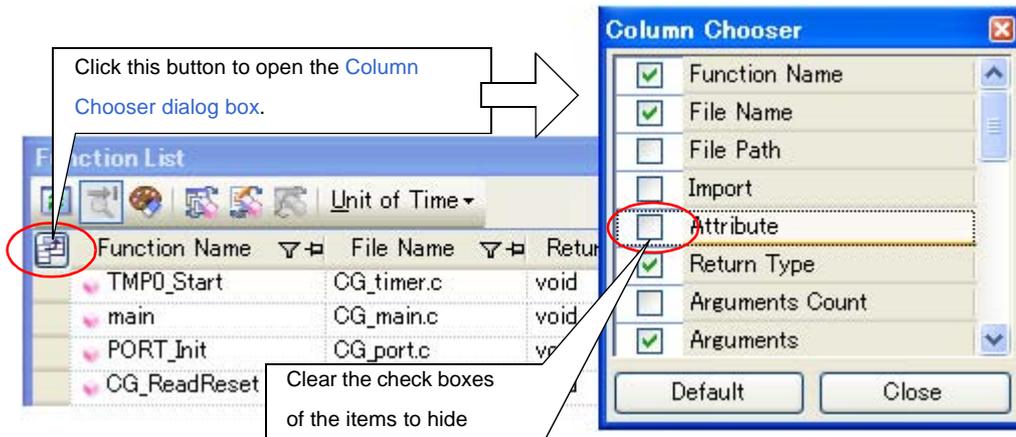
**Figure 2-9. Restricting Columns Displayed (Operation from Panel)**



**(b) Performing the operation from the Column Chooser dialog box**

On the left edge of the panel's header area, click the  button to open the [Column Chooser dialog box](#), and then clear the check boxes of the names of the items to hide.

Figure 2-10. Restricting Columns Displayed (Operation from Column Chooser Dialog Box)



(2) Adding columns to display

On the left edge of the panel's header area, click the  button to open the Column Chooser dialog box, and then select the check boxes of the names of the columns to display, or drag and drop them directly from the dialog box to the information display area on the panel using the mouse.

Figure 2-11. Adding Columns to Display (Clearing Check Boxes in Column Chooser Dialog Box)

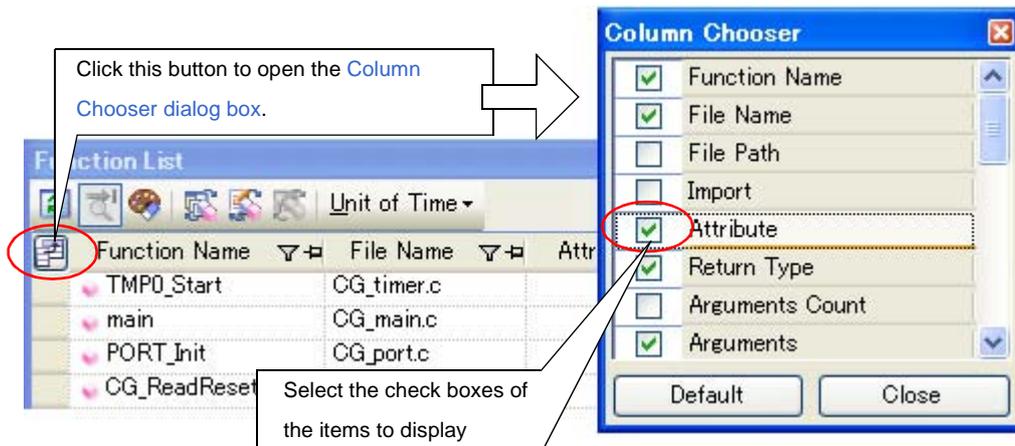
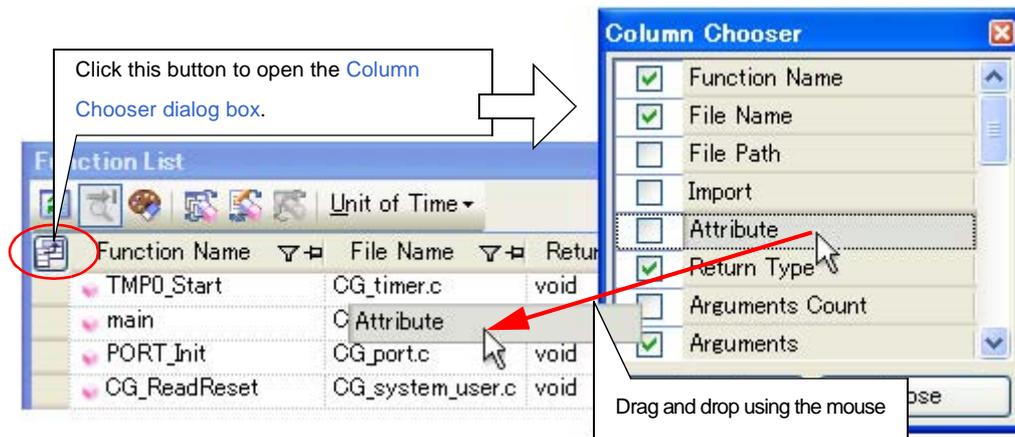


Figure 2-12. Adding Columns to Display (Dragging and Dropping from Column Chooser Dialog Box)



2.6.2 Change the order of the columns to display

You can change the display order of the items (columns) displayed on the [Function List panel/Variable List panel](#). There are two ways to perform this operation, as shown below.

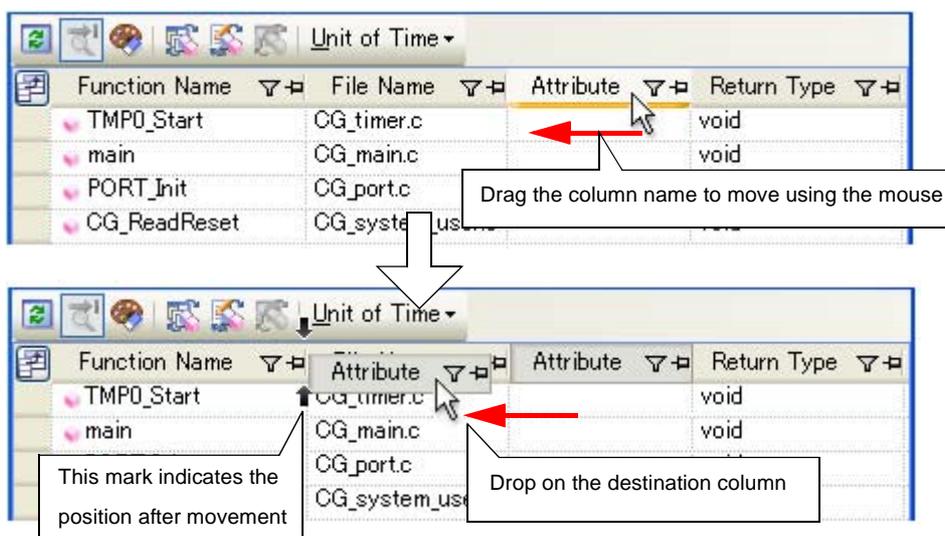
There are two ways to perform this operation, as shown below.

- (1) [Performing the operation from the panel](#)
- (2) [Performing the operation from the Column Chooser dialog box](#)

(1) Performing the operation from the panel

Use the mouse to drag and drop the column name to move directly to an arbitrary column (header area).

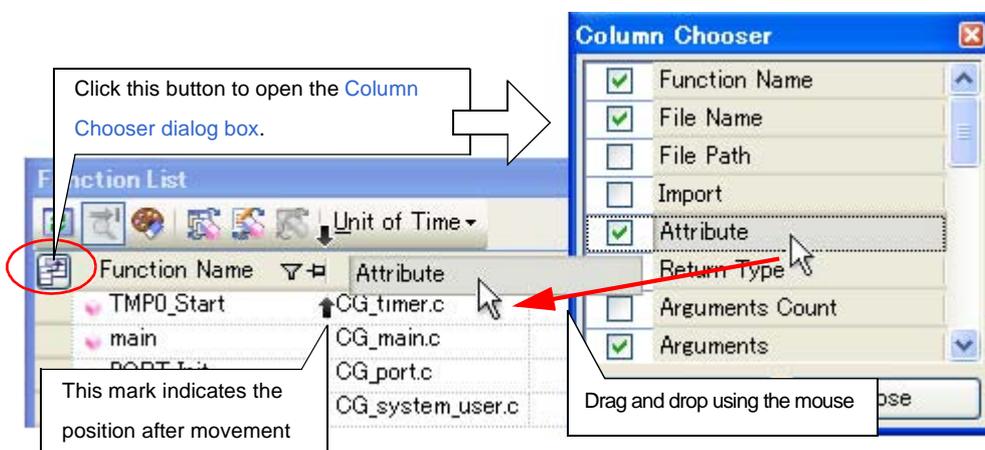
Figure 2-13. Change the Order of Displayed Items (Operation from Panel)



(2) Performing the operation from the Column Chooser dialog box

On the left edge of the panel's header area, click the  button to open the [Column Chooser dialog box](#), and then drag and drop the column names to move directly to the desired column (header area) on the panel using the mouse.

Figure 2-14. Change the Order of Displayed Items (Operation from Column Chooser Dialog Box)



**2.6.3 Lock the specific column to display**

You can set specified items (columns) to always be displayed on the [Function List panel/Variable List panel](#), even if the screen display is scrolled.

To do this, use the mouse to click the lock display icons (  /  ) of the desired column name (clicking repeatedly toggles the item display between locked and unlocked).

Columns configured for locked display are locked at the leftmost column, so they will not be scrolled even if the user scrolls horizontally.

The meanings of the lock display icons are as follows:

**Table 2-4. Lock Display Icons**

Icon	Description
	Indicates that the display is not locked (default).
	Indicates that the display is locked.

- Remarks 1.** You can also lock the display of a column by dragging and dropping it between two items with locked displays, or to the rightmost column.
- 2.** You can also unlock the display of a locked item by dragging and dropping it between two items with unlocked displays.

**2.6.4 Sort the analysis information**

You can sort the information values on the [Function List panel/Variable List panel](#) by column, and in ascending or descending order.

To do this, use the mouse to click the desired column name (clicking repeatedly toggles the sort order between ascending and descending).

If the items to sort are numbers (decimal or hexadecimal), then the information is sorted by numerical value. For other types of data (e.g. strings), the information is sorted by character code.

The following mark appears by the column name of the column that the information is sorted on:

**Table 2-5. Marks that Indicate Sorted Display**

Mark	Description
	Indicates that the information is being sorted in ascending order. Click again with the mouse to sort in descending order.
	Indicates that the information is being sorted in descending order. Click again with the mouse to sort in ascending order.

**Remark** Click while holding down the [Shift] key to sort on multiple columns.

**2.6.5 Filter the analysis information**

You can set filters for displaying information on the [Function List panel/Variable List panel](#).

The following types of filters are available.

- (1) [Filtering via custom settings for each column](#)
- (2) [Filtering results linked to panel](#)

**Caution** The [Filtering via custom settings for each column](#) and the [Filtering results linked to panel](#) are mutually exclusive functions. For this reason, the two filtering functions described here cannot be

enabled simultaneously (when one of the filters is active, setting the other filter will disable the first one).

**(1) Filtering via custom settings for each column**

Configure a custom filter for each column, and display the acquired information.

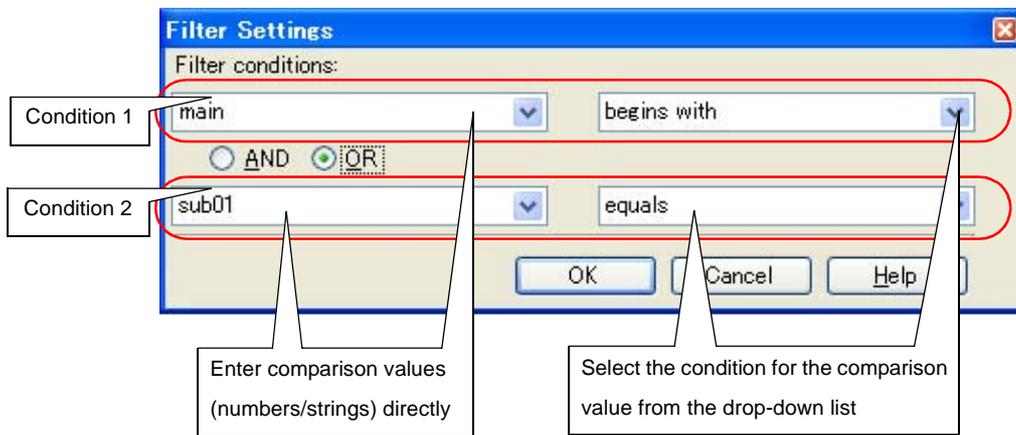
To do this, click on the filter icon (▼ / ▼) of the target column name, then select the following menu item.

**Table 2-6. Filter Display Settings**

Item	Description
(All)	Does not set filter (resets filter display). Displays all information.
(Custom)	Opens the <a href="#">Filter Settings dialog box</a> to set detailed filtering conditions. Only information matching the conditions specified in this dialog will be displayed.
(Blanks)	Displays empty ("-") fields.
(NonBlanks)	Does not display empty ("-") fields.
<i>Information list</i>	All acquired information is displayed in a list as strings. Only values matching the string selected from the list are displayed.

Select [(Custom)] to open the [Filter Settings dialog box](#). Use this dialog box to set up to two condition groups. Both conditions can be linked into a single filtering condition via the logical operator buttons ([AND] / [OR]). Use the top condition-specification area if you only wish to specify one filtering condition.

**Figure 2-15. Filtering via Custom Settings for Each Column (Filter Settings Dialog Box)**



The meanings of the filter icons are as follows:

**Table 2-7. Filter Icons**

Icon	Description
▼	Indicates that no filtering is taking place (default).
▼	Indicates that the information is being filtered.

(2) Filtering results linked to panel

- (a) Linking with the Project Tree panel
- (b) Linking with the Editor panel
- (c) Linking with the Debug Manager panel [RH850][V850E2]

**Caution** If "(1) Filtering via custom settings for each column" is configured while the information is being filtered via the method described below, linking displays that have been taking place will be reset.

**Remark** The three filtering functions described below can be used simultaneously.

(a) Linking with the Project Tree panel

Only display values for functions/variables in a file/category selected in the Project Tree panel. To do this, click the  button on the Function List panel/Variable List panel toolbar to enable this filtering function, then from the Project Tree panel, select the desired file(s)/category(s). Click this button again to disable this function. The relationship between the selections in the Project Tree panel and the filtered functions/variables is as follows:

Table 2-8. Filtering Linked to Project Tree Panel

Selection	Display
Single file in active project	Functions/variables defined in single file
Multiple files in active project	Functions/variables defined in multiple files
Single category in active project	Functions/variables defined in single category
Multiple categories in active project	Functions/variables defined in multiple categories
Combination of files and categories in active project	Functions/variables defined in the selected files and categories
Other than the above	All functions/variables defined in files included in the active project

**Caution** Assembler source files cannot be analyzed.

**Remark** If a header file is selected, then functions/variables defined in that header file are displayed.

(b) Linking with the Editor panel

Only display information for functions/variables starting with the word at the caret position in the Editor panel. To do this, click the  button on the Function List panel/Variable List panel toolbar to enable this filtering function, then move the caret to the desired function/variable name in the Editor panel. Click this button again to disable this function. Note, however, that if there is no word at the caret position (e.g. if it is at a space or tab character), then the display will not be filtered linked to the Editor panel.

(c) Linking with the Debug Manager panel [RH850][V850E2]

Only display information for functions/variables in the common region (Common) and PE*n* currently being selected in the Debug Manager panel.

To do this, click the  button on the Function List panel/Variable List panel toolbar to enable this filtering function. Click this button again to disable this function. Note, however, that if the selected microcontroller version does not support multi-core, then this function will be invalid.

## 2.7 Jump to Defined Location

You can jump to the location where the function, variable, or class **[CC-RX]**, etc. on the [Function List panel](#)/[Variable List panel](#)/[Call Graph panel](#)/[Class/Member panel](#) is defined.

### Cautions 1. **[CC-RH][CC-RX]**

If the target of the following operation is a member of an enumeration type, then a jump will be made to the defined location of the enumeration type. Note, however, that it is not possible to jump to the defined location of a member of an anonymous enumeration type via the member node.

### 2. **[CC-RX]**

When C++ source files are subject to analysis, the defined location information on member functions defined in a template function or a template class cannot be obtained. Therefore, the following operation is invalid for them; except however, when the template function is defined with a static declaration, and defined outside a class.

### 3. **[CX][NC30(Localised support)]**

It is not possible to jump to the defined location of a type via a struct type/union type/enum type node. Furthermore, it is not possible to jump to the defined location of a member of a struct type/union type via the member node.

### 4. **[CA850]**

This jump function may not be performed correctly if "#pragma directive" (like examples below) is described following "#include statement" in the source text.

In this case, move "#pragma directive" before the description of "#include statement".

- Examples 1. #pragma task TASK\_A  
2. #pragma interrupt INTPO functionA

### (1) Jump to the Editor panel

When the jump is executed, the source file in which the target is defined opens in the Editor panel, and the caret moves to the line where it is defined.

To do this, perform the following operation.

Panel	Target	Operation
<a href="#">Function List panel</a>	A line of function	Double-click
<a href="#">Variable List panel</a>	A line of variable	
<a href="#">Call Graph panel</a>	- A function box - A variable box	Double-click <sup>Note</sup>
<a href="#">Class/Member panel</a>	On the classes/categories tree area - A <i>namespace</i> node <b>[CC-RX]</b> - A <i>class name</i> node <b>[CC-RX]</b> - An <i>interface name</i> node <b>[CC-RX]</b> - A <i>struct type name</i> node - An <i>union type name</i> node - An <i>enum type name</i> node	Select [Jump to Source] from the context menu.
<a href="#">Class/Member panel</a>	On the functions/variables list area - A <i>function name</i> node - A <i>variable name</i> node - A <i>macro name</i> node - An <i>alternative name</i> node - An <i>enum member name</i> node	Double-click

**Note** If scrolling via the mouse dragging is being enabled by the  button on the panel control area, this operation cannot be used. In this case, select [Jump to Source] from the context menu after selecting the target, or disable the scroll function temporarily, then perform the operation above.

## (2) Jump to the Disassemble panel/Memory panel

When the jump is executed, the disassembly data/memory list corresponding to the starting address of the target opens in the Disassemble panel (Disassemble1)/Memory panel (Memory1), and the caret moves to the corresponding location (note, however, that this is only possible while connecting to the debug tool).

To do this, perform the following operation.

Panel	Target	Operation
Function List panel	A line of function	One of the following: - Jump to the Disassemble panel: Select [Jump to Disassemble] from the context menu. - Jump to the Memory panel: Select [Jump to Memory] from the context menu.
Variable List panel	A line of variable <sup>Note 1</sup>	
Call Graph panel	- A function box - A variable box	
Class/Member panel	On the functions/variables list area <sup>Note 2</sup> - A <i>function name</i> node - A <i>variable name</i> node - A <i>macro name</i> node - An <i>alternative name</i> node - An <i>enum member name</i> node	

**Notes 1.** You can jump to only the Memory panel.

- 2.** Neither a node on the classes/categories tree area nor a node of an instance variable can be target for this function.

## 2.8 Jump to Declared Location

You can jump to the location of the source text where the function, variable, or class **[CC-RX]**, etc. on the [Class/Member panel](#) is declared.

### Caution **[CC-RX]**

**When C++ source files are subject to analysis, the reference information for functions/variables that are referenced in a member function defined in a template function or a template class cannot be obtained. Therefore, the following operation is invalid for them; except however, when the template function is defined with a static declaration, and defined outside a class.**

When the jump is executed, the source file in which the target is declared opens in the Editor panel, and the caret moves to the line of the function prototype declaration (for C source files) or the line of the function prototype declaration inside class declaration (for C++ source files).

To do this, perform the following operation.

Target	Operation
On the classes/categories tree area - A <i>namespace</i> node <b>[CC-RX]</b> - A <i>class name</i> node <b>[CC-RX]</b> - An <i>interface name</i> node <b>[CC-RX]</b> - A <i>struct type name</i> node - An <i>union type name</i> node - An <i>enum type name</i> node	Select [Jump to Declaration of Source] from the context menu.
On the functions/variables list area - A <i>function name</i> node - A <i>variable name</i> node - A <i>macro name</i> node - An <i>alternative name</i> node - An <i>enum member name</i> node	

**Remark** The line to which the caret moves may be the same as the line of the result of selecting [Jump to Source] (see "2.7 [Jump to Defined Location](#)").

## 2.9 Set Break Events

You can set a break event in the debug tool at a function/variable on the [Function List panel](#)/[Variable List panel](#).

- [Set a breakpoint to a function](#)
- [Set a break event to a variable](#)

### 2.9.1 Set a breakpoint to a function

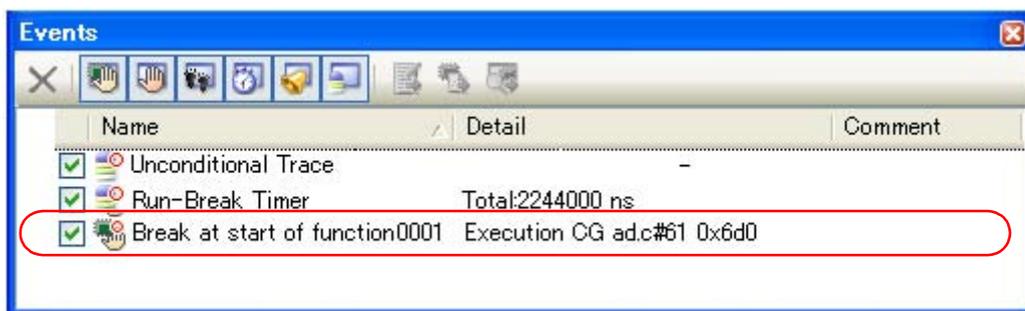
Set a breakpoint at the start of the function on the [Function List panel](#) (the first executable line in the function).

To do this, select the row in which the target function is displayed (make sure that the current row mark ( ▶ ) appears to the left of the selected row), then from the context menu, select [Set Break to Function].

**Caution** This operation is disabled while disconnecting from the debug tool.

A breakpoint set via this operation is managed in the Events panel as an event whose name is "Break at start of function".

Figure 2-16. Example of Setting Breakpoint in Events Panel



**Remarks 1.** If there is already a breakpoint set at the corresponding location, the behavior is as follows:

- If a breakpoint in a valid state is set: No effect
- If a breakpoint in an invalid state is set: Sets the breakpoint to a valid state

#### 2. [CC-RX]

If the target function is a template function and multiple addresses corresponding to its template function exist, then the breakpoints will be set to all the addresses.

### 2.9.2 Set a break event to a variable

Set a break event with access condition to a variable on the [Variable List panel](#).

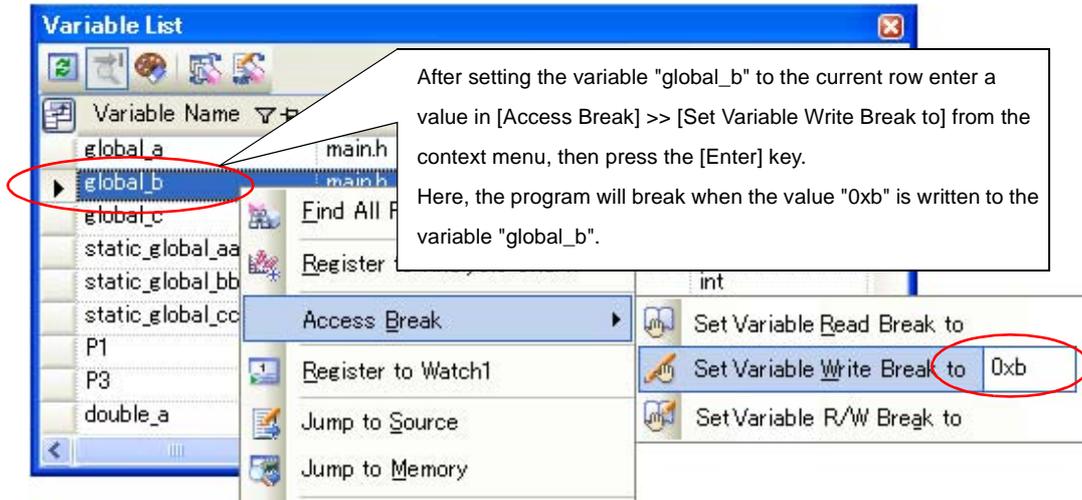
To do this, select the row in which the target variable is displayed (make sure that the current row mark ( ▶ ) appears to the left of the selected row), then from the context menu, select one of the following and then press the [Enter] key.

- [Access Break] >> [Set Variable Read Break to]: Sets a break event with read access condition.
- [Access Break] >> [Set Variable Write Break to]: Sets a break event with write access condition.
- [Access Break] >> [Set Variable R/W Break to]: Sets a break event with read/write access condition.

Note that at this time, you can specify the value in the text box in the context menu.

In this case, execution will only break if it is accessed with the specified value. If you press [Enter] key without entering a value in the text box, then execution will break when the target variable is accessed, regardless of the value.

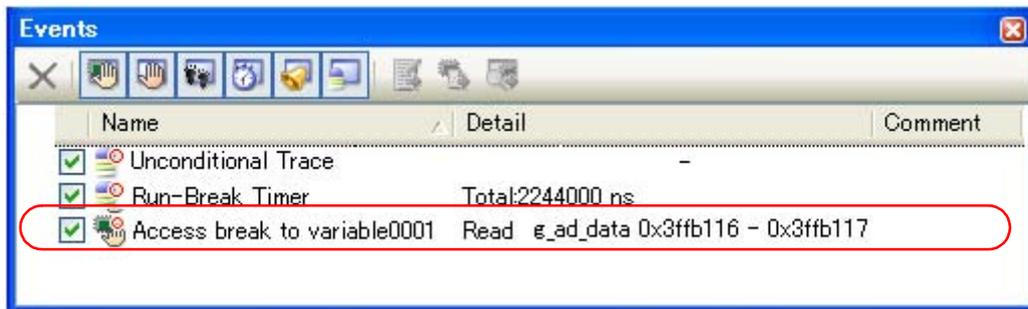
Figure 2-17. Example of Setting Break Event on Variable



- Cautions 1.** This operation is disabled while disconnecting from the debug tool.
- 2.** For combination break events, this function is available only when the combination condition is "OR".

A break event set via this operation is managed in the Events panel as an event whose name is "Access break to variable".

Figure 2-18. Example of Setting Break Event (Access Type) in Events Panel



**Remark** If there is already a breakpoint set at the corresponding location, the behavior is as follows:

- If a break event in a valid state is set: No effect
- If a break event in an invalid state is set: Sets the break event to a valid state

## 2.10 Register Watch-Expressions

You can register variables on the [Variable List panel](#) as watch-expressions in the Watch panel (Watch1).

To do this, drag and drop the row of the target variable (multiple selections possible) directly onto the Watch panel (Watch1).

- Cautions**
1. This operation is disabled while disconnecting from the debug tool.
  2. This operation is disabled for the [\*Total\*] row.

**Remark** The name of the watch-expression registered in the Watch panel differs depending on the type of the target variable, as follows:

- Global variable: *"variable name"*
- Static variable inside a file: *"file name#variable name"*
- Static variable inside a function: *"file name#function name#variable name"*
- Class variable: *"class name::variable name"*

## 2.11 Display List of Referencing Location

You can find and display a list of locations referencing a function/variable on the [Function List panel/Variable List panel](#).

To do this, select the row in which the target function/variable is displayed (multiple selections possible), then from the context menu, select [Find All References].

The results of the search are output to the next [Output panel's \[Find References\] tab](#).

**Figure 2-19. Sample List of Locations Referencing Function (Output Panel)**

```

Output
Start of finding all references
Target: sub01
Definition: sub01.c(9): int sub01(int arg_a, int arg_b, int arg_c)
Reference(Execution Count 0): sub01.h(3): int sub01(int, int, int);
Reference(Execution Count 0): main.c(38): int sub01(int, int, int);
Reference(Execution Count 0): main.c(98): result = sub01(local_a, local_b, local_c);
Reference(Execution Count 0): main.c(129): result = sub01(global_a, global_b, global_c);
A list of the functions which are being called by function sub01:
sub01_sub01
A list of the variables doing a reference (read / write) by function sub01:
global_a
End of finding all references
[EOF]
  
```

The following information about the referencing locations is output to the list as the search results.

For details on the output format, see the section of the [Output panel's \[Find References\] tab](#).

- Performing the operation from the [Function List panel](#)
  - Location where target function is defined
  - List of locations referencing the target function
  - List of functions called within the target function
  - List of variables referenced (read/write) inside the target function
- Performing the operation from the [Variable List panel](#)
  - Location where target variable is defined
  - List of locations referencing the target variable

**Caution** Code in C/C++ source files that is referenced but is eliminated by the preprocessor during compilation (e.g. via "#if" or "#ifndef" statements) is not output as referenced locations.

- Remarks 1.** When the function name/variable name on this panel is double-clicked, the Editor panel appears and the caret moves to the line where the function/variable is defined.
- 2.** You can save the contents displayed on this [Output panel's \[Find References\] tab](#) to a text file (\*.txt) by selecting [Save Output-Find References As...] from the [File] menu.

2.12 Import or Export Information File

Information for functions or variables defined outside the active project, and which are never referenced from the active project, is ordinary not displayed in the [Function List panel/Variable List panel](#).

In this case, it is possible to force this information to be displayed by importing an information file that saves the information about a function/variable you need.

The following types of information files are available.

Table 2-9. Types of Information Files

Information File Name	Contents
Function list file (*.mtfl)	Stores the information about the following items on the <a href="#">Function List panel</a> . [Function Name], [Class Name][ <b>CC-RX</b> ], [Namespace][ <b>CC-RX</b> ], [File Name], [File Path], [PM Information][ <b>RH850</b> ] <sup>Note</sup> , [PE Information][ <b>V850</b> ] <sup>Note</sup> , [Access Specifier][ <b>CC-RX</b> ], [Attribute], [Return Type], [Arguments Count], [Arguments], [Code Size[Bytes]], [Stack Size[Bytes]]
Variable list file (*.mtvl)	Stores the information about the following items on the <a href="#">Variable List panel</a> . [Variable Name], [Class Name][ <b>CC-RX</b> ], [Namespace][ <b>CC-RX</b> ], [File Name], [Function Name], [File Path], [PM Information][ <b>RH850</b> ] <sup>Note</sup> , [PE Information][ <b>V850</b> ] <sup>Note</sup> , [Access Specifier][ <b>CC-RX</b> ], [Attribute], [Type], [Members], [Size[Bytes]]

**Note** This item appears only when the selected microcontroller version supports multi-core.

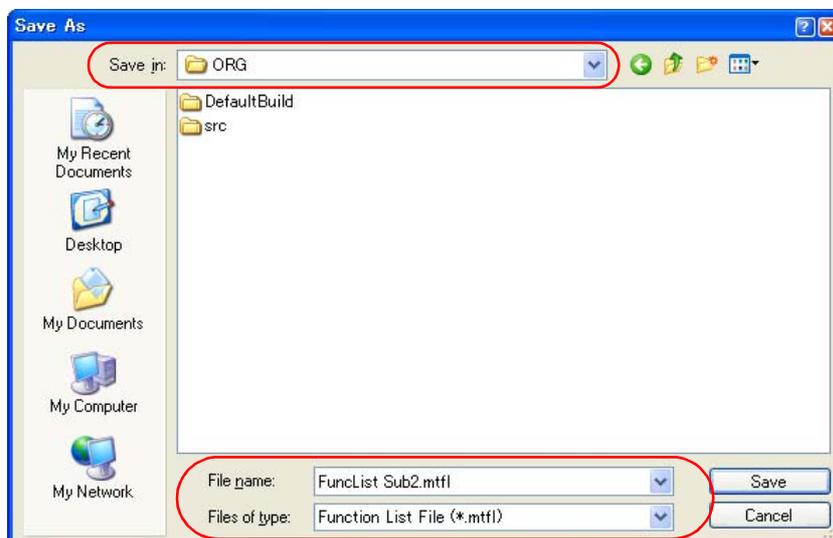
The procedure for doing this is as follows:

(1) Generate (Export) information file

(a) Export an information file manually

This operation will generate an information file with only the functions/variables needed to be imported. To do this, first make the project in which the target functions/variables are defined the active project, then display the necessary information of the target functions/variables in the [Function List panel/Variable List panel](#). Next, select all the rows displaying the target functions/variables in this panel (multiple rows can be selected by holding down the [Shift]/[Ctrl] key while selecting), then from the [File] menu, select [Save Function List Data As...]/[Save Variable List Data As...]. The following [Save As dialog box](#) opens.

Figure 2-20. Export Information File Manually (Example of Function List File)



In the [Save in] area of the above dialog box, select a folder of your choice in which to save the generated information file, then select "Function List File (\*.mtfl)" or "Variable List File (\*.mtvl)" from the drop-down list in the [Files of type] area.

Next, enter a file name of your choice for the information file in the [File name] area. Note, however, that the file extension must be "mtfl" for a function list file, and "mtvl" for a variable list file.

Click the [Save] button to generate the information file in the specified folder, with the specified file name.

**Remark** If you also need information about functions/variables defined outside the active project specified here, then generate a separate information file using the same procedure.

**(b) Export information files automatically**

This operation will automatically generate information files with all the functions/variables displayed in the [Function List panel](#)/[Variable List panel](#) each time a build/rebuild is executed.

To do this, specify [Yes] with the [Export the functions and variables] property in the [Import / Export] category on the [\[Settings\] tab](#) of the [Property panel](#), and then specify a file name of your choice for the information files in the [Export file name for functions]/[Export file name for variables] property. Note, however, that the file extension must be "mtfl" for a function list file, and "mtvl" for a variable list file. If you specify a relative path, specify the project folder as the base.

Figure 2-21. [Import / Export] Category



The above setting completes the automatic export of the information file.

Execute a build/rebuild to generate the information file automatically in the specified folder, with the specified file name.

**Caution** Only the current active project is subject to this export.

**Remark** The following placeholders are supported in the [Export file name for functions]/[Export file name for variables] property.

- %ProjectName%: Replaces itself with the project name.
- %ActiveProjectName%: Replaces itself with the active project name.

**(2) Import the information file**

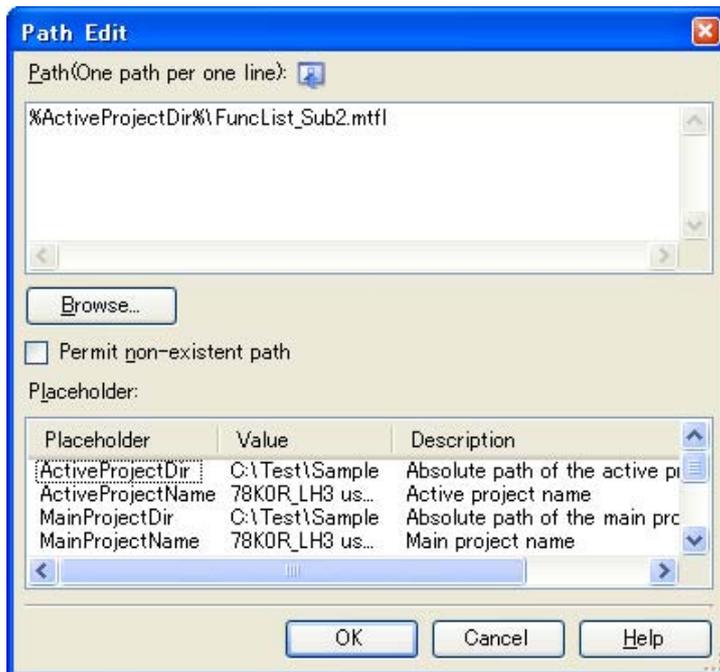
Import the information file you generated (exported).

To do this, set the [Import files] property in the [Import / Export] category on the [\[Settings\] tab](#) of the [Property panel](#). When the [Import files] property is selected, a [...] button appears. When this button is clicked the following [Path Edit dialog box](#) opens.

Figure 2-22. [Import files] Property in [Import / Export] Category



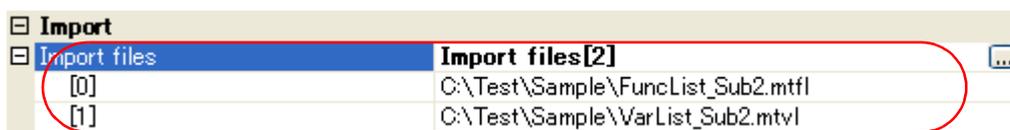
Figure 2-23. Specification of Import Files (Path Edit Dialog Box)



In the [Path Edit dialog box](#), in the [Path(One path per one line)] area, specify the names of the information files generated in (1) (including the path), one path per line (up to 259 characters per line/up to 64 lines). If you specify a relative path, specify the project folder as the base.

You can also use the [Browse...] button to specify information files. After you have entered the names of all the files to import, click the [OK] button. The specified paths appear as sub-properties of the [Import files] property.

Figure 2-24. [Import files] Property (After Adding Import Files)



The above setting completes the import of the information file.

- Remarks 1.** The following placeholders are supported in the [Path Edit dialog box](#).
- %ProjectName%: Replaces itself with the project name.
  - %MicomToolPath%: Replaces itself with the absolute path of the CubeSuite+ install folder.
- 2.** The information files that have been exported by CubeSuite (CubeSuite function list file (\*.csfl)/ CubeSuite variable list file (\*.csvl)) can also be imported.
- 3.** If a function/variable with the same name exists in a import file and an active project, the determination is made according to the following rules.
- **When C source files are used**
    - If the file names are different, and "static" (only lower case) is included in the attribute in the import file, then it is interpreted as a static function/static variable, and loaded as different function information/variable information.
    - If the file names are the same and the function names are different, and "static" (only lower case) is included in the attribute in the import file, then it is interpreted as a static variable in a function, and loaded as different variable information.

- Functions/variables other than the above are interpreted as a global function/global variable or a static function/static variable with the same name in a file, and merged as identical function information/variable information.
- **When C++ source files [CC-RX]**
  - [Class Name], [Namespace], [File Name], and [Arguments] are compared. If any one of these does not match, then it is loaded as different function information.
  - [Class Name], [Namespace], [Function Name], [File Name], and [Arguments] are compared. If any one of these does not match, then it is loaded as different variable information.
  - Functions/variables other than the above are merged as identical function information/variable information.

The rules for determining merging as identical function information/variable information are as follows:

Items		Rules
Function	Variable	
[Class Name][ <b>CC-RX</b> ] [Namespace][ <b>CC-RX</b> ] [Function Name] [File Pass] [PM Information][ <b>RH850</b> ] <sup>Note</sup> [PE Information][ <b>V850</b> ] <sup>Note</sup> [Stack Size[Bytes]] [Arguments Count] [Arguments] [Return Type] [Access Specifier][ <b>CC-RX</b> ] [Attribute]	[Class Name][ <b>CC-RX</b> ] [Namespace][ <b>CC-RX</b> ] [File Name] [Function Name] [File Path] [PM Information][ <b>RH850</b> ] <sup>Note</sup> [PE Information][ <b>V850</b> ] <sup>Note</sup> [Access Specifier][ <b>CC-RX</b> ] [Attribute] [Type] [Members]	The precedence is as follows: "Active project value" > "Import file value" Note, however, that if there is function information/variable information from multiple import files to merge, then the function information/variable information of the last file to be imported is loaded. If a value does not exist in either the active project or import file, then it is left blank ("-").
[Code Size[Bytes]]	[Size[Bytes]]	The precedence is as follows: "Import file value" > "Active project value" Note, however, that if there is function information/variable information from multiple import files to merge, then the function information/variable information of the last file to be imported is loaded. If a value does not exist in either the active project or import file, then it is left blank ("-").

**Note** This item appears only when the selected microcontroller version supports multi-core.

**(3) Cancel the file import**

To cancel the file import, delete the import files on the [Path Edit dialog box](#).

### 2.13 Display Analysis Information in Chart

You can display the acquired function information and variable information (including the data information for Smart Analog<sup>Note</sup>) in a chart.

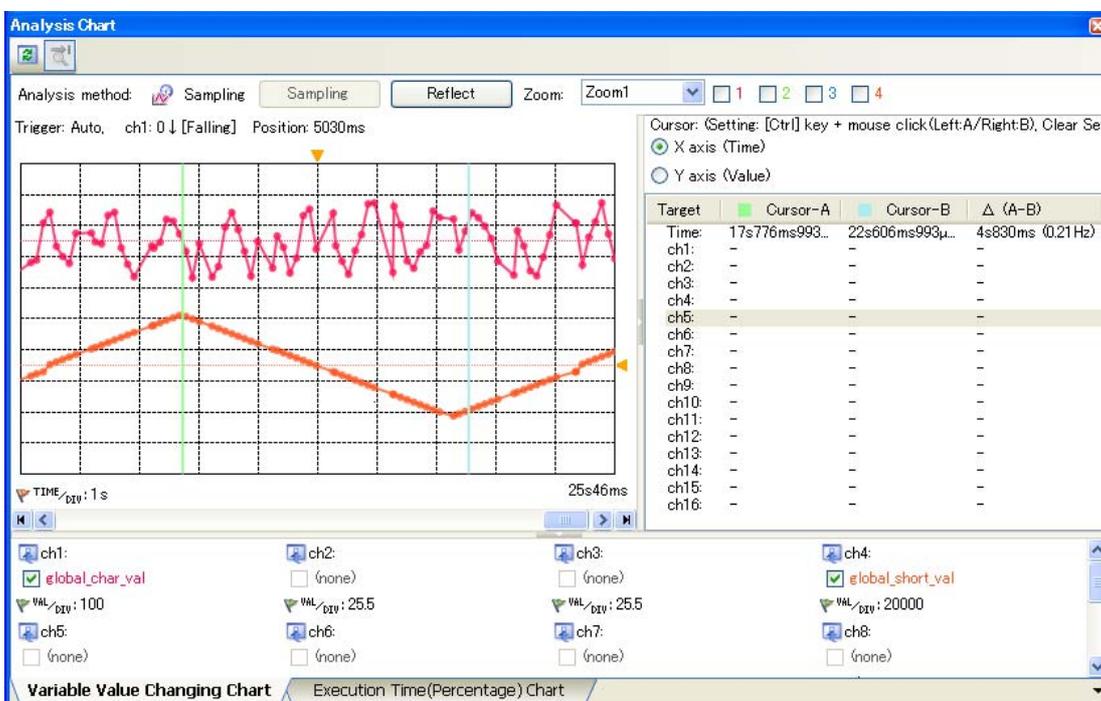
To display a chart, click the  button on the Main window's toolbar to open the Analysis Chart panel. The following content is graphed in the Analysis Chart panel.

- Graph transitions in values
- Chart the function execution time ratios

**Note [E1/E20[RL78]]**

This feature is supported only when the selected microcontroller incorporates a Smart Analog IC.

**Figure 2-25. Example of Graphs for Analysis Information (Analysis Chart Panel)**



#### 2.13.1 Graph transitions in values

Display the relationship between registered variable/register/address/other values and time by a graph (line chart).

Furthermore, when the selected microcontroller incorporates a Smart Analog IC [E1/E20[RL78]], by enabling the Data collection mode<sup>Note</sup> on the debug tool, it is possible to display the collected data for Smart Analog by a graph.

Graphs are displayed in the Analysis Chart panel's [Variable Value Changing Chart] tab.

See the [Variable Value Changing Chart] tab for details on each of the area displayed.

**Note** Data collection mode

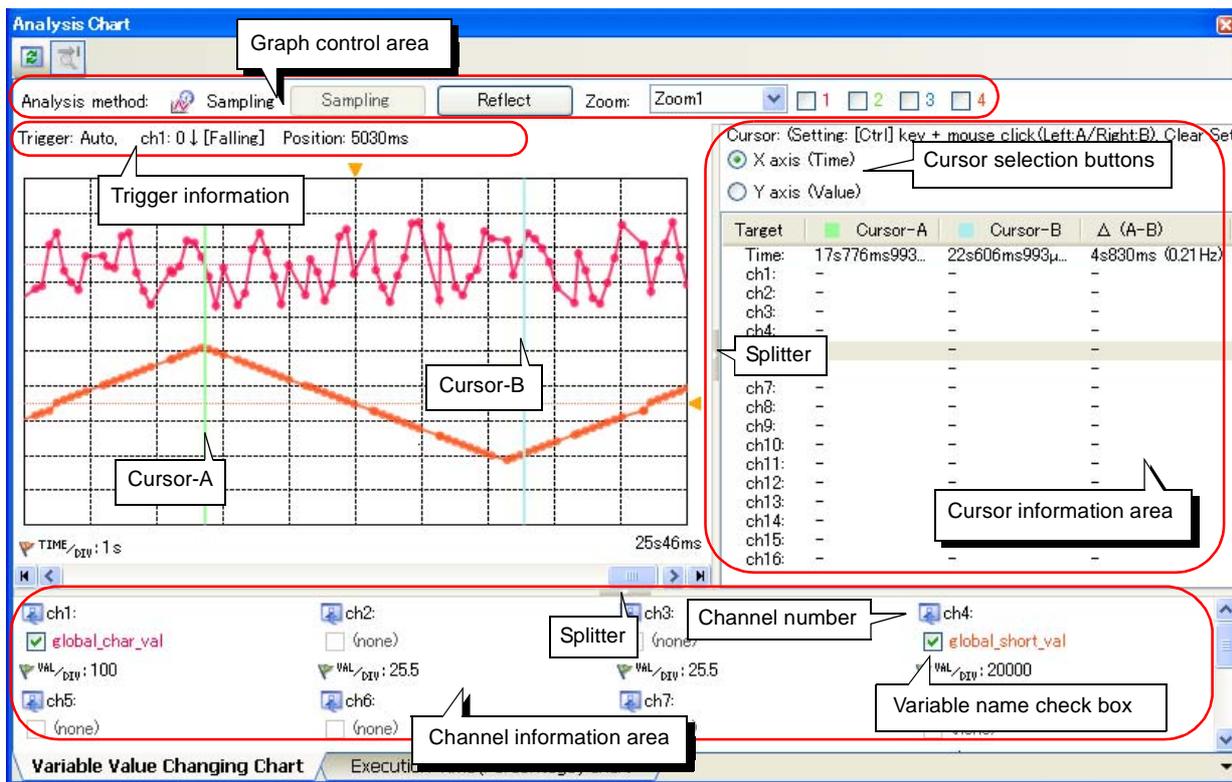
This mode is the debug tool's feature that is supported only when the selected microcontroller incorporates a Smart Analog IC [E1/E20[RL78]].

When the debug tool is in this mode, the data for Smart Analog can be collected while a program is executed.

This mode becomes valid by performing the following setting on the debug tool's Property panel:

[Debug Tool Settings] tab >> [Smart Analog] category >> [Collect data during the execution] property >> [Yes]

Figure 2-26. Graph of Transition in Variable Values



- Cautions 1.** A graph is displayed based on trace data or real-time RAM monitoring results acquired by the debug tool. To display a graph, see cautions of "(a) Trace function" or "(b) RRM function/RAM monitor (pseudo RRM) function".
2. When Real-time sampling analysis is specified as the method for acquiring the graph data, see "3.3 Real-time Sampling Analysis".
  3. When Trace data analysis is specified as the method for acquiring the graph data, see "3.4 Trace Data Analysis".
  4. [E1/E20[RL78]]  
To graph the collected data for Smart Analog, a monitor program for collecting data must be linked to the execution program.

**Remark** You can change the display range of the cursor information area and channel information area by moving the splitter with the mouse. In addition, these areas can be shown/hidden by clicking the mark on the center of their splitters.

The procedure for displaying the chart is as follows:

**(1) Register graphing targets**

Register the item to graph.

- (a) When graphing variable/register/address/other values
- (b) When graphing the data collected for Smart Analog [E1/E20[RL78]]

**Caution** Graphing targets cannot be registered in the condition below:  
- During execution of a program

- When the [Analysis method] property in the [General] category on the [Variable Value Changing] tab of the Property panel is set to [Load from file] (i.e. when graphs are restored from the analysis chart data file (\*.mtac))

(a) When graphing variable/register/address/other values

The following types of items can be registered.

- Global variable
- Static variable inside a file
- Static variable inside a function
- Class variable (only when C++ source files are subject to analysis)
- CPU register
- IOR [RH850][RX][V850]
- SFR [R8C(Localised support)][RL78][78K0R][78K0]
- Address

There are three ways to register graphing targets, shown below.

When the registration is completed, the name of the registered graphing target will be displayed at the Variable name check box corresponding to the channel number in the channel information area.

- <1> Registering targets separately (on the Property panel)
- <2> Registering targets separately (from the other panel)
- <3> Reflecting the Watch panel (auto registration)

**Caution** You can register graphing targets for up to 16 channels, one target per channel.

**Note**, however, that there may be restrictions on the number and size of targets that can be graphed, depending on the debug tool used and the method by which the graph data was acquired (see "(2) Specify the method for acquiring the graph data").

<1> Registering targets separately (on the Property panel)

Enter the name of the target directly from the keyboard into the [Variable/Address 1 - 16] property in the [Channel 1 - 16] category on the [Variable Value Changing] tab of the Property panel.

Figure 2-27. [Channel 1 - 16] Category

Channel 1	
Variable/Address 1	g_count_1 ms
Type/Size 1	Auto
Value per grid[Val/Div] 1	6553.5
Offset 1	0
Color 1	192, 255, 10, 79

In addition, the following detailed conditions can be specified with the [Channel 1 - 16] category for each target (see "(4) Display graphs").

- [Type/Size 1 - 16]: Type and size
- [Value per grid[Val/Div] 1 - 16]: Value per grid unit in graph
- [Offset 1 - 16]: Offset value of graph
- [Color 1 - 16] property: Rendering color of graph

**Remarks 1.** To delete the registered graphing target, delete the value of the above [Variable/Address 1 - 16] property.

2. The registered graphing target name is displayed at the [Variable name check box](#) on the [\[Variable Value Changing Chart\] tab](#).

### <2> Registering targets separately (from the other panel)

From the following panel, drag and drop the target directly onto the [Channel number](#) or [Variable name check box](#) in this tab.

- [Variable List panel](#)<sup>Note</sup>
- Editor panel
- CPU register panel
- IOR panel [\[RH850\]\[RX\]\[V850\]](#)
- SFR panel [\[R8C\(Localised support\)\]\[RL78\]\[78K0R\]\[78K0\]](#)
- Watch panel

**Note** The name of the graphing target displayed differs depending on the type of the variable, as follows:

- Global variable: `"variable name"`
- Static variable inside a file: `"file name#variable name"`
- Static variable inside a function: `"file name#function name#variable name"`
- Class variable: `"class name::variable name"`

- Remarks 1.** Other than the above, on the [Variable List panel](#) or the Editor panel, you can also register the selected variable as a graphing target, by selecting [\[Register to Analysis Chart\]](#) from the context menu.
- 2.** To delete the registered graphing target, delete the value of the [\[Variable/Address 1 - 16\]](#) property in the [\[Channel 1 - 16\]](#) category on the [\[Variable Value Changing\] tab](#) of the [Property panel](#).

### <3> Reflecting the Watch panel (auto registration)

By clicking the [\[Reflect\]](#) button on the graph control area, you can automatically register watch-expressions currently being registered in the Watch panel (Watch1) as graphing targets.

Up to 16 watch-expressions in the Watch panel (Watch1), from top to bottom can be registered. If fewer than 16 watch-expressions have been registered, only the registered number are registered.

**Caution** **Clicking the [\[Reflect\]](#) button will cause the information of graphing targets being registered to be deleted (the graph being displayed will be erased).**

- Remarks 1.** Categories themselves are not registered, but variables and other elements within categories are.
- 2.** If a variable has child nodes (e.g. arrays and structures), expanded variables and the like, such as structure members and array indices, are not registered.
- 3.** If a name matches a graphing target that has already registered, then it will be registered twice.

### (b) When graphing the data collected for Smart Analog [\[E1/E20\]\[RL78\]](#)

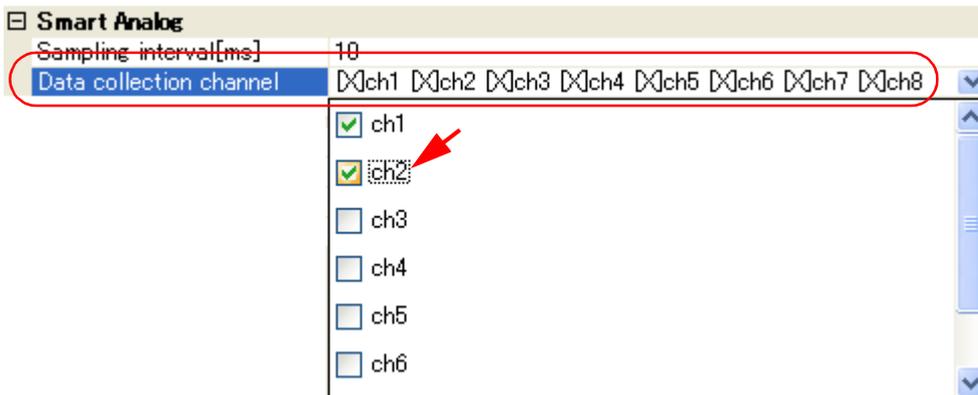
This feature is enabled only when the selected microcontroller incorporates a Smart Analog IC.

First, set the debug tool used to the [Data collection mode](#).

Next, specify the channel numbers in which the data is received from a device with the [\[Data collection channel\]](#) property in the [\[Smart Analog\]](#) category on the [\[Variable Value Changing\] tab](#) of the [Property panel](#). In the drop-down list, select the check box of the channel number to be used (up to 8 channels can be selected).

**Caution** The [Smart Analog] category appears only when the debug tool is set in the [Data collection mode](#).

Figure 2-28. [Smart Analog] Category



In addition, the following detailed conditions can be specified with the [Channel 1 - 16] category on the [\[Variable Value Changing\]](#) tab of the [Property panel](#) for each channel (see “(4) Display graphs”).

- [Variable/Address 1 - 16]: Character strings that are displayed at the [Variable name check box](#)
- [Value per grid[Val/Div] 1 - 16]: Value per grid unit in graph
- [Offset 1 - 16]: Offset value of graph
- [Color 1 - 16] property: Rendering color of graph

Figure 2-29. [Channel 1 - 16] Category [E1/E20[RL78]]

<b>Channel 1</b>	
Variable/Address 1	AN_01
Value per grid[Val/Div] 1	25.5
Offset 1	-2048
Color 1	 192, 255, 10, 79

**(2) Specify the method for acquiring the graph data**

There are three ways to acquire the data for graphing, as shown below.

Select the method in the [Analysis method] property in the [General] category on the [\[Variable Value Changing\]](#) tab of the [Property panel](#), in accordance with the purpose.

- Cautions 1.** The method for acquiring the graph data cannot be changed during execution of a program.
- 2.** If graph data has already been acquired, then changing this setting will cause the stored graph data to be deleted (the graph being displayed will be erased).

Figure 2-30. [Analysis method] Property

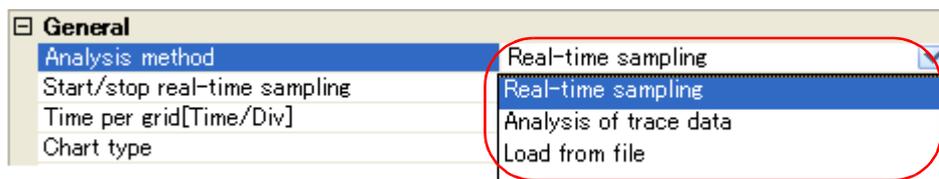


Table 2-10. Method for Acquiring Graph Data

Method for Acquiring Graph Data	Description
Real-time sampling analysis	<p>Select [Real-time sampling] in this property.</p> <p>This method displays graphs based on the data acquired via the <a href="#">RRM function/RAM monitor (pseudo RRM) function</a> of the debug tool (default).</p> <p><b>[E1/E20][RL78]</b></p> <p>If the debug tool is set in the <a href="#">Data collection mode</a>, then the data will be collected using a sampling method<sup>Note</sup> for Smart Analog. Therefore, select this item <a href="#">When graphing the data collected for Smart Analog [E1/E20][RL78]</a>.</p>
Trace data analysis	<p>Select [Analysis of trace data] in this property.</p> <p>This method displays graphs based on the trace data acquired via the <a href="#">Trace function</a> of the debug tool.</p> <p>This item does not appear when any one of the following:</p> <ul style="list-style-type: none"> <li>- The debug tool does not support the trace function.</li> <li>- The debug tool does not support the trace time tag feature.</li> <li>- The debug tool is set in the <a href="#">Data collection mode</a>. <b>[E1/E20][RL78]</b></li> </ul>
Load from file	<p>Select [Load from file] in this property.</p> <p>This method restores graphs by loading the analysis chart data file (*.mtac) that have been saved (see “(6) <a href="#">Save the graph data to restore the graph</a>”).</p>

**Note** The following property setting on the [Property panel](#) is required.  
 - [\[Variable Value Changing\] tab](#) >> [Smart Analog] category >> [Sampling interval[ms]] property

The differences depending on the method for acquiring graph data are as follows:

Table 2-11. Differences Depending on Method for Acquiring Graph Data

Differences	Real-time Sampling Analysis	Trace Data Analysis
Graph display	<p>Possible when the debug tool is in any of the following states:</p> <ul style="list-style-type: none"> <li>- <a href="#">RRM function/RAM monitor (pseudo RRM) function</a> is enabled</li> <li>- <a href="#">Data collection mode</a> <b>[E1/E20][RL78]</b></li> </ul>	<p>Possible when <a href="#">Trace function</a> of the debug tool is enabled</p>
Chart type	<p>One of the following<sup>Note 1</sup></p> <ul style="list-style-type: none"> <li>- Step-plot line chart</li> <li>- Normal line chart</li> </ul>	<p>Step-plot line chart (fixed)</p>
Registration of graphing targets	<p>Before execution of a program</p>	<p>Not dependent on the execution timing because analysis is based on trace data</p>
Number of targets that can be graphed	<p>Dependent on the size of the target area of RRM function<sup>Note 2</sup>.</p>	<p>If Point Trace events are used:                      Dependent on the maximum number of enabled events<sup>Note 3</sup> in the debug tool</p>
Size of targets that can be graphed	<ul style="list-style-type: none"> <li>- Less than or equal to 4 bytes</li> </ul>	<ul style="list-style-type: none"> <li>- Less than or equal to 4 bytes <b>[CC-RH][RX][V850]</b></li> <li>- Less than or equal to 2 bytes <b>[RL78][78K0R][78K0]</b></li> </ul>
Time display range	<p>The execution time, from the start to the end of program execution (Run-Break time)</p>	<p>Time registered as trace data</p>

Differences	Real-time Sampling Analysis	Trace Data Analysis
Time display format	XXXsXXXms	XXXsXXXmsXXXμsXXXns
Trigger function	Available (see "(3) Use the trigger function")	None
Real-time update	Possible	Impossible
Transition points of value	Not possible to identify accurate time/change points, because they depend on the specified sampling interval <sup>Note 4</sup> .	Matches actual timing (can be checked from <a href="#">Pop-up display</a> )
Target core [RH850]	See "(b) RRM function/RAM monitor (pseudo RRM) function".	See "(a) Trace function".
Note	See "3.3 Real-time Sampling Analysis".	See "3.4 Trace Data Analysis".

**Notes 1.** This depends on the following setting on the [Property panel](#):

- [\[Variable Value Changing\] tab](#) >> [General] category >> [Chart type] property
- 2. [IECUBE[V850]][MINICUBE2[78K0]][E20(JTAG)[RX600 Series]][EZ Emulator[78K0]]**  
Some restrictions apply to the size and the number of the target area for the RRM function. Therefore, CubeSuite+ automatically determines the target area within the restrictions. For details on "the target area for the RRM function", see "CubeSuite+ Integrated Development Environment User's Manual: Debug" of the microcontroller used.
- 3.** The number of events that can be set to "valid state" simultaneously differs depending on the microcontroller and the debug tool. For details on "Maximum number of enabled events", see "CubeSuite+ Integrated Development Environment User's Manual: Debug" of the microcontroller used.
- 4.** When RRM function/RAM monitor (pseudo RRM) function is used:
  - Debug tool's Property panel >> [Debug Tool Settings] tab >> [Access Memory While Running] category >> [Display update interval[ms]] property
 When the debug tool is in the [Data collection mode \[E1/E20\[RL78\]\]](#):
  - [Property panel](#) >> [\[Variable Value Changing\] tab](#) >> [Smart Analog] category >> [Sampling interval[ms]] property

**(3) Use the trigger function**

If you specify [Real-time sampling analysis](#), you can control the timing with which the graph is displayed in response to the specified trigger. You can use this trigger function to display the graph with the appearance of an oscilloscope.

The trigger function can be configured in the [Trigger] category on the [\[Variable Value Changing\] tab](#) of the [Property panel](#).

Your configuration contents will be listed on this tab as the trigger information.

**Caution** The properties in the [Trigger] category cannot be changed during execution of a program.

**Figure 2-31. Setting of Trigger Function ([Trigger] Category)**

Trigger	
Use trigger function	Yes
Trigger mode	Auto
Trigger source	ch1
Trigger level	0
Direction of trigger edge	Rising
Trigger position	0s
Trigger-mark color	 Orange

**(a) [Use trigger function]**

Select whether to use the trigger function.

Select [Yes] to use the trigger function (default: [No]).

**(b) [Trigger mode]**

Select the trigger mode (the timing at which graphs are updated).

Auto	<p>Updates the graph display cyclically in real time (default).</p> <p>When a trigger signal is received, the data immediately prior to the trigger signal is graphed to the left of the trigger position, and the data immediately after the trigger signal is graphed to the right of the trigger position.</p> <p>When the graph reaches the right edge of the graph area, the graph resumes scrolling to the left, and the graph display is updated.</p>
Single	<p>Only updates the graph display upon the first trigger signal from the start of sampling.</p> <p>When a trigger signal is received, the data immediately prior to the trigger signal is graphed to the left of the trigger position, and the data immediately after the trigger signal is graphed to the right of the trigger position.</p> <p>When the graph reaches the right edge of the graph area, the graph display stops updating and sampling<sup>Note</sup>.</p>
Normal	<p>Only updates the graph display upon trigger signals.</p> <p>When a trigger signal is received, the data immediately prior to the trigger signal is graphed to the left of the trigger position, and the data immediately after the trigger signal is graphed to the right of the trigger position.</p> <p>When the graph reaches the right edge of the graph area, the graph display stops updating. Because sampling does not stop, the graph display is updated upon the next trigger signal.</p>

**Note** The sampling is stopped regardless of whether [Sync] or [Manual] is selected in the [Start/stop real-time sampling] property in the [General] category on the [\[Variable Value Changing\]](#) tab of the [Property panel](#).

**Remark** Triggers are suspended while the graph is being updated in response to a trigger signal. A trigger signal received during this time will not cause the graph display to be updated. In [Auto] and [Single] modes, trigger signals are ignored while triggers are suspended. In [Normal] mode, the graph is updated in response to the latest trigger signal when trigger suspension is lifted.

**(c) [Trigger source]**

Select the variable (channel) that is the source of trigger signal.

Select a channel between ch1 and ch16 (default: [ch1]).

**(d) [Trigger level]**

Specify the threshold value for determining as a trigger signal.

The variable (channel) specified for [\[Trigger source\]](#) determines whether a trigger signal is recognized, according to whether the value exceeds the threshold set here.

Directly enter a value with a decimal or hexadecimal number in the range "minimum trigger source" to "maximum trigger source" (floating-point decimals allowed) (default: [0]).

**Remark** The trigger level is displayed on the right part of graphs as the trigger mark (  ).

The trigger level can be changed by dragging on this trigger mark (note, this operation is disabled during execution of a program).

**(e) [Direction of trigger edge]**

Select the direction of the threshold value specified with [Trigger level].

The variable (channel) specified for [Trigger source] determines whether a trigger signal is recognized, according to whether the value exceeds the threshold in the direction set here.

Rising	Generates a trigger signal when the value of [Trigger source] changes from less than [Trigger level] to greater than or equal to [Trigger level] (default).
Falling	Generates a trigger signal when the value of [Trigger source] changes from greater than [Trigger level] to less than or equal to [Trigger level].
Both	Generates a trigger signal when both "Rising" and "Falling".

**(f) [Trigger position]**

Specify the position on the X axis for rendering the location at which the trigger signal occurred (i.e. position where the data is graphed after a trigger signal occurs).

Directly enter a value with a decimal number in the following range (default: [0s]).

Note that if the unit (s, ms, us, ns: case insensitive) is omitted, it will be handled as "ms".

- 0 to "(the value of the [Time per grid[Time/Div]] property) x 10"

**Remark** The trigger position is displayed on the upper part of graphs as the trigger mark ( ▼ ).

The trigger position can be changed by dragging on this trigger mark (note, this operation is disabled during execution of a program).

**(g) [Trigger mark color]**

Specify the color of the trigger marks ( ◀ / ▼ ) that indicate the trigger level and the trigger position.

Specify with the color selection combo box or directly enter the value in decimal number/hexadecimal number (prefix "0x" is needed) or the color name (see "About the specification of the color") (default: [Color Orange]).

**(4) Display graphs**

After you have executed<sup>Note</sup> and stopped the program, the latest graph for the registered graphing target is displayed (default). Note, however, that no graphs will be displayed if the corresponding data could not be acquired.

Graphs are updated when:

**- Real-time sampling analysis**

Each time of the specified sampling interval, even if a program is running.

You can control (toggle) the start/stop of a real-time sampling manually via the [Sampling] button on this tab, by performing the following setting on the [Property panel](#).

- [\[Variable Value Changing\] tab](#) >> [General] category >> [Start/stop real-time sampling] property >> [Manual]

**- Trace data analysis**

Each time program execution is stopped. Note, however, that if the [Refresh at program stop] property in the [General] category on the [\[Settings\] tab](#) of the [Property panel](#) is set to other than [Yes] (default), the information will be updated in accordance with the setting on the [Property panel](#).

**Note [E1/E20[RL78]]**

Click the  button on the debug toolbar to operate the debug tool in the [Data collection mode](#) (all execution-related buttons other than the  button are disabled). For details on Smart Analog function, see "CubeSuite+ Integrated Development Environment User's Manual: Debug".

- Cautions 1.** If the acquired graph data exceeds the buffer capacity (10000 plots), then the oldest graph data is overwritten by the new graph data (ring buffer system).  
In this case, part of the graph rendering will be blank.
- 2.** When [Real-time sampling analysis](#) is selected, if acquiring graph data has failed, lines between transition points are not displayed but only the time information is displayed (see "(a) Graph").
- 3.** [E1/E20[RL78]]  
If you run the program by the debug tool in the [Data collection mode](#), change the [Data collection channel] property in the [Smart Analog] category on the [\[Variable Value Changing\] tab](#) of the [Property panel](#), and run the program again in the data collection mode, the first transition point in the graph may be incorrect.

**Remark** There are limitations to the trace memory area of the debug tool. Consequently, if the graph is displayed via [Trace data analysis](#), then we recommend setting a Point Trace event for the graphing target in the Watch panel, in order to display transitions of values over a wider range.

The display of this graph can be configured as follows:

**(a) Limitation of graphs displayed**

You can select the graph to be displayed.

In the default condition, all graphs of channels where the graphing target is being registered are displayed. To hide a graph, clear the [Variable name check box](#) corresponding to its channel number.

**(b) Selection of the chart type**

When [Real-time sampling analysis](#) is selected, you can select the chart type to be displayed.

To change the chart type, set the [Chart type] property in the [General] category on the [\[Variable Value Changing\] tab](#) of the [Property panel](#) (note, however, that if [Trace data analysis](#) is selected, then the chart type is fixed to [Step line chart]).

Line chart	Creates a graph linking each plot via straight lines (default).
Step line chart	Creates a graph linking each plot via perpendicular lines (step-plot chart).

**Figure 2-32. Chart Type**



**Remark** You can also specify the rendering color of a graph for each channel individually. To make the setting, set the [Color 1-16] property in the [Channel 1-16] category on the same tab.

**(c) Setting the display range**

The graph is displayed by dividing the X and Y axes into a 10 x 10 grid.

**<1> Use the auto adjustment function (default)**

According to the obtained graph data, optimum values are calculated for the time per unit grid (Time/Div) for the X axis and the value per unit grid (Val/Div)/offset value for the Y axis as described below (default) to display a graph.

- X axis (time)

The grids are automatically adjusted so that the specified number of value transition points<sup>Note</sup> on the graph fit inside the rendering area (between the left and right ends).

- Y axis (value)

The grids are automatically adjusted so that the upper and lower ends of the rendering area correspond to the maximum and minimum values of the graph data, respectively.

**Note** In the default condition, 20 points are specified. This value can be changed via the [Number of transition location for auto adjustment] property in the [General] category on the [Variable Value Changing] tab of the Property panel.

**Cautions 1.** The auto adjustment function is disabled when the trigger function is used or the debug tool is set in the Data collection mode.

**2.** The auto adjustment function becomes invalid when the following operation is performed (the [Auto adjustment] property in the [General] category on the [Variable Value Changing] tab of the Property panel will be changed to [None]).

- The [Time/Div] label or [Val/Div] label on the Analysis Chart panel is double-clicked.

- The display range is changed with the mouse operation (see "Set the display range manually").

**Remark** The timing at which the auto adjustment is conducted can be specified with the [Auto adjustment] property in the [General] category on the [Variable Value Changing] tab of the Property panel.

**<2> Set the display range manually**

First, specify [None] with the [Auto adjustment] property in the [General] category on the [Variable Value Changing] tab of the Property panel. Then, specify the following value.

- X axis (time)

You can specify the time per unit grid for all channels. To make the setting, set the [Time per grid[Time/Div]] property, in the [General] category on the [Variable Value Changing] tab of the Property panel.

- Y axis (value)

You can specify the value per unit grid or the offset value for each channel individually. To make the setting, set the [Value per grid[Val/Div] 1 - 16] property or the [Offset 1-16] property, in the [Channel 1 - 16] category on the [Variable Value Changing] tab of the Property panel.

You can also change the values above by any one of the following mouse operations.

Note, however, that these operations are disabled during execution of a program.

- **The time per unit grid (Time/Div)**

- Double-clicking on the [Time/Div] label in the Analysis Chart panel has the same effect as the automatic adjustment function on the X axis.

- In the graph area, move the mouse wheel forward or backward while holding down the [Ctrl] key.

- **The value per unit grid (Val/Div) or the offset value**

- Double-clicking on the [Val/Div] label in the Analysis Chart panel has the same effect as the automatic adjustment function on the Y axis.

- Select a graph (the mark of transition points transforms into ) , and then move the mouse wheel forward or backward while holding down the [Ctrl] key.
- **Offset value**
- Click and vertically drag the mouse while pressing the [Shift] key with the mouse cursor placed over any of the transition points in the graph (the mouse cursor transforms into the  icon). Pressing the [Esc] key during dragging the mouse cancels changing the offset value.

**(5) Validate the graph data**

You can perform the following operations on the displayed graph as needed.

**Caution** These operations below are disabled during execution of a program.

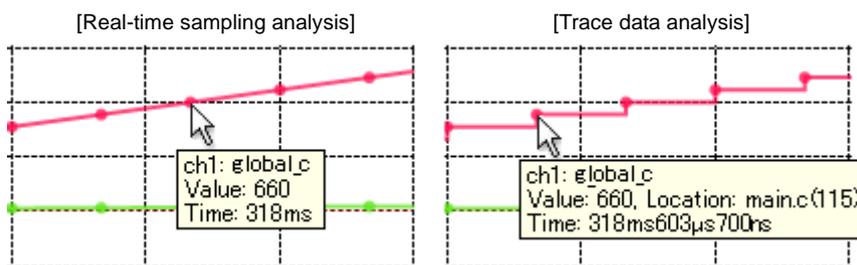
**(a) Pop-up display at a transition point**

Check the information about transition points in the graph.

When the mouse cursor is hovered over a transition point on the graph, information about that location appears in a pop-up.

Note, however, that the content displayed will differ depending on the method for acquiring the graph data (see "(i) Pop-up display").

**Figure 2-33. Pop-up Display at Transition Point**



**Remark** When the [Location] information is shown in a pop-up, you can display the applicable location in the Editor panel by double-clicking the transition point (except for a graph that was restored by loading the analysis chart data file (\*.mtac)).

**(b) Cursor measurement**

You can check the time and value at the position of Cursor-A/Cursor-B on the graph.

You can get a cursor measurement for the X axis (time) or Y axis (value).

Measurements appear in a list in the cursor information area on this tab (see "(4) Cursor information area").

To perform this measurement, first use the cursor selection button ([X axis (Time)] button/[Y axis (Value)] button) to select the target axis to measure. Then, set Cursor-A and Cursor-B to the desired location on the graph by using the following operations (the cursors are hidden by default).

Cursor	Setting (Show)	Clear Setting (Hide)
Cursor-A	[Ctrl] key + click	[Ctrl] key + double-click
Cursor-B	[Ctrl] key + right-click	[Ctrl] key + right-double-click

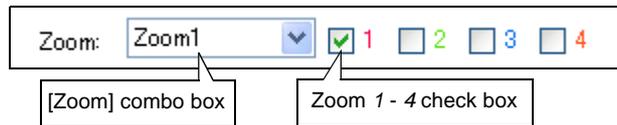
**Remark** You can change the color of each cursor via the [Cursor-A color]/[Cursor-B color] property in the [General] category on the [Variable Value Changing] tab of the Property panel.

(c) Zoom

You can zoom in on any location on the graph.

In the graph control area, select the zoom 1 - 4 check boxes (multiple selections allowed) to open the [Variable Value Changing Chart \(zoomed in\)](#) panel for the checked number(s), and view the specified range in a zoomed display (you can open up to four [Variable Value Changing Chart \(zoomed in\)](#) panels).

Figure 2-34. Zoom



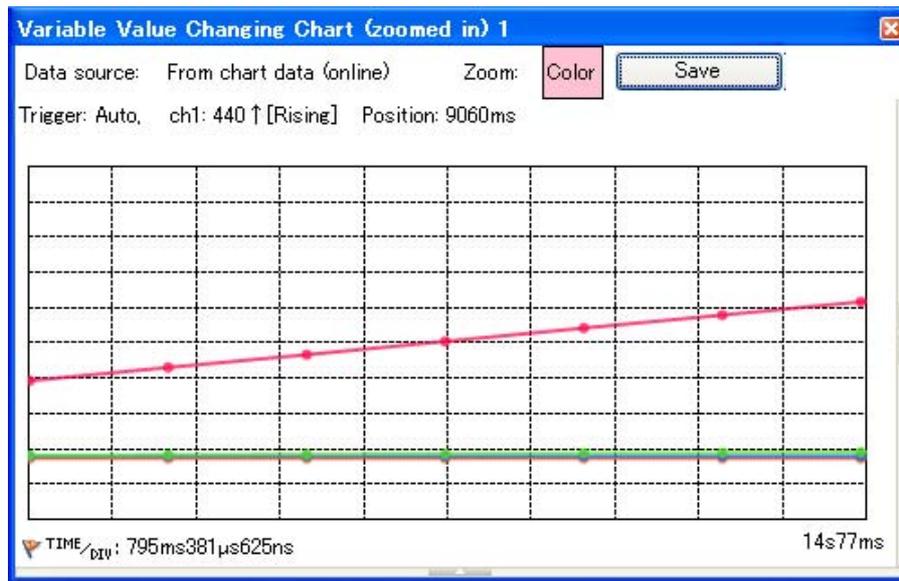
To set the zoom range, select the number of the [Variable Value Changing Chart \(zoomed in\)](#) panel in the [Zoom] combo box, then click and drag with the mouse to select the zoom area. During this time, you can reset the zoom range by using the same operation, as long as [Zoom frame](#) appears indicating the drag region displayed on the graph.

To cancel the zoom range setting, press the [ESC] key during dragging.

To clear the set range, double-click somewhere in the graph area (a transition point is not included).

See the [Variable Value Changing Chart \(zoomed in\)](#) panel for details on zoomed views.

Figure 2-35. Zoom (Variable Value Changing Chart (zoom in) Panel)



- Remarks 1. You can set separate zoom ranges for each of four [Variable Value Changing Chart \(zoomed in\)](#) panel views.
- 2. You can change the color of each zoom frame via the [Zoom-1 - 4 color] property in the [General] category on the [\[Variable Value Changing\]](#) tab of the [Property panel](#).

(6) Save the graph data to restore the graph

You can restore the graph currently being displayed by loading the analysis chart data file (\*.mtac) which saves the data of the graph.

The procedure for restoring the graph is as follows:

**(a) Save the graph data**

- For the graph on the [\[Variable Value Changing Chart\] tab](#)

When the graph to be saved are being displayed on this tab, select [Save Analysis Chart Data As...] from the [File] menu to open the [Save As dialog box](#).

On this dialog box, select "Analysis Chart Data (\*.mtac)" in the [Save as type] area and specify the name of new file in the [File name] area, and then click the [Save] button (note that the file extension must be ".mtac").

- For the graph on the [Variable Value Changing Chart \(zoomed in\) panel](#)

When the graph to be saved are being displayed on this panel, select [Save] button to open the [Save As dialog box](#).

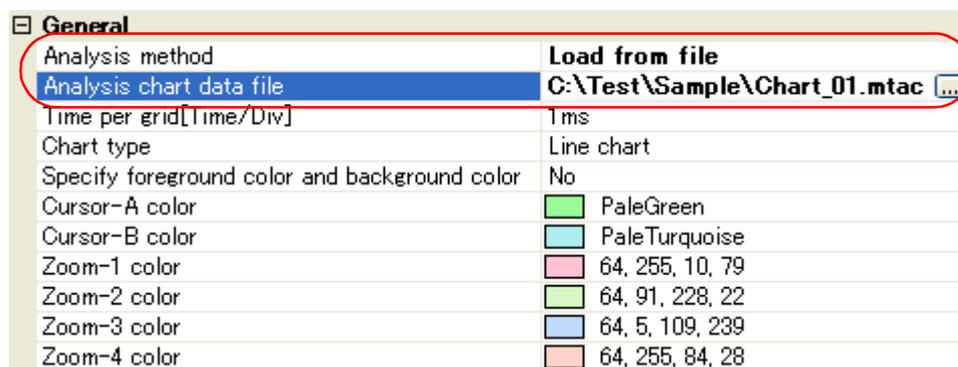
On this dialog box, select "Analysis Chart Data (\*.mtac)" in the [Save as type] area and specify the name of new file in the [File name] area, and then click the [Save] button (note that the file extension must be ".mtac"). However, the graph data saved on the [Variable Value Changing Chart \(zoomed in\) panel](#) is limited to the zoom range.

**(b) Load the graph data**

On the [\[Variable Value Changing\] tab](#) of the [Property panel](#), select [Load from file] in the [Analysis method] property in the [General] category (see "(2) [Specify the method for acquiring the graph data](#)").

Then, specify the analysis chart data file (\*.mtac) that was saved previously with the [Analysis chart data file] property in the same category.

**Figure 2-36. Load Graph Data ([General] Category)**



At this time, data items to be saved/restored are as follows:

**[Graph data for each channel]**

Data Item	Restored Location
Value	- The graph display area
Time	- The channel information area
Information about display or non-display	

**[Values of properties on the [Variable Value Changing] tab in the Property panel]**

Data Item	Restored Location
Time per unit grid	[General] category >> [Time per grid[Time/Div]] property

Data Item	Restored Location
The name of variable registered in each channel	[Channel 1 - 16] category >> [Variable/Address 1 - 16] property
Type/size for each channel	[Channel 1 - 16] category >> [Type/Size 1 - 16] property
Value per unit grid for each channel	[Channel 1 - 16] category >> [Value per grid[Val/Div] 1 - 16] property
Offset value for each channel	[Channel 1 - 16] category >> [Offset 1 - 16] property
Sampling interval [E1/E20[RL78]]	[Smart Analog] category >> [Sampling interval[ms]] property Note that saving is possible only when the selected microcontroller incorporates a Smart Analog IC and the debug tool has collected graph data in the <a href="#">Data collection mode</a> .

**Caution** Channels without graph data are not saved, regardless of whether the graph is visible. In this case, the default values of the properties for that channel are applied.

**Remark** See "2.14 Save Analysis Information" for details about saving graph data for purposes other than restoring graphs.

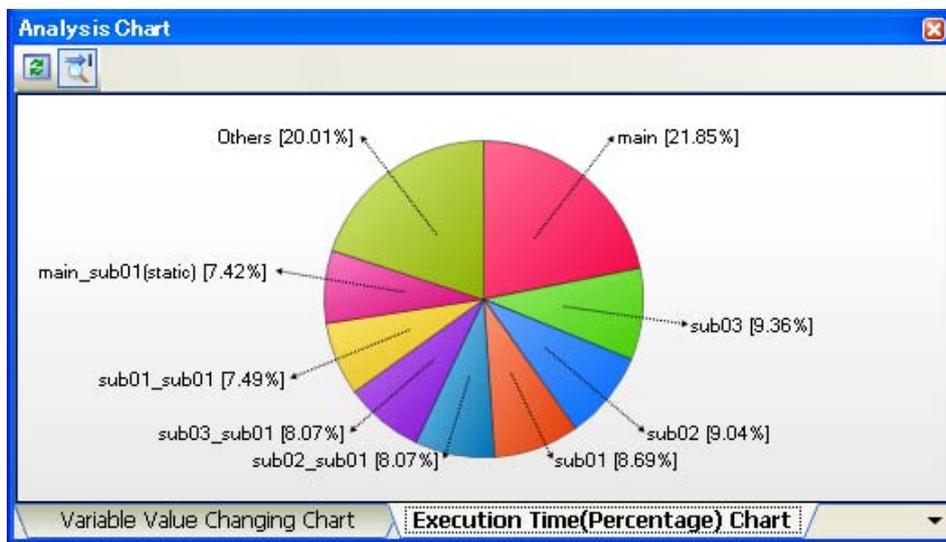
**2.13.2 Chart the function execution time ratios**

Display a pie chart of the function execution time ratios.

The chart is displayed in the [Analysis Chart](#) panel's [\[Execution Time\(Percentage\) Chart\]](#) tab, based on the currently acquired [Dynamic analysis information](#) (equivalent to [Execution Time(Percentage)[%]] in the [Function List](#) panel).

See the [\[Execution Time\(Percentage\) Chart\]](#) tab for details on each of the area displayed.

**Figure 2-37. Chart of Function Execution Time Ratios**



- Cautions 1.** When the debug tool to use does not support the [Trace function](#), or when the debug tool's trace function is not enabled, this chart cannot be displayed. Even if the trace function is enabled-condition, if there is no trace data in the trace memory, no chart is displayed on this tab, and then the following message will be displayed on the [Output panel](#): "There is no execution time information."
- To display this chart, see cautions of "(a) [Trace function](#)".
  - [IECUBE[78K0]][E1/E20[RX]][EZ Emulator[RX]]  
Because the trace time tag function is not supported, this chart cannot be displayed.

The following operations can be performed on the call graph displayed.

**(1) Specify the number of functions to display**

You can change the number of functions displayed in the chart.

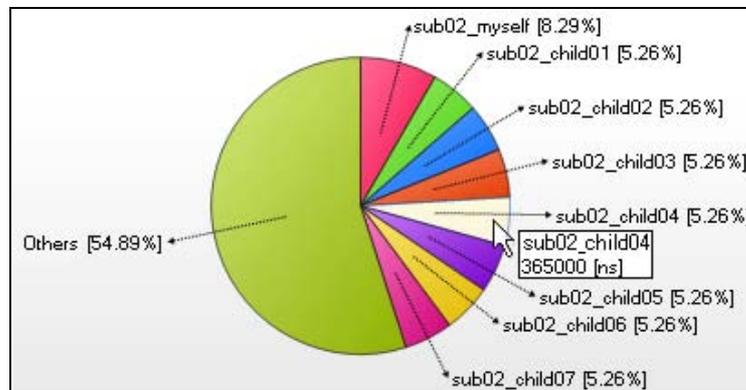
To do this, specify the number in the [The number of functions displaying in the Execution Time(Percentage) Chart] property in the [Analysis Chart] category on the [Settings] tab of the Property panel (set to [10] by default).

The targets are graphed in ranking order of highest proportion of execution time taken. Functions exceeding the number specified here are displayed together under "Others".

**(2) Pop-up display of execution time**

When the mouse cursor is hovered over a function, information about that function's execution time appears in a pop-up.

**Figure 2-38. Pop-up Display of Execution Time**



**Remark** The display contents are updated each time program execution is stopped.

Note, however, that if the [Refresh at program stop] property in the [General] category on the [Settings] tab of the Property panel is set to other than [Yes] (default), the information will be updated in accordance with the setting on the Property panel.

## 2.14 Save Analysis Information

You can save the contents of the [Function List panel](#), [Variable List panel](#), [Analysis Chart panel](#), [Call Graph panel](#), and [Variable Value Changing Chart \(zoomed in\) panel](#) to a file.

### (1) Save function information

To do this, with the focus on the [Function List panel](#), select [Save Function List Data As...] from the [File] menu. The [Save As dialog box](#) opens; perform the operation from this dialog.

The following file formats can be specified when saving the information.

Text file (*.txt)	Text format
CSV(Comma-Separated Variables) (*.csv)	CSV format
Microsoft Office Excel Workbook (*.xls)	Microsoft Office Excel book format
Function List file (*.mtfl)	File format to import the function information (see " <a href="#">2.12 Import or Export Information File</a> ".)

**Caution** Only the items/analysis information currently displayed in the panel can be saved.

### (2) Save variable information

To do this, with the focus on the [Variable List panel](#), select [Save Variable List Data As...] from the [File] menu. The [Save As dialog box](#) opens; perform the operation from this dialog.

The following file formats can be specified when saving the information.

Text file (*.txt)	Text format
CSV(Comma-Separated Variables) (*.csv)	CSV format
Microsoft Office Excel Workbook (*.xls)	Microsoft Office Excel book format
Variable List file (*.mtvl)	File format to import the variable information (see " <a href="#">2.12 Import or Export Information File</a> ".)

**Caution** Only the items/analysis information currently displayed in the panel can be saved.

### (3) Save graph information

To do this, with the focus on the [Analysis Chart panel](#), select [Save Analysis Chart Data As...] from the [File] menu, or with the focus on the [Variable Value Changing Chart \(zoomed in\) panel](#), click the [Save] button. The [Save As dialog box](#) opens; perform the operation from this dialog.

The following file formats can be specified when saving the information.

Text file (*.txt)	Text format
CSV(Comma-Separated Variables) (*.csv)	CSV format
Microsoft Office Excel Workbook (*.xls)	Microsoft Office Excel book format
Analysis Chart Data (*.mtac) <sup>Note 1</sup>	Analysis chart data file
Bitmap (*.bmp)	Bitmap format (32-bit) (graphic file format)
JPEG (*.jpg)	JPEG format (graphic file format)
PNG (*.png)	PNG format (graphic file format)
EMF(*.emf) <sup>Note 2</sup>	EMF format (Graphic file format)

- Notes**
1. This format is only applied to the [\[Variable Value Changing Chart\] tab](#) of the [Analysis Chart panel](#).
  2. Only the area of graph rendering is saved (trigger marks, channel information, etc. cannot be saved).

- Cautions**
1. For the [Analysis Chart panel](#), only the contents of the currently displayed tab will be saved.
  2. When the graphic file format is selected, only the part currently being displayed in the panel will be saved.

#### (4) Save call graph information

To do this, with the focus on the [Call Graph panel](#), select [Save Call Graph Data As...] from the [File] menu. The [Save As dialog box](#) opens; perform the operation from this dialog.

The following file formats can be specified when saving the information.

Note that when "(Only the visible part)" is selected, only the part currently being displayed in the panel will be saved.

Bitmap(Only the visible part) (*.bmp)	Bitmap format (32-bit) (graphic file format)
JPEG(Only the visible part) (*.jpg)	JPEG format (graphic file format)
PNG(Only the visible part) (*.png)	PNG format (graphic file format)
Bitmap (*.bmp)	Bitmap format (32-bit) (graphic file format)
JPEG (*.jpg)	JPEG format (graphic file format)
PNG (*.png)	PNG format (graphic file format)
EMF (*.emf)	EMF format (graphic file format)

**Caution** All part of the call graph may not successfully be saved as a graphic file format if the project is too big.

**Remark** If the zoom function has been applied, then the image will be saved at the current zoom ratio (except for EMF format).

## CHAPTER 3 CAUTIONS

This chapter describes cautions for using the analyze tool.

### 3.1 Changing Active Project

If you change the active project, it is possible that nothing will be shown in the [Function List panel/Variable List panel/Call Graph panel/Class/Member panel](#), because no cross reference file has been generated.

If this happens, rebuild after setting the [Enable static analysis information] property in the [General] category on the [Settings] tab of the [Property panel](#) to [Yes]. The contents of the panel will be updated.

### 3.2 Coverage Results

The results of code coverage and data coverage store the program's execution results. If you repeatedly download, execute, and stop your program, then the results for the repeated actions will be displayed.

To clear the coverage results, select [Clear Coverage Information] from the context menu of the Editor panel/Disassemble panel, and then click the  button on the [Function List panel/Variable List panel](#).

If you change the program and execute a build, the allocation addresses of the functions and variables may differ from those of the previous builds. If this happens, coverage ratio will appear for functions that have not been executed, and for variables that have not been read or written.

### 3.3 Real-time Sampling Analysis

Notes on acquisition of graph data through [Real-time sampling analysis](#) are listed below.

- It may not be possible to identify the sampling interval, depending on the number of graphing targets registered.
- It may not be possible to acquire graph data.  
If acquiring data has failed, lines between transition points are not displayed but only the time information is displayed (see "(a) [Graph](#)").
- When graph data of a 2-, 4-, or 8-byte variable is to be acquired, the process of assigning a value to the variable may be divided into two steps (see "[Example When using the RL78 microcontroller](#)").  
If sampling of the variable takes place between the two steps, an incorrect value may be read out because the assignment is not completed.

#### Example When using the RL78 microcontroller

In this example, if sampling takes place before "(2)" is executed, the value of variable "value\_a" in which only the assignment to the two lower-order bytes has been completed is read out.

#### [C source text]

```
long int    value_a = 0;    // Definition of a 4-byte variable

void func(void)
{
    value_a = 4000000000;    // Assignment to a 4-byte variable
}
```

**[Assembly instructions for the assignment processing above]**

```

MOVW    AX, #2800H
MOVW    !_value_a, AX    ;(1): Assignment to the two lower-order bytes of variable "value_a"
MOVW    AX, #0EE6BH
MOVW    !_value_a+2, AX  ;(2): Assignment to the two upper-order bytes of variable "value_a"

```

**3.4 Trace Data Analysis**

Notes on acquisition of graph data through [Trace data analysis](#) are listed below.

- It is not possible to display changes in values in the following case:
  - In sections where variables have been assigned to registers via compiler optimization (because reads/writes of variables in such sections cannot be acquired as trace data.)
  - When a 2-byte region is written in units of 1 byte, or a 4-byte region is written in units of 1 or 2 bytes
  - When the debug tool or OCD internal trace cannot acquire trace data of the access type (read/write) (because the graph is rendered by analyzing trace data of the access type.)
- If you mix a Point Trace event and a Trace event specified sections, then the instruction line at the end of the Trace event specified section may be applied to subsequent Point Trace event. For this reason, although [Location] in the pop-up display on the graph should be "-", an invalid file name and line number may appear.
- bit-type, Boolean-type, and \_Bool-type variables and structure bit-fields are analyzed in byte units. In cases where bit-type, Boolean-type, and \_Bool-type variables and structure bit-fields are allocated to the same address, even a point where access to a single variable or structure bit-field occurs is recorded as a transition point in the graph. Double-clicking on the transition point will jump to the source line of another variable that is allocated to the same address. If a structure bit-field is assigned to several bytes, some of the information on the byte access is not included in the output trace data. This portion is displayed as a lost section (see "(a) Graph") in the graph since the value of the variable cannot be analyzed.
- **[IECUBE[V850E1]][V850ES]][E20[RX]]**  
If [Stop Trace]/[Start Trace] of the context menu on the Trace panel is selected during a program execution, an invalid chart may be displayed.
- **[E20[RX]]**  
In the debug tool's Property panel, when the [Trace data type] property in the [Trace] category on the [Debug Tool Settings] tab is set to [Branch], the acquisition time of trace data will increase because complement of instructions between branch instructions is performed. If you wish to display graphs focusing only on variables, we recommend selecting [Data access] in this property.

**3.5 When the Panel Is Opened during Execution of Program****(1) Function List panel/Variable List panel**

The updated static analysis information will be displayed. The dynamic analysis information is not updated.

**(2) Analysis Chart panel**

- If this panel has been never opened since CubeSuite+ was launched:

[Variable Value Changing Chart] tab	Nothing will be displayed.
[Execution Time(Percentage) Chart] tab	Nothing will be displayed.

- If this panel has been opened once since CubeSuite+ was launched:

[Variable Value Changing Chart] tab	<ul style="list-style-type: none"> <li>- <a href="#">Real-time sampling analysis</a> The graph will be updated at the specified sampling interval.</li> <li>- <a href="#">Trace data analysis</a> The previous display contents will be displayed.</li> </ul>
[Execution Time(Percentage) Chart] tab	The previous display contents will be displayed.

### (3) Call Graph panel

- If this panel has been never opened since CubeSuite+ was launched:

The updated static analysis information will be displayed. The dynamic analysis information is not updated.

- If this panel has been opened once since CubeSuite+ was launched:

The previous display contents will be displayed.

### (4) Class/Member panel

The updated static analysis information will be displayed. The dynamic analysis information is not updated.

### (5) Variable Value Changing Chart (zoomed in) panel

The previous display contents will be displayed.

## 3.6 When CC-RX (C++ Source Files) Is Used

### (1) Function List panel

- The following cautions apply to template functions and member functions defined in template classes.
  - "(No Definition)" will appear in the [File Name] column.
  - Only the types will be displayed in the Arguments column. The argument names will not be displayed.
  - "-" will appear in the [Start Address] and [End Address] columns of member functions defined in template classes. If "-" appears in the [Start Address] column, you will not be able to jump to the Editor panel, Disassemble panel, or Memory panel.
  - [Find All References] from the context menu does not display the locations of definitions. Information about the referencing functions and variables is also not displayed.
  - This feature does not count the number of function references in template functions and member functions defined in template classes. Similarly, reference information does not appear via [Find All References] from the context menu.
  - It is not possible to set breakpoints at the start of member functions defined in template classes via [Set Break to Function] from the context menu.
- If a member function defined in a class declaration is only declared and not used, the filename will not be displayed. It will be treated as a function with no defined location.
- If you specify a function parameter with a class type, "-" will be displayed in the [Start Address], [End Address] and [Code Size[Bytes]] columns.
- If you define a function with an argument of type signed char, and an overloaded function with an argument of type char, "-" will be displayed in the [Start Address], [End Address] and [Code Size[Bytes]] columns.

**(2) Variable List panel**

- This feature does not display static variables defined in template functions or member functions defined in template classes.
- This feature does not count the number of variable references in template functions and member functions defined in template classes.
- The compiler changes the types of const variables without an extern/volatile declaration to constants. As a result, they will not appear in the Variable List as variables.
- Global variables with the same name defined in anonymous namespaces in different files will be treated as having the same type.
- [Address] and [Size[Bytes]] of anonymous structures and anonymous unions cannot be displayed.

**(3) Call Graph panel**

- By default, template functions and member functions defined in template classes do not appear in this panel. To display them, on the [Settings] tab of the Property panel, set the [Display the function/variable without definition at Call Graph panel] property of the [General] category to [Yes].
- Functions called from/variables referenced from template functions and member functions defined in template classes do not appear in this panel.
- A constructor/destructor of a class type that was implicitly called in C++ source files does not appear in this panel.

**(4) Class/Member panel**

- Namespace aliases are not displayed.
- [Jump to Source] or [Jump to Declaration of Source] from the context menu is disabled when template functions or member functions defined in template classes are selected.

## APPENDIX A WINDOW REFERENCE

Appendix A provides detailed explanations of windows/panels/dialog boxes used by the analyze tool.

### A.1 Description

The analyze tool has the following windows, panels and dialog boxes.

**Table A-1. Window/Panel/Dialog Box List**

Window/Panel/Dialog Box Name	Description
<a href="#">Main window</a>	This is the first window to open when CubeSuite+ is launched.
<a href="#">Project Tree panel</a>	Displays the components of the project in a tree structure.
<a href="#">Property panel</a>	Displays detailed information on the analyze tool and changes the settings of the information.
<a href="#">Function List panel</a>	Displays the acquired function information.
<a href="#">Variable List panel</a>	Displays the acquired variable information.
<a href="#">Analysis Chart panel</a>	Displays charts of the acquired function information and variable information.
<a href="#">Call Graph panel</a>	Displays calling relationships between functions (call graph).
<a href="#">Class/Member panel</a>	Displays the acquired class information <b>[CC-RX]</b> <sup>Note</sup> , function information and variable information in a tree structure.
<a href="#">Variable Value Changing Chart (zoomed in) panel</a>	Zooms in on the specified range on the chart.
<a href="#">Output panel</a>	Displays operation logs for various components provided by CubeSuite+ and the reference list for functions/variables.
<a href="#">Select Files not To Analyze dialog box</a>	Specifies files not to analyze.
<a href="#">Select Files To Analyze dialog box</a>	Specifies files to analyze.
<a href="#">Path Edit dialog box</a>	Specifies the information file (function list file (*.mtfl)/variable list file (*.mtvl)) to import.
<a href="#">Open File dialog box</a>	Selects a file to restore graphs.
<a href="#">Column Chooser dialog box</a>	Changes the order of the display items and the setting of display/non-display for the <a href="#">Function List panel</a> / <a href="#">Variable List panel</a> .
<a href="#">Call Graph Search dialog box</a>	Searches for a function/variable exists in the call graph displayed in the <a href="#">Call Graph panel</a> .
<a href="#">Filter Settings dialog box</a>	Specifies the filter conditions to display the information on the <a href="#">Function List panel</a> / <a href="#">Variable List panel</a> .
<a href="#">Save As dialog box</a>	Saves the contents of the <a href="#">Function List panel</a> / <a href="#">Variable List panel</a> / <a href="#">Analysis Chart panel</a> / <a href="#">Call Graph panel</a> to a file with a name. Generates the information file (function list file (*.mtfl)/variable list file (*.mtvl)).

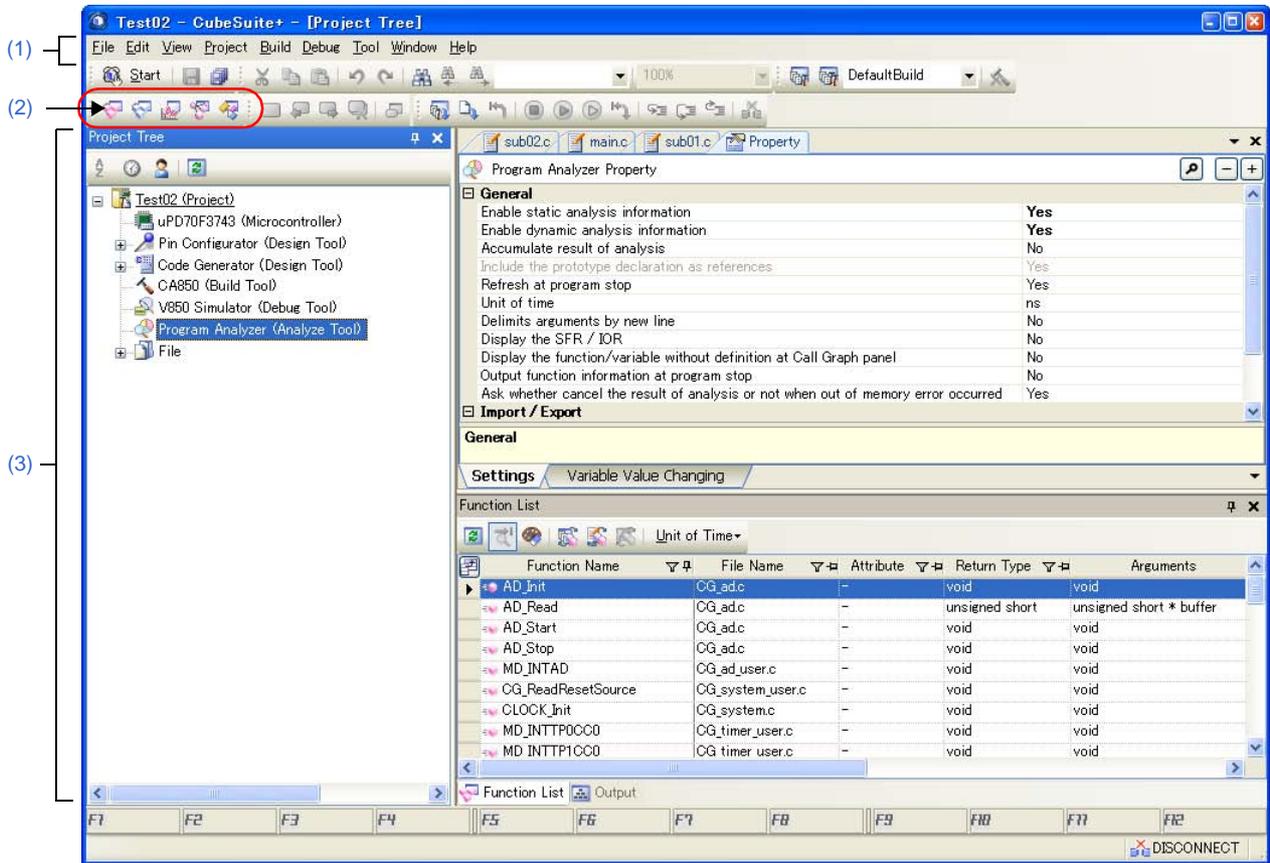
**Note [CC-RX]**

The class information is provided only when C++ source files are subject to analysis.

**Main window**

This is the first window to open when CubeSuite+ is launched.  
 In this window, you can open panels for the analyze tool.

**Figure A-1. Main Window**



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)

**[How to open]**

- From the Windows [start] menu, select [All Programs] >> [Renesas Electronics CubeSuite+] >> [CubeSuite+].

**[Description of each area]**

**(1) Menubar**

**(a) [View]**

The [View] menu for the analyze tool provides the following items and functions (default).

Output	Opens the <a href="#">Output panel</a> .
--------	--

Program Analyzer	The following cascade menus are displayed to open panels for the analyze tool.
Function List	Opens the <a href="#">Function List panel</a> .
Variable List	Opens the <a href="#">Variable List panel</a> .
Analysis Chart	Opens the <a href="#">Analysis Chart panel</a> .
Call Graph	Opens the <a href="#">Call Graph panel</a> .
Class/Member	Opens the <a href="#">Class/Member panel</a> .

**(2) Toolbar**

The toolbar for the analyze tool provides the following items and functions (default).

	Opens the <a href="#">Function List panel</a> . The function of this item is the same as that of [Function List] in the [View] menu.
	Opens the <a href="#">Variable List panel</a> . The function of this item is the same as that of [Variable List] in the [View] menu.
	Opens the <a href="#">Analysis Chart panel</a> . The function of this item is the same as that of [Analysis Chart] in the [View] menu.
	Opens the <a href="#">Call Graph panel</a> . The function of this item is the same as that of [Call Graph] in the [View] menu.
	Opens the <a href="#">Class/Member panel</a> . The function of this item is the same as that of [Class/Member] in the [View] menu.

**(3) Panel display area**

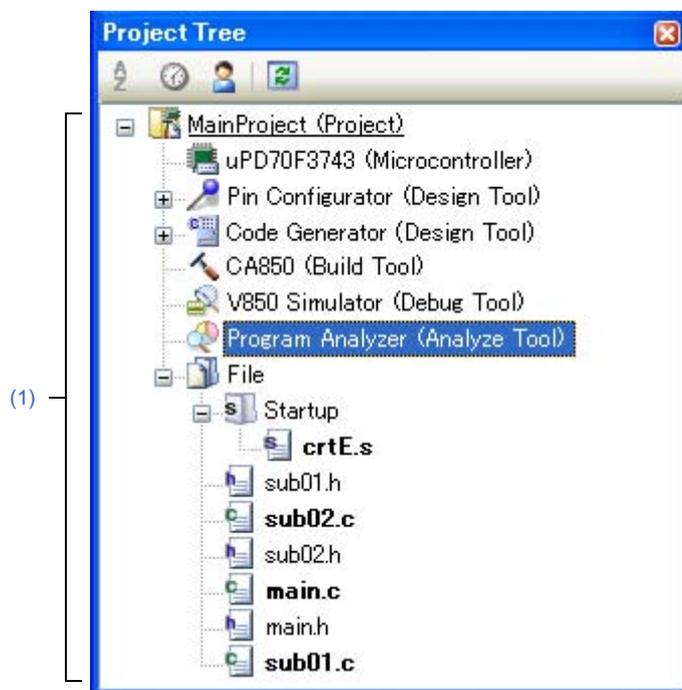
This area consists of multiple panels, each dedicated to a different purpose. See the following sections for details on a panel used by the analyze tool.

- [Project Tree panel](#)
- [Property panel](#)
- [Function List panel](#)
- [Variable List panel](#)
- [Analysis Chart panel](#)
- [Call Graph panel](#)
- [Class/Member panel](#)
- [Variable Value Changing Chart \(zoomed in\) panel](#)
- [Output panel](#)

**Project Tree panel**

This panel is used to display components of the project (microcontroller, design tool, build tool, debug tool, etc.) in a tree structure.

**Figure A-2. Project Tree Panel**



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Context menu\]](#)

**[How to open]**

- From the [View] menu, select [Project Tree].

**[Description of each area]**

**(1) Project tree area**

Project components are displayed in tree view with the following given node.

Node	Description
Program Analyzer (Analyze Tool)	This is the analyze tool to use.

**Remark** When the node is selected, the detailed information (property) is displayed in the [Property panel](#), and you can change the settings.

The [Property panel](#) can be opened by double-clicking on the node when the Property panel is not opened.

**[Context menu]**

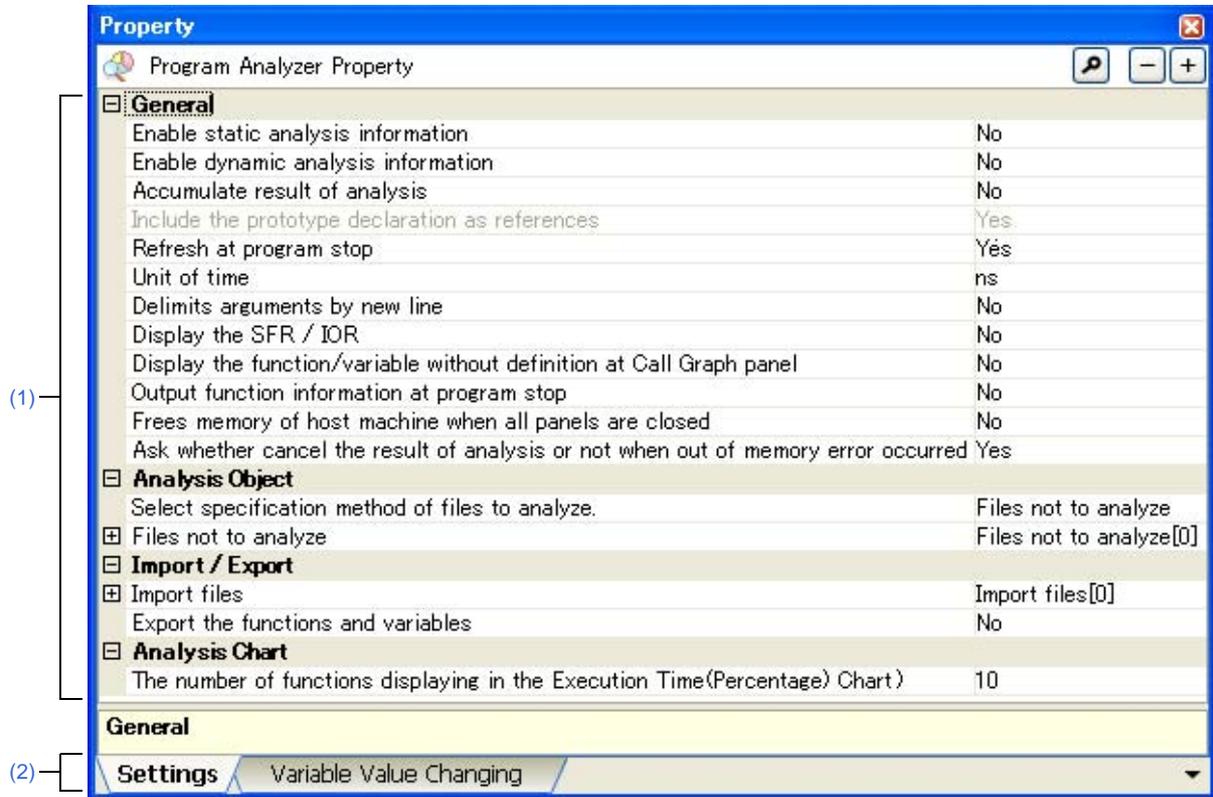
The following context menu items are displayed by right clicking the mouse on [Program Analyzer (Analyze Tool)] node.

Function List	Opens the <a href="#">Function List panel</a> .
Variable List	Opens the <a href="#">Variable List panel</a> .
Analysis Chart	Opens the <a href="#">Analysis Chart panel</a> .
Call Graph	Opens the <a href="#">Call Graph panel</a> .
Class/Member	Opens the <a href="#">Class/Member panel</a> .
Property	Opens the <a href="#">Property panel</a> containing the information for the analyze tool.

## Property panel

This panel is used to display the detailed information on the analyze tool and change the settings of the information.

Figure A-3. Property Panel (Example of Using [V850])



The following items are explained here.

- [How to open]
- [Description of each area]
- [[Edit] menu (Property panel-dedicated items)]
- [Context menu]

### [How to open]

- On the [Project Tree panel](#), select the [Program Analyzer (Analyze Tool)] node, and then select [Property] from the [View] menu.
- On the [Project Tree panel](#), select the [Program Analyzer (Analyze Tool)] node, and then select [Property] from the context menu.

**Remark** If the Property panel has been opened, the detailed information on the analyze tool is displayed by selecting the [Program Analyzer (Analyze Tool)] node on the [Project Tree panel](#).

### [Description of each area]

#### (1) Detailed information display/change area

In this area, the detailed information on the analyze tool is displayed by category in the list. Also, you can directly change its settings.

The  mark indicates all the items in the category are expanded. The  mark indicates all the items are shrink. You can expand/shrink the items by clicking these marks or double-clicking the category name.

For details on the information/how to setup in the category and property items contained in it, see the section explaining the corresponding tab.

## (2) Tab selection area

Categories for the display of the detailed information are changed when each tab is selected.

In this panel, a following tab is contained (see the section explaining the tab for details on the display/setting).

- [\[Settings\] tab](#)
- [\[Variable Value Changing\] tab](#)

## [[Edit] menu (Property panel-dedicated items)]

The [Edit] menu for this panel provides the following items and functions.

Undo	Undoes the latest property value editing being done.
Cut	Deletes the selected character string(s) and copies them to the clipboard while editing the property value.
Copy	Copies the contents of the selected range to the clipboard as character string(s).
Paste	Pastes the contents of the clipboard to the property value while editing the property value.
Delete	Deletes the selected character string(s) while editing the property value.
Select All	Selects all the character strings in the selected property while editing the property value.

## [Context menu]

The context menu displayed by right-clicking on this panel provides the following items and functions.

### (1) While not editing the property value

Reset to Default	Restores the selected setting of the property item to default value.
Reset All to Default	Restores all the selected settings of the property items on the tab to default value.

### (2) While editing the property value

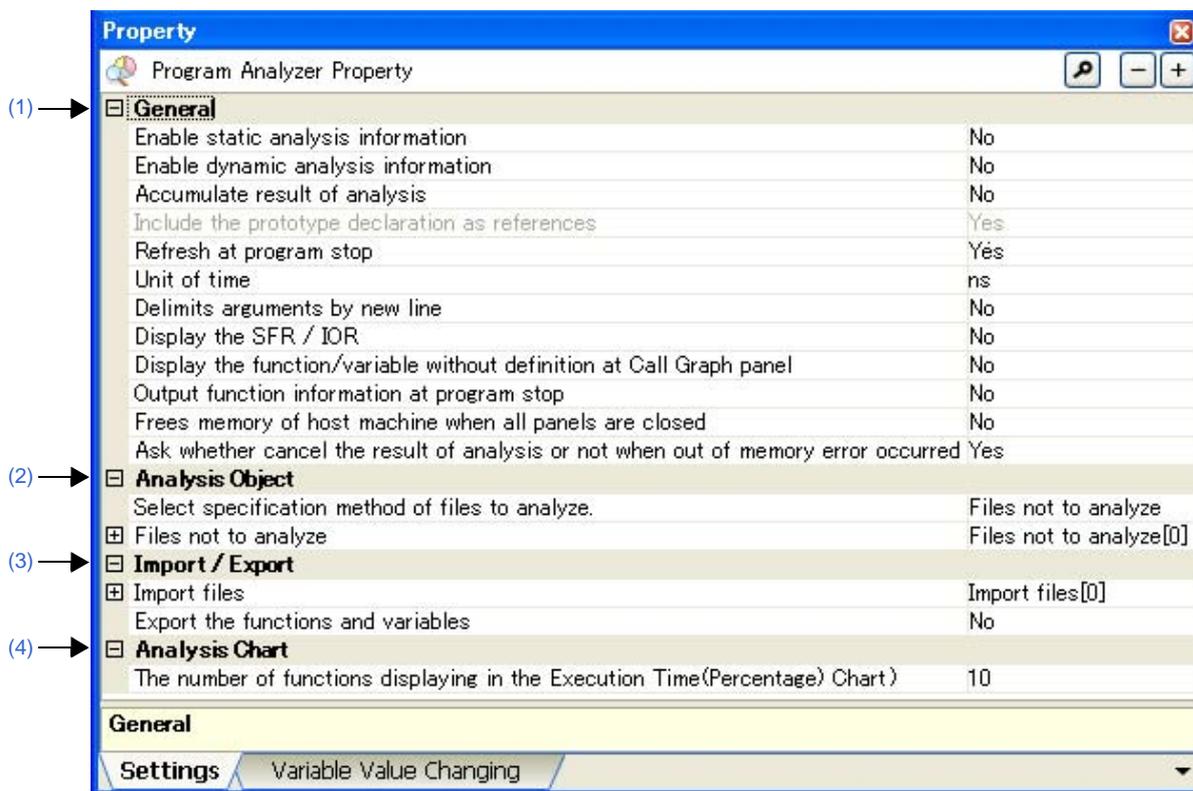
Undo	Undoes the latest property value editing being done.
Cut	Deletes the selected character string(s) and copies them to the clipboard while editing the property value.
Copy	Copies the contents of the selected range to the clipboard as character string(s).
Paste	Pastes the contents of the clipboard to the property value while editing the property value.
Delete	Deletes the selected character string(s) while editing the property value.
Select All	Selects all the character strings in the selected property while editing the property value.

[Settings] tab

The [Setting] tab is used to display the detailed information categorized by the following and the configuration can be changed.

- (1) [General]
- (2) [Analysis Object]
- (3) [Import / Export]
- (4) [Analysis Chart]

Figure A-4. Property Panel: [Settings] Tab (Example of Using [V850])



[Description of each category]

(1) [General]

The general information on the analyze tool is displayed and its configuration can be changed.

Enable static analysis information	Select whether the analyze tool should force output of cross reference information when performing a build in order to obtain the cross-reference information necessary to acquire <a href="#">Static analysis information</a> , ignoring the property setting <sup>Note 1</sup> on the build tool specifying whether to output cross reference information.	
	Default	No
	Modifying	Select from the drop-down list.
	Available values	Yes
	No	Gives priority to the build tool's property settings.

Enable dynamic analysis information	Select whether to forcibly enable the functions of the debug tool <sup>Note 2</sup> ignoring the property settings (enabled/disabled) of those functions that are necessary for the analyze tool to acquire the <a href="#">Dynamic analysis information</a> .				
	Default	No			
	Modifying	Select from the drop-down list.			
	Available values	<table border="1"> <tr> <td>Yes</td> <td>Ignores the debug tool's property settings, and enables the debug tool's functions forcibly.</td> </tr> <tr> <td>No</td> <td>Gives priority to the debug tool's property settings.</td> </tr> </table>	Yes	Ignores the debug tool's property settings, and enables the debug tool's functions forcibly.	No
Yes	Ignores the debug tool's property settings, and enables the debug tool's functions forcibly.				
No	Gives priority to the debug tool's property settings.				
Accumulate result of analysis	<p>Select whether to display the cumulative value for each program execution for the execution count and execution time as the analysis information.</p> <p>The target items for this property are as follows:</p> <ul style="list-style-type: none"> <li>- <a href="#">Function List panel</a> [Execution Count]/[Execution Time[<i>unit</i>]]</li> <li>- <a href="#">Variable List panel</a> [Read Count]/[Write Count]/[Read/Write Count]/[Minimum Value]/[Maximum Value]</li> <li>- <a href="#">Call Graph panel</a> Execution count/Read count/Write count</li> </ul>				
	Default	No			
	Modifying	Select from the drop-down list.			
	Available values	Yes	Displays the sum of the measurements from the last program execution and from this one.		
		No	Displays the measurements for each program execution. If the value is changed from [Yes] to [No], then the current analysis results will be cleared, and the measured values displayed.		
	Include the prototype declaration as references	Select whether to target the prototype declaration as the information of the function references when displaying a list of locations referencing a function (see "2.11 <a href="#">Display List of Referencing Location</a> ").			
Default		Yes			
Modifying		<p><b>[CC-RH][CC-RX][CX][NC30(Localised support)]</b> Select from the drop-down list.</p> <p><b>[CA850][CA78K0R][CA78K0]</b> Changes not allowed</p>			
Available values		Yes	Displays the prototype declaration.		
		No	Does not display the prototype declaration.		
Refresh at program stop	Select whether to update the contents of the <a href="#">Function List panel</a> / <a href="#">Variable List panel</a> / <a href="#">Analysis Chart panel</a> / <a href="#">Call Graph panel</a> when the program stops executing.				
	Default	Yes			
	Modifying	Select from the drop-down list.			
	Available values	Yes	Updates the contents of the panel after the program execution is stopped.		
		No	Does not update the contents of the panel even if the program execution is stopped.		
		Specify individually	Enables the  button on each panel (updates the contents of the panel by clicking this button).		

Unit of time	Select the time unit for the analyze tool.		
	Default	ns	
	Modifying	Select from the drop-down list.	
	Available values	ns	Displays the time in nanosecond units. The value is rounded to integer number.
		µs	Displays the time in microsecond units. The value is rounded to three decimal places.
		ms	Displays the time in millisecond units. The value is rounded to three decimal places.
		s	Displays the time in second units. The value is rounded to three decimal places.
h:min:s	Displays the time as "hours, minutes (0 - 59), seconds (0 - 59)".		
Delimits arguments by new line	Select whether to display the [Arguments] items in the <a href="#">Function List panel</a> with newlines.		
	Default	No	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Displays in multiple lines, one value per line.
No		Displays without newlines, comma (" , ") separated.	
Display the SFR / IOR	Select whether to display the SFR/IOR on the <a href="#">Variable List panel</a> by treating it as a variable.		
	Default	No	
	Modifying	<b>[CC-RH][CC-RX][NC30(Localised support)]</b> Changes not allowed <b>[CA850][CX][CA78K0R][CA78K0]</b> Select from the drop-down list.	
	Available values	Yes	Displays the SFR/IOR.
No		Does not display the SFR/IOR.	
Display the function/variable without definition at Call Graph panel	Select whether to display the function/variable without definition (i.e. a function/variable whose source file does not exist) in the <a href="#">Call Graph panel</a> .		
	Default	No	
	Modifying	Select from the drop-down list.	
	Available values	Yes	Displays the function without definition.
No		Does not display the function without definition.	

Output function information at program stop <b>[RH850]</b> <b>[V850]</b>	Select whether to output the information file for STF <sup>Note 3</sup> to the folder specified with "[Common Options] tab >> [Output File Type and Path] category >> [Intermediate file output folder] property" on the property panel of the build tool to use.				
	Default	No			
	Modifying	Select from the drop-down list.			
	Available values	<table border="1"> <tr> <td>Yes</td> <td>When [Yes] is selected, the contents of the <a href="#">Function List panel</a> currently being displayed is output to a file (if the <a href="#">Function List panel</a> is not displayed, the information acquired from the last trace data is output).  Thereafter, the information acquired from the last trace data is output to the file each time program execution is stopped.  Note that the file is always overwritten.</td> </tr> <tr> <td>No</td> <td>Does not output the information file for STF.</td> </tr> </table>	Yes	When [Yes] is selected, the contents of the <a href="#">Function List panel</a> currently being displayed is output to a file (if the <a href="#">Function List panel</a> is not displayed, the information acquired from the last trace data is output).  Thereafter, the information acquired from the last trace data is output to the file each time program execution is stopped.  Note that the file is always overwritten.	No
Yes	When [Yes] is selected, the contents of the <a href="#">Function List panel</a> currently being displayed is output to a file (if the <a href="#">Function List panel</a> is not displayed, the information acquired from the last trace data is output).  Thereafter, the information acquired from the last trace data is output to the file each time program execution is stopped.  Note that the file is always overwritten.				
No	Does not output the information file for STF.				
Frees memory of host machine when all panels are closed	Select whether to free memory of the host machine when all of the panels provided by the analyze tool ( <a href="#">Function List panel</a> , <a href="#">Variable List panel</a> , <a href="#">Analysis Chart panel</a> , <a href="#">Call Graph panel</a> , <a href="#">Class/Member panel</a> , and <a href="#">Variable Value Changing Chart (zoomed in) panel</a> ) are closed.				
	Default	Yes			
	Modifying	Select from the drop-down list.			
	Available values	<table border="1"> <tr> <td>Yes</td> <td>Frees the memory of the host machine when all panels provided by the analyze tool are closed so that the other CubeSuite+ plug-in tools operate stably.  Note that after memory is freed, it may take some time to open a panel of the analyze tool next time.</td> </tr> <tr> <td>No</td> <td>Does not free the memory of the host machine even when all panels provided by the analyze tool are closed.  The time required to open a panel of the analyze tool next time will be reduced.</td> </tr> </table>	Yes	Frees the memory of the host machine when all panels provided by the analyze tool are closed so that the other CubeSuite+ plug-in tools operate stably.  Note that after memory is freed, it may take some time to open a panel of the analyze tool next time.	No
Yes	Frees the memory of the host machine when all panels provided by the analyze tool are closed so that the other CubeSuite+ plug-in tools operate stably.  Note that after memory is freed, it may take some time to open a panel of the analyze tool next time.				
No	Does not free the memory of the host machine even when all panels provided by the analyze tool are closed.  The time required to open a panel of the analyze tool next time will be reduced.				
Ask whether cancel the result of analysis or not when out of memory error occurred	Select whether to save the memory capacity by deleting the result of analysis or to display the incomplete result of analysis without deleting it when an insufficient memory error occurs during a processing of analysis.  Note, however, that the operation of CubeSuite+ may become unstable if the result of analysis is not deleted.				
	Default	Yes			
	Modifying	Select from the drop-down list.			
	Available values	<table border="1"> <tr> <td>Yes</td> <td>Displays a message dialog box prompting you to select whether or not to delete the result of analysis when an insufficient memory error occurs.</td> </tr> <tr> <td>No</td> <td>Does not display a message dialog box and deletes the result of analysis when an insufficient memory error occurs.</td> </tr> </table>	Yes	Displays a message dialog box prompting you to select whether or not to delete the result of analysis when an insufficient memory error occurs.	No
Yes	Displays a message dialog box prompting you to select whether or not to delete the result of analysis when an insufficient memory error occurs.				
No	Does not display a message dialog box and deletes the result of analysis when an insufficient memory error occurs.				

**Notes 1.** The settings on the following property of your build tool's Property panel.

- **[CC-RH][CX]**  
[Common Options] tab >> [Output File Type and Path] category >> [Output cross reference information] property
- **[CC-RX][NC30(Localised support)]**  
[Compile Options] tab >> [Others] category >> [Output cross reference information] property

- [CA850]  
[Cross Reference Options] tab >> [Cross Reference Tool] category >> [Output cross reference information] property
  - [CA78K0R][CA78K0]  
[Compile Options] tab >> [List File] category >> [Output cross reference list file] property
2. This applies to the following debug tool's functions (in the order of priority).
- Trace function
  - Real-time display update function (RRM function/RAM monitor (pseudo RRM) function)
  - Coverage function
- The corresponding property settings differ depending on the selected microcontroller and debug tool. For details on how to enable these functions, see "CubeSuite+ Integrated Development Environment User's Manual: Debug" of the microcontroller used.
3. The same information as the content on the [Function List panel](#) is output to the STF information file (FuncInfo.csv) (including information for items that are currently hidden).

**(2) [Analysis Object]**

The detailed information on analysis objects is displayed and its configuration can be changed. For details on the analysis objects, see "[1.1.1 What is analyzed](#)".

Select specification method of files to analyze	Select the specification method of files to analyze by the analyze tool.	
	Default	Files not to analyze
	Modifying	Select from the drop-down list.
	Available values	Files not to analyze
Files to analyze		Specifies files to analyze.
Files not to analyze	Specify files that are not the subject of analysis. This property appears only when the <a href="#">[Select specification method of files to analyze]</a> property is set to [Files not to analyze].	
	Default	Files not to analyze[0]
	Modifying	Specify with the <a href="#">Select Files not To Analyze dialog box</a> . The Select files not to analyze dialog box is opened by clicking the [...] button that appears at right edge in the column when this property is selected (no files can be specified on the Property panel).
Files to analyze	Specify files that are the subject of analysis. This property appears only when the <a href="#">[Select specification method of files to analyze]</a> property is set to [Files to analyze].	
	Default	Files to analyze[0]
	Modifying	Specify with the <a href="#">Select Files To Analyze dialog box</a> . The Select files to analyze dialog box is opened by clicking the [...] button that appears at right edge in the column when this property is selected (no files can be specified on the Property panel).

**(3) [Import / Export]**

The detailed information on import/export functions is displayed and its configuration can be changed. For details on the import/export function, see "[2.12 Import or Export Information File](#)".

Import files	Specify the file to import. The following placeholders are supported: %ProjectName%: Replaces itself with the project name. %MicomToolPath%: Replaces itself with the absolute path of the CubeSuite+ install folder. The reference point of the relative path is the project folder. Note that if the same file is specified, only the file that is first specified will be imported. The name of the file to import is displayed as the subproperty.	
	Default	Import files [0]
	Modifying	Specify with the <a href="#">Path Edit dialog box</a> . The Path Edit dialog box is opened by clicking the [...] button that appears at right edge in the column when this property is selected (no files can be specified on the Property panel).
Available values	Up to 64 files	
Export the functions and variables	Select whether to generate the information files (function list file (*.mtfl)/variable list file (*.mtvl)) with the contents of the <a href="#">Function List panel/Variable List panel</a> when a build or rebuild is performed.	
	Default	No
	Modifying	Select from the drop-down list.
	Available values	Yes
	No	Does not Generate the information files.
Export file name for functions	Specify the name of the function list file (*.mtfl) to be generated. The extension (*.mtfl) cannot be changed. If the extension is omitted, "mtfl" is appended. Note that if this property is left blank, no files will be generated. The following placeholders are supported: %ProjectName%: Replaces itself with the project name. %ActiveProjectName%: Replaces itself with the active project name. The reference point of the relative path is the project folder. Note that if this property is left blank, no files will be generated. This property appears only when the <a href="#">[Export the functions and variables]</a> property is set to [Yes].	
	Default	%ProjectName%.mtfl
	Modifying	Directly enter from the keyboard.
	Available values	Up to 259 characters

Export file name for variables	Specify the name of the variable list file (*.mtvl) generated. The extension (*.mtvl) cannot be changed. If the extension is omitted, "mtvl" is appended. Note that if this property is left blank, no files will be generated. The following placeholders are supported: %ProjectName%: Replaces itself with the project name. %ActiveProjectName%: Replaces itself with the active project name. The reference point of the relative path is the project folder. This property appears only when the <a href="#">[Export the functions and variables]</a> property is set to [Yes].	
	Default	%ProjectName%.mtvl
	Modifying	Directly enter from the keyboard.
	Available values	Up to 259 characters

**(4) [Analysis Chart]**

The detailed information on analysis graphs is displayed and its configuration can be changed.

For details on the analysis graph, see ["2.13 Display Analysis Information in Chart"](#).

**Caution** The contents of the [Analysis Chart panel](#) is updated automatically if you change the setting of the property in this category.

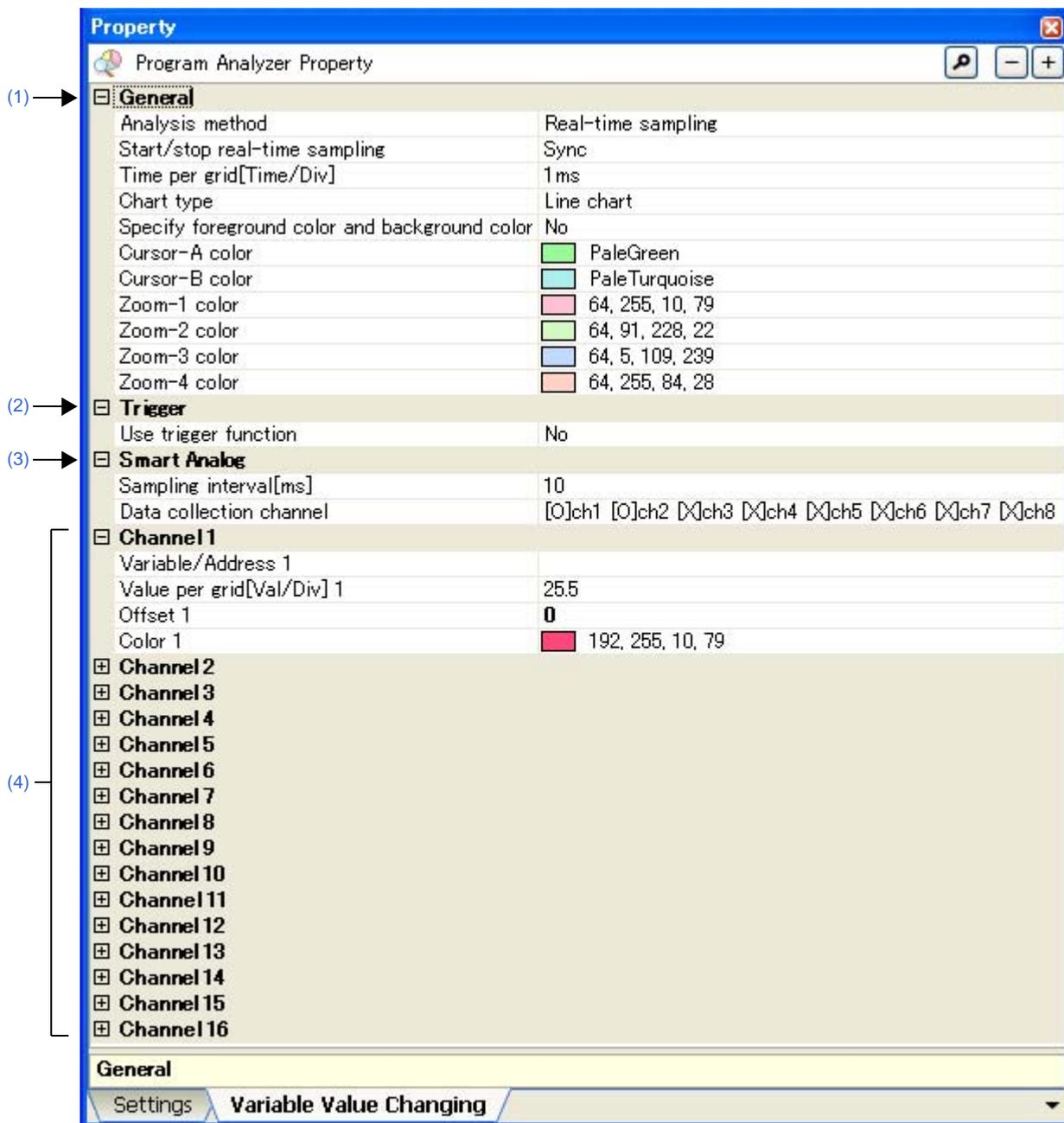
The number of functions displaying in the Execution Time(Percentage) Chart	Specify the number of the functions displayed on the <a href="#">[Execution Time(Percentage) Chart] tab</a> of the <a href="#">Analysis Chart panel</a> . The function with a large ratio of the execution time is displayed in the graph by priority, and the remainder is collectively displayed as "Others".	
	Default	10
	Modifying	Directly enter from the keyboard.
	Available values	Integer number between 1 and 100

**[Variable Value Changing] tab**

The [Variable Value Changing] tab is used to display the detailed information on graphs displayed in the [Variable Value Changing Chart] tab of the Analysis Chart panel categorized by the following and the configuration can be changed.

- (1) [General]
- (2) [Trigger]
- (3) [Smart Analog] [E1/E20[RL78]]
- (4) [Channel 1 - 16]

**Figure A-5. Property Panel: [Variable Value Changing] Tab (Example of Using [E1[RL78]])**



[Description of each category]

(1) [General]

The general information on graphs is displayed and its configuration can be changed.

Analysis method	Select the method for acquiring the graph data (see “(2) <a href="#">Specify the method for acquiring the graph data</a> ”).		
	Default	Real-time sampling	
	Modifying	Select from the drop-down list. Note that changes cannot be made during execution of a program.	
	Available values	Real-time sampling	Displays graphs based on the data acquired via the <a href="#">RRM function/RAM monitor (pseudo RRM) function</a> of the debug tool. Note, however, that if the debug tool is set in the <a href="#">Data collection mode</a> <sup>Note 1</sup> , then the data will be collected using a sampling method for Smart Analog.
		Analysis of trace data	Displays graphs based on the trace data acquired via the <a href="#">Trace function</a> of the debug tool. This does not appear when any one of the following: - The debug tool does not support the trace function. - The debug tool does not support the trace time tag feature. - The debug tool is set in the <a href="#">Data collection mode</a> <sup>Note 1</sup> .
Load from file	Displays graphs by loading the analysis chart data file (*.mtac) that has been saved.		
Analysis chart data file	Specify the analysis chart data file (*.mtac) to restore graphs. The following placeholders are supported: %ProjectName%: Replaces itself with the project name. %MicomToolPath%: Replaces itself with the absolute path of the CubeSuite+ install folder. The reference point of the relative path is the project folder. This property appears only when the [ <a href="#">Analysis method</a> ] property is set to [Load from file].		
	Default	Blank	
	Modifying	Specify with the <a href="#">Open File dialog box</a> . The Open File dialog box is opened by clicking the [...] button that appears at right edge in the column when this property is selected (no files can be specified on the Property panel).	
	Available values	One file (*.mtac)	
Start/stop real-time sampling	Select whether the start/stop of the real-time sampling synchronizes to the start/stop of a program execution. This property appears only when the [ <a href="#">Analysis method</a> ] property is set to [Real-time sampling].		
	Default	Sync	
	Modifying	Select from the drop-down list.	
	Available values	Sync	Synchronizes to the start/stop of a program execution.
Manual		Does not synchronize to the start/stop of a program execution. It can be controlled via the [Sampling] button on the [ <a href="#">Variable Value Changing Chart</a> ] tab.	

Auto adjustment	Select whether to calculate the optimum values from the obtained graph data and specify them in the [Time per grid[Time/Div]] property and [Value per grid[Val/Div] 1 - 16][Offset 1 - 16] properties for the X and Y axes of the graph (see "(c) Setting the display range"). <b>[E1/E20[RL78]]</b> This property does not appear when the debug tool is set in the <a href="#">Data collection mode</a> <sup>Note 1</sup> .						
	Default	Only during program stop Note that [None] is fixed to this property if the [Analysis method] property is set to [Load from file] or the [Use trigger function] property is set to [Yes].					
	Modifying	Select from the drop-down list.					
	Available values	<table border="1"> <tr> <td>Always</td> <td>Always adjusts the graph display automatically.</td> </tr> <tr> <td>Only during program stop</td> <td>Automatically adjusts the graph display only when program execution stops.</td> </tr> <tr> <td>None</td> <td>Does not adjust the graph display automatically.</td> </tr> </table>	Always	Always adjusts the graph display automatically.	Only during program stop	Automatically adjusts the graph display only when program execution stops.	None
Always	Always adjusts the graph display automatically.						
Only during program stop	Automatically adjusts the graph display only when program execution stops.						
None	Does not adjust the graph display automatically.						
Number of transition location for auto adjustment	Specify the number of value transition points to be displayed in the rendering area when the graph display is automatically adjusted. This property does not appear when the [Auto adjustment] property is set to [None].						
	Default	20					
	Modifying	Directly enter from the keyboard.					
	Available values	The value in decimal number within the following range. - 1 to 1000					
Time per grid[Time/Div]	Specify the time for each of the 10 units (grid segments) that the grid area is divided into. This property is enabled only when the [Auto adjustment] property is set to [None].						
	Default	1ms Note that the value loaded from the file is set to this property if [Load from file] is specified with the [Analysis method] property.					
	Modifying	Directly enter from the keyboard.					
	Available values	Any one of the following (1ns to 10s: in decimal number) <sup>Note 2</sup> . - 1 to 10s - 1 to 10000ms - 1 to 10000000µs - 1 to 10000000000ns					
Chart type	Select the chart type (i.e. the format of lines between transition points).						
	Default	Depends on the specification of the [Analysis method] property. - When [Real-time sampling] is specified Line chart - When [Analysis of trace data] is specified Step line chart (fixed)					
	Modifying	Depends on the specification of the [Analysis method] property. - When [Real-time sampling] is specified Select from the drop-down list. - When [Analysis of trace data] is specified Changes not allowed					
	Available values	<table border="1"> <tr> <td>Line chart</td> <td>Displays the graph in a normal line chart.</td> </tr> <tr> <td>Step line chart</td> <td>Displays the graph in a step-plot line chart.</td> </tr> </table>	Line chart	Displays the graph in a normal line chart.	Step line chart	Displays the graph in a step-plot line chart.	
Line chart	Displays the graph in a normal line chart.						
Step line chart	Displays the graph in a step-plot line chart.						

Specify foreground color and background color	Select whether to select the foreground color and background color of the graph.	
	Default	No
	Modifying	Select from the drop-down list.
	Available values	Yes
No		Does not select the foreground color and background color of the graph (i.e. uses the setting of background color for the [Default] item in the Option dialog box).
Background color	Specify the background color <sup>Note 3</sup> of the graph. This property appears only when the [ <a href="#">Specify foreground color and background color</a> ] property is set to [Yes].	
	Default	The setting of background color for the [Default] item in the Option dialog box.
	Modifying	Specify with the color selection combo box or directly enter from the keyboard.
	Available values	When directly enter from the keyboard The value in decimal number/hexadecimal number (prefix "0x" is needed) or the color name (see " <a href="#">About the specification of the color</a> ").
Background color(Lost section)	Specify the background color <sup>Note 3</sup> of the graph in a lost section (see "(a) Graph"). This property appears only when the [ <a href="#">Specify foreground color and background color</a> ] property is set to [Yes].	
	Default	The setting of background color for the [Lost] item in the Option dialog box.
	Modifying	Specify with the color selection combo box or directly enter from the keyboard.
	Available values	When directly enter from the keyboard: The value in decimal number/hexadecimal number (prefix "0x" is needed) or the color name (see " <a href="#">About the specification of the color</a> ").
Foreground color	Specify the foreground color <sup>Note 3</sup> of the graph. This property appears only when the [ <a href="#">Specify foreground color and background color</a> ] property is set to [Yes].	
	Default	The setting of font color for the [Default] item in the Option dialog box.
	Modifying	Specify with the color selection combo box or directly enter from the keyboard.
	Available values	When directly enter from the keyboard: The value in decimal number/hexadecimal number (prefix "0x" is needed) or the color name (see " <a href="#">About the specification of the color</a> ").
Cursor-A -B color	Specify the color of the Cursor-A and Cursor-B.	
	Default	Cursor-A: PaleGreen Cursor-B: PaleTurquoise
	Modifying	Specify with the color selection combo box or directly enter from the keyboard.
	Available values	When directly enter from the keyboard: The value in decimal number/hexadecimal number (prefix "0x" is needed) or the color name (see " <a href="#">About the specification of the color</a> ").

Zoom-1 - 4 color	Specify the color of the Zoom-1 to the Zoom-4.	
	Default	Zoom-1: 64, 255,10,79 Zoom-2: 64, 91, 228, 22 Zoom-3: 64, 5, 109, 239 Zoom-4: 64, 255, 84, 28
	Modifying	Specify with the color selection combo box or directly enter from the keyboard.
	Available values	When directly enter from the keyboard: The value in decimal number/hexadecimal number (prefix "0x" is needed) or the color name (see " <a href="#">About the specification of the color</a> ").

**Notes 1. [E1/E20[RL78]]**

- This feature is supported only when the selected microcontroller incorporates Smart Analog IC.
- 2. If the unit (case insensitive) is omitted, when [Real-time sampling] is selected in the [[Analysis method](#)] property, it will be handled as "ms". In other cases, it will be handled as "ns".  
When this property is changed, if the value of the [[Trigger position](#)] property is greater than "the value of this property x 10", then "the value of this property x 10" is automatically specified to the [Trigger position] property.
- 3. The color specified in this property will also be reflected to the foreground color/background color of the [Variable Value Changing Chart \(zoomed in\) panel](#).

**(2) [Trigger]**

The information on the trigger function is displayed and its configuration can be changed (see "(3) [Use the trigger function](#)").

Use trigger function	Select whether to display the graphs by using the trigger function.	
	Default	No
	Modifying	Depends on the specification of the [ <a href="#">Analysis method</a> ] property. - When [Real-time sampling] is specified Select from the drop-down list. - When [Analysis of trace data] or [Load from file] is specified Changes not allowed Note that changes cannot be made during execution of a program.
	Available values	Yes    Uses the trigger function. No     Does not use the trigger function.
Trigger mode	Select the trigger mode (the timing at which graphs are updated). This property appears only when the [ <a href="#">Use trigger function</a> ] property is set to [Yes].	
	Default	Auto
	Modifying	Select from the drop-down list. Note that changes cannot be made during execution of a program.
	Available values	Auto    Re-renders the graph cyclically. Also clears and re-renders the graph upon receipt of a trigger signal. Single   Only renders the graph upon the first trigger signal from the start of sampling. Normal   Only re-renders the graph upon receipt of a trigger signal.

Trigger source	Select the variable (channel) that is the source of a trigger signal. This property appears only when the <a href="#">[Use trigger function]</a> property is set to [Yes].		
	Default	ch1	
	Modifying	Select from the drop-down list. Note that changes cannot be made during execution of a program.	
	Available values	Any one of the channels (ch1 to ch16)	
Trigger level	Select the trigger level (i.e. the threshold value for determining as a trigger signal). This property appears only when the <a href="#">[Use trigger function]</a> property is set to [Yes].		
	Default	0	
	Modifying	Select from the drop-down list. Note that changes cannot be made during execution of a program.	
	Available values	The value in decimal number/hexadecimal number within the following range. "The minimum value of the trigger variable ( <a href="#">Trigger source</a> )" to "The maximum value of the trigger variable ( <a href="#">Trigger source</a> )" Values in floating point notation can also be specified.	
Direction of trigger edge	Select the direction of the trigger edge. This property appears only when the <a href="#">[Use trigger function]</a> property is set to [Yes].		
	Default	Rising	
	Modifying	Select from the drop-down list. Note that changes cannot be made during execution of a program.	
	Available values	Rising	Generates a trigger signal when the value of <a href="#">Trigger source</a> changes from less than <a href="#">Trigger level</a> to greater than or equal to <a href="#">Trigger level</a> .
		Falling	Generates a trigger signal when the value of <a href="#">Trigger source</a> changes from greater than <a href="#">Trigger level</a> to less than or equal to <a href="#">Trigger level</a> .
Both	Generates a trigger signal when both "Rising" and "Falling".		
Trigger position	Specify the trigger position (i.e. the horizontal position that indicates the location at which a trigger signal has been generated). This property appears only when the <a href="#">[Use trigger function]</a> property is set to [Yes].		
	Default	0s	
	Modifying	Directly enter from the keyboard. Note that changes cannot be made during execution of a program.	
	Available values	The value in decimal number within the following range <sup>Note</sup> . - 0 to "(the value of the <a href="#">[Time per grid[Time/Div]]</a> property) x 10"	
Trigger mark color	Specify the color of the trigger marks that indicate trigger level and the trigger position. This property appears only when the <a href="#">[Use trigger function]</a> property is set to [Yes].		
	Default	Color Orange	
	Modifying	Specify with the color selection combo box or directly enter from the keyboard. Note that changes cannot be made during execution of a program.	
	Available values	When directly enter from the keyboard: The value in decimal number/hexadecimal number (prefix "0x" is needed) or the color name (see " <a href="#">About the specification of the color</a> ").	

**Note** If the unit (s, ms, us/μs, ns: case insensitive) is omitted, it will be handled as "ms".  
 If the specified value is greater than "the value of [Time per grid[Time/Div]] x 10", then "the value of [Time per grid[Time/Div]] x 10" is automatically specified to this property.

**(3) [Smart Analog] [E1/E20[RL78]]**

The information on the Smart Analog function is displayed and its configuration can be changed.

**Caution** This category appears only when the selected microcontroller incorporates Smart Analog IC and the debug tool is set to the **Data collection mode**.

Sampling interval[ms]	Specify the sampling interval in ms unit to collect the data by using Smart Analog function. Before the debug tool starts acquiring data in the <b>Data collection mode</b> , the value of the symbol (r_dbg_graph.c#static_e1_waveout_rate) is replaced with the value specified for this property.	
	Default	10
	Modifying	Directly enter from the keyboard. Note that changes cannot be made during execution of a program.
	Available values	0 to 1000 in decimal number
Data collection channel	Select channels to collect the data. Before the debug tool starts acquiring data in the <b>Data collection mode</b> , the value of the symbol (r_dbg_graph.c#static_e1_waveout_flag) is replaced with the value specified for this property.	
	Default	All check boxes are not selected
	Modifying	Select check boxes from the drop-down list. Note that changes cannot be made during execution of a program.
	Available values	ch1 to ch8 (multiple selections possible)

**(4) [Channel 1 - 16]**

The information on the graphing target that is registered in each channel is displayed and its configuration can be changed (see "(1) Register graphing targets").

**Caution** If [Load from file] has been specified with the [Analysis method] property, then the values loaded from the file are reflected to the properties in this category and these values cannot be changed.

Variable/Address 1 - 16	Specify the name of a variable or an address expression that is registered as a graphing target (see "(1) Register graphing targets"). Note that the values of all properties in this category will be changed to the default value if the value of this property is changed. <b>[E1/E20[RL78]]</b> When the debug tool is in the <b>Data collection mode</b> , character strings specified in this property are only displayed as a label for the <b>Variable name check box</b> (they are not handled as a graphing target).	
	Default	Blank
	Modifying	Directly enter from the keyboard. Note that changes cannot be made during execution of a program.
	Available values	Up to 2046 characters <sup>Note 1</sup>

Type/Size 1 - 16	Select the type and size of the variable/address to register. <b>[E1/E20[RL78]]</b> This property does not appear when the debug tool is in the <a href="#">Data collection mode</a> .	
	Default	Auto
	Modifying	Select from the drop-down list. Note that changes cannot be made during execution of a program.
	Available values	Any one of the following: - Auto <sup>Note 2</sup> - signed 1byte(8bits) - signed 2bytes(16bits) - signed 4bytes(32bits) - unsigned 1byte(8bits) - unsigned 2bytes(16bits) - unsigned 4bytes(32bits)
Value per grid[Val/Div] 1 - 16	Select the value for each of the 10 units (grid segments) that the grid area is divided into. This property is enabled only when the <a href="#">[Auto adjustment]</a> property is set to [None].	
	Default	(“ <i>The maximum value of the variable</i> ” - “ <i>The minimum value of the variable</i> ”) / 10 Note that the value loaded from the file is set to this property if [Load from file] is specified with the <a href="#">[Analysis method]</a> property. <b>[E1/E20[RL78]]</b> When the debug tool is in the <a href="#">Data collection mode</a> : 409.5
	Modifying	Select from the drop-down list. Note that changes cannot be made during execution of a program.
	Available values	The value in decimal number/hexadecimal number within the following range: “ <i>The minimum positive value of the variable</i> ” to (“ <i>The maximum value of the variable</i> ” - “ <i>The minimum value of the variable</i> ”) Values in floating point notation can also be specified.
Offset 1 - 16	Specify the offset of the graph. The graph of the variable value to which this offset value is added is rendered. This property is enabled only when the <a href="#">[Auto adjustment]</a> property is set to [None].	
	Default	0 <b>[E1/E20[RL78]]</b> When the debug tool is in the <a href="#">Data collection mode</a> : -2048
	Modifying	Directly enter from the keyboard.
	Available values	The value in decimal number within the following range: “ <i>The minimum value of the float type</i> (Approx. -3.4028235e+38)” to “ <i>The maximum value of the float type</i> (Approx. 3.4028235e+38)” Values with decimals can also be specified.
Color 1 - 16	Specify the rendering color of the graph.	
	Default	Depends on the number (1 - 16) <sup>Note 3</sup>
	Modifying	Specify with the color selection combo box or directly enter from the keyboard.
	Available values	When directly enter from the keyboard: The value in decimal number/hexadecimal number (prefix "0x" is needed) or the color name (see <a href="#">"About the specification of the color"</a> ).

- Notes 1.** The input format is as follows (same as the input format of the Watch panel).  
 Note the following, however, when registering items.
- If the variable name of a structure, union, or array is registered, it cannot be graphed. For structures, unions, and arrays, the member name or element must be specified.
  - Immediate addresses are registered with a size of 1 byte.

Input Format	Value to be Acquired
Variable name of C language	Value of C language variable
<i>Variable-expression</i> [ <i>Variable-expression</i> ]	Element of array
<i>Variable-expression</i> .Member name	Member of structures/unions
<i>Variable-expression</i> ->Member name	Member of structures/unions that pointer designates
* <i>Variable-expression</i>	Value of pointer variable
CPU register name	Value of the CPU register
I/O register name [RH850][RX][V850]	I/O register value
SFR register name [R8C(Localised support)][RL78][78K0R][78K0]	SFR register value
Label, EQU symbol and immediate address	Values of label, EQU symbol and immediate address
Bit symbol	Bit symbol value

2. A judgment is automatically made in order of address expression, register name, IOR/SFR name, variable name by the debug tool. If a address expression is determined, it will be handled as "signed 1byte(8bits)".
3. The relationship between channel numbers and colors is as follows:

Number	Color	αRGB value in 32 bits	Number	Color	αRGB value in 32 bits
1	Red	0xC0FF0A4F	9	Yellowgreen	0xC0BEE02F
2	Green	0xC05BE416	10	Blueviolet	0xC05510FF
3	Blue	0xC0056DFF	11	Pink	0xC0FF97E4
4	Orange	0xC0FF541C	12	Brown	0xC0913A37
5	Lightblue	0xC04FC1FF	13	Ocher	0xC0C68E15
6	Purple	0xC0A932FF	14	Darkgreen	0xC0317F0C
7	Yellow	0xC0FFD91C	15	Darkbrown	0xC060493E
8	Redviolet	0xC0FF30A5	16	Grey	0xC072808E

**Remark About the specification of the color**

You can specify one of the following.

Note, however, that if you specify the alpha value indicating transparency, then it is assumed that "255(0xff)" (opaque) was specified.

- 32-bit value (assigning each 8-bit segment, from high to low, to the αRGB values)  
 Example: 0xC0FF0A4F
- 24-bit value (assigning each 8-bit segment, from high to low, to the RGB values)  
 Example: 0xFF0A4F
- Four eight-bit values, delimited by commas (assigning values from left to right to αRGB)  
 Example: 192, 255, 10, 79

- Three eight-bit values, delimited by commas (assigning values from left to right to RGB)

Example: 255, 10, 79

- Using general English color words (case insensitive)

Example: Blue

### Function List panel

This panel is used to display the acquired function information.

The following types of function are displayed in this panel.

- Global function
- Static function
- Member function (only when C++ source files are subject to analysis)

Note, however, that the function information in the files not to be analyzed is not displayed in this panel (see "1.1.1 What is analyzed").

See "2.1 Overview" for details on how to display the function information.

#### Cautions 1. [E20[RX]]

In the debug tool's Property panel, when the [Trace data type] property in the [Trace] category on the [Debug Tool Settings] tab is set to [Branch], the acquisition time of the **Dynamic analysis information** by trace data will increase because complement of instructions between branch instructions is performed (the processing time is proportional to the set value of the [Trace memory size[MByte]] property in the same category above).

#### 2. [CC-RX]

When C++ source files are subject to analysis, also see "3.6 When CC-RX (C++ Source Files) Is Used".

#### 3. [CC-RX][CX][NC30(Localised support)]

Unused static functions deleted via compiler optimization cannot be displayed in the panel.

#### 4. [CA850][CA78K0R][CA78K0]

When a clean is executed in the build tool, the contents currently being displayed in this panel will be cleared.

#### 5. [CA78K0R][CA78K0]

If a source file with the same name exists in a project, the build tool rewrites the cross reference information of them. Therefore, the information of other than the source file that has been compiled last in files with the same name cannot be acquired.

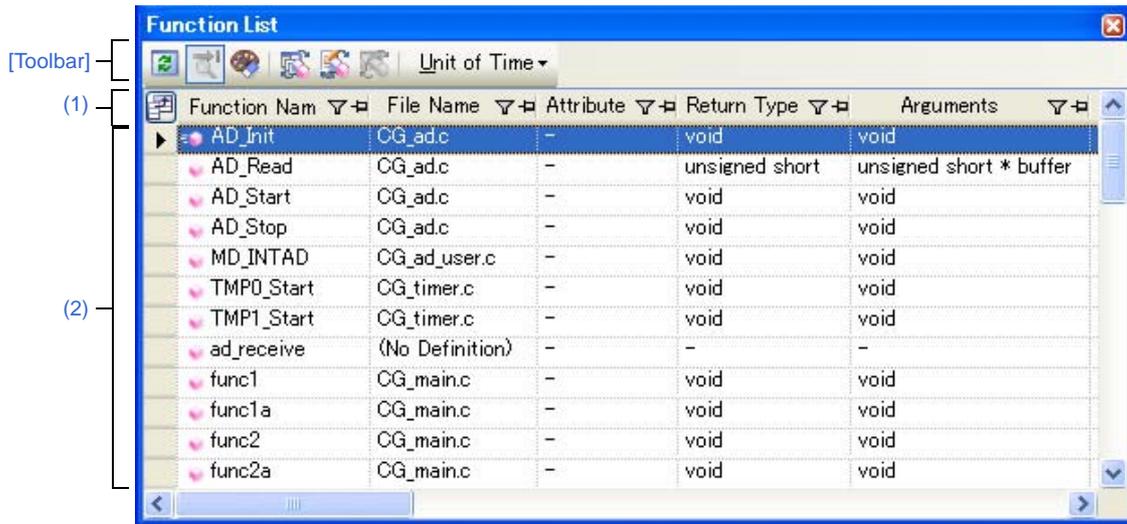
**Remarks 1.** You can set filters for displaying function information on this panel (see "2.6.5 Filter the analysis information").

2. This panel can be zoomed in and out by  in the tool bar, or by moving the mouse wheel forward or backward while holding down the [Ctrl] key.

#### 3. [RH850]

See "(a) Trace function" and "(c) Coverage function [IECUBE][IECUBE2][E20[RX]][Simulator]" for details on the relationship between the **Dynamic analysis information** in this panel and PE<sub>n</sub> currently being selected in the Debug Manager panel.

Figure A-6. Function List Panel



The following items are explained here.

- [How to open]
- [Description of each area]
- [Toolbar]
- [[File] menu (Function List panel-dedicated items)]
- [[Edit] menu (Function List panel-dedicated items)]
- [Context menu]

**[How to open]**

- On the toolbar in the **Main window**, click the  button.
- From the [View] menu, select [Program Analyzer] >> [Function List].

**[Description of each area]**

**(1) Header area**

**(a) Column name**

The name of the item of the acquired function information is displayed.

Marks (icons) that are displayed at the name of the item indicate as follows:

Mark (Icon)	Meaning
	Indicates whether there is a sort order setting (see "2.6.4 Sort the analysis information").
	Indicates whether there is a filter display setting (see "2.6.5 Filter the analysis information").
	Indicates whether display is locked (see "2.6.3 Lock the specific column to display").
	Indicates that a message relating to information about this item has been output to the <b>Output panel</b> . Hover the mouse over it to display a pop-up with the last message to be output.

**Remark** Columns to display can be customized via mouse operations in this area.

- Set the columns to display
- Change the order of the columns to display
- Lock the specific column to display

(b) Button

	Opens the <a href="#">Column Chooser dialog box</a> in order to sort and show/hide the items (columns) displayed in the panel, and return customized settings to the defaults (see " <a href="#">2.6 Customize Display Method</a> ").
---	---

(2) Information area

This area displays the acquired function information.

The analyze tool acquires two types of function information: [Static analysis information](#) and [Dynamic analysis information](#). The timing when each type of information can be displays differs (see "[1.1.2 Types of analysis information](#)").

When the active project is changed while displaying the function information, the function information of the target project is displayed. Note, however, that if the cross reference information has not been generated in the project or the analyze tool is not supported by the project, nothing is displayed on this panel.

The value of the information that has been changed because of the execution of a program is shown highlighted (the color depends on the configuration in the [General - Font and Color] category of the Option dialog box). To reset the highlighting, click the  button on the toolbar.

The items and details acquired as function information are as follows:

Item	Type	Contents
Function Name	Static analysis	<p>Displays the name of the global function, the file-internal static function or the member function (function that is defined with class type) that is defined/referenced in the C/C++ source file.</p> <p>The maximum display characters is as follows:</p> <ul style="list-style-type: none"> <li>- [CC-RH][CC-RX]: 1024 characters</li> <li>- [CA850]: 1022 characters</li> <li>- [CX]: 2046 characters</li> <li>- [NC30(Localised support)]:1024 characters</li> <li>- [CA78K0R][CA78K0]: 249 characters</li> </ul> <p>The meanings of an icon displayed is as follows:</p> <ul style="list-style-type: none"> <li>-  : Function</li> <li>-  : Member function [CC-RX]</li> <li>-  : Template function [CC-RX]</li> </ul> <p><b>[CC-RX]</b></p> <p>If the function is a const member function/volatile member function, then "const"/"volatile" is displayed immediately after the function name.</p>
Class Name <b>[CC-RX]</b>	Static analysis	<p>Displays the name of the class to which the function belongs.</p> <p>If it is a template class, then the number of template arguments are also displayed in the format of "<i>class name</i>&lt;T: <i>the number of template arguments</i>&gt;".</p> <p>Note, however, that if the function is not a member function, or if analysis was not completed, then this will be blank.</p>
Namespace <b>[CC-RX]</b>	Static analysis	<p>Displays the name of the namespace to which the function belongs.</p> <p>If the namespace is nested, then it is displayed in the format of "<i>namespace name::namespace name</i>". If it is an anonymous namespace, then "&lt;unnamed&gt;" is displayed.</p> <p>Note, however, that if analysis was not completed, then this will be blank.</p>

Item	Type	Contents
File Name	Static analysis	Displays the file name (without path) of the C/C++ source file where the function is defined. Note, however, that if it is not defined in a C/C++ source file in project, or if analysis was not completed, then this will display "(No Definition)". If it is defined in a header file, then the name of the header file is displayed.
File Pass <sup>Note 1</sup>	Static analysis	Displays the absolute path of the C/C++ source file where the function is defined. Note, however, that if it is not defined in a C/C++ source file in project, or if analysis was not completed, then this will be blank. If it is defined in a header file, then the absolute path of the header file is displayed.
PM Information [RH850] PE Information [V850E2]	Static analysis	Displays the following information about PE in which the function is executed. - In PE $n$ : PM $n$ /PE $n$ - In common PE: Common - Unknown: - Note that this item appears only when the selected microcontroller version supports multi-core.
Import <sup>Note 1</sup>	Static analysis	Displays the following information about the source of the function information. - If it was acquired from inside an active project "Original" is displayed. - If it was acquired from an import file The names of all import files are displayed. - If it was acquired from inside an active project and an import file "Original" and the names of all import files are displayed. See "2.12 Import or Export Information File" for details on the import function.
Access Specifier [CC-RX]	Static analysis	Displays the access specifier being declared to the member function. Note, however, that this will display "-" if analysis is not complete. The following access specifiers can be displayed. public, private, protected
Attribute	Static analysis	Displays the function's symbol attributes/symbol-modifier attributes. If there are multiple attributes, then they are displayed separated by commas (","). Note, however, that this will display "-" if analysis is not complete. The following attributes can be displayed. - [CC-RH][CX] static, interrupt, inline - [CC-RX] static, interrupt, inline, template, virtual, abstract - [CA850] static - [NC30(Localised support)] static, interrupt, inline - [CA78K0R] static, callt, interrupt, near, far, rtos task, rtos interrupt - [CA78K0] static, callt, callf, noauto, norec, interrupt, bank, rtos task, rtos interrupt

Item	Type	Contents
Return Type	Static analysis	<p>Displays the return type of the function<sup>Note 2</sup>.</p> <p>Note, however, that this will display "-" if analysis is not complete.</p> <p>The maximum number of pointers that can be displayed is as follows:</p> <ul style="list-style-type: none"> <li>- [CC-RH][CC-RX]: No limit</li> <li>- [CA850]: 6</li> <li>- [CX]: 8</li> <li>- [NC30(Localised support)]: No limit</li> <li>- [CA78K0R][CA78K0]: 7</li> </ul>
Arguments Count <sup>Note 1</sup>	Static analysis	<p>Displays the number of arguments of the function in decimal number notation.</p> <p>If the function has a variable argument parameter, then the number of arguments defined at the location where the function is defined is displayed.</p> <p>Note, however, that this will display "-" if analysis is not complete.</p>
Arguments	Static analysis	<p>Displays the argument type and formal argument name of the function<sup>Note 2</sup>.</p> <p>If there are multiple arguments, then they are displayed separated by commas (",").</p> <p>If the function has a variable argument parameter, then the argument type and argument name defined at the location where the function is defined are displayed.</p> <p>If there are no arguments, then "void" is displayed.</p> <p>Note, however, that this will display "-" if analysis is not complete.</p> <p>The maximum number of pointers that can be displayed is as follows (the first dimension of an array is handled as a pointer):</p> <ul style="list-style-type: none"> <li>- [CC-RH][CC-RX]: No limit</li> <li>- [CA850]: 6</li> <li>- [CX]: 8</li> <li>- [NC30(Localised support)]: No limit</li> <li>- [CA78K0R][CA78K0]: 8</li> </ul>
Code Size[Bytes]	Static analysis	<p>Displays the function's code size in decimal number notation.</p> <p>Note, however, that this will display "-" if analysis is not complete.</p>
Stack Size[Bytes] [V850] [RL78] [78K0R] [78K0]	Static analysis	<p>Displays the function's stack size in decimal number notation.</p> <p>Note, however, that this will display "-" if analysis is not complete.</p> <p><b>[CA78K0R][CA78K0]</b></p> <p>The value displayed here is stack size secured by the compiler in the first function or first basic block. For this reason, it will be different from the stack size displayed by the stack usage tracer. It also does not include the stack size used by CALL/PUSH/POP instruction inside the function.</p>
Start Address	Static analysis	<p>Displays the function's start address in hexadecimal number notation. The number of digits to display is equivalent to the maximum address value of the selected microcontroller.</p> <p>Note, however, that this will display "-" if analysis is not complete.</p> <p><b>[NC30(Localised support)]</b></p> <p>Since the start address of a system library function cannot be acquired, this will always display "-".</p>
End Address <sup>Note 1</sup>	Static analysis	<p>Displays the end address of the function aligned in ROM in hexadecimal number notation. The number of digits to display is equivalent to the maximum address value of the selected microcontroller.</p> <p>Note, however, that this will display "-" if analysis is not complete.</p>

Item	Type	Contents
Reference Count	Static analysis	<p>Displays the number of times the function is referenced in the program, in decimal number notation. Prototype declarations are also counted as references.</p> <p>Code in C/C++ source files that is referenced but is eliminated by the preprocessor during compilation (e.g. via "#if" or "#ifdef" statements) is not included in the number of references (it is also not output by the search results from [Find All References] in the context menu).</p> <p>Note, however, that this will display "-" if analysis is not complete.</p> <p><b>[CC-RH][CC-RX][CX][NC30(Localised support)]</b></p> <p>References to functions via assignment to function pointers are not included in the reference count.</p> <p><b>[CA850][CA78K0R][CA78K0]</b></p> <p>If a prototype declaration of function C is placed or function C is referenced via assignment to the pointer to the function somewhere between the definitions of functions A and B, function C is counted as a function referenced by function A.</p> <p><b>[CA78K0R][CA78K0]</b></p> <p>The function names included in "#pragma directive" are also counted as references.</p>
Execution Count <b>[Full-spec emulator]</b> <b>[IECUBE]</b> <b>[IECUBE2]</b> <b>[E1/E20[RX]]</b> <b>[E1/E20[RL78]]</b> <b>[EZ Emulator[RL78]]</b> <b>[Simulator]</b>	Dynamic analysis	<p>Displays the number of times the function was executed (called) as a result of program execution, in decimal number notation<sup>Note 3</sup>.</p> <p>This function counts the number of times that the instruction located at the address allocated to the function label is executed. For this reason, an invalid value may be displayed if the measurement starts midway through the execution of a function.</p> <p>Note, however, that this will display "-" if the debug tool's <a href="#">Trace function</a> is disabled, or analysis is not complete.</p> <p><b>[E1/E20[RL78]][EZ Emulator[RL78]]</b></p> <p>The execution count is calculated according to the following conditions, because only the branch origin address can be acquired from the trace data.</p> <p>For this reason, the execution count of a function whose size is unknown or a function which is being called from an interrupt processing cannot be calculated.</p> <ul style="list-style-type: none"> <li>- If the result of disassembling the branch instruction is a CALL/CALLT instruction, then the execution count of the function containing the branch origin address of the next trace frame is added.</li> </ul>
Execution Time[ <i>unit</i> ] <b>[Full-spec emulator]</b> <b>[IECUBE[V850]]</b> <b>[IECUBE[RL78]]</b> <b>[IECUBE[78K0R]]</b> <b>[IECUBE2]</b> <b>[Simulator]</b>	Dynamic analysis	<p>Displays the execution time of the function (the time executing the code in the function body, not including subroutines)<sup>Note 3, 4</sup>.</p> <p>The unit of time can be changed by selecting [Unit of Time] from the <a href="#">[Toolbar]</a>, or from the <a href="#">Property panel's [Settings] tab</a>, in the [General] category, changing the [Unit of time] property (see the [Unit of time] property in the <a href="#">Property panel</a> for details on the time-display format).</p> <p>Note, however, that this will display "-" if the debug tool's <a href="#">Trace function</a> is disabled, or analysis is not complete.</p>
Execution Time(Percentage)[%] <b>[Full-spec emulator]</b> <b>[IECUBE[V850]]</b> <b>[IECUBE[RL78]]</b> <b>[IECUBE[78K0R]]</b> <b>[IECUBE2]</b> <b>[Simulator]</b>	Dynamic analysis	<p>Displays the proportion of total execution time (range that could be obtained as trace data) taken by the execution time of the target function, rounded to the nearest two decimal places, in the range 0.00 to 100.00<sup>Note 3, 4</sup>.</p> <p>The level of shading of the cell's background color indicates the proportion.</p> <p>Note, however, that this will display "-" if the debug tool's <a href="#">Trace function</a> is disabled, or analysis is not complete.</p>

Item	Type	Contents
Execution Time(Average)[unit] <b>[Full-spec emulator]</b> <b>[IECUBE[V850]]</b> <b>[IECUBE[RL78]]</b> <b>[IECUBE[78K0R]]</b> <b>[IECUBE2]</b> <b>[Simulator]</b>	Dynamic analysis	Displays the average execution time of the function (" <i>execution time</i> " / " <i>number of executions</i> ") <sup>Note 3, 4</sup> . The results of calculation are displayed rounded to the nearest nanosecond. The unit of time can be changed by selecting [Unit of Time] from the [Toolbar], or from the Property panel's [Settings] tab, in the [General] category, changing the [Unit of time] property. The unit of time can be changed by selecting [Unit of Time] from the [Toolbar], or from the Property panel's [Settings] tab, in the [General] category, changing the [Unit of time] property (see the [Unit of time] property in the Property panel for details on the time-display format). Note, however, that this will display "-" if the debug tool's Trace function is disabled, or analysis is not complete.
Code Coverage[%] <b>[IECUBE]</b> <b>[IECUBE2]</b> <b>[E20[RX]]</b> <sup>Note 5</sup> <b>[Simulator]</b>	Dynamic analysis	Displays the code coverage rates of the function (C0: " <i>number of bytes of code executed in the address range</i> " / " <i>function's code size</i> " x 100) (see "3.2 Coverage Results"). The level of shading of the cell's background color indicates the code coverage. Note, however, that this will display "-" if the debug tool's Coverage function [IECUBE][IECUBE2][E20[RX]][Simulator] is disabled, or analysis is not complete.

**Notes 1.** This item does not appear by default.

See "2.6.1 Set the columns to display" for details on how to display this item.

**2. [CC-RH][CC-RX][CA850][CA78K0R][CA78K0]**

If including an alternative name by using "#define" or "typedef", then not the alternative name but the original type is displayed.

**[NC30(Localised support)]**

If including an alternative name by using "#define", then not the alternative name but the original type is displayed. If including an alternative name by using "typedef", then the alternative name is displayed.

**[CX]**

If including an alternative name by using "typedef", the following types are displayed.

Original Type of "typedef"	Type Displayed
- unsigned long - unsigned int	unsigned long
- signed long - signed int - long - int	long
- unsigned short	unsigned short
- signed short - short	short
- unsigned char - char (when "-Xchar=unsigned" option is specified)	unsigned char
- signed char - char (when "-Xchar=unsigned" option is not specified)	char

- Set the [Accumulate result of analysis] property in the [General] category on the [Settings] tab of the Property panel to [Yes] to display the cumulative value for each program execution.

4. The execution time does not include the execution time of the runtime libraries provided by the compiler. If the current setting of the debug tool does not allow clearing of the trace memory before tracing starts, the execution time value will be incorrect.
5. **[E20[RX]]**  
This item appears only when the selected microcontroller supports the coverage function and also CubeSuite+ is connected to the debug tool.

In the bottom of the [Function Name] item, the following information for each file is shown as [\*Total\*]. Note that "(No Definition)" displayed in the [File Name] item is treated as one file.

**Figure A-7. [\*Total\*] Display (Function List Panel)**

Function Name	File Name	Code Size[Bytes]	Reference Count	Execution Count
*Total*	CG_timer.c	412	12	2
*Total*	(No Definition)	0	4	1
*Total*	CG_main.c	114	8	0

Item	Contents
File Name	The target file name
File Pass <sup>Note</sup>	File pass of the target file
Code Size[Bytes]	Total of the function's code size in the target file
Reference Count	Total number of times functions are referenced in the target file
Execution Count	Total number of times functions were executed in the target file
Execution Time[ <i>unit</i> ]	Total of the execution time of functions in the target file
Execution Time(Percentage)[%]	The proportion of total execution time (range that could be obtained as trace data) taken by the execution time of functions in the target file
Code Coverage[%]	The code coverage rates of functions in the target file

**Note** This item does not appear by default.

See "2.6.1 Set the columns to display" for details on how to display this item.

- Cautions 1. Items of the Dynamic analysis information do not appear while disconnecting from the debug tool (default).**
2. **[IECUBE[V850]][IECUBE2]**  
If step execution (step in and step over execution) was performed, the values output in the time tags of the trace data will be invalid.  
As a result, the [Execution Time[*unit*]] / [Execution Time(Percentage)[%]] / [Execution Time(Average)[*unit*]] items will be invalid.  
Moreover, the [Code Coverage[%]] item does not appear when the coverage board is not mounted on IECUBE to be used.
  3. **[IECUBE[RL78]][IECUBE[78K0R]]**  
"0" is output as the first trace-data time tag during program execution.  
For this reason, if the user repeatedly starts and stops execution, or performs step execution, the [Execution Time[*unit*]] / [Execution Time(Percentage)[%]] / [Execution Time(Average)[*unit*]] items will be invalid.

## 4. [IECUBE[78K0]]

The following items are not supported.

[Execution Time[*unit*]] / [Execution Time(Percentage)[%]] / [Execution Time(Average)[*unit*]]

## 5. Values of the following items for the system library functions cannot be acquired.

[Return Type]/[Arguments Count]/[Arguments]/[Code Size[Bytes]]/[Stack Size[Bytes]]/[End Address]/[Execution Time[*unit*]]/[Execution Time(Percentage)[%]]/[Execution Time(Average)[*unit*]]/[Code Coverage[%]]

**Remarks 1.** The following will be displayed if you define a static function in a header file, and include it from more than one source file.

- [CC-RH][CC-RX][CX][NC30(Localised support)][CA78K0R][CA78K0]

Information for the header file and all source files is displayed on one line.

- [CA850]

Information for the header file is displayed on one line, and information for all source files is displayed on another line.

## 2. [CC-RX]

The display contents of the line for the definition of a template function and the line for the function using a template function are as follows:

```
//
// Function using the template function
//
int templatet_use(void)
{
    short    result = 0;
    short    s = 100;
    char     c = 200;

    result += template_func(s, c);

    return result;
}

//
// Definition of the template function
//
template <typename T1, typename T2> T1 template_func(T1 t1, T2 t2)
{
    T1      result = 10;
    result += t1 + t2;
    return result;
}
```

**Examples 1.** Display contents for the line for the definition of the template function

[Return Type] : -  
 [Arguments]: -  
 [Function Name]: Template\_func<T:2>

**2.** Display contents for the line for the function using the template function

[Return Type]: short  
 [Arguments]: short t1, char t2  
 [Function Name]: Template\_func<T:2>

3. The display of each type of information can be customized as follows:
  - [Sort the analysis information](#)
  - [Filter the analysis information](#)
4. The current row mark ( ▶ ) on the left edge of this area indicates that the column in question is the current row.  
 The following operations can be performed on the current row:
  - [Jump to Defined Location](#)
  - [Set Break Events](#)
  - [Display List of Referencing Location](#)

**[Toolbar]**

The toolbar provides the following items and functions.

	Acquires the latest data from the debug tool, and updates the contents of this panel.
	Displays the latest information automatically by acquiring the information each time program execution stops. Note that this item is disabled when the [Refresh at program stop] property in the [General] category on the [Settings] tab of the <a href="#">Property panel</a> is set to other than [Specify Individually] (this button is locked in accordance with the setting on the <a href="#">Property panel</a> ).
	Resets highlighting of values that have been changed by executing a program. Note that this item is disabled during execution of a program.
	Displays only the information for functions in a file(s)/category(s) currently selected in the <a href="#">Project Tree panel</a> (see "(a) <a href="#">Linking with the Project Tree panel</a> ").
	Displays only the information for functions starting with the word at the caret position in the Editor panel (see "(b) <a href="#">Linking with the Editor panel</a> ").
 [RH850] [V850E2]	Displays only the information for functions in the common region and PE currently selected in the Debug Manager panel (see "(c) <a href="#">Linking with the Debug Manager panel [RH850][V850E2]</a> "). Note that this item is disabled when the selected microcontroller version does not support multi-core. or while disconnecting from the debug tool.
Unit of Time	The following cascade menus are displayed to specify the time unit for the analyze tool. The setting of the [Unit of time] property in the [General] category on the [Settings] tab of the <a href="#">Property panel</a> is specified by default. The time unit set in this toolbar is reflected in the <a href="#">Property panel</a> .
H:M:S	Displays the time as "hours, minutes (0 - 59), seconds (0 - 59)".
Second	Displays the time in second units. The value is rounded to three decimal places.
Millisecond	Displays the time in millisecond units. The value is rounded to three decimal places.
Microsecond	Displays the time in microsecond units. The value is rounded to three decimal places.
Nanosecond	Displays the time in nanosecond units. The value is rounded to integer number.

**[[File] menu (Function List panel-dedicated items)]**

The [File] menu for this panel provides the following items and functions.

Save Function List Data	Overwrites the contents of this panel to the previously saved file (see "2.14 <a href="#">Save Analysis Information</a> "). Note that when the file has never been saved or the file is write disabled, the same operation is applied as the selection in [Save Function List Data As...].
-------------------------	---

Save Function List Data As...	Opens the <a href="#">Save As dialog box</a> to newly save the contents of this panel to the specified text file (see " <a href="#">2.14 Save Analysis Information</a> ").
Print...	Opens the Windows dialog box to print the contents currently being displayed in this panel.

### [[Edit] menu (Function List panel-dedicated items)]

The [Edit] menu for this panel provides the following items and functions.

Copy	Copies the contents of the selected column (multiple selections possible) to the clipboard as character strings separated by tabs. Note that the contents of the clipboard cannot be pasted to this panel.
Select All	Selects all the columns being displayed in this panel.

### [Context menu]

The context menu displayed by right-clicking on this panel provides the following items and functions.

Find All References	Displays a list of the locations referencing the function(s) of the selected row(s), in the <a href="#">Output panel's [Find References] tab</a> (see " <a href="#">2.11 Display List of Referencing Location</a> ").
Set Break to Function	Sets a breakpoint at the first line of the function at the current row (the first executable line in the target function) (see " <a href="#">2.9.1 Set a breakpoint to a function</a> "). Note that this item is disabled while disconnecting from the debug tool.
Jump to Source	Opens the Editor panel and displays the source file in which the function of the current row is defined (see " <a href="#">2.7 Jump to Defined Location</a> ").
Jump to Disassemble	Opens the Disassemble panel (Disassemble1) and displays the disassemble data corresponding to the start address of the function of the current row (see " <a href="#">2.7 Jump to Defined Location</a> "). Note that this item is disabled while disconnecting from the debug tool.
Jump to Memory	Opens the Memory panel (Memory1) and displays the memory list corresponding to the start address of the function of the current row (see " <a href="#">2.7 Jump to Defined Location</a> "). Note that this item is disabled while disconnecting from the debug tool.
Copy	Copies the contents of the selected row(s) to the clipboard as character strings separated by tabs. Note that the contents of the clipboard cannot be pasted to this panel.

### Variable List panel

This panel is used to display the acquired variable information.

The following types of variable are displayed in this panel.

- Global variable
- Static variable inside a file
- Static variable inside a function
- IOR [V850]
- SFR [R8C(Localised support)][RL78][78K0R][78K0]
- Class variable (only when C++ source files are subject to analysis)

Note, however, that the variable information in the files not to be analyzed is not displayed in this panel (see "1.1.1 What is analyzed").

See "2.1 Overview" for details on how to display the variable information.

#### Cautions 1. [E20[RX]]

In the debug tool's Property panel, when the [Trace data type] property in the [Trace] category on the [Debug Tool Settings] tab is set to [Branch], the acquisition time of the **Dynamic analysis information** by trace data will increase because complement of instructions between branch instructions is performed. If you focus only on variables, we recommend selecting [Data access] in this property.

#### 2. [CC-RX]

When C++ source files are subject to analysis, also see "3.6 When CC-RX (C++ Source Files) Is Used".

#### 3. [CC-RX][CX][NC30(Localised support)]

Unused variables deleted via compiler optimization are not displayed.

#### 4. [CA850]

If there are assembler instructions coded between "#pragma asm" and "#pragma endasm", then the registers and instructions coded in that location will be displayed as variables.

#### 5. [CA850][CA78K0R][CA78K0]

When a clean is executed in the build tool, the contents currently being displayed in this panel will be cleared.

#### 6. [CA78K0R][CA78K0]

If a source file with the same name exists in a project, the build tool rewrites the cross reference information of them. Therefore, the information of other than the source file that has been compiled last in files with the same name cannot be acquired.

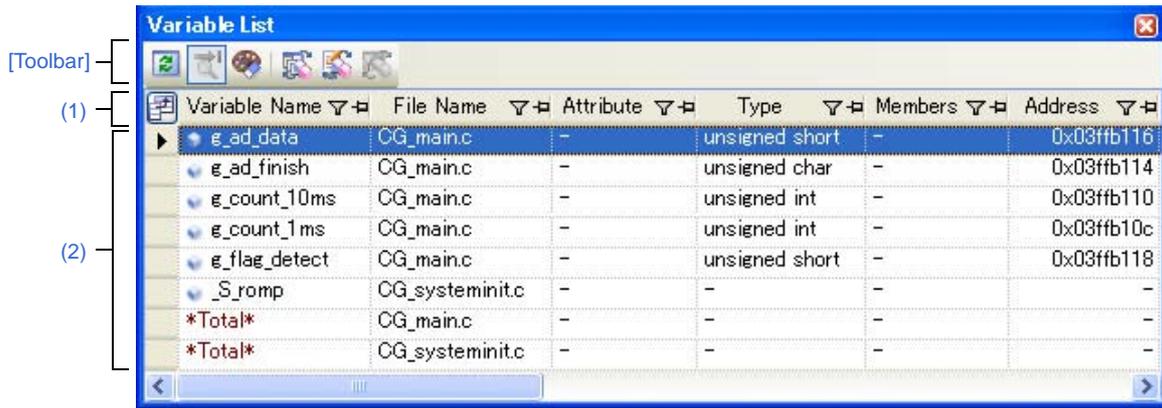
**Remarks 1.** You can set filters for displaying variable information on this panel (see "2.6.5 Filter the analysis information").

2. This panel can be zoomed in and out by  in the tool bar, or by moving the mouse wheel forward or backward while holding down the [Ctrl] key.

#### 3. [RH850]

See "(a) Trace function" for details on the relationship between the **Dynamic analysis information** in this panel and PE $n$  currently being selected in the Debug Manager panel.

Figure A-8. Variable List Panel



The following items are explained here.

- [How to open]
- [Description of each area]
- [Toolbar]
- [[File] menu (Variable List panel-dedicated items)]
- [[Edit] menu (Variable List panel-dedicated items)]
- [Context menu]

**[How to open]**

- On the toolbar in the **Main window**, click the  button.
- From the [View] menu, select [Program Analyzer] >> [Variable List].

**[Description of each area]**

**(1) Header area**

**(a) Column name**

The name of the item of the acquired variable information is displayed.

Marks (icons) that are displayed at the name of the item indicate as follows:

Mark (Icon)	Meaning
 / 	Indicates whether there is a sort order setting (see "2.6.4 Sort the analysis information").
 / 	Indicates whether there is a filter display setting (see "2.6.5 Filter the analysis information").
 / 	Indicates whether display is locked (see "2.6.3 Lock the specific column to display").
	Indicates that a message relating to information about this item has been output to the <b>Output panel</b> . Hover the mouse over it to display a pop-up with the last message to be output.

**Remark** Columns to display can be customized via mouse operations in this area.

- [Set the columns to display](#)
- [Change the order of the columns to display](#)
- [Lock the specific column to display](#)

(b) Button

	Opens the <a href="#">Column Chooser dialog box</a> in order to sort and show/hide the items (columns) displayed in the panel, and return customized settings to the defaults (see " <a href="#">2.6 Customize Display Method</a> ").
---	---

(2) Information area

This area displays the acquired variable information.

The analyze tool acquires two types of function information: [Static analysis information](#) and [Dynamic analysis information](#). The timing when each type of information can be displays differs (see "[1.1.2 Types of analysis information](#)").

When the active project is changed while displaying the variable information, the variable information of the target project is displayed. Note, however, that if the cross reference information has not been generated in the project or the analyze tool is not supported by the project, nothing is displayed on this panel.

The value of the information that has been changed because of the execution of a program is shown highlighted (the color depends on the configuration in the [General - Font and Color] category of the Option dialog box). To reset the highlighting, click the  button on the toolbar.

The items and details acquired as variable information are as follows:

Item	Type	Contents
Variable Name	Static analysis	<p>Displays the name of the global variable, the static variable inside a file, the static variable inside a function or the class variable (static member variable of a class type) that is defined/referenced in the C/C++ source file.</p> <p>Moreover, displays the name of SFR/IOR being used in the program when the [Display the SFR/IOR] property in the [General] category on the <a href="#">[Settings] tab</a> of the <a href="#">Property panel</a> is set to [Yes] (except for [CC-RH][CC-RX][NC30(Localised support)]).</p> <p>The maximum display characters is as follows:</p> <ul style="list-style-type: none"> <li>- [CC-RH][CC-RX]: 1024 characters</li> <li>- [CA850]: 1022 characters</li> <li>- [CX]: 2046 characters</li> <li>- [NC30(Localised support)]: 1024 characters</li> <li>- [CA78K0R][CA78K0]: 249 characters</li> </ul> <p>The meanings of icons displayed are as follows:</p> <ul style="list-style-type: none"> <li>-  : Variable</li> <li>-  : Member variable <b>[CC-RX]</b></li> <li>-  : IOR <b>[V850]/SFR [RL78][78K0R][78K0]</b></li> </ul> <p><b>[CC-RX]</b></p> <p>If an anonymous union is used, then this will display "&lt;unnamed_N&gt;" as the variable name (N: A number from 1 automatically provided in appearance order).</p>
Class Name <b>[CC-RX]</b>	Static analysis	<p>Displays the name of the class to which the variable belongs.</p> <p>If it is a template class, then the number of template arguments are also displayed in the format of "<i>class name</i>&lt;T: <i>the number of template arguments</i>&gt;".</p> <p>Note, however, that if the variable is not a member variable, or if analysis was not completed, then this will be blank.</p>
Namespace <b>[CC-RX]</b>	Static analysis	<p>Displays the name of the namespace to which the variable belongs.</p> <p>If the namespace is nested, then it is displayed in the format of "<i>namespace name::namespace name</i>". If it is an anonymous namespace, then "&lt;unnamed&gt;" is displayed.</p> <p>Note, however, that if analysis was not completed, then this will be blank.</p>

Item	Type	Contents
File Name	Static analysis	<p>Displays the file name (without path) of the C/C++ source file where the variable is defined<sup>Note 1</sup>.</p> <p>Note, however, that if it is not defined in a C/C++ source file in project, or if analysis was not completed, then this will display "(No Definition)".</p> <p>If it is defined in a header file, then the name of the header file is displayed.</p> <p><b>[CA78K0R][CA78K0]</b></p> <p>If a line with a variable described in "extern declaration" and a line with the variable definition exist in the same file, the information of the variable definition cannot be acquired. Consequently, in this case, this item will display "(No Definition)".</p>
Function Name <sup>Note 1</sup>	Static analysis	<p>Displays the name of the function where the variable is defined.</p> <p>Note, however, this will be blank for other than a static variable inside a function.</p> <p><b>[CC-RX]</b></p> <p>In addition to the above, the argument types of the function are also displayed in parentheses "()". If the function is a const member function/volatile member function, then "const"/"volatile" is displayed immediately after the function name.</p>
File Path <sup>Note 1</sup>	Static analysis	<p>Displays the absolute path of the C/C++ source file where the variable is defined.</p> <p>Note, however, that if it is not defined in a C/C++ source file in project, or if analysis was not completed, then this will be blank.</p> <p>If it is defined in a header file, then the absolute path of the header file is displayed.</p>
PM Information <b>[RH850]</b> PE Information <b>[V850E2]</b>	Static analysis	<p>Displays the following information about PE from which the variable can be accessed.</p> <ul style="list-style-type: none"> <li>- From PE<math>n</math>: PM<math>n</math>/PE<math>n</math></li> <li>- From common PE: Common</li> <li>- Unknown: -</li> </ul> <p>Note that this item appears only when the selected microcontroller version supports multi-core.</p>
Import <sup>Note 1</sup>	Static analysis	<p>Displays the following information about the source of the variable information.</p> <ul style="list-style-type: none"> <li>- If it was acquired from inside an active project "Original" is displayed.</li> <li>- If it was acquired from an import file The names of all import files are displayed.</li> <li>- If it was acquired from inside an active project and an import file "Original" and the names of all import files are displayed.</li> </ul> <p>See "2.12 Import or Export Information File" for details on the import function.</p>
Access Specifier <b>[CC-RX]</b>	Static analysis	<p>Displays the access specifier being declared to the member variable.</p> <p>Note, however, that this will display "-" if analysis is not complete.</p> <p>The following access specifiers can be displayed.</p> <p>public, private, protected</p>

Item	Type	Contents
Attribute	Static analysis	<p>Displays the variable's symbol attributes/symbol-modifier attributes.</p> <p>If there are multiple attributes, then they are displayed separated by commas (",").</p> <p>Note, however, that this will display "-" if analysis is not complete.</p> <p>The following attributes can be displayed.</p> <ul style="list-style-type: none"> <li>- [CC-RH][CX] static, ior, const, volatile</li> <li>- [CC-RX] static, const, volatile, restrict</li> <li>- [CA850] static</li> <li>- [NC30(Localised support)] static, const, volatile</li> <li>- [CA78K0R] static, const, volatile, sreg, rwsfr, rosfr, wosfr, near, far</li> <li>- [CA78K0] - static, const, volatile, sreg, rwsfr, rosfr, wosfr</li> </ul>
Type	Static analysis	<p>Displays the type of the variable<sup>Note 2</sup>.</p> <p>If including an alternative name by using "#define statement" or "typedef statement", then not the alternative name but the type is displayed.</p> <p>Note, however, that this will display "-" if analysis is not complete.</p> <p>The maximum number of pointers that can be displayed is as follows (up to 4 dimensions for an array can be displayed):</p> <ul style="list-style-type: none"> <li>- [CC-RH][CC-RX]: No limit</li> <li>- [CA850]: 6</li> <li>- [CX]: 8</li> <li>- [NC30(Localised support)]: No limit</li> <li>- [CA78K0R][CA78K0]: 8</li> </ul>
Members	Static analysis	<p>Displays the members of the structure/union.</p> <p>If there are multiple members, then they are displayed separated by commas (",").</p> <p>Note, however, that this will display "-" if other than structures/unions or analysis is not complete.</p> <p><b>[NC30(Localised support)]</b></p> <p>The members of an anonymous structure (structure without a tag) cannot be displayed.</p>
Address	Static analysis	<p>Displays the variable's allocated address in hexadecimal number notation.</p> <p>The number of digits to display is equivalent to the maximum address value of the selected microcontroller.</p> <p>Note, however, that this will display "-" if analysis is not complete.</p>
Size[Bytes]	Static analysis	<p>Displays the variable's size in decimal number notation.</p> <p>Note, however, that this will display "-" for bit variables or the like that cannot be displayed in bytes, or if analysis is not complete.</p> <p><b>[CC-RX(V1.xx.xx)][NC30(Localised support)]</b></p> <p>Variables that are defined only and not referenced will be eliminated by compiler optimization. For this reason, this will display "0" for such variables.</p>

Item	Type	Contents
Reference Count	Static analysis	<p>Displays the number of times the variable is referenced in the program, in decimal number notation.</p> <p>The location where the variable is defined is also counted. The totals for structures, unions, and arrays are displayed at the variable level (references to individual members and array elements are not shown).</p> <p>Code in C/C++ source files that is referenced but is eliminated by the preprocessor during compilation (e.g. via "#if" or "#ifdef" statements) is not included in the number of references (it is also not output by the search results from [Find All References] in the context menu).</p> <p>Note, however, that this will display "-" if analysis is not complete.</p> <p><b>[CC-RH][CC-RX][CX][NC30(Localised support)]</b></p> <p>In the definition of variables, a line with an assignment statement (e.g. "int variable = 10") is counted.</p> <p>"variable++;" is interpreted as "variable = variable + 1". Therefore, this variable is counted twice.</p>
Read Count [Full-spec emulator] [IECUBE] [IECUBE2] [E1/E20[RX]] [Simulator]	Dynamic analysis	<p>Displays the number of times the variable was read, in decimal number notation<sup>Note 3, 4</sup>.</p> <p>For structures and unions, reads are counted at the structure/union variable level (the number of reads at the individual member and element levels are not shown).</p> <p>Note, however, that this will display "-" if the debug tool's <a href="#">Trace function</a> is disabled, or if analysis is not complete.</p>
Write Count [Full-spec emulator] [IECUBE] [IECUBE2] [E1/E20[RX]] [Simulator]	Dynamic analysis	<p>Displays the number of times the variable was written, in decimal number notation<sup>Note 3, 4</sup>.</p> <p>For structures and unions, writes are counted at the structure/union variable level (the number of writes at the individual member and element levels are not shown).</p> <p>Note, however, that this will display "-" if the debug tool's <a href="#">Trace function</a> is disabled, or if analysis is not complete.</p>
Read/Write Count <sup>Note 1</sup> [Full-spec emulator] [IECUBE] [IECUBE2] [E1/E20[RX]] [Simulator]	Dynamic analysis	<p>Displays the number of times the variable was read/written, in decimal number notation<sup>Note 3, 4</sup>.</p> <p>For structures and unions, reads/writes are counted at the structure/union variable level (the number of reads/writes at the individual member and element levels are not shown).</p> <p>It may not be possible to analyze reads/writes of variables in segments where variables have been assigned to registers via compiler optimization. For this reason, reads/writes in such sections will not be counted.</p> <p>Note, however, that this will display "-" if the debug tool's <a href="#">Trace function</a> is disabled, or if analysis is not complete.</p>
Minimum Value [Full-spec emulator] [IECUBE] [IECUBE2] [E1/E20[RX]] [Simulator]	Dynamic analysis	<p>Displays the minimum measurement time from the results of program execution, in decimal number notation<sup>Note 3</sup>.</p> <p>Note, however, that this will display "-" if for bit type variables/boolean type variables/Bool type variables/structures/unions/arrays/pointer, if the debug tool's <a href="#">Trace function</a> is disabled, or if analysis is not complete.</p> <p><b>[CC-RH][CC-RX][CA850][CX]</b></p> <p>Only a variable/IOR less than or equal to 4 bytes can be displayed.</p> <p><b>[CA78K0R][CA78K0]</b></p> <p>Only a variable/SFR less than or equal to 2 bytes can be displayed.</p>

Item	Type	Contents
Maximum Value [Full-spec emulator] [IECUBE] [IECUBE2] [E1/E20[RX]] [Simulator]	Dynamic analysis	Displays the maximum measurement time from the results of program execution, in decimal number notation <sup>Note 3</sup> . Note, however, that this will display "-" if for bit type variables/boolean type variables/Bool type variables/structures/unions/arrays/pointer, if the debug tool's <a href="#">Trace function</a> is disabled, or if analysis is not complete. [CC-RH][CC-RX][CA850][CX] Only a variable/IOR less than or equal to 4 bytes can be displayed. [CA78K0R][CA78K0] Only a variable/SFR less than or equal to 2 bytes can be displayed.
Data Coverage[%] [IECUBE[RL78]] [IECUBE[78K0R]] [IECUBE[78K0]] [Simulator]	Dynamic analysis	Displays the data coverage rates of the variable (" <i>number of bytes accessed in the address range</i> " / " <i>variable size</i> " x 100) (see " <a href="#">3.2 Coverage Results</a> "). The level of shading of the cell's background color indicates the data coverage. Note, however, that this will display "-" if the debug tool's <a href="#">Coverage function</a> [IECUBE][IECUBE2][E20[RX]][Simulator] is disabled, or if analysis is not complete.

**Notes 1.** This item does not appear by default.

See "[2.6.1 Set the columns to display](#)" for details on how to display this item.

**2. [CC-RH][CC-RX][CA850][CA78K0R][CA78K0]**

If including an alternative name by using "#define" or "typedef", then not the alternative name but the original type is displayed.

**[NC30(Localised support)]**

If including an alternative name by using "#define", then not the alternative name but the original type is displayed. If including an alternative name by using "typedef", then the alternative name is displayed.

**[CX]**

If including an alternative name by using "typedef", the following types are displayed.

Original Type of "typedef"	Type Displayed
- unsigned long - unsigned int	unsigned long
- signed long - signed int - long - int	long
- unsigned short	unsigned short
- signed short - short	short
- unsigned char - char (when "-Xchar=unsigned" option is specified)	unsigned char
- signed char - char (when "-Xchar=unsigned" option is not specified)	signed char

**3.** Set the [Accumulate result of analysis] property in the [General] category on the [\[Settings\] tab](#) of the [Property panel](#) to [Yes] to display the cumulative value for each program execution.

**4.** The calculation is based on the acquired trace data. So for example, if one write to a 4-byte area was output in the trace data as the upper 2 bytes and the lower 2 bytes, then two times will be displayed.

In the bottom of the [Variable Name] item, the following information for each file is shown as [\*Total\*].

Note that "(No Definition)" displayed in the [File Name] item is treated as one file.

Figure A-9. [\*Total\*] Display (Variable List Panel)

Variable Name	File Name	Size[Bytes]	Reference Count	Read Count	Write Count
*Total*	CG_main.c	13	9	0	4
*Total*	CG_systeminit.c	0	1	0	0

Item	Contents
File Name	The target file name
File Pass <sup>Note</sup>	File pass of the target file
Size[Bytes]	Total of the variable's size in the target file
Reference Count	Total number of times variables are referenced in the target file
Read Count	Total number of times variables were read in the target file
Write Count	Total number of times variables were written in the target file
Read/Write Count	Total number of times variables were read/written in the target file
Data Coverage[%]	The data coverage rates of variables in the target file

**Note** This item does not appear by default.

See "2.6.1 Set the columns to display" for details on how to display this item.

- Cautions 1.** Items of the **Dynamic analysis information** do not appear while disconnecting from the debug tool (default).
- 2.** The [Read Count]/[Write Count]/[Read/Write Count]/[Data Coverage[%]] item for a bit type variable/boolean type variable/\_Bool type variable/bit field structure is measured by counting the number of accesses to the address that the variable is being allocated. Consequently, if bit type variables/boolean type variables/\_Bool type variables/bit field structures are allocated to the same address, these items above will display the same value.

**Remarks 1.** If static variables inside a function with the same name are declared in a function, they are handled as follows:

**[CC-RH][CC-RX][NC30(Localised support)]**

- [Members]/[Address]/[Size[Bytes]]/[Read Count]/[Write Count]/[Read/Write Count]/[Minimum Value]/[Maximum Value]/[Data Coverage[%]]

The information of the variable that is first declared in a function is displayed.

- Items other than listed above

The information of the variable that is first declared in a function is displayed.

**[CX]**

- [Type]/[Members]/[Address]/[Size[Bytes]]/[Read Count]/[Write Count]/[Read/Write Count]/[Minimum Value]/[Maximum Value]/[Data Coverage[%]]

The information of the variable that is first declared in a function is displayed.

- Items other than listed above

The information of the variable that is first declared in a function is displayed.

**[CA850][CA78K0R][CA78K0]**

- [Variable Name]/[File Name]/[Function Name]/[File Path]/[Import]

The information of the variable that is first declared in a function is displayed.

- [Reference Count]

All number of times the variable declared in a function is referenced is displayed.

- Items other than listed above  
The information of the variable that is last declared in a function is displayed.
- 2. The following will be displayed if you define a static variable in a header file, and include it from more than one source file.
  - **[CC-RH][CC-RX][CX][NC30(Localised support)][CA78K0R][CA78K0]**  
Information for the header file and all source files is displayed on one line.  
"-" will be displayed in the [Type] item (except **[CC-RH][CC-RX][NC30(Localised support)]**).
  - **[CA850]**  
Information for the header file is displayed on one line, and information for all source files is displayed on another line.
- 3. The display of each type of information can be customized as follows:
  - [Sort the analysis information](#)
  - [Filter the analysis information](#)
- 4. The current row mark ( ▶ ) on the left edge of this area indicates that the column in question is the current row.  
The following operations can be performed on the current row:
  - [Jump to Defined Location](#)
  - [Register Watch-Expressions](#)
  - [Display List of Referencing Location](#)
  - [Graph transitions in values](#)

**[Toolbar]**

The toolbar provides the following items and functions.

	Acquires the latest data from the debug tool, and updates the contents of this panel.
	Displays the latest information automatically by acquiring the information each time program execution stops. Note that this item is disabled when the [Refresh at program stop] property in the [General] category on the <a href="#">[Settings] tab</a> of the <a href="#">Property panel</a> is set to other than [Specify Individually] (this button is locked in accordance with the setting on the <a href="#">Property panel</a> ).
	Resets highlighting of values that have been changed by executing a program. Note that this item is disabled during execution of a program.
	Displays only the information for variables in a file(s)/category(s) currently selected in the <a href="#">Project Tree panel</a> (see "(a) <a href="#">Linking with the Project Tree panel</a> ").
	Displays only the information for variables starting with the word at the caret position in the Editor panel (see "(b) <a href="#">Linking with the Editor panel</a> ").
 <b>[RH850]</b> <b>[V850E2]</b>	Displays only the information for variables in the common region and PE <i>n</i> currently selected in the Debug Manager panel (see "(c) <a href="#">Linking with the Debug Manager panel [RH850][V850E2]</a> "). Note that this item is disabled when the selected microcontroller version does not support multi-core. or while disconnecting from the debug tool.

**[[File] menu (Variable List panel-dedicated items)]**

The [File] menu for this panel provides the following items and functions.

Save Variable List Data	Overwrites the contents of this panel to the previously saved file (see "2.14 <a href="#">Save Analysis Information</a> "). Note that when the file has never been saved or the file is write disabled, the same operation is applied as the selection in [Save Variable List Data As...].
-------------------------	---

Save Variable List Data As...	Opens the <a href="#">Save As dialog box</a> to newly save the contents of this panel to the specified text file (see "2.14 <a href="#">Save Analysis Information</a> ").
Print...	Opens the Windows dialog box to print the contents currently being displayed in this panel.

### [[Edit] menu (Variable List panel-dedicated items)]

The [Edit] menu for this panel provides the following items and functions.

Copy	Copies the contents of the selected column (multiple selections possible) to the clipboard as character strings separated by tabs. Note that the contents of the clipboard cannot be pasted to this panel.
Select All	Selects all the columns being displayed in this panel.

### [Context menu]

The context menu displayed by right-clicking on this panel provides the following items and functions.

Find All References	Displays a list of the locations referencing the variable(s) of the selected row(s), in the <a href="#">Output panel's [Find References] tab</a> (see "2.11 <a href="#">Display List of Referencing Location</a> ").
Register to Analysis Chart	Registers the variable(s) of the selected row(s) to the <a href="#">Analysis Chart panel</a> (see "2.13.1 <a href="#">Graph transitions in values</a> "). Note that this item is disabled while disconnecting from the debug tool.
Access Break	The following cascade menus are displayed to set a break event <sup>Note</sup> with access condition (see "2.9.2 <a href="#">Set a break event to a variable</a> "). Note that this item is disabled while disconnecting from the debug tool.
Set Variable Read Break to	Sets a break event with read access condition to the variable of the current row.
Set Variable Write Break to	Sets a break event with write access condition to the variable of the current row.
Set Variable R/W Break to	Sets a break event with read/write access condition to the variable of the current row.
Register to Watch1	Registers the variable(s) of the selected row(s) to the Watch panel (Watch1) as watch-expression(s) (see "2.10 <a href="#">Register Watch-Expressions</a> "). Note that this item is disabled while disconnecting from the debug tool.
Jump to Source	Opens the Editor panel and displays the source file in which the variable of the current row is defined (see "2.7 <a href="#">Jump to Defined Location</a> ").
Jump to Memory	Opens the Memory panel (Memory1) and displays the memory list from the start address of the variable of the current row (see "2.7 <a href="#">Jump to Defined Location</a> "). Note that this item is disabled while disconnecting from the debug tool.
Copy	Copies the contents of the selected row(s) to the clipboard as character strings separated by tabs. Note that the contents of the clipboard cannot be pasted to this panel.

#### Note [RX]

For combination break events, this item is valid only when the combination condition is "OR".

**Analysis Chart panel**

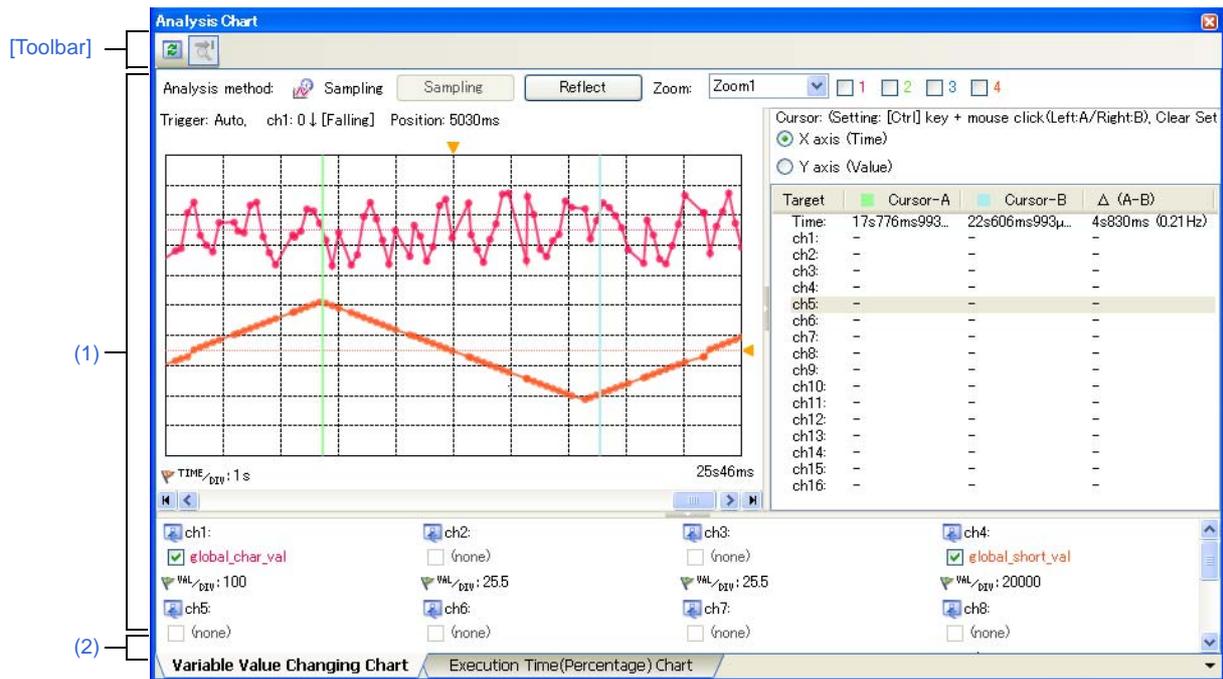
This panel is used to display charts of the acquired function information and variable information (including the data information for Smart Analog<sup>Note</sup>).

See "2.13 Display Analysis Information in Chart" for details on how to display the chart in this panel.

**Note [E1/E20[RL78]]**

This feature is supported only when the selected microcontroller incorporates Smart Analog IC.

**Figure A-10. Analysis Chart Panel**



The following items are explained here.

- [How to open]
- [Description of each area]
- [Toolbar]
- [[File] menu (Analysis Chart panel-dedicated items)]

**[How to open]**

- On the toolbar in the **Main window**, click the  button.
- From the [View] menu, select [Program Analyzer] >> [Analysis Chart].

**[Description of each area]**

**(1) Chart area**

This area displays a chart relating to the acquired function information/variable information.

**(2) Tab selection area**

The type of chart is switched when a tab is selected.

In this panel, a following tab is contained (see the section explaining the tab for details on the display/setting).

- [Variable Value Changing Chart] tab
- [Execution Time(Percentage) Chart] tab

**Remark** A  mark is displayed on the tab if a message relating to the chart information has been output to the [Output panel](#).

**[Toolbar]**

The toolbar provides the following items and functions.

	Acquires the latest data from the debug tool, and updates the contents of this panel. Note that this item is disabled if no registered variables exist.
	Displays the latest information automatically by acquiring the information each time program execution stops. Note that this item is disabled if the [Refresh at program stop] property in the [General] category on the <a href="#">[Settings] tab</a> of the <a href="#">Property panel</a> is set to other than [Specify Individually] (this button is locked in accordance with the setting on the <a href="#">Property panel</a> ).

**[[File] menu (Analysis Chart panel-dedicated items)]**

The [File] menu for this panel provides the following items and functions.

Save Analysis Chart Data	Overwrites the contents that are displayed on the currently selected tab to the previously saved file (see " <a href="#">2.14 Save Analysis Information</a> "). Note that when the file has never been saved or the file is write disabled, the same operation is applied as the selection in [Save Analysis Chart Data As...].
Save Analysis Chart Data As...	Opens the <a href="#">Save As dialog box</a> to newly save the contents that are displayed on the currently selected tab to the specified file (see " <a href="#">2.14 Save Analysis Information</a> ").

[Variable Value Changing Chart] tab

Use this tab to display the relationship between registered variable/register/address/other values and time by a graph (line chart).

Furthermore, when the selected microcontroller incorporates Smart Analog IC, by enabling [Data collection mode](#) on the debug tool, it is possible to display the collected data for Smart Analog by a graph [E1/E20[RL78]].

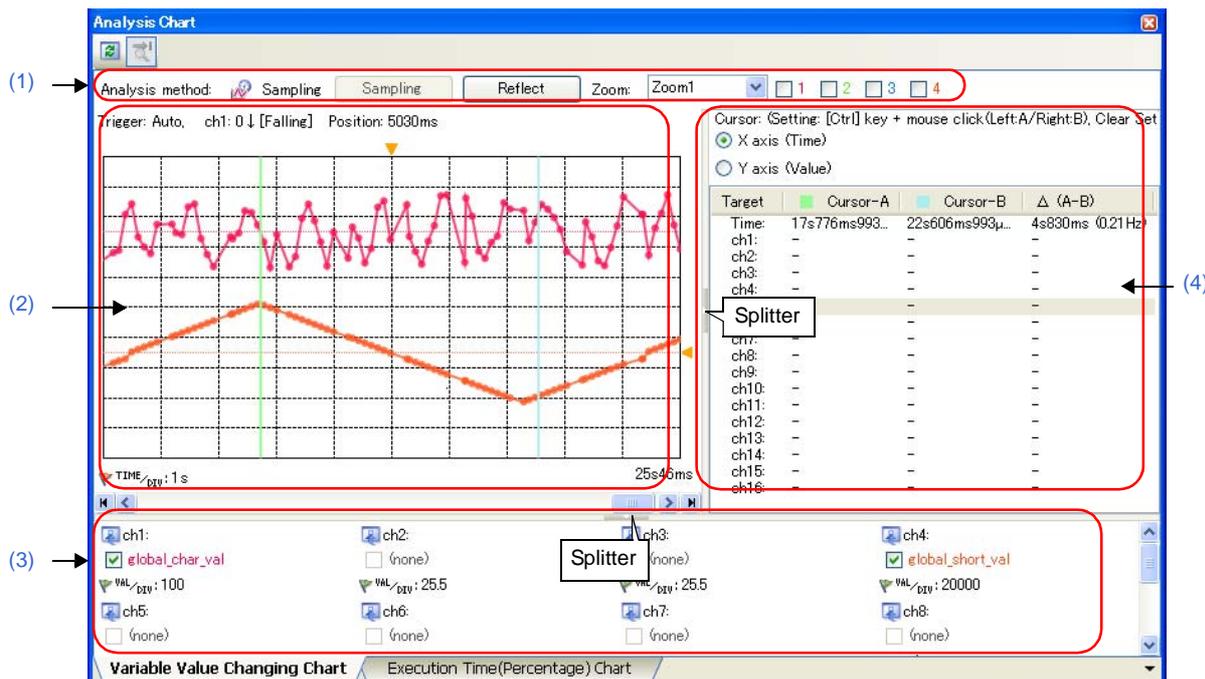
You can zoom in on the range specified in this panel, using the [Variable Value Changing Chart \(zoomed in\) panel](#).

See "2.13.1 [Graph transitions in values](#)" for details on how to display graphs on this tab.

- Cautions 1.** A graph is displayed based on trace data or real-time RAM monitoring results acquired by the debug tool. To display a graph, also see cautions of "(a) [Trace function](#)" or "(b) [RRM function/ RAM monitor \(pseudo RRM\) function](#)".
2. When [Real-time sampling analysis](#) is specified as the method for acquiring the graph data, also see "3.3 [Real-time Sampling Analysis](#)".
  3. When [Trace data analysis](#) is specified as the method for acquiring the graph data, also see "3.4 [Trace Data Analysis](#)".
  4. [E1/E20[RL78]]  
To graph the collected data for Smart Analog, a monitor program for collecting data must be linked to the execution program.

- Remarks 1.** See "(4) [Display graphs](#)" for details on when graphs on this tab are updated.
2. You can restore graphs currently being displayed on this tab by loading the analysis chart data file (\*.mtac) which saves the data of the graphs (see "(6) [Save the graph data to restore the graph](#)").
  3. If the acquired graph data exceeds the buffer capacity (10000 plots), then the oldest graph data is overwritten by the new graph data (ring buffer system). In this case, part of the graph rendering will be blank.
  4. The [Channel information area](#) and [Cursor information area](#) can be shown/hidden by clicking the mark on the center of their splitter.

Figure A-11. Analysis Chart Panel: [Variable Value Changing Chart] Tab



The following items are explained here.

- [How to open]
- [Description of each area]

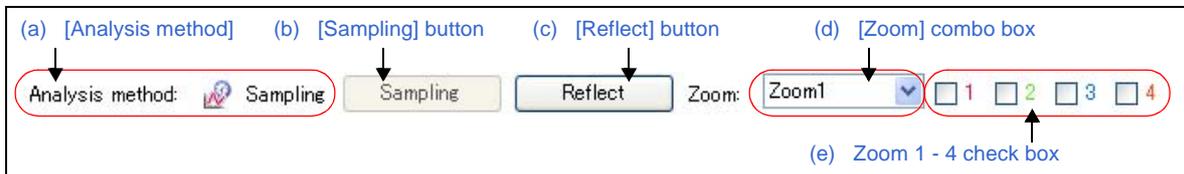
**[How to open]**

- On the toolbar in the **Main window**, click the  button, and then select the [Variable Value Changing Chart] tab.
- From the [View] menu, select [Program Analyzer] >> [Analysis Chart], and then select the [Variable Value Changing Chart] tab.

**[Description of each area]**

**(1) Graph control area**

**Figure A-12. Graph Control Area**



**(a) [Analysis method]**

Displays the analysis method currently being specified with the [Analysis method] property in the [General] category on the [Variable Value Changing] tab of the **Property panel** (see “(2) Specify the method for acquiring the graph data”).

Display Contents	Description
Sampling	Indicates that [Real-time sampling] is selected.
Trace	Indicates that [Analysis of trace data] is selected.
File	Indicates that [Load from file] is selected.

**(b) [Sampling] button**

Manually controls (toggles) the start/stop of the real-time sampling.

Note, however, that this button is available only when the following conditions are being satisfied:

- The debug tool is connected to CubeSuite+.
- **Real-time sampling analysis** is selected as the analysis method.
- [Manual] is selected in the [Start/stop real-time sampling] property in the [General] category on the [Variable Value Changing] tab of the **Property panel**.

**Caution** If sampling is restarted, the stored graph data will be deleted (the graph being displayed will be erased).

**(c) [Reflect] button**

Automatically registers watch-expressions currently being registered in the Watch panel (Watch1) as graphing target.

Up to 16 watch-expressions in the Watch panel (Watch1), from top to bottom are registered (see “<3> Reflecting the Watch panel (auto registration)”).

Note, however, that this button is available only when the following conditions are being satisfied:

- The debug tool is connected to CubeSuite+.
- The program is halted.
- [Real-time sampling analysis](#) or [Trace data analysis](#) is selected as the analysis method.
- **[E1/E20[RL78]]**  
[Data collection mode](#) is invalid on the debug tool.

**Caution** If this button is clicked, the information on the registered graphing targets will be deleted (the graph being displayed will be erased).

**(d) [Zoom] combo box**

When selecting the zoom range, select the number of the [Variable Value Changing Chart \(zoomed in\) panel](#) (Zoom1 - 4) (see "(c) Zoom").

Note, however, that this combo box is disabled during execution of a program.

**(e) Zoom 1 - 4 check box**

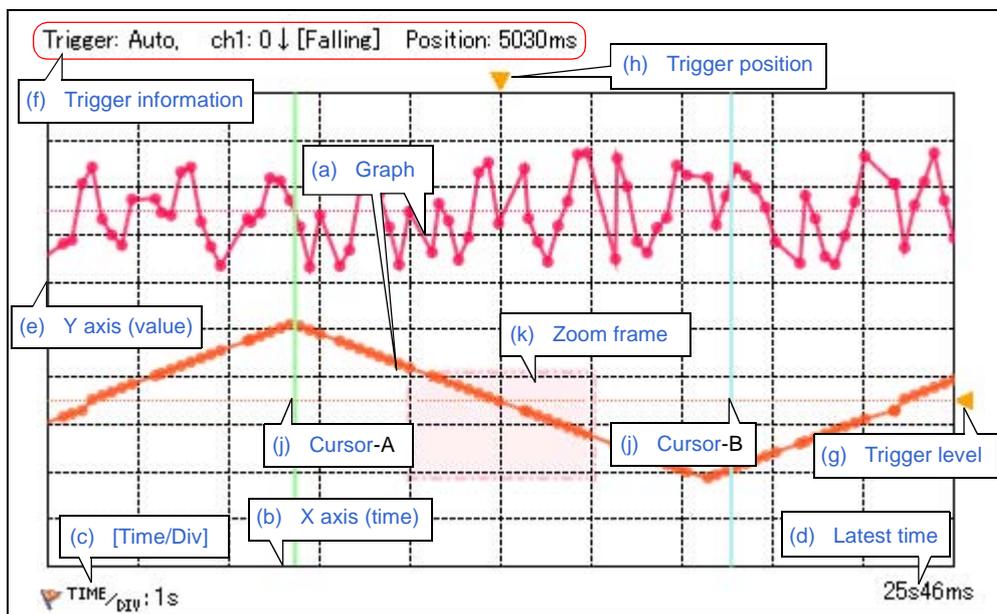
Specifies the [Variable Value Changing Chart \(zoomed in\) panel](#) to be displayed.

Select the check box of the corresponding number to open the [Variable Value Changing Chart \(zoomed in\) panel](#) (Variable Value Changing Chart (zoomed in) 1 - 4) (multiple selections possible).

In the default condition, no check boxes are selected.

**(2) Chart area**

**Figure A-13. Chart Area**



This area displays the relationship between values (Y axis (value) and X axis (time)) in the registered graphing target, in a line chart.

The function of each item displayed in this area is as follows:

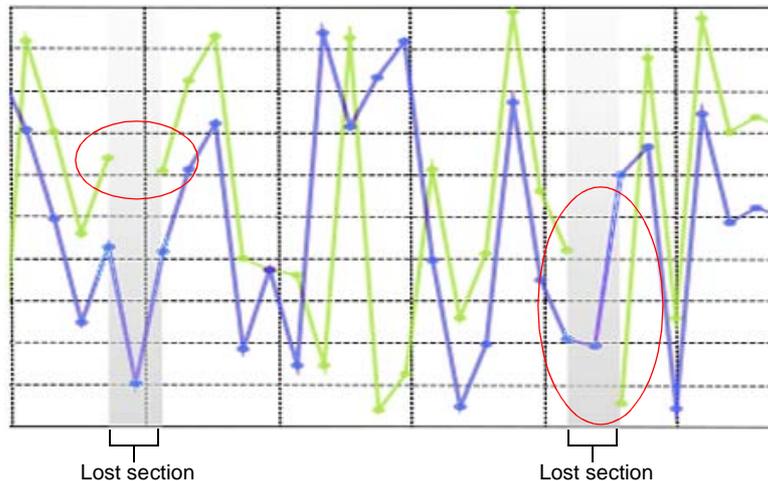
**(a) Graph**

The graph shows how the value of a target registered for each channel changes as the program execution proceeds. If acquisition of graph data through [Real-time sampling analysis](#) fails, however, only time information is provided with no connection lines. Such period is called a lost section and appears in the graph after the pro-

gram has stopped as shown below (the background color of graphs depends on the [Background color(Lost section)] property in the [General] category on the [Variable Value Changing] tab of the Property panel). If the debug tool has failed to acquire any single unit of data regardless of the current display range of graphs, the following message appears in the Output panel.

“RRM lost or buffer overflow has occurred, during data sampling.“

**Figure A-14. View of Section Where Acquisition of Graph Data Has Failed**



- Remarks 1.** Acquisition of graph data may fail for the following reasons.
- The target variable without a scope definition is outside the current scope at the time of sampling because it is a static variable inside a file or function.
  - **[E1/E20[RL78]]**  
Some data has been lost while the debug tool is in the [Data collection mode](#).
  - Other than the above (see "[3.4 Trace Data Analysis](#)")
- 2.** The rendering color of the graph for each channel can be changed individually via the [Color 1-16] property in the [Channel 1-16] category on the [Variable Value Changing] tab of the Property panel.
- 3.** If [Real-time sampling analysis](#) is selected, the format of the line chart can be changed via the [Chart type] property in the [General] category on the [Variable Value Changing] tab of the Property panel (see "(b) Selection of the chart type").

**(b) X axis (time)**

Displays the passage of time.

Grid lines divided in ten are displayed. The time per unit grid (Time/Div) for all channels can be specified with the [Time per grid[Time/Div]] property in the [General] category on the [Variable Value Changing] tab of the Property panel (see "(c) Setting the display range"). The time display range depends on the method for acquiring the graph data (see "[Table 2-11. Differences Depending on Method for Acquiring Graph Data](#)").

**(c) [Time/Div]**

Displays the value specified in the [Time per grid[Time/Div]] property in the [General] category on the [Variable Value Changing] tab of the Property panel.

Note that only while program execution is stopped, double-clicking on this label automatically adjusts the [Time per grid[Time/Div]] property to an optimum value so that the specified number of value transition points of the graph fit inside the rendering area (between the left and right ends) (see "(c) Setting the display range").

**(d) Latest time**

Displays the latest time corresponding to the graph data.

**(e) Y axis (value)**

Displays the values of the registered graphing target.

Grid lines divided in ten are displayed. The value per unit grid (Val/Div) for each channel can be specified with the [Value per grid[Val/Div] 1 - 16] property in the [Channel 1 - 16] category on the [Variable Value Changing] tab of the Property panel.

While the program is stopped, the axis line representing the origin (Value = 0) will be displayed as a dotted line in the rendering color of each channel.

**Caution** The display range of Y axis (value) depends on the [Value per grid[Val/Div] 1 - 16] property and [Offset 1 - 16] property setting, but if the obtained value is greater than the maximum or less than the minimum, nothing is displayed on the graph for that segment. Regardless of a range of current graph display, if a channel whose obtained values are always outside of the display range exists, then the message will be displayed on the **Output panel**.

**Remark** The offset value for each channel can be specified in unit grids of Y axis with the [Offset 1 - 16] property in the [Channel 1 - 16] category on the [Variable Value Changing] tab of the Property panel.

**(f) Trigger information**

Displays the contents currently being set in the [Trigger] category on the [Variable Value Changing] tab of the Property panel if the trigger function is used (see "Use the trigger function").

"-" will be displayed in this area if the trigger function is not used.

**(g) Trigger level**

Indicates the trigger level when the trigger function is used (see "Use the trigger function").

The value of the trigger level can be changed by dragging this trigger mark with the mouse.

Note, however, that this operation is disabled during execution of a program.

**(h) Trigger position**

Indicates the trigger position when the trigger function is used (see "Use the trigger function").

The value of the trigger position can be changed by dragging this trigger mark with the mouse.

Note, however, that this operation is disabled during execution of a program.

**(i) Pop-up display**

When the mouse cursor is hovered over a transition location on the graph, the information about that location appears in a pop-up.

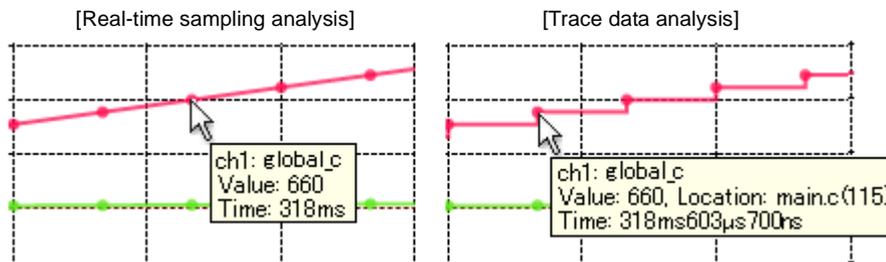
The display format is as follows:

[Real-time sampling analysis]

<p>ch1 - 16: Value name  Value: Data value  Time: Data value</p>
--

[Trace data analysis]

ch1 - 16: Value name  
 Value: Data value, Location: File name(Line number)  
 Time: Data value



**Caution** This function is disabled during execution of a program.

- Remarks 1.** The [Location] information is only displayed both when the graph data is acquired via [Trace data analysis](#) and when the transition location information exists in the graph data (if the information does not exist, "-" will be displayed). In this case, furthermore, you can display the applicable location in the Editor panel by double-clicking the transition point (except for a graph that was restored by loading the analysis chart data file (\*.mtac)). Note, however, that the [Location] information may be invalid value when the graph data is acquired by simultaneously using a Trace event and a Point Trace event in the debug tool.
- 2.** The [Time] information is displayed in the following format:
- Real-time sampling analysis: XXXsXXXms
  - Trace data analysis: XXXsXXXmsXXXµsXXXns

**(j) Cursor**

These two cursors (Cursor-A and Cursor-B) are used to check the time and value on the X axis (time) or Y axis (value). Under [Cursor selection buttons](#), select the [X axis (Time)] button to perform a cursor measurement for the X axis (time), and the [Y axis (Value)] button to perform it for the Y axis (value). The measurement results are listed in the [Cursor information area](#). Use the following operations to show or hide the cursors (the cursors are hidden by default).

Cursor	Setting (Show)	Clear Setting (Hide)
Cursor-A	[Ctrl] key + mouse click	[Ctrl] key + mouse double-click
Cursor-B	[Ctrl] key + mouse right-click	[Ctrl] key + mouse right-double-click

- Cautions 1.** **Cursor-A and Cursor-B are not displayed during execution of a program.**
- 2.** **When the [X axis (Time)] button is selected for the target axis, you cannot set the cursors in the area on the left of "0s", regardless of whether the graph data exists.**

**Remark** The color of each cursor can be changed via the [Cursor-A color]/[Cursor-B color] property in the [General] category on the [Variable Value Changing](#) tab of the [Property panel](#).

**(k) Zoom frame**

Indicates the zoom range displayed in the [Variable Value Changing Chart \(zoomed in\) panel](#) (see "(c) [Zoom](#)").

You can specify different zoom ranges for four different [Variable Value Changing Chart \(zoomed in\)](#) panels by selecting the panel numbers in the [\[Zoom\]](#) combo box.

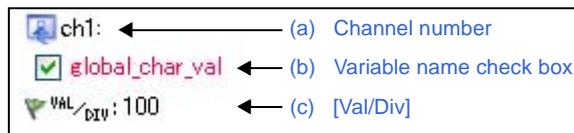
**Remark** The color of each zoom frame can be changed via the [\[Zoom-1 - 4 color\]](#) property in the [\[General\]](#) category on the [\[Variable Value Changing\]](#) tab of the [Property panel](#).

### (3) Channel information area

The display range of this area can be changed by moving the splitter with the mouse.

In addition, this area can be shown/hidden by clicking the mark on the center of the splitter.

**Figure A-15. Channel Information Area**



#### (a) Channel number

Displays the channel number (ch1 to ch16).

**Remark** [\[E1/E20\[RL78\]\]](#)

If the debug tool is in [Data collection mode](#), the icon displayed here changes to  from .

#### (b) Variable name check box

Displays the name of a variable (including register name/address expression) currently being registered in the channel as a graphing target, in a check box format ("none" will be displayed if no graphing target is registered).

The character color of the variable name is the same as the rendering color of the graph.

Select the check box to display the graph corresponding to the variable (multiple selections possible).

In the default condition, all check boxes where the graphing target is being registered are selected.

**Caution** [\[E1/E20\[RL78\]\]](#)

**When the debug tool is in [Data collection mode](#), the variable name displayed here is not target for graphing.**

**Remarks 1.** See "(1) [Register graphing targets](#)" for details on how to register the graphing target.

**2.** The color of the graph for each channel can be changed individually via the [\[Color 1-16\]](#) property in the [\[Channel 1-16\]](#) category on the [\[Variable Value Changing\]](#) tab of the [Property panel](#).

#### (c) [\[Val/Div\]](#)

Displays "*the value per unit grid*" currently being specified with the [\[Value per grid\[Val/Div\] 1 - 16\]](#) property in the [\[Channel 1-16\]](#) category on the [\[Variable Value Changing\]](#) tab of the [Property panel](#).

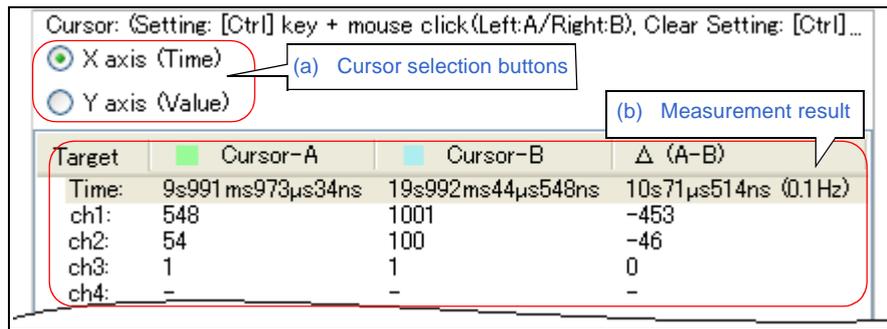
When the program is not running, double-clicking this label automatically adjusts the values of the [\[Value per grid\[Val/Div\] 1 - 16\]](#) and [\[Offset 1-16\]](#) properties so that the graph for the selected channel will be shown in the entire rendering area (see "(c) [Setting the display range](#)").

### (4) Cursor information area

The display range of this area can be changed by moving the splitter with the mouse.

In addition, this area can be shown/hidden by clicking the mark on the center of the splitter.

Figure A-16. Cursor Information Area



**(a) Cursor selection buttons**

Select the target axis to perform a **Cursor** measurement.

Note, however, that this button is disabled during execution of a program.

X axis (Time)	Set the cursor measurement target to the X axis (time) (default).
Y axis (Value)	Set the cursor measurement target to the Y axis (value).

**(b) Measurement result**

This area displays the following measurement result from the cursor-A and cursor-B positions.

Note, however, that the information below disappears during execution of a program.

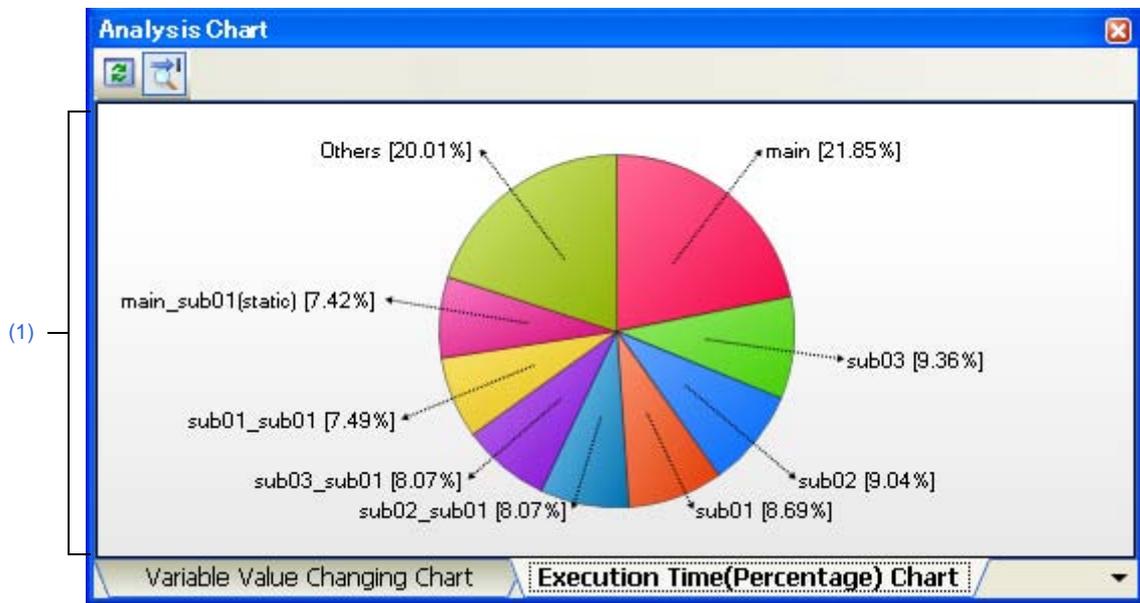
Time	<ul style="list-style-type: none"> <li>- When the target is the X axis (time) Shows the time indicated by cursor-A, the time indicated by cursor-B, and the difference between the two (the cycle [Hz] calculated from the difference).</li> <li>- When the target is the Y axis (value) Shows "-"</li> </ul>
ch1 - 16	<ul style="list-style-type: none"> <li>- When the target is the X axis (time) Shows the value at the time when cursor-A indicates, the value at the time when cursor-B indicates, and the difference between the two.</li> <li>- When the target is the Y axis (value) Shows the value indicated by cursor-A, the value indicated by cursor-B, and the difference between the two.</li> </ul> <p>If no value can be obtained, "-" will be displayed.</p>

**[Execution Time(Percentage) Chart] tab**

Use this tab to display the acquired function information in a pie chart of the function execution times.  
See "2.13.2 [Chart the function execution time ratios](#)" for details on how to display graphs on this tab.

- Cautions 1.** If the debug tool used does not support the [Trace function](#), or if the debug tool's trace function is not enabled, then this chart cannot be displayed.  
Even if the trace function is enabled-condition, if there is no trace data in the trace memory, no chart is displayed on this tab, and then the following message will be displayed on the [Output panel](#): "There is no execution time information."
- 2.** To display this chart, also see cautions of "(a) [Trace function](#)".
- 3.** [\[IECUBE\[78K0\]\]\[E1/E20\[RX\]\]\[EZ Emulator\[RX\]\]](#)  
Because the trace time tag function is not supported, this chart cannot be displayed.

Figure A-17. Analysis Chart Panel: [Execution Time(Percentage) Chart] Tab



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)

**[How to open]**

- On the toolbar in the [Main window](#), click the  button, and then select the [Execution Time(Percentage) Chart] tab.
- From the [View] menu, select [Program Analyzer] >> [Analysis Chart], and then select the [Execution Time(Percentage) Chart] tab.

**[Description of each area]****(1) Chart area**

This area displays a pie chart indicating function execution time ratios.

You can change the number of functions to display by setting from the [The number of functions displaying in the Execution Time(Percentage) Chart 1-4] property in the [General] category on the [Settings] tab of the Property panel (set to [10] by default).

The targets are charted in ranking order of highest proportion of execution time taken. Functions exceeding the number specified here are displayed together under "Others".

- Remarks 1.** The colors of labels and background of a chart depend on the settings of "font color" and "background color" for the [Default] item in the [General - Font and Color] category of the Option dialog box.
- 2.** You can expand or shrink the graph display from 10 to 100 % by rotating the mouse wheel forward or backward while pressing the [Ctrl] key (note that this does not apply to the font size of each label).

**(a) Label display**

Each function label is displayed in the format "*Function-name*[*proportion of execution time (%)*]".

For static files, the file name is also included inside the parentheses ( ).

**Remark** The *proportion of execution time* is the same as the [Execution Time(Percentage)[%]] in the [Function List panel](#).

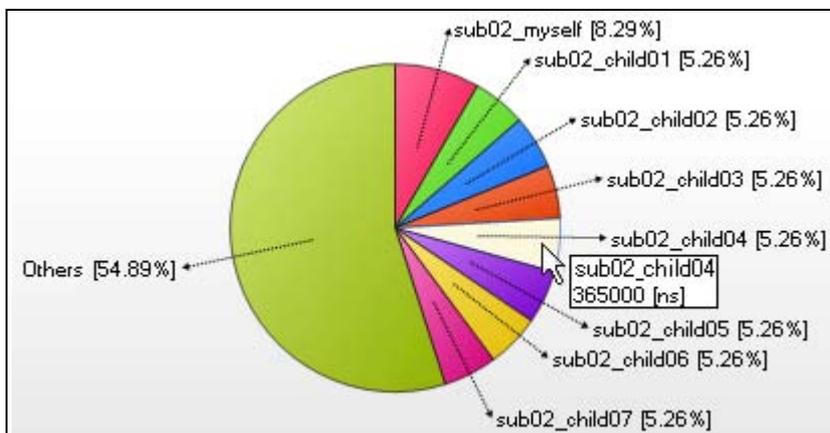
**(b) Pop-up display**

When the mouse cursor is hovered over the chart, information about that function appears in a pop-up.

The display format is as follows:

<i>Function name</i>
<i>Execution time [unit]</i>

<i>Function name</i>	Indicates the name of the target function. <b>[CC-RX]</b> If the target function is a member function, then this will be displayed in the format of " <i>Class name::Function name</i> ".
<i>Execution time [unit]</i>	Indicates the execution time of the target function (this is the same value as the [Execution Time[unit]] in the <a href="#">Function List panel</a> ). The [Unit] can be changed by the [Unit of time] property in the [General] category on the [Settings] tab of the <a href="#">Property panel</a> .



### Call Graph panel

This panel is used to display the tree-structured chart (call graph) that indicates calling relationships between functions. The following types of function/variable are displayed in this panel.

- Global function
- Static function
- Member function (only when C++ source files are subject to analysis) [CC-RX]
- Global variable
- Static variable inside a file
- Static variable inside a function
- IOR [V850]
- SFR [R8C(Localised support)][RL78][78K0R][78K0]
- Class variable (only when C++ source files are subject to analysis) [CC-RX]

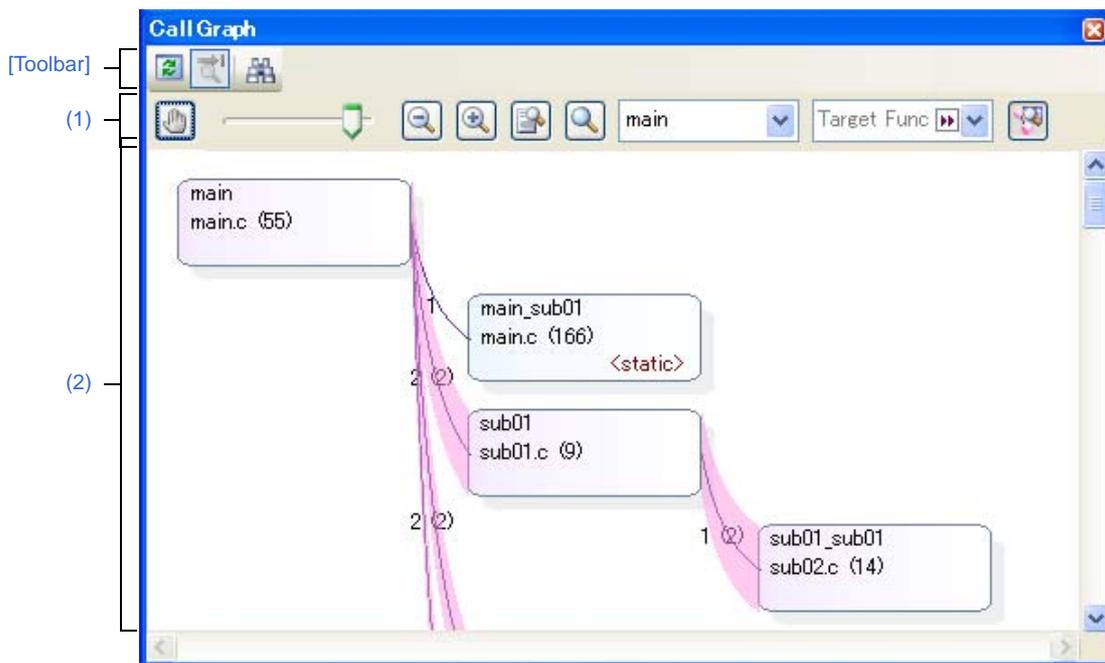
Note, however, that the function information and variable information in the files not to be analyzed is not displayed in this panel (see "1.1.1 What is analyzed").

See "2.4 Display Calling Relationships between Functions (Call Graph)" for details on how to display the call graph.

- Cautions 1.** If the debug tool to use does not support the trace function, or if the debug tool's trace function is not enabled, then the **Dynamic analysis information** (e.g. the execution count/read count/write count) in the call graph cannot be displayed. For the dynamic analysis information, also see the caution of "(a) Trace function".
2. [CC-RX]  
When C++ source files are subject to analysis, also see "3.6 When CC-RX (C++ Source Files) Is Used".
  3. [CA850][CA78K0R][CA78K0]  
When a clean is executed in the build tool, the contents currently being displayed in this panel will be cleared.

- Remarks 1.** A function/variable whose source file does not exist in the project can be displayed in the call graph if the [Display the function/variable without definition at Call Graph panel] property in the [General] category on the [Settings] tab of the **Property panel** is set to [Yes].
2. This panel can be zoomed in and out (10 to 109 %) by  in the **Panel control area**, or by moving the mouse wheel forward or backward while holding down the [Ctrl] key.

Figure A-18. Call Graph Panel (Whole View)



The following items are explained here.

- [How to open]
- [Description of each area]
- [Toolbar]
- [[File] menu (Call Graph panel-dedicated items)]
- [[Edit] menu (Call Graph panel-dedicated items)]
- [Context menu]

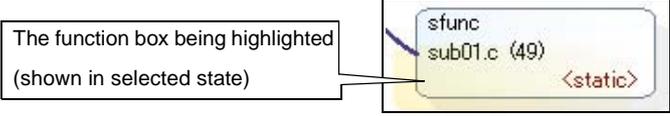
**[How to open]**

- On the toolbar in the [Main window](#), click the  button.
- From the [View] menu, select [Program Analyzer] >> [Call Graph].

**[Description of each area]**

**(1) Panel control area**

	<p>Toggles whether scrolling of content is enabled by dragging the panel directly. By default, scrolling is disabled.</p> <p>Note that if scrolling is enabled, then the shape of the mouse cursor will change, and a function/variable will be neither highlighted (shown in selected state) nor reflected in the <a href="#">[Target Function Name] combo box</a> even if clicking on a <a href="#">Function box/Variable box</a> with mouse. There will also not be a <a href="#">Pop-up display</a> with information about the function.</p>
	<p>Changes the call graph's zoom ratio. A number between 10 and 109 can be selected (by default, 100 is specified).</p> <p>You can also change the zoom by rotating the mouse wheel forward or backward while pressing the [Ctrl] key.</p>
	<p>Changes the call graph's zoom ratio. Each click of one of the buttons decrements/increments the value of the zoom slider by 1.</p>

	Automatically changes the call graph's zoom ratio. The call graph is enlarged or shrunk in accordance with the size of the panel, so that the entire call graph can be displayed.
	Resets the zoom ratio of the call graph to the default value of 100.
 [Parent Function Name] combo box (Left combo box)	Selects the parent function to target for display in the call graph (it will be at the top of the call graph) from the drop-down list <sup>Note 1</sup> . By default, the first function to appear named "main"/"reset"/"start" <sup>Note 2</sup> or containing such word is specified as the parent function (if there is no corresponding function, it will be empty).
 [Target Function Name] combo box (Right combo box)	Selects the function to highlight (show in selected state) on the call graph from the following drop-down list. - If the [Parent Function Name] combo box is empty Whole view: <i>The names of all functions in the program</i> Details view: <i>The names of all functions currently being shown</i> - If a parent function is specified with the [Parent Function Name] combo box Whole view: <i>The names of functions called by that parent function (including child and grandchild functions)</i> Details view: <i>The names of all functions currently being shown</i> When a function box is clicked in the call graph, the function is highlighted (unless scrolling via the mouse dragging is enabled by the  button), and then the function's name is also reflected in this combo box. <div style="text-align: center; margin-top: 20px;">  </div>
	Switches to the details view that displays the parent functions and child functions of the function currently being selected (specified with the [Target Function Name] combo box) (toggle). Note that this item is disabled during execution of a program.

**Notes 1. [RH850][V850E2]**

If the selected microcontroller version supports multi-core, you can select "PMn"<sup>[RH850]/</sup> "PEn"<sup>[V850E2]</sup>. In this case, the call graph will only be displayed for the functions executed in the selected PMn/PEn.

2. The function name differs with the selected microcontroller as follows:

- <sup>[RH850][V850][RL78][78K0R][78K0]</sup>: main
- <sup>[RX]</sup>: reset
- <sup>[R8C(Localised support)]</sup>: start

**(2) Call graph area**

This area displays a call graph indicating calling relationships between functions that have been acquired from the cross reference information.

A call graph has the following two view modes that can be toggled by clicking the  button.

**- Whole view (default)**

By default, the first function to appear named "main"/"reset"/"start"<sup>Note 1</sup> or containing such word is considered to be the parent function, and that function is placed on the left edge of the call graph (the corresponding parent function name will be displayed in the [Parent Function Name] combo box). If there is no corresponding function name, then the functions in the program that are not called by any other functions (reference count = 0) will be considered to be parent functions, and they will be placed on the leftmost edge (the [Parent Function Name] combo box will be empty).

Next, the child functions, grandchild functions, and so on will be placed on the call graph, from left to right (they will be arranged from top to bottom, with the functions to appear first shown topmost).

If an arbitrary parent function is specified in the [Parent Function Name] combo box<sup>Note 2</sup>, then only the functions called by that function will appear in the call graph. If the combo box is empty, then all functions in the program will appear in the call graph.

**Notes 1.** The function name differs with the selected microcontroller as follows:

- [RH850][V850][RL78][78K0R][78K0]: main
- [RX]: reset
- [R8C(Localised support)]: start

**2.** [RH850][V850E2]

If the selected microcontroller version supports multi-core and "PMn"[RH850]/"PEn"[V850E2] is selected in the [Parent Function Name] combo box, the call graph will only be displayed for the functions executed in the selected PMn/PEn.

**- Details view**

The parent functions and child functions of the function currently being specified with the [Target Function Name] combo box are displayed.

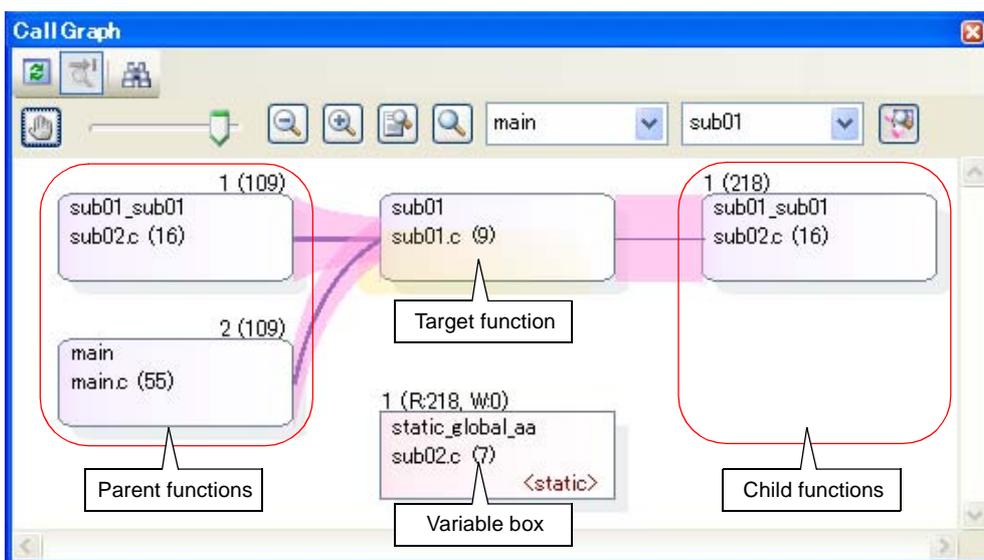
The parent functions, the target function and the child functions are placed on the call graph, from left to right (they will be arranged from top to bottom, with the functions to appear first shown topmost).

Furthermore, if global variables, static variables inside a file or static variables inside a function accessed from the target function exist, the variables also is placed immediately below the target function, as variable boxes (when the multiple variables exist, they will be arranged from top to bottom, with the variables to appear first shown topmost).

**Remark**

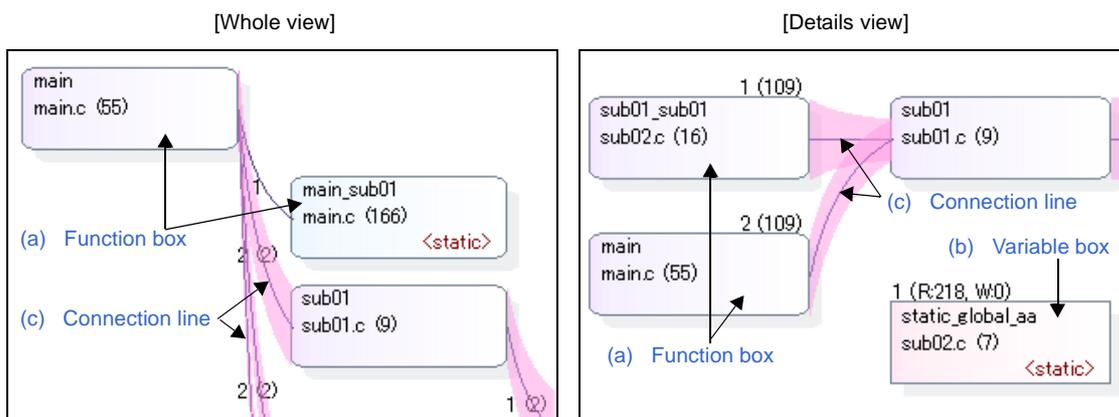
- Any one of the following events will cause that the call graph is switched to whole view mode.
  - The value of the [Parent Function Name] combo box has been changed (including the result of a build/rebuild in the build tool).
  - A clean has been executed in the build tool.

**Figure A-19. Call Graph Panel (Details View)**



The composition of the call graph is as follows:

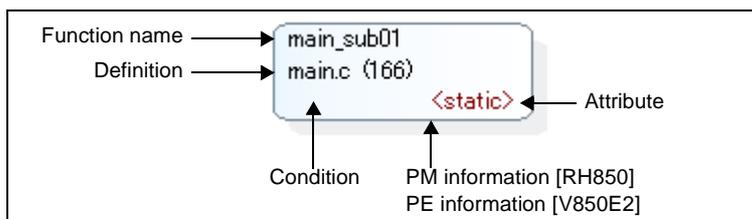
Figure A-20. Composition of Call Graph



(a) Function box

The function is displayed in a box format.  
The following information is displayed in a function box.

Figure A-21. Information of Function Box



Function name	Indicates the name of the target function <sup>Note</sup> .
Definition	Indicates the location at which the target function is defined with "File name(Line number)". Note, however, that if the definition information does not exist, then this will display "(No Definition)".
Attribute	Indicates the attribute of the target function. - Static function: <static> - Template function: <template> [CC-RX] - Virtual function: <virtual> [CC-RX] - Pure virtual function: <abstract> [CC-RX] If the attribute other than the above, nothing is displayed in this area.
Condition	The current state of the target function is shown in the following background colors. - Light blue: Not executed - Purple: Executed
PM Information [RH850] PE Information [V850E2]	If the selected microcontroller version supports multi-core, PE in which the target function is assigned is shown in the following shadow colors. - Gray: Common - Red: PM1/PE1 - Green: PM2/PE2 - Blue: PM3/PE3 - Orange: PM4/PE4

**Note [CC-RX]**

If the target function is a overloaded function or a template function, then the argument types of the function are also displayed in parentheses "( )".

If the target function is a const member function or volatile member function, then "const" or "volatile" is displayed immediately after the function name.

**Remark** By double-clicking a function box, you can jump to the location of the source text where the function is defined (see "2.7 Jump to Defined Location").

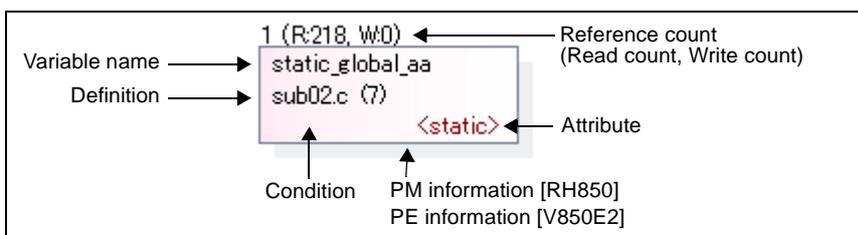
Note, however, that this function cannot be used while scrolling via the mouse dragging is being enabled by the  button. In this case, select the target function with the [Target Function Name] combo box, then select [Jump to Source] from the context menu.

**(b) Variable box**

In the details view mode, a global variable, a static variable inside a file or a static variable inside a function accessed from the target function is displayed in a box format.

The following information is displayed in a variable box.

**Figure A-22. Information of Variable Box**



Variable name	Indicates the name of the target variable. If the target variable is a static variable inside a function, then this will be displayed in the format of "Variable name#Function name".
Definition	Indicates the location at which the target variable is defined with "File name(Line number)". Note, however, that if the definition information does not exist, then this will display "(No Definition)".
Attribute	Indicates the attribute of the target variable. - Static variable: <static> - Static variable inside a function: <static local> If the attribute other than the above, nothing is displayed in this area.
Reference count	Indicates the number of times of static references.
Read count, Write count	Indicates the number of times dynamic references (R: read accesses, W: write accesses) have been made. Note that this information appears only when the trace data exist. This information is the cumulative value for each program execution when the [Accumulate result of analysis] property in the [General] category on the [Settings] tab of the Property panel is set to [Yes].
Condition	The current state of the target function is shown in the following background colors. - Green: Not referenced - Red purple: Referenced

PM Information [RH850] PE Information [V850E2]	<p>If the selected microcontroller version supports multi-core, PE in which the target variable is assigned is shown in the following shadow colors.</p> <ul style="list-style-type: none"> <li>- Gray: Common</li> <li>- Red: PM1/PE1</li> <li>- Green: PM2/PE2</li> <li>- Blue: PM3/PE3</li> <li>- Orange: PM4/PE4</li> </ul>
---	---

**Remark** By double-clicking a variable box, you can jump to the location of the source text where the variable is defined (see "2.7 Jump to Defined Location").

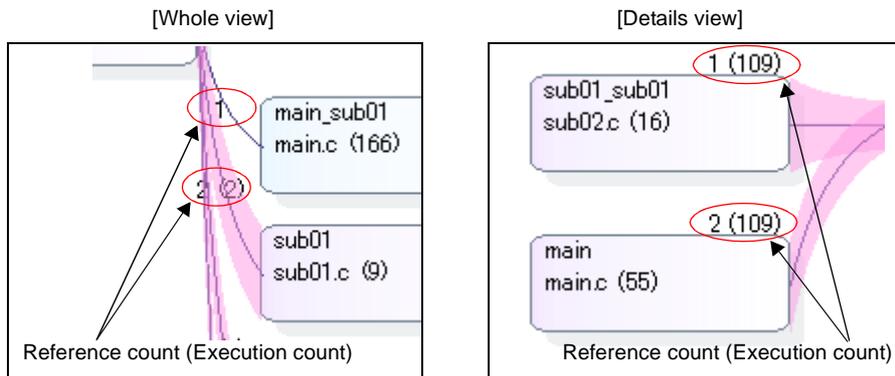
Note, however, that this function cannot be used while scrolling via the mouse dragging is being enabled by the  button. In this case, disable the scroll function temporarily, then double click the target variable.

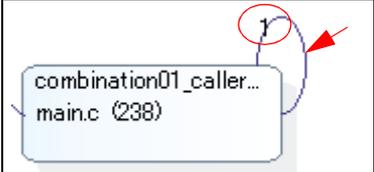
**(c) Connection line**

If a given function is called statically by another function, then the function boxes of the two functions will be shown linked by a connection line. Connection lines are only displayed for static function calls. Connection lines are not displayed for dynamic function calls.

The following information is displayed by each connection line.

**Figure A-23. Information of Connection Line**



Reference count	Indicates the number of times of static calls.
Execution count	Indicates the number of times dynamic calls have been made, in parentheses "( )". Note that this information appears only when the trace data exist. This information is the cumulative value for each program execution when the [Accumulate result of analysis] property in the [General] category on the [Settings] tab of the Property panel is set to [Yes].
Recursive call	<p>A connection line and reference count is displayed as follows when a function calls itself.</p> 

<p>Circular call</p>	<p>As an example, say that there are functions A, B, and C. If A calls B, which calls C, which then calls A (A -&gt; B -&gt; C -&gt; A), then although connection lines will be displayed for the calls of A to B and B to C, no connection line will be shown for the call of C to A. Instead, the following line segment will only be shown, to indicate a circular call.</p> <p>You can check information about functions in a circular call via the <a href="#">Pop-up display</a>.</p> <div data-bbox="753 434 1121 568" style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p>circular02_caller05 main.c (97)</p>  </div>
----------------------	--

- Remarks 1.** If there are no static function calls but there are dynamic function calls (e.g. if function calls are only made using function pointers), then this information can be confirmed in the [Pop-up display](#).
- 2. [CA850][CA78K0R][CA78K0]**
- If a prototype declaration of function C is placed between the definitions of functions A and B, a connection line appears between functions A and C, indicating that function A calls function C. Also, for other than a prototype declaration of a function, a reference to a variable (assignment to a variable pointer) or a reference to a function (assignment to a function pointer variable) is the same operation as the above.

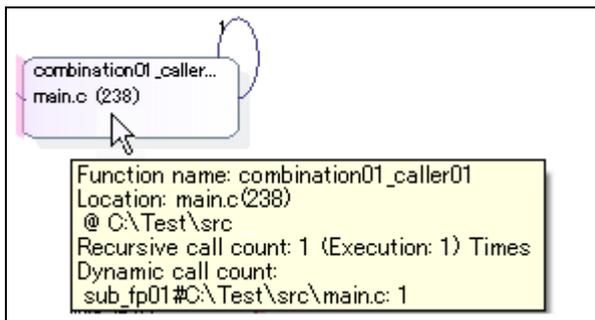
**(d) Pop-up display**

When the mouse cursor is hovered over a function box/variable box, information about that function/variable appears in a pop-up.

The display format is as follows:

**Caution** This function cannot be used while scrolling via the mouse dragging is being enabled by the  button.

**<1> Function box**



<p>Function name: <i>Function name</i></p> <p>Location: <i>File name(Line number)</i>  <i>@ absolute path of the file</i></p> <p>Recursive call count: <i>Reference count(Execution: Execution count)</i></p> <p>Circular call count  <i>Function name: Reference count(Execution: Execution count)</i>                  ...</p> <p>Dynamic call count  <i>Function name: Count</i>                  ...</p>
--

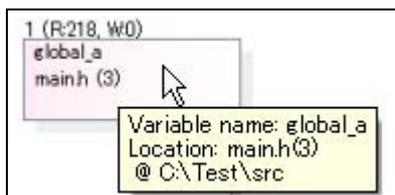
Function name	Indicates the name of the target function <sup>Note</sup> .
Location	Indicates the location at which the target function is defined. Note, however, that if the definition information does not exist, then this will display "(No Definition)".
Recursive call count	This item is displayed only if the target function is a recursive call. Reference count: Number of recursive calls made Execution count: Number of executions (only if there is trace data)
Circular call count	This item is displayed only if there are circular function calls from the target function. If there are multiple target functions, that at most 4 will be listed. Function name: Name of function in circular call Reference count: Number of circular calls made Execution count: Number of executions (only if there is trace data)
Dynamic call count	This item is displayed only if there are only dynamic function calls, and no static function calls. If there are multiple target functions, that at most 4 will be listed. Function name: Name of dynamic call function Count: Number of executions

**Note [CC-RX]**

- When the target function is a global function or static function:  
If the target function belongs to a namespace (except for a global namespace/anonymous namespace), then this will be displayed in the format of "*Namespace name::Function name*". Moreover, if the target function is a overloaded function or a template function, then this will be followed by the argument types of the function in parentheses "( )".
- When the target function is a member function:  
If the target function belongs to a namespace (except for a global namespace/anonymous namespace), then this will be displayed in the format of "*Namespace name::Class name::Function name*". Otherwise this will be displayed in the format of "*Class name::Function name*". Moreover, if the target function is a overloaded function or a template function, then Function name will be followed by the argument types of the function in parentheses "( )".  
If the target function is a const member function or volatile member function, then "const" or "volatile" is displayed immediately after the function name.

**Remark** The cumulative value for each program execution is displayed as the execution count when the [Accumulate result of analysis] property in the [General] category on the [Settings] tab of the Property panel is set to [Yes].

**<2> Variable box**



Variable name: *Variable name*  
Location: *File name(Line number)*  
*@ absolute path of the file*

Variable name	Indicates the name of the target variable <sup>Note</sup> . If the target variable is a static variable inside a function, then this will be displayed in the format of " <i>Function name#Variable name</i> ".
Location	Indicates the location at which the target variable is defined. Note, however, that if the definition information does not exist, then this will display "(No Definition)".

**Note [CC-RX]**

- When the target variable is a global variable or static variable inside a file:  
If the target variable belongs to a namespace (except for a global namespace/anonymous namespace), then this will be displayed in the format of "*Namespace name::Variable name*".
- When the target variable is a class variable:  
If the target variable belongs to a namespace (except for a global namespace/anonymous namespace), then this will be displayed in the format of "*Namespace name::Class name::Variable name*". Otherwise this will be displayed in the format of "*Class name::Variable name*".
- When the target variable is a static variable inside a function:  
If the target variable belongs to a namespace (except for a global namespace/anonymous namespace), then this will be displayed in the format of "*Namespace name::Function name#Variable name*". Otherwise this will be displayed in the format of "*Function name#Variable name*".
- If the target variable is a class variable, then this will be displayed in the format of "*Class name::Function name#Variable name*".

**[Toolbar]**

The toolbar provides the following items and functions.

	Acquires the latest data from the debug tool, and updates the contents of this panel. Note that this item is disabled during execution of a program.
	Displays the latest information automatically by acquiring the information each time program execution stops. Note that this item is disabled if the [Refresh at program stop] property in the [General] category on the [Settings] tab of the <a href="#">Property panel</a> is set to other than [Specify Individually] (this button is locked in accordance with the setting on the <a href="#">Property panel</a> ).
	Opens the <a href="#">Call Graph Search dialog box</a> to search for a function/variable existing on this panel.

**[[File] menu (Call Graph panel-dedicated items)]**

The [File] menu for this panel provides the following items and functions.

Save Call Graph Data	Overwrites the contents of this panel to the previously saved file (see " <a href="#">2.14 Save Analysis Information</a> "). Note that when the file has never been saved or the file is write disabled, the same operation is applied as the selection in [Save Call Graph Data As...].
Save Call Graph Data As...	Opens the <a href="#">Save As dialog box</a> to newly save the contents of this panel to the specified text file (see " <a href="#">2.14 Save Analysis Information</a> ").

**[[Edit] menu (Call Graph panel-dedicated items)]**

The [Edit] menu for this panel provides the following items and functions.

Find...	Opens the <a href="#">Call Graph Search dialog box</a> to search for a function/variable currently existing on this panel.
---------	--

**[Context menu]**

The context menu displayed by right-clicking on this panel provides the following items and functions.

Jump to Source	Opens the Editor panel and displays the source file in which the function/variable currently being selected is defined (see " <a href="#">2.7 Jump to Defined Location</a> ").
Jump to Disassemble	Opens the Disassemble panel (Disassemble1) and displays the disassemble data corresponding to the start address of the function/variable currently being selected (see " <a href="#">2.7 Jump to Defined Location</a> "). Note that this item is disabled while disconnecting from the debug tool.
Jump to Memory	Opens the Memory panel (Memory1) and displays the memory list corresponding to the start address of the function/variable currently being selected (see " <a href="#">2.7 Jump to Defined Location</a> "). Note that this item is disabled while disconnecting from the debug tool.
Jump to Function/Variable List	Opens the <a href="#">Function List panel/Variable List panel</a> and selects the function/variable currently being selected.
Detail View	Switches to the details view that displays the parent functions and child functions of the function currently being selected (specified with the <a href="#">[Target Function Name] combo box</a> ). Note that this item is disabled during execution of a program.

**Class/Member panel**

This panel is used to display the class information [CC-RX], function information and variable information in a tree structure.

The following types of function/variable are displayed in this panel.

- Global function
- Static function
- Member function (only when C++ source files are subject to analysis) [CC-RX]
- Global variable
- Static variable inside a file
- Class variable (only when C++ source files are subject to analysis) [CC-RX]
- Instance variable (only when C++ source files are subject to analysis) [CC-RX]

Note, however, that the class information [CC-RX], function information and variable information in the files not to be analyzed is not displayed in this panel (see "1.1.1 What is analyzed").

See "2.5 Display Class/Function/Variable Information" for details on how to display the class information [CC-RX], function information and variable information.

**Cautions 1. [CC-RX]**

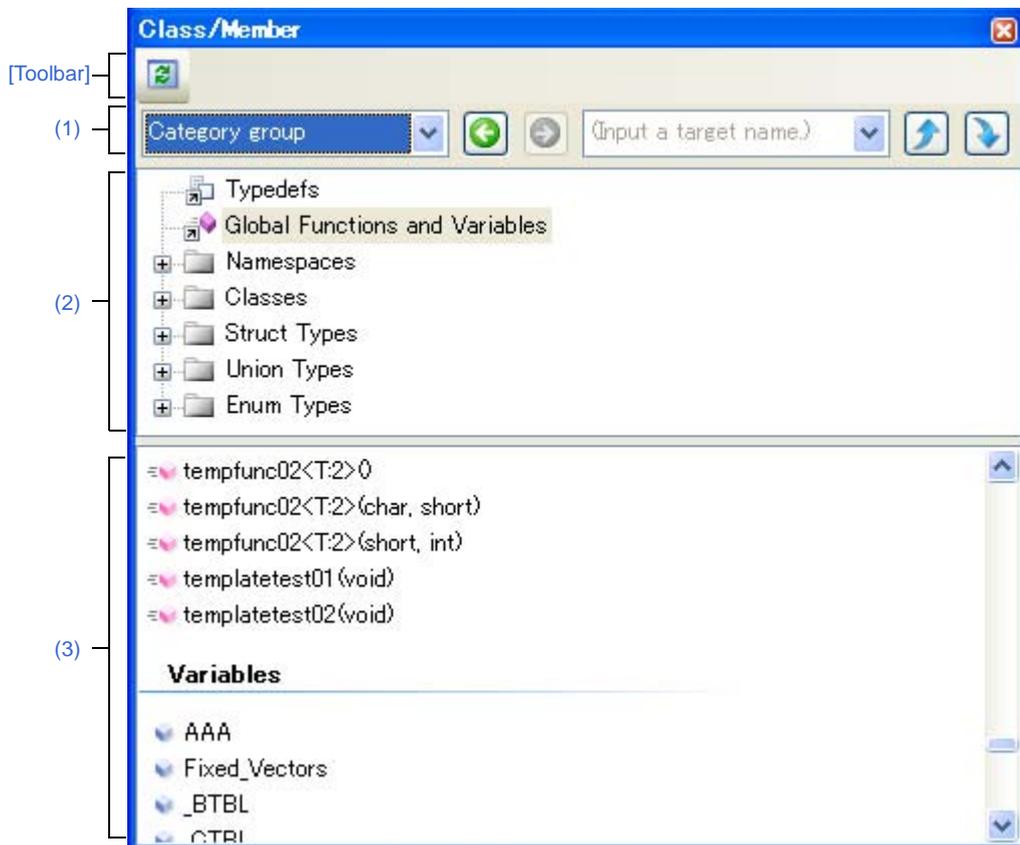
The class information is provided only when C++ source files are subject to analysis.

When C++ source files are subject to analysis, also see "3.6 When CC-RX (C++ Source Files) Is Used".

**2. [CA850][CA78K0R][CA78K0]**

If a clean is executed in the build tool, the contents of this panel being displayed will be erased.

Figure A-24. Class/Member Panel



The following items are explained here.

- [How to open]
- [Description of each area]
- [Toolbar]
- [[Edit] menu (Class/Member panel-dedicated items)]
- [Context menu]

**[How to open]**

- On the toolbar in the [Main window](#), click the  button.
- From the [View] menu, select [Program Analyzer] >> [Class/Member].

**[Description of each area]**

**(1) Panel control area**

 [View Setting] combo box (Left combo box)	Selects a classification method for the tree from the drop-down list. - <a href="#">Category group</a> (default) - <a href="#">Access group</a> - <a href="#">Namespace group</a> - <a href="#">File group</a> - <a href="#">Alphabet group</a>
	Selects the last node that was selected. Note that this item is disabled if there are no node selections in the history, or during execution of a program.
	Selects the node that was selected before the  button was clicked. Note that this item is disabled if there are no node selections in the history that were selected via  button, or during execution of a program.
 [Target Name] combo box (Left combo box)	Specifies the character strings to search via the  /  button (case-insensitive). You can either type character strings directly from the key board (up to 512 characters), or select one from the input history via the drop-down list (up to 10 items). Note that this item is disabled during execution of a program.
	Searches up in the last tree to have focus for nodes containing the string specified in the <a href="#">[Target Name] combo box</a> , and selects the node that is found. Note that this button is disabled if the <a href="#">[Target Name] combo box</a> is left blank, or during execution of a program.
	Searches down in the last tree to have focus for nodes containing the string specified in <a href="#">[Target Name] combo box</a> , and selects the node that is found. Note that this button is disabled if the <a href="#">[Target Name] combo box</a> is left blank, or during execution of a program.

**Remark** After typing character strings to search, to press the [Enter] key is the same function as clicking the  button, and to press the [Shift] + [Enter] key is the same function as clicking the  button.

**(2) Classes/Categories tree area**

This area used to display the class and other information defined by the program in tree format. Information about the node being selected in this tree appears in the [Functions/Variables list area](#). You can use the [\[View Setting\] combo box](#) to classify the display into the following five groups.

- Category group (default)
- Access group
- Namespace group
- File group
- Alphabet group

The node name and their order in the tree are described below.

Note however, that if information could not be obtained, or there are no child nodes to display, then that node will not appear in the tree.

You can expand/shrink the node with a child node by double-clicking it.

**(a) Category group (default)**

This classifies and displays classes, interfaces and other types.

Node	Note
Typedefs	
Global Functions and Variables	
Define Macros and Constants	[CA78K0R][CA78K0]
Namespaces { } <i>Namespace name</i>	[CC-RX]
Class <i>Class name</i> <sup>Note 1</sup> Base Types <i>Base type (Class/Interface/Struct type) name</i> Derived Types <i>Derived type (Class/Interface/Struct type) name</i> Inner Types <i>Inner type (Class/Interface/Struct type/Union type/Enum type) name</i>	[CC-RX]
Interfaces <i>Interface name</i> <sup>Note 1</sup> Base Types <i>Base type (Class/Interfaces/Struct type) name</i> Derived Types <i>Derived type (Class/Interface/Struct type) name</i> Inner Types <i>Inner type (Class/Interface/Struct type/Union type/Enum type) name</i>	[CC-RX]
Struct Types <i>Struct type name</i> <sup>Note 1, 2</sup>	Except [CA850]
Union Types <i>Union type name</i> <sup>Note 3</sup>	
Enum Types <i>Enum type name</i>	

**Notes 1. [CC-RX]**

These are displayed in the following access specifier order, together with icons denoting the access specifier (nodes with the same access specifier are displayed in alphabetical order).

Access Specifier	Icon
public	None
protected	
private	

2. **[CC-RX]**  
Base Types node, Derived Types node and Inner Types node are displayed as necessary.
3. **[CC-RX]**  
Inner Types node is displayed as necessary.

**(b) Access group**

This classifies and displays nodes by their access specifiers specified in their classes, etc..  
Nodes of the same type are displayed in alphabetical order.

Node	Note
Typedefs	
Global Functions and Variables	
Define Macros and Constants	<b>[CA78K0R][CA78K0]</b>
public <i>Class name</i> <sup>Note 1</sup> <i>Interface name</i> <sup>Note 1</sup> <i>Struct type name</i> <sup>Note 1</sup> <i>Union type name</i> <sup>Note 2</sup> <i>Enum type name</i>	<b>[CC-RX]</b>
protected <i>Class name</i> <sup>Note 1</sup> <i>Interface name</i> <sup>Note 1</sup> <i>Struct type name</i> <sup>Note 1</sup> <i>Union type name</i> <sup>Note 2</sup> <i>Enum type name</i>	<b>[CC-RX]</b>
private <i>Class name</i> <sup>Note 1</sup> <i>Interface name</i> <sup>Note 1</sup> <i>Struct type name</i> <sup>Note 1</sup> <i>Union type name</i> <sup>Note 2</sup> <i>Enum type name</i>	<b>[CC-RX]</b>

- Notes 1. [CC-RX]**  
Base Types node, Derived Types node and Inner Types node are displayed as necessary.
2. **[CC-RX]**  
Inner Types node is displayed as necessary.

**(c) Namespace group**

This classifies and displays nodes by the namespaces in which the class or other elements are defined.  
Nodes of the same type are displayed in the access specifier order (see "(a) Category group (default)"), and nodes with the same access specifier are displayed in alphabetical order.

Node	Note
 <i>Namespace name</i> <sup>Note 1</sup>	
 Typedefs	
 Global Functions and Variables	
 Define Macros and Constants	[CA78K0R][CA78K0]
 <i>Class name</i> <sup>Note 2</sup>	[CC-RX]
 <i>Interface name</i> <sup>Note 2</sup>	
 <i>Struct type name</i> <sup>Note 2</sup>	Except [CA850]
 <i>Union type name</i> <sup>Note 3</sup>	
 <i>Enum type name</i>	

- Notes 1.** Function, classes, and other elements that do not belong to any namespace (including when they are in C source file) are displayed in the same manner, in the "(Global)" instead of "*Namespace name*". If an element is defined in an anonymous namespace, its "*Namespace name*" will be "(Anonymous:<*File name*>)", and it will be displayed in the same way.
- 2. [CC-RX]**  
Base Types node, Derived Types node and Inner Types node are displayed as necessary.
  - 3. [CC-RX]**  
Inner Types node is displayed as necessary.

**(d) File group**

This classifies and displays the files in which the class or other elements are defined. Nodes of the same type are displayed in the access specifier order (see "(a) [Category group \(default\)](#)"), and nodes with the same access specifier are displayed in alphabetical order.

Node	Note
 <i>File name</i> <sup>Note 1</sup>	
 Typedefs	
 Global Functions and Variables	
 Define Macros and Constants	[CA78K0R][CA78K0]
 <i>Class name</i> <sup>Note 2</sup>	[CC-RX]
 <i>Interface name</i> <sup>Note 2</sup>	
 <i>Struct type name</i> <sup>Note 2</sup>	Except [CA850]
 <i>Union type name</i> <sup>Note 3</sup>	
 <i>Enum type name</i>	

- Notes 1.** The icon displayed differs depending on the type of the file.
- 2. [CC-RX]**  
Base Types node, Derived Types node and Inner Types node are displayed as necessary.
  - 3. [CC-RX]**  
Inner Types node is displayed as necessary.

(e) Alphabet group

This displays nodes in alphabetical order, regardless of their classes or access specifiers.

Node	Note	
Typedefs		
Global Functions and Variables		
Define Macros and Constants	[CA78K0R][CA78K0]	
<i>Class name</i> <sup>Note 1</sup>	[CC-RX]	These are displayed in alphabetical order.
<i>Interface name</i> <sup>Note 1</sup>		
<i>Struct type name</i> <sup>Note 1</sup>	Except [CA850]	
<i>Union type name</i> <sup>Note 2</sup>		
<i>Enum type name</i>		

Notes 1. [CC-RX]

Base Types node, Derived Types node and Inner Types node are displayed as necessary.

2. [CC-RX]

Inner Types node is displayed as necessary.

Cautions 1. Typedefs node/Global Functions and Variables node/Define Macros and Constants node is not displayed if no information displayed in the [Functions/Variables list area](#) exist.

2. [CC-RX]

When C++ source files are subject to analysis, the alternative name of a namespace is not displayed.

3. [CX]

The bit field structure type and union type cannot be displayed.

4. [CA850][NC30(Localised support)]

Define Macros and Constants node is not displayed because constants (i.e. variables with a const declaration) cannot be distinguished (these are treated as variables).

Remarks 1. When the mouse cursor is hovered over a *xxx name* node (except for a *file name* node), the namespace name to which it belongs appears in a pop-up. However, if it belongs to a global namespace or an anonymous namespace, then "-" or "<unnamed>" will appear.

If the mouse cursor is hovered over a *namespace name* node, the namespace name to which it belongs (upper namespace name) will appear.

2. About *Namespace name* node:

If an element contains an internal namespace, it is displayed in the form "*enveloping Namespace name::internal Namespace name*".

- Examples 1.** namespace Name: Name  
 2. For an internal namespace: Name::SubName

3. About *Class name* node:

If it is a template class, then the type information is added.

If an element is an internal class, it is displayed in the form "*enveloping Class name::internal Class name*".

- Examples 1.** class Sub: Sub  
 2. template<class T> class List: List<T>  
 3. For an internal class: Main::SubInMain

4. About *Class/Interface/Struct* type:  
Base Types, Derived Types, and Inner Types are only shown for classes/interfaces that are direct descendants (if there are multiple such types, they are all shown).
5. If *Class name/Interface name/Struct type name/Union type name/Enum type name* is an anonymous (i.e. it has no tag name), then it will be displayed in the format of "<unnamed\_<math>N</math>\*" (<math>N</math>: A number from 1 automatically provided in appearance order).
6. You can jump to the corresponding node (*Class name* node, *Interface name* node, etc. in the same tree) by double-clicking a node displayed under the Base Types node/Derived Types node/Inner Types node.
7. You can jump to the location where the node currently being selected is defined or declared (see "[2.7 Jump to Defined Location](#)"/"[2.8 Jump to Declared Location](#)").

**(3) Functions/Variables list area**

This area displays the information of the node currently being selected in the [Classes/Categories tree area](#) (e.g. function names/variable names that are defined in the node).

The relationship between the selection in the [Classes/Categories tree area](#) and the display contents of this area is as follows:

Node Selection	Display Contents of This area	Note
Typedefs	<i>Alternative name</i>	
Global Functions and Variables	<i>Global function name</i>	
	<i>Static function name</i>	
	<i>Global variable name</i>	
	<i>Static variable inside a file name</i>	
Define Macros and Constants	<i>Macro name</i>	[CA78K0R][CA78K0]
	<i>Constant</i>	
<i>Class name</i> <i>Interface name</i> <i>Struct type name (C++ source file)</i>	<i>Alternative name</i>	[CC-RX]
	<i>Member function name</i>	
	<i>Class variable name</i>	
	<i>Instance variable name</i>	
	<i>Constant</i>	
<i>Union type name (C++ source file)</i>	<i>Alternative name</i>	[CC-RX]
	<i>Member function name</i>	
	<i>Instance variable name</i>	
	<i>Constant</i>	
<i>Struct type name (C source file)</i>	<i>Member variable name</i>	Except [CA850]
<i>Union type name (C source file)</i>	<i>Member variable name</i>	
<i>Enum type name</i>	<i>Enum member name</i> <sup>Note</sup>	
<i>Namespace name</i> <i>File name</i> Other than above	None	

**Note [CX][NC30(Localised support)]**

Members of enumeration cannot be displayed.

**Caution** Nothing is displayed in this area if functions/variables to be displayed do not exist or nothing is being selected in the **Classes/Categories tree area**.

**Remarks 1.** When the mouse cursor is hovered over a *xxx name* node (except for a *file name* node), the namespace name to which it belongs appears in a pop-up. However, if it belongs to a global namespace or an anonymous namespace, then "-" or "<unnamed>" will appear.  
If the mouse cursor is hovered over a *namespace name* node, the namespace name to which it belongs (upper namespace name) will appear.

**2. [CC-RH][CC-RX][CX][NC30(Localised support)]**

Argument types are added to function names.

- Examples 1.**
- 1. int main(void): main(void)
  - 2. void main\_sub01(int local\_a, int local\_b, int local\_c): main\_sub01(int, int, int)
  - 3. int function01(int arg01) const: function01(int) const

For template functions **[CC-RX]**, the line for the definition of a template function and the line for the function using a template function are displayed individually.

- Examples 1.**
- 1. template<class T> T max(T a, T b) (on a line where a template is defined):  
max<T:1>()
  - 2. int max(int a, int b) (on a line where a template is used): max<T:1>(int, int)

**3. [CC-RX]**

These are displayed in the following access specifier order, together with icons denoting the access specifier type (nodes with the same access specifier are displayed in alphabetical order). However, if the tree is classified by **File group**, then they are all displayed in alphabetical order.

Access Specifier	Icon
public	None
protected	
private	

**4. [CC-RX]**

For template functions, the line for the definition of a template function and the line for the function using a template function are displayed individually.

**5. [CA78K0][CA78K0R]**

If "only a type-definition (structure/union/enumeration), without a variable-declaration" and "an anonymous type (structure/union/enumeration)" is described consecutively, "a member of the anonymous type" is treated as a member of "only a type-definition, without a variable-declaration". In the case of the following example, mem01, mem02, num01, and num02 are displayed in this area as the members of the structure "def\_only\_str".

```

struct def_only_str {
    char    mem01;
    short   mem02;
};
struct {
    short   num01;
    int     num02;
} anonymous_struct
    
```

**6. [CA78K0][CA78K0R]**

If a structure/union/enumeration is defined in a structure/union/enumeration, a member described after the definition line is treated as a member of the structure/union/enumeration defined in the

structure/union/enumeration.

In the case of the following example, mem01 (as the member of the structure "SType") and inn01, mem02, and mem03 (as the member of the structure "InnerType") are displayed in this area.

```

struct SType {
    char    mem01;
    struct  InnerType {
        char    inn01;
    } mem02;
    int     mem03;
} struct01;
    
```

7. You can jump to the location where the node currently being selected is defined or declared (see "2.7 Jump to Defined Location"/"2.8 Jump to Declared Location").

**[Toolbar]**

The toolbar provides the following items and functions

	Acquires the latest data from the debug tool, and updates the contents of this panel.
---	---

**[[Edit] menu (Class/Member panel-dedicated items)]**

The [Edit] menu for this panel provides the following items and functions.

Copy	Copies the contents of the selected node to the clipboard as character strings.
Select All	Selects all the nodes being displayed in the <a href="#">Functions/Variables list area</a> .

**[Context menu]**

The context menu displayed by right-clicking on this panel provides the following items and functions.

Jump to Source	Opens the Editor panel and displays the source file in which the node currently being selected is defined (see "2.7 Jump to Defined Location").
Jump to Declaration of Source	Opens the Editor panel and displays the source file in which the node currently being selected is declared (see "2.8 Jump to Declared Location").
Jump to Disassemble	Opens the Disassemble panel (Disassemble1) and displays the disassemble data corresponding to the start address of the node <sup>Note</sup> currently being selected in the <a href="#">Functions/Variables list area</a> (see "2.7 Jump to Defined Location"). Note that this item is disabled while disconnecting from the debug tool.
Jump to Memory	Opens the Memory panel (Memory1) and displays the memory list corresponding to the start address of the node <sup>Note</sup> currently being selected in the <a href="#">Functions/Variables list area</a> (see "2.7 Jump to Defined Location"). Note that this item is disabled while disconnecting from the debug tool.
Jump to Function/Variable List	Opens the <a href="#">Function List panel</a> / <a href="#">Variable List panel</a> and selects the function/variable of the node <sup>Note</sup> currently being selected.
Copy	Copies the contents of the selected node to the clipboard as character strings.

**Note [CC-RX]**

This item is disabled if a node that indicates a instance variable is selected.

**Variable Value Changing Chart (zoomed in) panel**

This panel is used to zoom in on the specified range on the chart displayed in the [Variable Value Changing Chart] tab of the Analysis Chart panel.

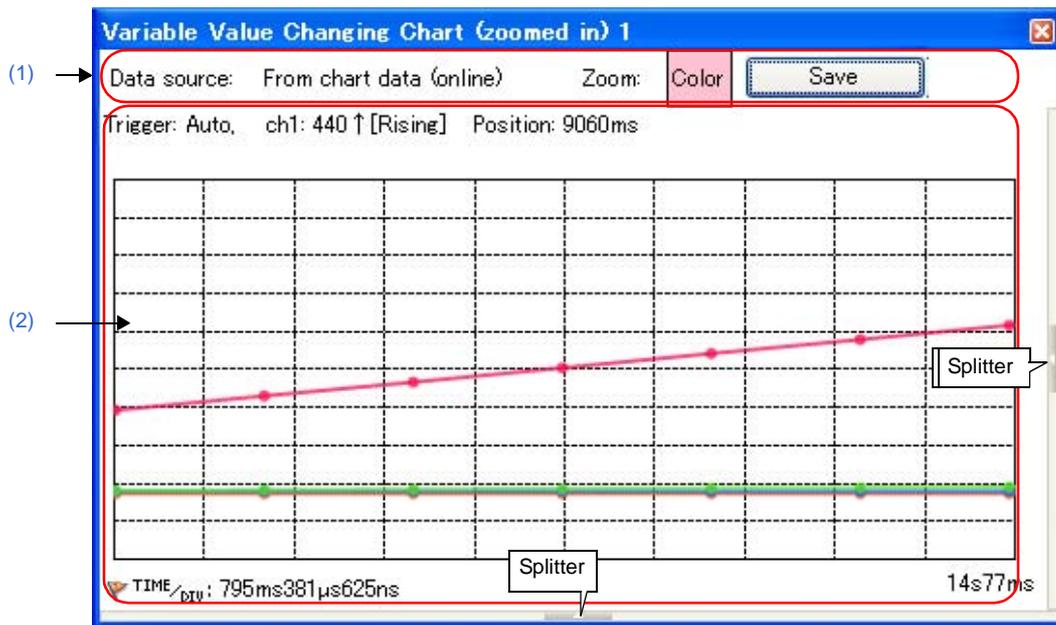
Up to a maximum of four of these panels can be opened. Each panel is identified by the names "Variable Value Changing Chart (zoomed in) 1 - 4" on the title bar. From these panels, you can zoom in on the specified range individually.

See "(c) Zoom" for details on how to display the chart on this panel.

- Cautions 1.** The graph on this panel cannot be updated during execution of a program.
- 2.** The graphing target cannot be registered/deleted on this panel.

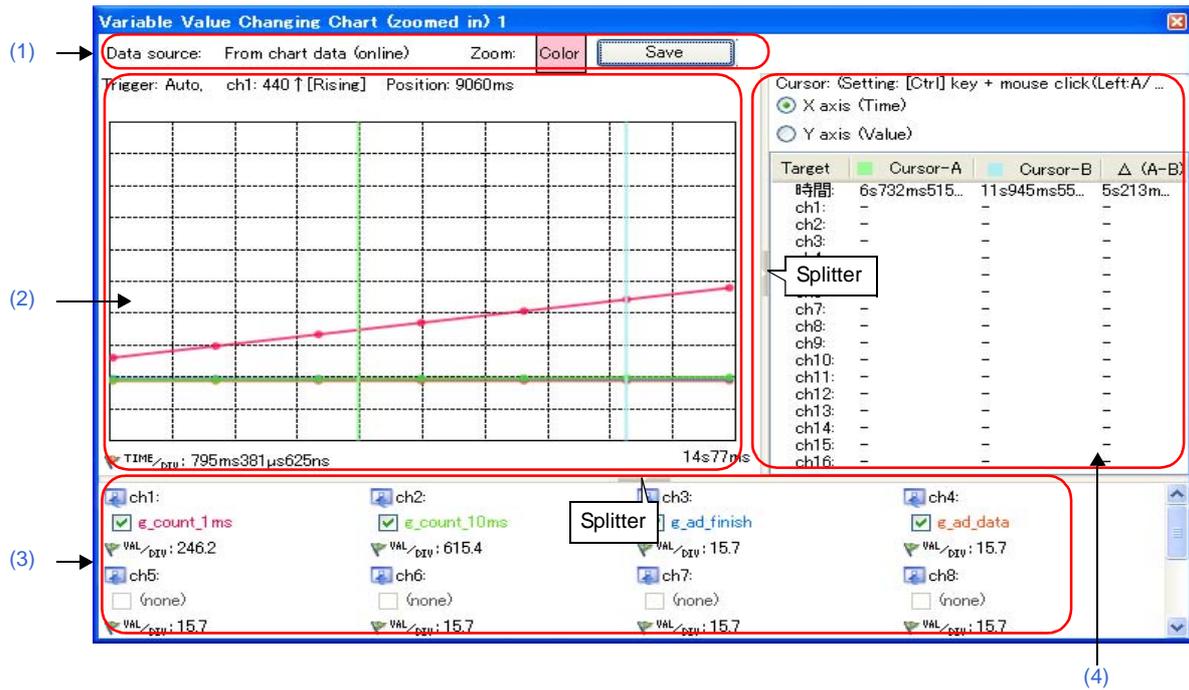
- Remarks 1.** The display contents of this panel are updated when:
  - Each time the graph displayed in the [Variable Value Changing Chart] tab of the Analysis Chart panel is updated (except for the real-time display update)
  - The zoom range is set or cleared
  - The analysis chart data file (\*.mtac) is loaded
- 2.** You can restore graphs currently being displayed on this panel by loading the analysis chart data file (\*.mtac) which saves the data of the graphs (see "(6) Save the graph data to restore the graph").

**Figure A-25. Variable Value Changing Chart (zoomed in) Panel (Default)**



You can display/hide the Channel information area and the Cursor information area by clicking the mark on the center of the splitter on this panel.

Figure A-26. Variable Value Changing Chart (zoomed in) Panel (Whole Area View)



The following items are explained here.

- [How to open]
- [Description of each area]

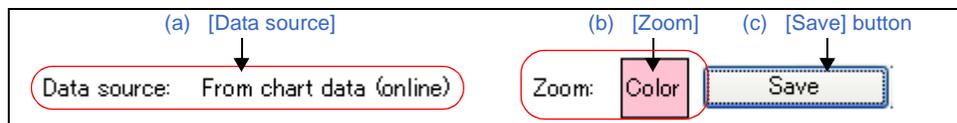
**[How to open]**

- On the [Variable Value Changing Chart] tab of the Analysis Chart panel, select the zoom 1 - 4 check box.

**[Description of each area]**

**(1) Graph control area**

Figure A-27. Graph Control Area



**(a) [Data source]**

Displays the data source of the graph currently being displayed.

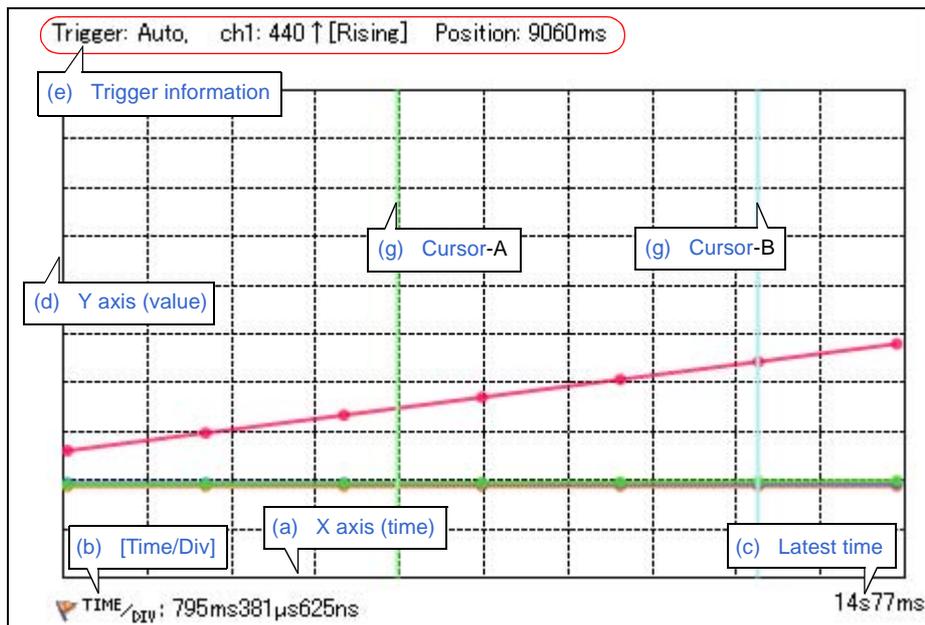
Display Contents	Description
From chart data (online)	Indicates that the data source is the graph currently being displayed in the [Variable Value Changing Chart] tab.
From file (offline)	Indicates that the data source is the information form the loaded analysis chart data file (*.mtac).

**(b) [Zoom]**

Indicates the color of the zoom frame currently being specified with the [Zoom-1 - 4 color] property in the [General] category on the [Variable Value Changing] tab of the Property panel.

**(c) [Save] button**

Opens the [Save As dialog box](#) to save the contents currently being displayed on this panel to the specified file (see "2.14 Save Analysis Information").

**(2) Chart area****Figure A-28. Chart Area**

This area displays a zoomed view of the range selected on the graph in the [Variable Value Changing Chart] tab of the Analysis Chart panel.

The function of each item displayed in this area is as follows:

**(a) X axis (time)**

Displays the passage of time.

**(b) [Time/Div]**

Displays the time per grid unit (10% of the total time for the zoom range set in the [Variable Value Changing Chart] tab of the Analysis Chart panel). This value cannot be changed.

**(c) Latest time**

Displays the latest time corresponding to the zoom range set in the [Variable Value Changing Chart] tab of the Analysis Chart panel.

**(d) Y axis (value)**

Displays the values of the registered graphing target.

The value per grid unit (Val/Div) is 10% of the value (difference between maximum and minimum) for the zoom range set in the [Variable Value Changing Chart] tab of the Analysis Chart panel for each channel (cannot be changed).

If the origin (value = 0) exists in the zoom range, the axis line represented it will be displayed as a dotted line in the rendering color of each channel.

**Caution** If the graph data obtained from the data source exceeds the minimum/maximum of the display range above, then nothing will be displayed on the graph for that segment.

**(e) Trigger information**

Displays the contents currently being set in the [Trigger] category on the [Variable Value Changing] tab of the Property panel if the trigger function is used (see "Use the trigger function").

"-" will be displayed in this area if the trigger function is not used.

**(f) Pop-up display**

When the mouse cursor is hovered over a transition location on the graph, the information about that location appears in a pop-up.

See "(i) Pop-up display" for details on its display format.

**Caution** This function is disabled during execution of a program.

**(g) Cursor**

These two cursors (Cursor-A and Cursor-B) are used to check the time and value on the X axis (time) or Y axis (value).

Under **Cursor selection button**, select the [X axis (Time)] button to perform a cursor measurement for the X axis (time), and the [Y axis (Value)] button to perform it for the Y axis (value).

The measurement results are listed in the **Cursor information area**.

Use the following operations to show or hide the cursors (the cursors are hidden by default).

Cursor	Setting (Show)	Clear Setting (Hide)
Cursor-A	[Ctrl] key + mouse click	[Ctrl] key + mouse double-click
Cursor-B	[Ctrl] key + mouse right-click	[Ctrl] key + mouse right-double-click

**Caution** Cursor-A and Cursor-B are not displayed during execution of a program.

**(3) Channel information area**

The display range of this area can be changed by moving the splitter with the mouse.

In addition, this area can be shown/hidden by clicking the mark on the center of the splitter (by default, this area is hidden).

**Figure A-29. Channel Information Area**



**(a) Variable name check box**

Displays the name of a variable (including register name/address expression) currently being registered in a channel as a graphing target, in a check box format ("none" will be displayed if no graphing target is registered).

The character color of the variable name is the same as the rendering color of the graph.

Select the check box to display the graph corresponding to the variable (multiple selections possible). In the default condition, the settings of the data source are reflected to these check boxes. Note, however, that this operation is disabled during execution of a program.

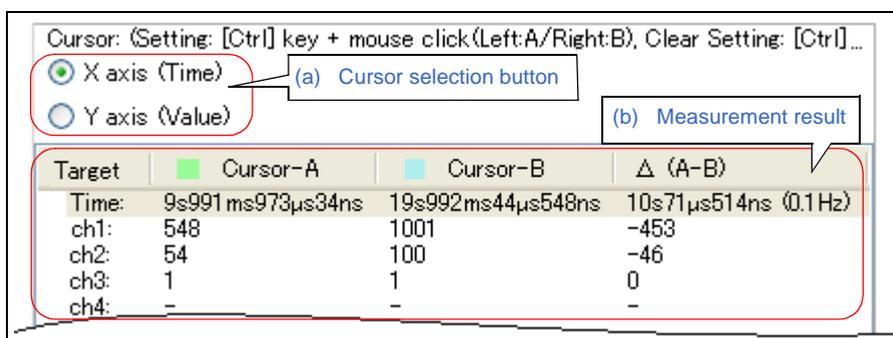
**(b) [Val/Div]**

Displays “the value per unit grid” currently being specified with the [Value per grid[Val/Div] 1 - 16] property in the [Channel 1-16] category on the [Variable Value Changing] tab of the Property panel.

**(4) Cursor information area**

The display range of this area can be changed by moving the splitter with the mouse. In addition, this area can be shown/hidden by clicking the mark on the center of the splitter (by default, this area is hidden).

**Figure A-30. Cursor Information Area**



**(a) Cursor selection button**

Select the target axis to perform a **Cursor** measurement. Note, however, that this button is disabled during execution of a program.

X axis (Time)	Set the cursor measurement target to the X axis (time) (default).
Y axis (Value)	Set the cursor measurement target to the Y axis (value).

**(b) Measurement result**

This area displays the following measurement result from the cursor-A and cursor-B positions. Note, however, that the information below disappears during execution of a program.

Time	<ul style="list-style-type: none"> <li>- When the target is the X axis (time) Shows the time indicated by cursor-A, the time indicated by cursor-B, and the difference between the two (the cycle [Hz] calculated from the difference).</li> <li>- When the target is the Y axis (value) Shows "-"</li> </ul>
ch1 - 16	Shows the value at the time when cursor-A indicates, the value at the time when cursor-B indicates, and the difference between the two. If no value can be obtained, "-" will be displayed.

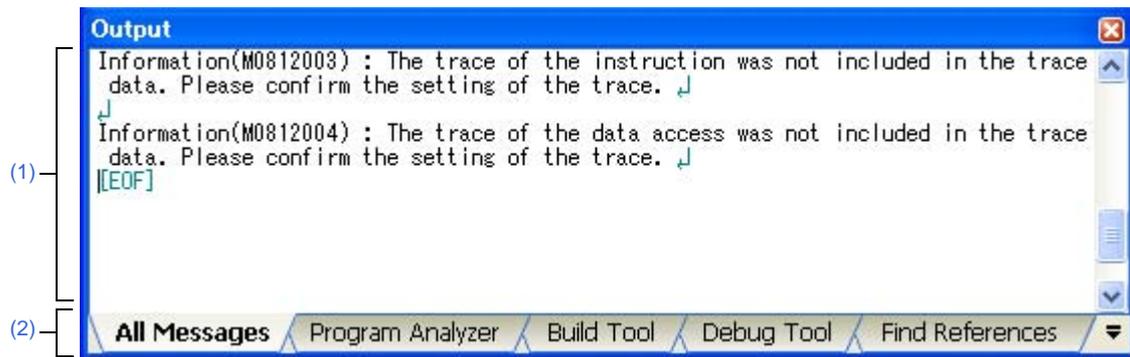
## Output panel

This panel is used to display operation logs for various components (analyze tool, design tool, build tool, debug tool, etc.) provided by CubeSuite+ and a list of locations that refer to the specified function/variable.

See "2.11 Display List of Referencing Location" for details on how to output lists of locations referencing a function or variable.

**Remark** This panel can be zoomed in and out by  in the tool bar, or by moving the mouse wheel forward or backward while holding down the [Ctrl] key.

Figure A-31. Output Panel



The following items are explained here.

- [How to open]
- [Description of each area]
- [[File] menu (Output panel-dedicated items)]
- [[Edit] menu (Output panel-dedicated items)]
- [Context menu]

### [How to open]

- From the [View] menu, select [Output].

### [Description of each area]

#### (1) Message area

This area displays operation logs for various components (analyze tool, design tool, build tool, debug tool, etc.) provided by CubeSuite+ and a list of places that refer to the specified function/variable.

For details on the contents displayed in this area, see the section explaining the corresponding tab.

#### (2) Tab selection area

The origin of messages is changed when each tab is selected.

The following tabs are available for the analyze tool.

- [All Messages] tab
- [Program Analyzer] tab
- [Find References] tab

**Remark** When the new message is output, "\*" mark is displayed to the left of the tab name.

**[[File] menu (Output panel-dedicated items)]**

The [File] menu for this panel provides the following items and functions.

Save Output- <i>Tab Name</i>	Overwrites the contents that are displayed on the currently selected tab to the preciously saved text file (*.txt). Note that when the file has never been saved or the file is write disabled, the same operation is applied as the selection in [Save Output- <i>Tab Name</i> As...].
Save Output- <i>Tab Name</i> As...	Opens the <a href="#">Save As dialog box</a> to newly save the contents that are displayed on the currently selected tab to the specified text file (*.txt).

**[[Edit] menu (Output panel-dedicated items)]**

The [Edit] menu for this panel provides the following items and functions.

Copy	Copies the contents of the selected range to the clipboard as character string(s).
Select All	Selects all the messages displayed on the currently selected tab.
Find...	Opens the Find and Replace dialog box.
Replace...	Opens the Find and Replace dialog box.

**[Context menu]**

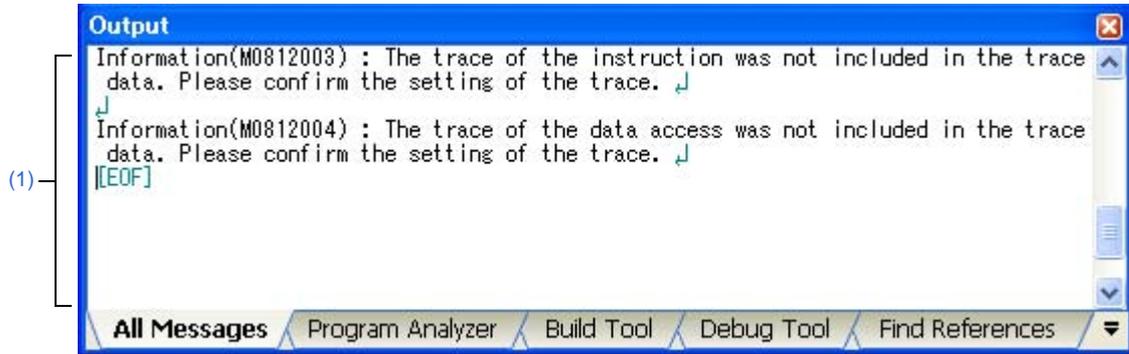
The context menu displayed by right-clicking on this panel provides the following items and functions.

Copy	Copies the contents of the selected range to the clipboard as character string(s).
Select All	Selects all the messages displayed on the currently selected tab.
Clear	Deletes all the messages displayed on the currently selected tab.
Tag Jump	Opens the Editor panel and jumps to the number of the corresponding line in the corresponding file of the message at the caret position.
Stop Searching	Cancels the current search operation. However, this item is not displayed when a search is not being executed.
Help for Message	Displays help for the message on the current caret position. This item only applies to warning messages and error messages.

**[All Messages] tab**

This tab is used to display operation logs for all components (analyze tool, design tool, build tool, debug tool, etc.) provided by CubeSuite+.

**Figure A-32. Output Panel: [All Messages] Tab**



The following items are explained here.

- [How to open]
- [Description of each area]

**[How to open]**

- From the [View] menu, select [Output].

**[Description of each area]**

**(1) Message area**

This area displays operation logs for all components (analyze tool, design tool, build tool, debug tool, etc.) provided by CubeSuite+. Note that messages that the analysis tool output during analysis are not displayed on this tab (these messages are displayed only on the [Program Analyzer] tab).

The colors of message display differ with the type of message as shown below (character colors and background colors depend on the configuration in the [General - Font and Color] category of the Option dialog box).

Message Type	Display Example (Default)		Description	
Normal message	AaBbCc	Character color	Black	Displayed with information notices
		Background color	White	
Warning message	AaBbCc	Character color	Blue	Displayed with warnings about operations
		Background color	Standard color	
Error message	AaBbCc	Character color	Red	Displayed when there is a critical error, or when execution is not possible due to a operational mistake
		Background color	Light gray	

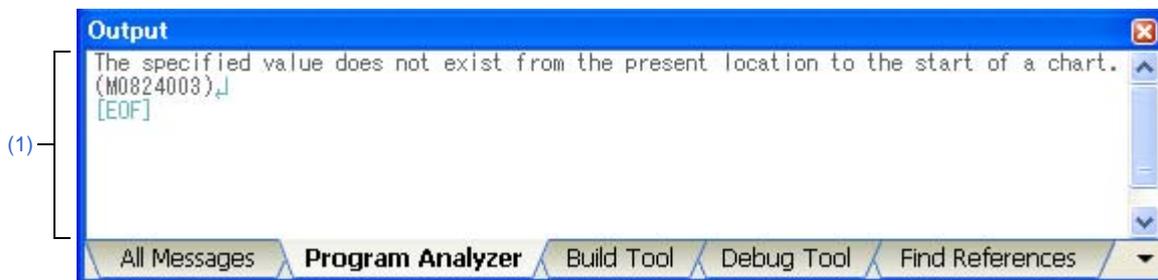
**Remarks 1.** When the output message is double-clicked, or the [Enter] key is pressed with the caret over the message, the Editor panel appears and the destination line number of the file is displayed.

2. Help with regard to the message in the line is shown by selecting [Help for Message] in the context menu or pressing the [F1] key while the caret is in the line where the warning message or the error message is displayed.
3. You can save the contents displayed on this tab to a text file (\*.txt) by selecting [Save Output-All Messages As...] from the [File] menu.

**[Program Analyzer] tab**

This tab is used to display only operation logs for the analyze tool out of those for various components (analyze tool, design tool, build tool, debug tool, etc.) provided by CubeSuite+.

**Figure A-33. Output Panel: [Program Analyzer] Tab**



The following items are explained here.

- [How to open]
- [Description of each area]

**[How to open]**

- From the [View] menu, select [Output].

**[Description of each area]**

**(1) Message area**

This area displays only operation logs for the analyze tool out of those for various components (analyze tool, design tool, build tool, debug tool, etc.) provided by CubeSuite+.

The colors of message display differ with the type of message as shown below (character colors and background colors depend on the configuration in the [General - Font and Color] category of the Option dialog box).

Message Type	Display Example (Default)		Description
Normal message	AaBbCc	Character color	Displayed with information notices
		Background color	
Warning message	AaBbCc	Character color	Displayed with warnings about operations
		Background color	
Error message	AaBbCc	Character color	Displayed when there is a critical error, or when execution is not possible due to a operational mistake
		Background color	

- Remarks 1.** When the output message is double-clicked, or the [Enter] key is pressed with the caret over the message, the Editor panel appears and the destination line number of the file is displayed.
- 2.** Help with regard to the message in the line is shown by selecting [Help for Message] in the context menu or pressing the [F1] key while the caret is in the line where the warning message or the error message is displayed.
- 3.** You can save the contents displayed on this tab to a text file (\*.txt) by selecting [Save Output-Program Analyzer As...] from the [File] menu.

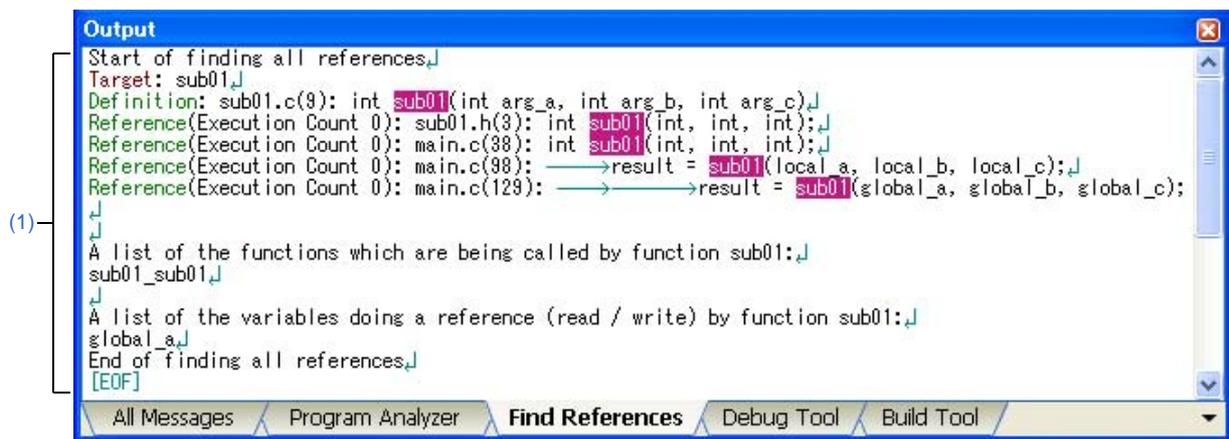
**[Find References] tab**

Use this tab to display a list of locations referencing the specified function/variable.

See "2.11 Display List of Referencing Location" for details on how to output lists of locations referencing a function or variable.

- Cautions 1.** This tab does not appear if the [Display List of Referencing Location](#) operation is never performed.
2. Code in C/C++ source files that is referenced but is eliminated by the preprocessor during compilation (e.g. via "#if" or "#ifdef" statements) is not output as referenced locations.
  3. [CC-RH][CC-RX][CX][NC30([Localised support](#))]  
Reference of a function that is assignment to function pointers is not output as referenced locations.
  4. Each time a search is made for referencing locations, the output list of referencing locations is cleared.

Figure A-34. Output Panel: [Find References] Tab



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)

**[How to open]**

- From the [View] menu, select [Output].

**[Description of each area]****(1) Message area**

This area displays the following list of locations referencing a function/variable on the [Function List panel/Variable List panel](#) (see "2.11 Display List of Referencing Location").

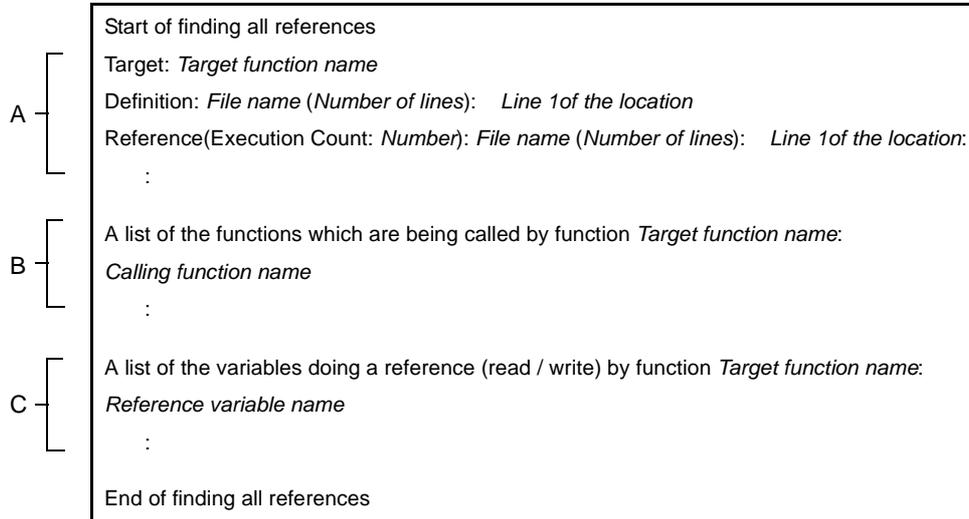
- Remarks 1.** When the function name/variable name on this tab is double-clicked, the Editor panel appears and the caret moves to the line where the function/variable is defined.
2. The color of the highlighted target function name and target variable name depends on the [Highlight] item in the [General - Font and Color] category of the Option dialog box.

- You can save the contents displayed on this tab to a text file (\*.txt) by selecting [Save Output-Find References As...] from the [File] menu.

**(a) Output format of list of locations referencing a function**

The contents of the search results consist of the following parts:

**Figure A-35. Output Format of List of Locations Referencing Function**



A	The location at which the target function is defined and a list of locations referencing the target function		
	Target	Indicates the name of the target function.	
	Definition	Indicates the location at which the target function is defined. Note that if the information cannot be acquired from the cross reference file, then "None" is displayed.	
		<i>File name (Number of lines)</i>	Indicates the file name <sup>Note 1</sup> at which the location in question exists. The number in parentheses indicates the number of lines (line number) in the file.
		<i>Line 1 of the location</i>	Displays a 1-line excerpt from the file at the location. At this time, the target function name is highlighted.
	Reference	Lists the locations referencing the target function. If there is the <a href="#">Dynamic analysis information</a> , then the number of times the target function is executed is included in parentheses ( ). Note that if the information cannot be acquired from the cross reference file, then "None" is displayed.	
		<i>File name (Number of lines)</i>	Indicates the file name <sup>Note 1</sup> at which the location in question exists. The number in parentheses indicates the number of lines (line number) in the file.
		<i>Line 1 of the location</i>	Displays a 1-line excerpt from the file at the location. At this time, the target function name is highlighted.
	B	A list of functions called within the target function	
		<i>Calling function name</i>	Lists the names of functions called within the target function. If no functions are called, then "None" is displayed.
C	A list of variables referenced (read/written) inside the target function		
	<i>Reference variable name</i>	Lists the names of variables referenced (read/written) inside the target function <sup>Note 2</sup> . If no variables are referenced, then "None" is displayed.	



Definition	Indicates the location at which the target variable is defined. Note that if the information cannot be acquired from the cross reference file, then "None" is displayed.	
	File name (Number of lines)	Indicates the file name <sup>Note</sup> at which the location in question exists. The number in parentheses indicates the number of lines (line number) in the file.
	Line 1 of the location	Displays a 1-line excerpt from the file at the location. At this time, the target variable name is highlighted.
Reference	Lists the locations referencing the target variable. If there is the <a href="#">Dynamic analysis information</a> , then the number of times the target variable has been read/written is included in parentheses (.). Note that if the information cannot be acquired from the cross reference file, then "None" is displayed.	
	File name (Number of lines)	Indicates the file name <sup>Note</sup> at which the location in question exists. The number in parentheses indicates the number of lines (line number) in the file.
	Line 1 of the location	Displays a 1-line excerpt from the file at the location. At this time, the target variable name is highlighted.

**Note** When the [File path] item is being displayed on the [Variable List panel](#) (this item is not displayed by default), the absolute path of the file appears.

- Remarks 1.** How lines with variables described in "extern declaration" are handled depending on the compiler currently used is as follows:
- **[CC-RH][CC-RX][CX][NC30(Localised support)]**  
A line with a variable described in "extern declaration" is handled as "Reference".
  - **[CA850]**  
A line with a variable described in "extern declaration" is handled as "Definition".
  - **[CA78K0R][CA78K0]**  
A line with a variable described in "extern declaration" is handled as "Reference".  
Furthermore, if a line with a variable described in "extern declaration" and a line with the variable definition exist in the same file, then "None" is displayed in "Definition".
- 2. [CX]**  
In the definition of variables, a line with an assignment statement (e.g. "int variable = 10") is handled as "Reference".

**Figure A-39. Example of List of Locations Referencing Variable (without Dynamic Analysis Information)**

```

Target: global_a;
Definition: main.h(3): extern int global_a;
Definition: main.c(10): int global_a = 10;
Reference: main.c(95): →global_pointer = &global_a;
Reference: main.c(112): →global_a = 0;
Reference: main.c(125): →global_a++;
Reference: main.c(129): →result = sub01(global_a, global_b, global_c);
Reference: sub01.c(15): →result = tmp + global_a;
    
```

**Figure A-40. Example of List of Locations Referencing Variable (with Dynamic Analysis Information)**

```

Target: global_a;
Definition: main.h(3): extern int global_a;
Definition: main.c(10): int global_a = 10;
Reference(Read/Write Count 0): main.c(95): →global_pointer = &global_a;
Reference(Read/Write Count 0): main.c(112): →global_a = 0;
Reference(Read/Write Count 0): main.c(125): →global_a++;
Reference(Read/Write Count 0): main.c(129): →result = sub01(global_a, global_b, global_c);
Reference(Read/Write Count 0): sub01.c(15): →result = tmp + global_a;
    
```

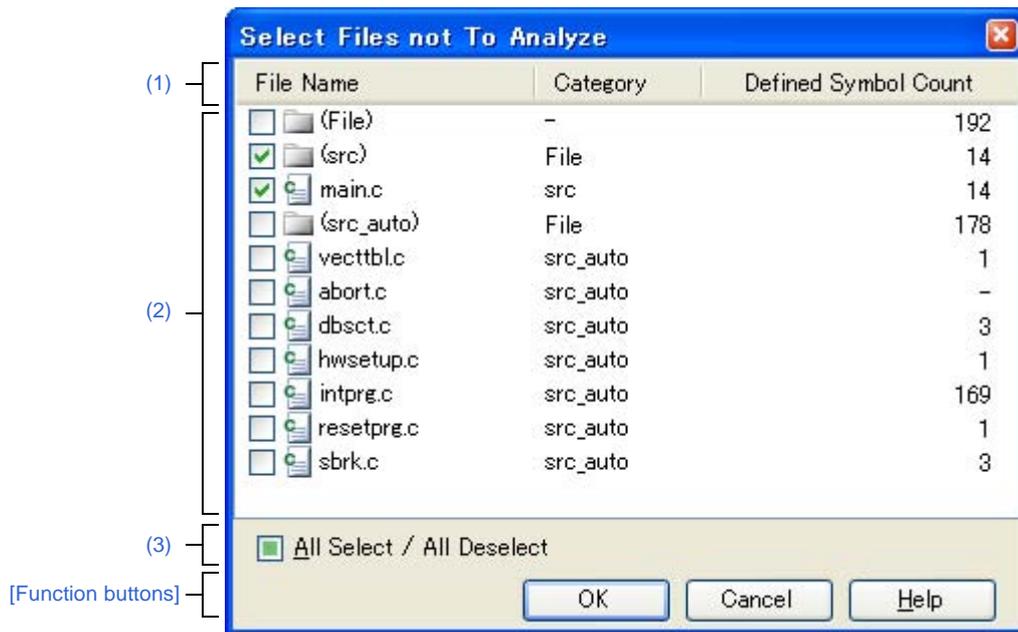
### Select Files not To Analyze dialog box

This dialog box is used to specify the files not to be analyzed by the analyze tool (see "1.1.1 What is analyzed").

The function information and variable information in the files specified not to be analyzed on this dialog box are not displayed in the following panels:

- [Function List panel](#)
- [Variable List panel](#)
- [Call Graph panel](#)
- [Class/Member panel](#)

Figure A-41. Select Files not To Analyze Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

#### [How to open]

- On the [\[Settings\]](#) tab in the [Property panel](#), select the [\[Files not to analyze\]](#) property in the [\[Analysis Object\]](#) category, and then click the [\[...\]](#) button.

#### [Description of each area]

##### (1) Header area

This area displays the column name (File Name, Category, and Defined Symbol Count) of the file information.

You can change the display order of the columns displayed by dragging and dropping the column name to an arbitrary column.

Furthermore, you can sort the contents in the [File information display area](#) by clicking each column name (clicking repeatedly cycles the sort order between ascending, descending and the default display (i.e. in the order on the project tree)). In this case, if the items to sort are numbers (decimal or hexadecimal), then the file information is sorted by numerical value, for other types of data (e.g. strings), the file information is sorted by character code.

**Caution** The display order of the columns and the order of the contents in the file information area cannot be restored. This dialog box is always opened in the default condition.

**(2) File information display area**

This area displays a list of C/C++ source file names currently being registered in the project and their information.

**(a) [File Name]**

Displays the C/C++ source file name or the category name (surrounded by "()").

Specify the check boxes of each file name not to analyze that file.

<input checked="" type="checkbox"/>	Does not analyze this file.
<input type="checkbox"/>	Analyzes this file.

- Remarks 1.** If a check box of a category name is cleared, then all of the check boxes of the file names that belong to the category will be cleared.
- 2.** When the mouse cursor is hovered over a file name, the full path of the file and the path including the tree node name that is displayed in the project tree appear in a pop-up.

**(b) [Category]**

Displays the category name to which the corresponding C/C++ source file is registered in the project tree. When a category name is displayed in the [File Name], "File" is displayed (if it is a root category, "-" is displayed).

**(c) [Defined Symbol Count]**

Displays the total number (decimal number) of the functions and variables that are defined in the corresponding C/C++ source file or category.

However, if the number of the defined symbols cannot be obtained, then "-" will be displayed.

**(3) [All Select / All Deselect] check box**

Select this check box to select all of the check boxes in [File] column.

Conversely, remove the check of this check box to remove all of the checks in [File] column.

<input checked="" type="checkbox"/>	Indicates that all of the check boxes in [File] column is being selected.
<input type="checkbox"/>	Indicates that all of the check boxes in [File] column is not being selected or all of the check boxes in [File] column is not being cleared.
<input type="checkbox"/>	Indicates that no check box in [File] column is being selected (default).

**[Function buttons]**

Button	Function
OK	Sets the specified file(s) to the file(s) not to be analyzed then closes the dialog box.
Cancel	Ignores the setting and closes this dialog box.
Help	Displays the help of this dialog box.

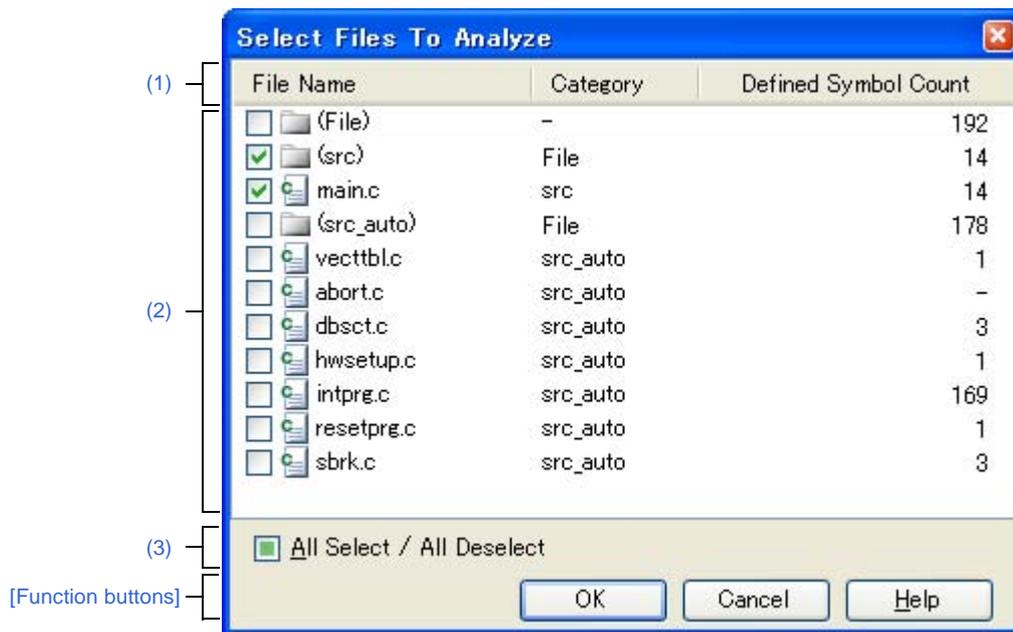
### Select Files To Analyze dialog box

This dialog box is used to specify the files to be analyzed by the analyze tool (see "1.1.1 What is analyzed").

Only the function information and variable information in the files specified to be analyzed on this dialog box are displayed in the following panels:

- [Function List panel](#)
- [Variable List panel](#)
- [Call Graph panel](#)
- [Class/Member panel](#)

Figure A-42. Select Files To Analyze Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

#### [How to open]

- On the [\[Settings\]](#) tab in the [Property panel](#), select the [\[Files to analyze\]](#) property in the [\[Analysis Object\]](#) category, and then click the [\[...\]](#) button.

#### [Description of each area]

##### (1) Header area

This area displays the column name (File Name, Category, and Defined Symbol Count) of the file information.

You can change the display order of the columns displayed by dragging and dropping the column name to an arbitrary column.

Furthermore, you can sort the contents in the [File information display area](#) by clicking each column name (clicking repeatedly cycles the sort order between ascending, descending and the default display (i.e. in the order on the project tree)). In this case, if the items to sort are numbers (decimal or hexadecimal), then the file information is sorted by numerical value, for other types of data (e.g. strings), the file information is sorted by character code.

**Caution** The display order of the columns and the order of the contents in the file information area cannot be restored. This dialog box is always opened in the default condition.

**(2) File information display area**

This area displays a list of C/C++ source file names currently being registered in the project and their information.

**(a) [File Name]**

Displays the C/C++ source file name or the category name (surrounded by "()").

Specify the check boxes of each file name to analyze that file.

<input checked="" type="checkbox"/>	Analyzes this file.
<input type="checkbox"/>	Does not analyze this file.

- Remarks 1.** If a check box of a category name is cleared, then all of the check boxes of the file names that belong to the category will be cleared.
- 2.** When the mouse cursor is hovered over a file name, the full path of the file and the path including the tree node name that is displayed in the project tree appear in a pop-up.

**(b) [Category]**

Displays the category name to which the corresponding C/C++ source file is registered in the project tree. When a category name is displayed in the [File Name], "File" is displayed (if it is a root category, "-" is displayed).

**(c) [Defined Symbol Count]**

Displays the total number (decimal number) of the functions and variables that are defined in the corresponding C/C++ source file or category.

However, if the number of the defined symbols cannot be obtained, then "-" will be displayed.

**(3) [All Select / All Deselect] check box**

Select this check box to select all of the check boxes in [File] column.

Conversely, remove the check of this check box to remove all of the checks in [File] column.

<input checked="" type="checkbox"/>	Indicates that all of the check boxes in [File] column is being selected.
<input type="checkbox"/>	Indicates that all of the check boxes in [File] column is not being selected or all of the check boxes in [File] column is not being cleared.
<input type="checkbox"/>	Indicates that no check box in [File] column is being selected (default).

**[Function buttons]**

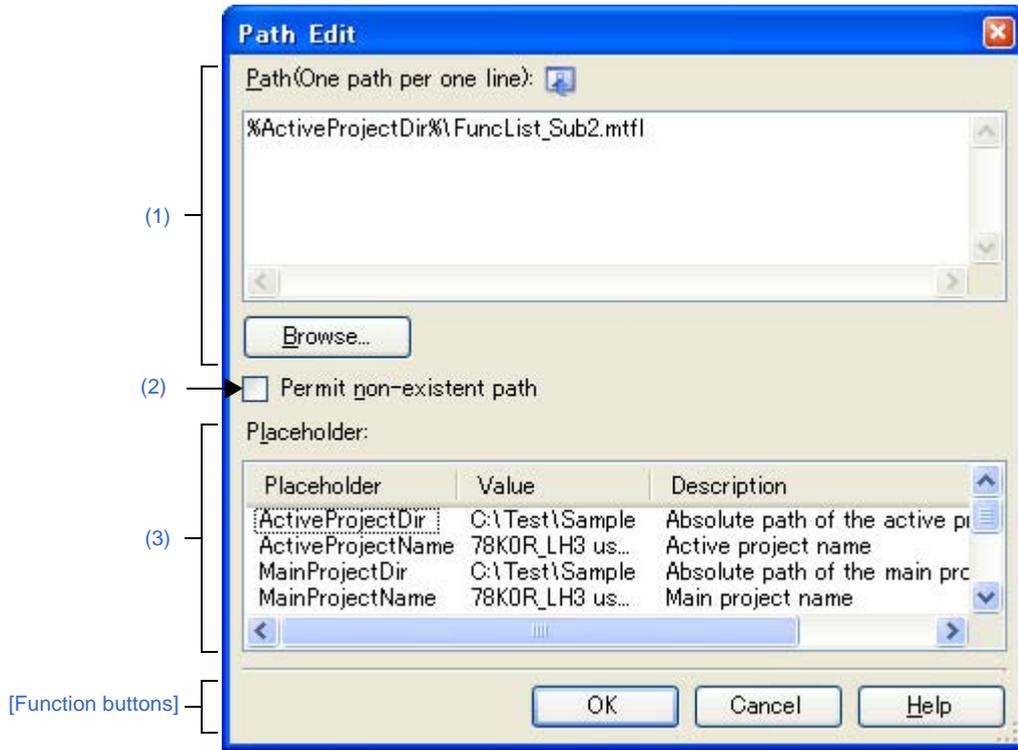
Button	Function
OK	Sets the specified file(s) to the file(s) to be analyzed then closes the dialog box.
Cancel	Ignores the setting and closes this dialog box.
Help	Displays the help of this dialog box.

**Path Edit dialog box**

This dialog box is used to specify the file when importing an information file (function list file (\*.mtfl)/variable list file (\*.mtvl)).

See "2.12 Import or Export Information File" for details on the import function.

Figure A-43. Path Edit Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

**[How to open]**

- On the [Settings] tab in the Property panel, select the [Import files] property in the [Import / Export] category, and then click the [...] button.

**[Description of each area]**

**(1) Path edit area**

This area is used to edit or add the file name including the path.

**(a) [Path(One path per one line)]**

Specify each function list file (\*.mtfl)/variable list file (\*.mtvl) to be imported, one to a line, including the path (up to 259 characters per line, and up to 64 lines can be specified). If you specify a relative path, specify the project folder as the base.

The file names including the path can be added by one of the following methods.

- Click the [Browse...] button, and then select a file.

- Drag and drop a file using such as Explorer.

**(b) [Browse...] button**

Opens a dialog box to specify files to be imported.  
The selected file name is added to [Path(One path per one line)].

**Caution** If an extremely long absolute path is specified as a relative path, an error could occur when clicking the [OK] button. In this case, designate the absolute path.

**(2) [Permit non-existent path] check box**

<input checked="" type="checkbox"/>	Checks whether the specified path and file name exist when the [OK] button is clicked.
<input type="checkbox"/>	Does not check whether the specified path and file name exist when the [OK] button is clicked (default).

**(3) [Placeholder] area**

This area is used to list the placeholders that can be specified to the calling property of this dialog box (ascending order). Double-click a line to display the placeholder adding "%" as a pre-fix/post-fix in the path edit area.

**(a) [Placeholder]**

Displays the placeholder.

**(b) [Value]**

Displays the character strings that is replaced by the placeholder.

**(c) [Description]**

Displays the description of the placeholder.

**Caution** This area appears only when the calling property of this dialog box supports the placeholder function.

**Remark** The available placeholders depend on the calling property of this dialog box. See the explanation of the calling property for details on the available placeholders.

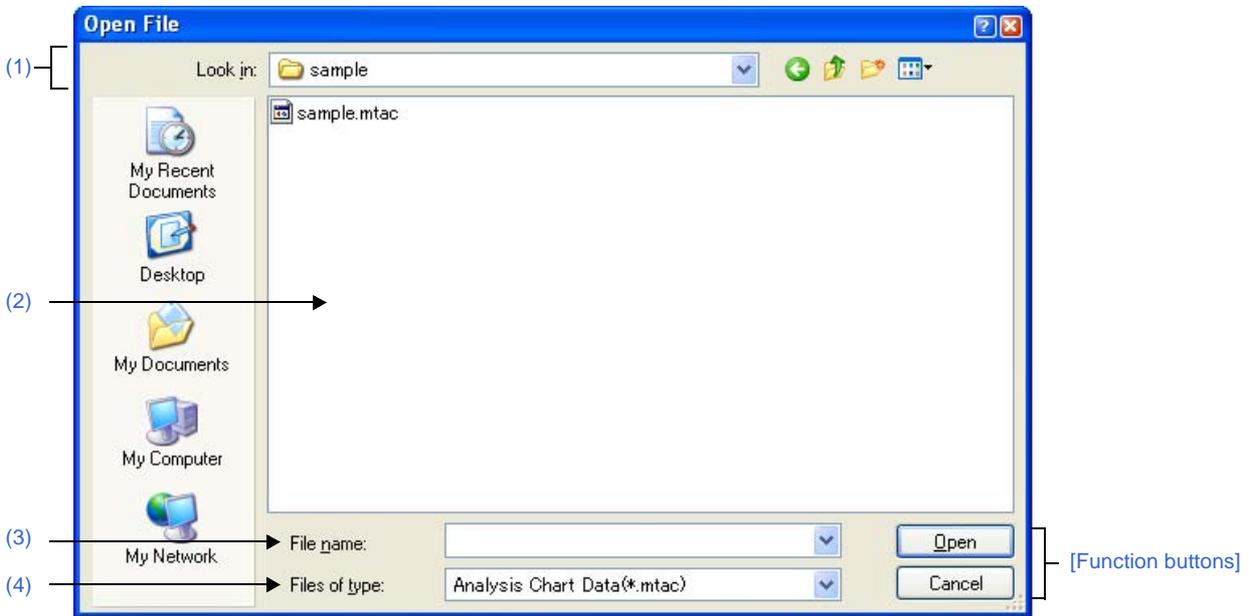
**[Function buttons]**

Button	Function
OK	Sets the specified file(s) to the [Import files] property then closes the dialog box.
Cancel	Ignores the setting and closes this dialog box.
Help	Displays the help of this dialog box.

**Open File dialog box**

This dialog box is used to select the analysis chart data file (\*.mtac) to restore graphs.

**Figure A-44. Open File Dialog Box**



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

**[How to open]**

- On the [Variable Value Changing] tab in the Property panel, select the [Analysis chart data file] property in the [General] category, and then click the [...] button.

**[Description of each area]**

**(1) [Look in] area**

Select the folder which contains the file you want to open.

**(2) List of files area**

This area shows the list of files that meet the conditions specified in [Look in] and [Files of type] areas.

**(3) [File name] area**

Specify the name of the file to be opened.

**(4) [Files of type] area**

Select the type of the file to be opened.

Analysis Chart Data (*.mtac)	Analysis chart data file
------------------------------	--------------------------

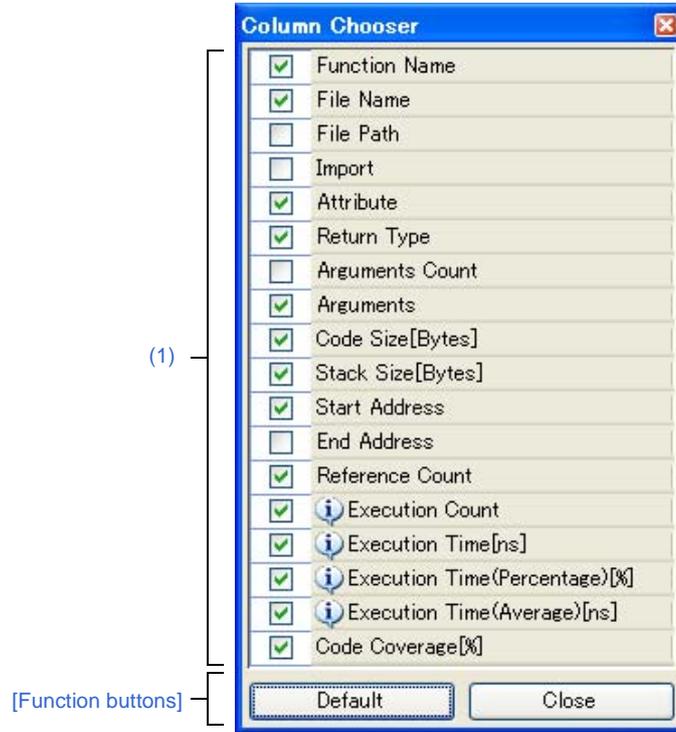
**[Function buttons]**

Button	Function
Open	Loads the specified file, and then restores graphs.
Cancel	Closes this dialog box.

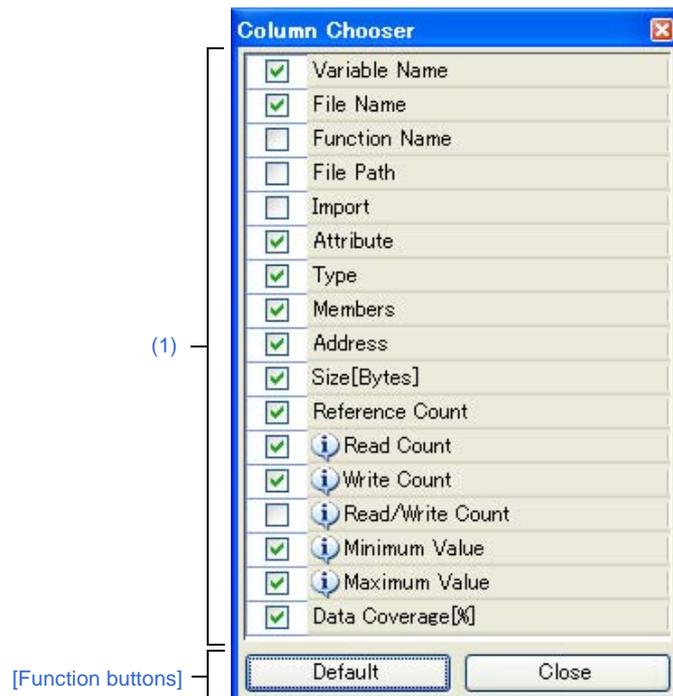
**Column Chooser dialog box**

This dialog box is used to sort, show/hide the items (columns) displayed on the [Function List panel](#)/[Variable List panel](#). If the [Customize Display Method](#) operation has been performed on one of the panels, then this dialog can also be used to reset these customizations to the default settings.

**Figure A-45. Column Chooser Dialog Box (For Function List Panel)**



**Figure A-46. Column Chooser Dialog Box (For Variable List Panel)**



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

**[How to open]**

- On the [Function List panel](#), click the  button.
- On the [Variable List panel](#), click the  button.

**[Description of each area]**

**(1) Item list area**

This area displays a list of all items (columns) that can be displayed in the [Function List panel/Variable List panel](#) (the items (columns) that can be displayed differ according to the microcontroller and the debug tool used). The display order of the items in this list, and the state of checkboxes, are the same as the current sort order and visible/hidden status in the corresponding panel. Specify the check boxes of each item to show or hide that item on the panel.

<input checked="" type="checkbox"/>	Displays this item on the panel.
<input type="checkbox"/>	Does not display this item on the panel.
	Indicates that a message relating to information about this item has been output to the <a href="#">Output panel</a> . Hover the mouse over it to display a pop-up with the last message to be output.

**[Function buttons]**

Button	Function
Default	Resets the display order and visible/hidden settings for each item in the <a href="#">Function List panel/Variable List panel</a> to their default values <sup>Note</sup> . Note that all display locking, display filtering, and other customization will be cleared.
Close	Closes this dialog box.

**Note** The default display status for each item is as follows:  
The order of [Item] in the table corresponds to the display order of items (columns) on each panel.  
Note that items displayed differ depending on the microcontroller and debug tool used (see "(2) [Information area](#)" of the Function List panel/"(2) [Information area](#)" of the Variable List panel).

**Table A-2. Default Display Status (Function List Panel)**

Item	Status	
	While Disconnecting from Debug Tool	While Connecting to Debug Tool
Function Name	Display	Display
Class Name	Display	Display
Namespace	Non-display	Non-display
File Name	Display	Display
File Path	Non-display	Non-display

Item	Status	
	While Disconnecting from Debug Tool	While Connecting to Debug Tool
PM Information [RH850] PE Information [V850E2]	Display	Display
Import	Non-display	Non-display
Access Specifier	Display	Display
Attribute	Display	Display
Return Type	Display	Display
Arguments Count	Non-display	Non-display
Arguments	Display	Display
Code Size[Bytes]	Display	Display
Stack Size[Bytes]	Display	Display
Start Address	Display	Display
End Address	Non-display	Non-display
Reference Count	Display	Display
Execution Count	Non-display	Display
Execution Time[ <i>unit</i> ]	Non-display	Display
Execution Time(Percentage)[%]	Non-display	Display
Execution Time(Average)[ <i>unit</i> ]	Non-display	Display
Code Coverage[%]	Non-display	Display

**Table A-3. Default Display Status (Variable List Panel)**

Item	Status	
	While Disconnecting from Debug Tool	While Connecting to Debug Tool
Variable Name	Display	Display
Class Name	Display	Display
Namespace	Non-display	Non-display
File Name	Display	Display
Function Name	Non-display	Non-display
File Path	Non-display	Non-display
PM Information [RH850] PE Information [V850E2]	Non-display	Display
Import	Non-display	Non-display
Access Specifier	Display	Display
Attribute	Display	Display
Type	Display	Display
Members	Display	Display
Address	Display	Display
Size[Bytes]	Display	Display
Reference Count	Display	Display

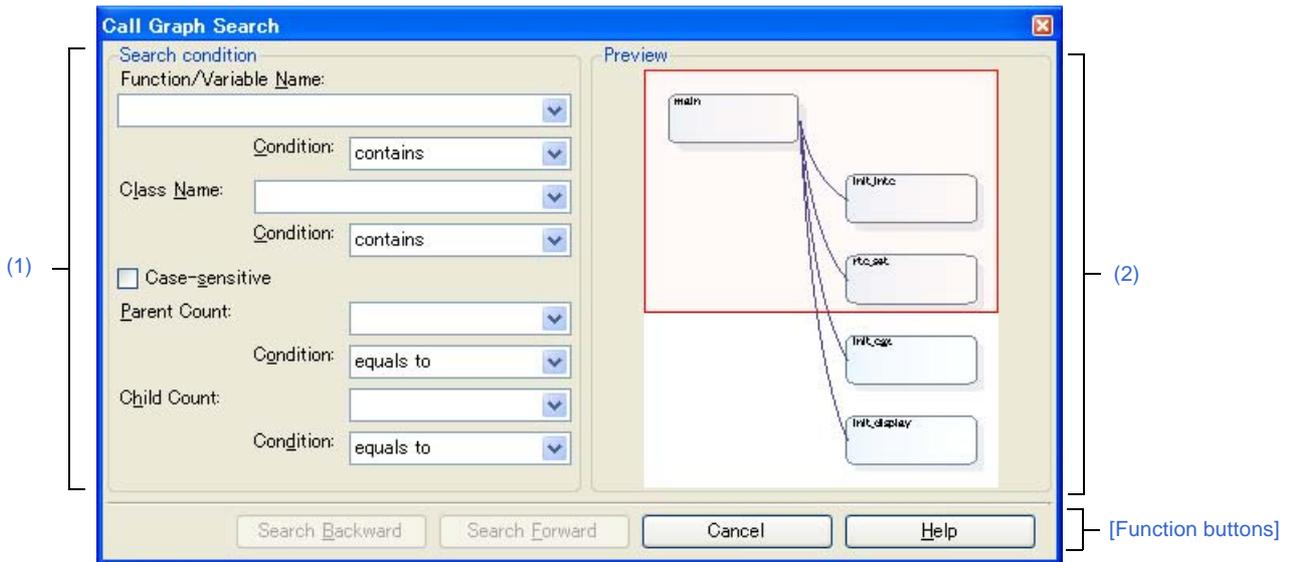
Item	Status	
	While Disconnecting from Debug Tool	While Connecting to Debug Tool
Read Count	Non-display	Display
Write Count	Non-display	Display
Read/Write Count	Non-display	Non-display
Minimum Value	Non-display	Display
Maximum Value	Non-display	Display
Data Coverage[%]	Non-display	Display

**Call Graph Search dialog box**

This dialog box is used to search for a function/variable existing in the call graph displayed in the [Call Graph panel](#).

**Caution** Searches can be performed only for functions/variables (function boxes/variable boxes) that appear in a call graph.

**Figure A-47. Call Graph Search Dialog Box**



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

**[How to open]**

- On the toolbar of the [Call Graph panel](#), click the  button.
- Focus the [Call Graph panel](#), and then select [Find...] from the [Edit] menu.

**[Description of each area]**

**(1) [Search condition] area**

Configure the search conditions.

If multiple search conditions are specified, then only a function/variable meeting all conditions will be searched.

**(a) [Function/Variable Name]**

Specify a function name/variable name to search.

You can either type character strings directly into the text box (up to 2046 characters), or select one from the input history via the drop-down list (up to 10 items).

The previous specified function name/variable name is displayed by default (if it does not exist, then this is left blank).

<1> [Condition]

From the following drop-down list, select the condition for the character strings specified with [Function/Variable Name].

equals	Searches for the function names/variable names that exactly matches the specified string.
begins with	Searches for function names/variable names starting with the specified string.
ends with	Searches for function names/variable names ending with the specified string.
contains	Searches for function names/variable names containing the specified string (default).

**Remark** You can also specify the function name/variable name with any one of the following methods.

- Drag and drop an arbitrary row of the [Function List panel/Variable List panel](#) to this area.
- Drag and drop an arbitrary character strings to this area.

(b) [Class Name] [CC-RX]

If required as a search parameter for a function/variable, you can specify the class name to which a member function/member variable to search belongs.

You can either type character strings directly into the text box (up to 2046 characters), or select one from the input history via the drop-down list (up to 10 items).

The previous specified class name is displayed by default (if it does not exist, then this is left blank).

<1> [Condition]

From the following drop-down list, select the condition for the character strings specified with [Class Name].

equals	Searches for the function names/variable names that belong to the class exactly matching the specified string.
begins with	Searches for function names/variable names that belong to the class starting with the specified string.
ends with	Searches for function names/variable names that belong to the class ending with the specified string.
contains	Searches for function names/variable names that belong to the class containing the specified string (default).

**Remark** You can also specify the class name with any one of the following methods.

- Drag and drop an arbitrary character strings to this area.

(c) [Case-sensitive]

Specify whether to search for characters specified in [Function/Variable Name]/[Class Name] [CC-RX] in case-sensitive.

<input checked="" type="checkbox"/>	Searches with the specified characters in case-sensitive.
<input type="checkbox"/>	Searches with the specified characters in not case-sensitive (default).

(d) [Parent Count]

If required as a search parameter for a function, you can specify the number of parent functions to search.

You can either type a value directly into the text box (settable range: 0 to 65535), or select one from the input history via the drop-down list (up to 10 items).

The previous specified value is displayed by default (if it does not exist, then this is left blank).

<1> **[Condition]**

From the following drop-down list, select the condition for the numeric value specified with [Parent Count].

is greater than	Searches for the names of functions with more (greater than) the specified number of parent functions.
is greater than or equals to	Searches for the names of functions with at least as many (greater than or equal to) the specified number of parent functions.
equals to	Searches for the names of functions with the same number of (equal to) the specified number of parent functions (default).
is less than or equals to	Searches for the names of functions with at least as less (greater than or equal to) the specified number of parent functions.
is less than	Searches for the names of functions with less (greater than) the specified number of parent functions.

(e) **[Child Count]**

If required as a search parameter for a function, you can specify the number of child functions to search. You can either type a value directly into the text box (settable range: 0 to 65535), or select one from the input history via the drop-down list (up to 10 items).  
The previous specified value is displayed by default (if it does not exist, then this is left blank).

<1> **[Condition]**

From the following drop-down list, select the condition for the numeric value specified with [Child Count].

is greater than	Searches for the names of functions with more (greater than) the specified number of child functions.
is greater than or equals to	Searches for the names of functions with at least as many (greater than or equal to) the specified number of child functions.
equals to	Searches for the names of functions with the same number of (equal to) the specified number of child functions (default).
is less than or equals to	Searches for the names of functions with at least as less (greater than or equal to) the specified number of child functions.
is less than	Searches for the names of functions with less (greater than) the specified number of child functions.

(2) **[Preview] area**

This area displays all part of the call graph.  
If the area currently displayed in the [Call Graph panel](#) is a portion of the entire call graph, then that area is shown in a red frame.

[Function buttons]

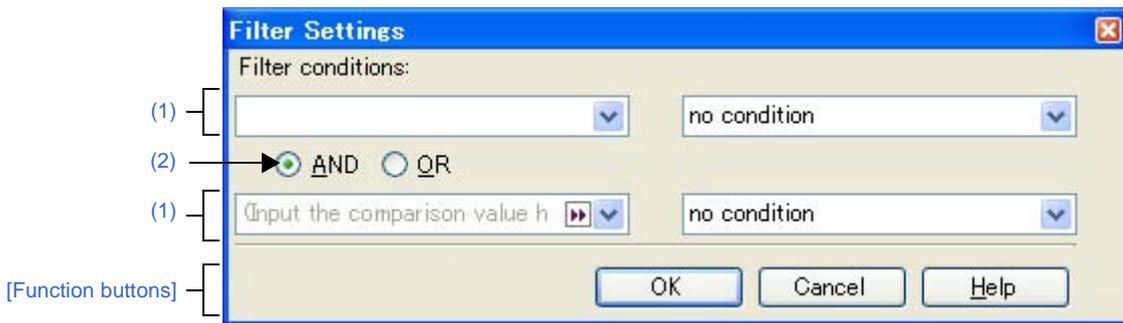
Button	Function
Search Backward	<p>The search uses the specified conditions, starting from the function name/variable name at the bottom of the call graph, and proceeding upward. The corresponding function box/variable box is highlighted.</p> <p>If a function box/variable box has been selected on the <a href="#">Call Graph panel</a>, then the search starts from the corresponding function/variable, moving upward.</p> <p>Note that if an illegal value is specified, or the program is executing, then a message will appear, and the function/variable search will not be performed. If focus moves to this dialog box from a panel other than the <a href="#">Call Graph panel</a> or the Call Graph panel is hidden, then this button will be disabled.</p>
Search Forward	<p>The search uses the specified conditions, starting from the function name/variable name at the top of the call graph, and proceeding downward. The corresponding function box/variable box is highlighted.</p> <p>If a function box/variable box has been selected on the <a href="#">Call Graph panel</a>, then the search starts from the corresponding function/variable, moving downward.</p> <p>Note that if an illegal value is specified, or the program is executing, then a message will appear, and the function/variable search will not be performed. If focus moves to this dialog box from a panel other than the <a href="#">Call Graph panel</a> or the Call Graph panel is hidden, then this button will be disabled.</p>
Cancel	Cancels the function search and closes this dialog box.
Help	Displays the help for this dialog box.

### Filter Settings dialog box

This dialog box is used to set the filter conditions when filtering the display of analysis information on the [Function List panel/Variable List panel](#).

- Caution** The filtering configured via this dialog box and the [Filtering results linked to panel](#) are mutually exclusive functions.
- For this reason, the two filtering functions here cannot be enabled simultaneously (when one of the filters is active, setting the other filter will disable all filters set to date).
- See "[2.6.5 Filter the analysis information](#)" for details on the display-filter.

Figure A-48. Filter Settings Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

#### [How to open]

- On the [Function List panel/Variable List panel](#), click the filter icon ( / ), and then select the [(Custom)].

#### [Description of each area]

Configure the filter conditions.

In the [First condition setting area \(top\)/Second condition setting area \(bottom\)](#), you can specify up to two conditions. Use the [Logical operator buttons](#) ([AND]/[OR]) to set the two conditions as a single filter condition.

##### (1) First condition setting area (top)/Second condition setting area (bottom)

###### (a) Comparison value (left combo box)

You can either type the comparison value of for the filter (number or string) directly via the keyboard (up to 2048 characters), or select one from the input history via the drop-down list (maximum 10 items). By default, the comparison value set in the current target item is displayed.

###### (b) Condition (right combo box)

Select the condition for the comparison value specified in (a) above, from the drop-down list. By default, the condition set in the current target item is selected.

Item	Comparison Value	
	Numerical Value	Character String
no condition <sup>Note</sup>	Compared as numerical value	Compared as character string
equals		
does not equal		
is greater than		
is greater than or equal to		
is less than		
is less than or equal to		
begins with	Compared as character string	
does not begin with		
ends with		
does not end with		
contains		
does not contain		

**Note** If [no condition] is selected, then the comparison value will be ignored (not set as a condition).

**Caution** Use the top condition-specification area if you only wish to specify one filtering condition.

**(2) Logical operator buttons**

Select the logical relationship holding between the conditions specified in the [First condition setting area \(top\)](#)/[Second condition setting area \(bottom\)](#) via the following option button.

AND	Only information meeting both condition 1 and condition 2 is displayed. This is mutually exclusive to the [OR] button.
OR	Only information meeting either condition 1 or condition 2 is displayed. This is mutually exclusive to the [AND] button.

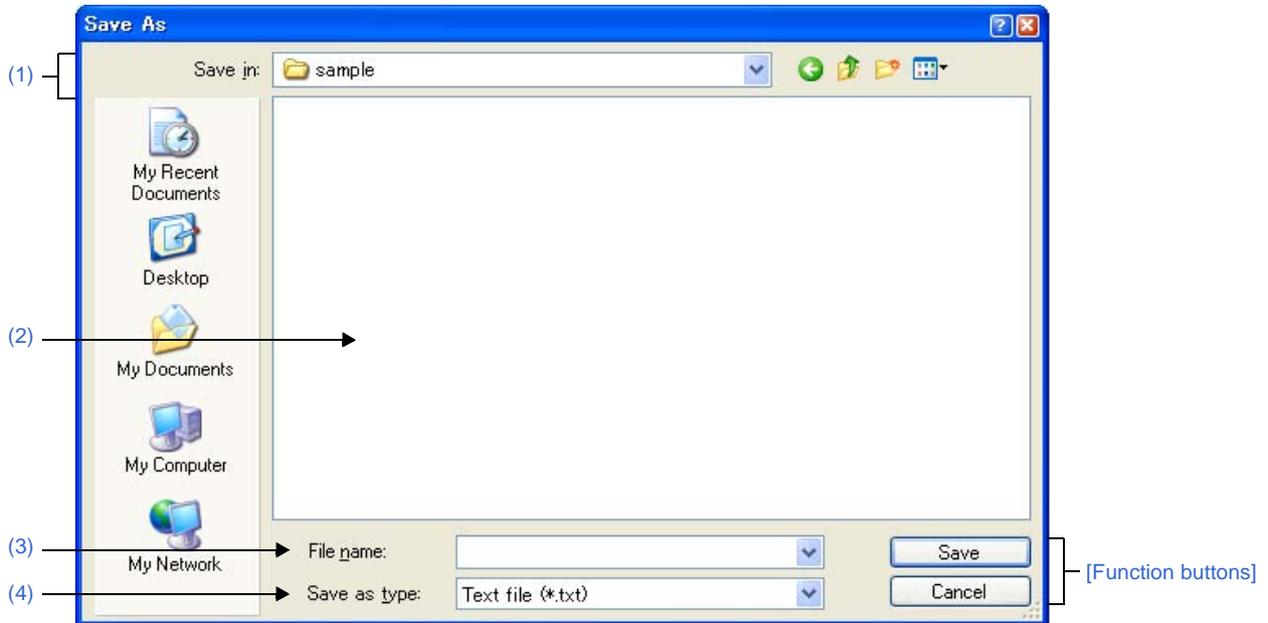
**[Function buttons]**

Button	Function
OK	Displays the analysis information in the <a href="#">Function List panel/Variable List panel</a> under the specified filtering conditions. If the <a href="#">Filtering results linked to panel</a> was performed, then all display-filters linked to the panel conducted prior to this are deleted.
Cancel	Ignores the setting and closes this dialog box.
Help	Displays the help of this dialog box.

### Save As dialog box

This dialog box is used to save the contents of the [Function List panel](#)/[Variable List panel](#)/[Analysis Chart panel](#)/[Call Graph panel](#)/[Variable Value Changing Chart \(zoomed in\) panel](#)/[Output panel](#) to a specified file.

Figure A-49. Save As Dialog Box



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Function buttons\]](#)

#### [How to open]

- Focus the [Function List panel](#), and then select [Save Function List Data As...] from the [File] menu.
- Focus the [Variable List panel](#), and then select [Save Variable List Data As...] from the [File] menu.
- Focus the [Analysis Chart panel](#), and then select [Save Analysis Chart Data As...] from the [File] menu.
- Focus the [Call Graph panel](#), and then select [Save Call Graph Data As...] from the [File] menu.
- Focus the [Variable Value Changing Chart \(zoomed in\) panel](#), and then select the [Save] button.
- Focus the [Output panel](#), and then select [Save Output - *Tab Name* As...] from the [File] menu.

#### [Description of each area]

##### (1) [Save in] area

Select the folder to save the panel contents in the file.

##### (2) List of files area

This area displays a list of files matching the conditions selected in [Save in] and [Save as type].

##### (3) [File name] area

Specify the file name to save.

**(4) [Save as type] area**

The following file types are displayed. Select the type of file to save.

Note that when the CSV format is selected, the data is saved with entries separated by commas (.). If the data contains commas, each entry is surrounded by double quotes (" ") in order to avoid illegal formatting.

**(a) From the [Function List panel](#)**

Text file (*.txt)	Text format
CSV(Comma-Separated Variables) (*.csv)	CSV format
Microsoft Office Excel Workbook (*.xls)	Microsoft Office Excel book format
Function List file (*.mtfl)	File format to import the function information (see " <a href="#">2.12 Import or Export Information File</a> ".)

**Caution** Only items (column) and analysis information (rows) currently being displayed in the panel will be saved.

**(b) From the [Variable List panel](#)**

Text file (*.txt)	Text format
CSV(Comma-Separated Variables) (*.csv)	CSV format
Microsoft Office Excel Workbook (*.xls)	Microsoft Office Excel book format
Variable List file (*.mtvl)	File format to import the variable information (see " <a href="#">2.12 Import or Export Information File</a> ".)

**Caution** Only items (column) and analysis information (rows) currently being displayed in the panel will be saved.

**(c) From the [Analysis Chart panel](#)/[Variable Value Changing Chart \(zoomed in\) panel](#)**

Text file (*.txt)	Text format
CSV(Comma-Separated Variables) (*.csv)	CSV format
Microsoft Office Excel Workbook (*.xls)	Microsoft Office Excel book format
Analysis Chart Data (*.mtac) <sup>Note 1</sup>	Analysis chart data file
Bitmap (*.bmp)	Bitmap format (32-bit) (graphic file format)
JPEG (*.jpg)	JPEG format (graphic file format)
PNG (*.png)	PNG format (graphic file format)
EMF (*.emf) <sup>Note 2</sup>	EMF format (Graphic file format)

- Notes 1.** This format is only applied to the [\[Variable Value Changing Chart\] tab](#) of the [Analysis Chart panel](#).
- 2.** Only the area of graph rendering is saved (trigger marks, channel information, etc. cannot be saved).

**Cautions 1.** For the [Analysis Chart panel](#), only the contents of the currently displayed tab will be saved.

2. When the graphic file format is selected, only the part currently being displayed in the panel will be saved.

The items to be saved in the format other than the graphic file are as follows (except for the analysis chart data file (\*.mtac), see "(6) Save the graph data to restore the graph"):

- [Variable Value Changing Chart] tab

- Real-time sampling analysis

[Items]: Time Value(Variable name) Value(Variable name) ...

**Remark** Value(Variable name): Outputs always the value of the variable (if the value designates an unknown, this field is left blank).

- Trace data analysis

[Items]: Time Value(Variable name) Value(Variable name) ... Location

**Remark** Value(Variable name): Outputs only the variable whose value has been changed (if the value designates an unknown, this field is left blank).

Location: If the information does not exist, this field is left blank.

- [Execution Time(Percentage) Chart] tab

[Items]: Function Name Percentage[%] Time

(d) From the Call Graph panel

Bitmap(Only the visible part) (*.bmp)	Bitmap format (32-bit) (graphic file format)
JPEG(Only the visible part) (*.jpg)	JPEG format (graphic file format)
PNG(Only the visible part) (*.png)	PNG format (graphic file format)
Bitmap (*.bmp)	Bitmap format (32-bit) (graphic file format)
JPEG (*.jpg)	JPEG format (graphic file format)
PNG (*.png)	PNG format (graphic file format)
EMF (*.emf)	EMF format (graphic file format)

**Caution** All part of the call graph may not successfully be saved as a graphic file format if the project is too big.

- Remarks**
1. When "(Only the visible part)" is selected, only the part currently being displayed in the panel will be saved.
  2. If the zoom function has been applied, then the image will be saved at the current zoom ratio (except for EMF format).

(e) For the Output panel

Text file (*.txt)	Text format
-------------------	-------------

**[Function buttons]**

Button	Function
Save	Saves the file with the specified name.
Cancel	Closes this dialog box.

## APPENDIX B INDEX

**A**

Active project ... 44  
Analysis Chart panel ... 115  
    [Execution Time(Percentage) Chart] tab ... 125  
    [Variable Value Changing Chart] tab ... 117  
Analysis method ... 52  
Ascending order ... 34  
Auto adjustment function ... 58

**B**

Break event ... 40  
Breakpoint ... 40

**C**

Call Graph panel ... 127  
Call Graph Search dialog box ... 173  
Circular call ... 134  
Class/Member panel ... 138  
Code coverage ratio ... 100  
Column Chooser dialog box ... 169  
Coverage function ... 10  
Cross reference information ... 8, 77  
Current row mark ... 40, 103, 113

**D**

Data collection mode ... 48  
Data coverage ratio ... 111  
Declared location ... 39  
Defined location ... 37  
Descending order ... 34  
Details view ... 130  
Display range ... 57  
Dynamic analysis information ... 8

**E**

Export ... 44

**F**

Features ... 12

Filter icon ... 35  
Filter Settings dialog box ... 177  
Filters for displaying information ... 34  
Function execution time ratios ... 62  
Function list file ... 165  
Function List panel ... 94

**G**

Graphing target ... 49

**I**

Import ... 44, 165  
Information file ... 44

**J**

Jump to the defined location ... 37, 39

**L**

List of referencing location ... 43, 157  
Lock display icon ... 34

**M**

Main window ... 71  
Menubar ... 71  
Method for acquiring graph data ... 53

**O**

Open File dialog box ... 167  
Output panel ... 152  
    [All Messages] tab ... 154  
    [Find References] tab ... 157  
    [Program Analyzer] tab ... 156

**P**

Path Edit dialog box ... 165  
Pie chart ... 62, 125  
Pop-up display ... 59  
Project Tree panel ... 73  
Property panel ... 75

[Settings] tab ... 77

Window reference ... 70

[Variable Value Changing] tab ... 84

Pseudo RRM function ... 10

## R

RAM monitor function ... 10

Real-time sampling analysis ... 53

Recursive call ... 133

RRM function ... 10, 53

Runtime library ... 101

## S

Save As dialog box ... 179

Select Files not To Analyze dialog box ... 161

Select Files To Analyze dialog box ... 163

Smart Analog ... 48

Static analysis information ... 8

Structure ... 109

## T

Toolbar ... 72

Trace data analysis ... 53

Trace function ... 9, 53

Trigger edge ... 56

Trigger function ... 54

Trigger level ... 55

Trigger mark ... 55, 56, 121

Trigger mode ... 55

Trigger position ... 56

Trigger source ... 55

## U

Union ... 109

## V

Variable list file ... 165

Variable List panel ... 105

Variable Value Changing Chart (zoomed in) panel ... 147

## W

Watch-expression ... 42

Whole view ... 129

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Feb 01, 2014	-	First Edition issued

---

CubeSuite+ V2.02.00 User's Manual:  
Analysis

Publication Date: Rev.1.00 Feb 01, 2014

Published by: Renesas Electronics Corporation

---

**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.**  
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.  
Tel: +1-408-588-6000, Fax: +1-408-588-6130

**Renesas Electronics Canada Limited**  
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada  
Tel: +1-905-898-5441, Fax: +1-905-898-3220

**Renesas Electronics Europe Limited**  
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.  
Tel: +44-1628-651-700, Fax: +44-1628-651-804

**Renesas Electronics Europe GmbH**  
Arcadialstrasse 10, 40472 Düsseldorf, Germany  
Tel: +49-211-65030, Fax: +49-211-6503-1327

**Renesas Electronics (China) Co., Ltd.**  
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, 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 Rd., Putuo District, Shanghai, China  
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

**Renesas Electronics Hong Kong Limited**  
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong  
Tel: +852-2886-9318, Fax: +852-2886-9022/9044

**Renesas Electronics Taiwan Co., Ltd.**  
13F, No. 363, Fu Shing North Road, Taipei, 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 906, 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 Korea Co., Ltd.**  
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea  
Tel: +82-2-558-3737, Fax: +82-2-558-5141

CubeSuite+ V2.02.00