

# FR Family

CONFORMING TO  $\mu$ T-Kernel SPECIFICATIONS

## SOFTUNE™ $\mu$ T-REALOS/FR ANALYZER GUIDE



**FR Family**  
CONFORMING TO  $\mu$ T-Kernel SPECIFICATION  
**SOFTUNE<sup>TM</sup>  $\mu$ T-REALOS/FR**  
**ANALYZER GUIDE**

**FUJITSU MICROELECTRONICS LIMITED**



# Preface

## ■ Objective of This Manual

This document is designed for those who develop application programs for SOFTUNE  $\mu$ T-REALOS/FR (hereinafter called " $\mu$ T-REALOS"), and describes the  $\mu$ T-REALOS analyzer. Refer to this manual, when necessary, to check how to operate the  $\mu$ T-REALOS analyzer. It is also recommended to read "SOFTUNE  $\mu$ /FR User's Guide" (hereinafter called "User's Guide") before reading this document.

$\mu$ T-REALOS is a real-time OS based on the  $\mu$ T-Kernel specifications, which operates in a 32-bit RISC controller FR family product.

The  $\mu$ T-Kernel specifications are open-style real-time OS specifications, which have been developed by the T-Engine Forum. The  $\mu$ T-Kernel Specification Manual is available on the T-Engine Forum's website (<http://www.t-engine.org/>). The original copyright of  $\mu$ T-Kernel belongs to Mr. Ken Sakamura, while the copyright of the  $\mu$ T-Kernel specifications belongs to the T-Engine Forum.

Note: FR, the abbreviation of FUJITSU RISC controller, is a line of products of FUJITSU MICROELECTRONICS Limited.

## ■ Trademarks

SOFTUNE is a trademark of FUJITSU MICROELECTRONICS Limited.

REALOS (REALtime Operating System) is a trademark of FUJITSU MICROELECTRONICS Limited.

TRON is an abbreviation of "The Real-time Operating system Nucleus".

ITRON is an abbreviation of "Industrial TRON".

$\mu$ ITRON is an abbreviation of "Micro Industrial TRON".

T-Kernel and  $\mu$ T-Kernel are names to indicate the family of products and not specific products.

The names of products and systems appearing in this manual are <sup>TM</sup> or ® of their respective companies.

The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

## ■ Organization of this Manual

This manual consists of 3 chapters and an appendix, as listed below.

### CHAPTER 1 GENERAL DESCRIPTION

This chapter describes the precautions for using the SOFTUNE  $\mu$ T-REALOS Analyzer and its functions.

### CHAPTER 2 EXPLANATION of FUNCTIONS

This chapter describes the each of the  $\mu$ T-REALOS Analyzer functions and windows.

### CHAPTER 3 TASK ANALYSIS MODULE

This chapter describes the "task analysis module" used in sampling by the module use logs.

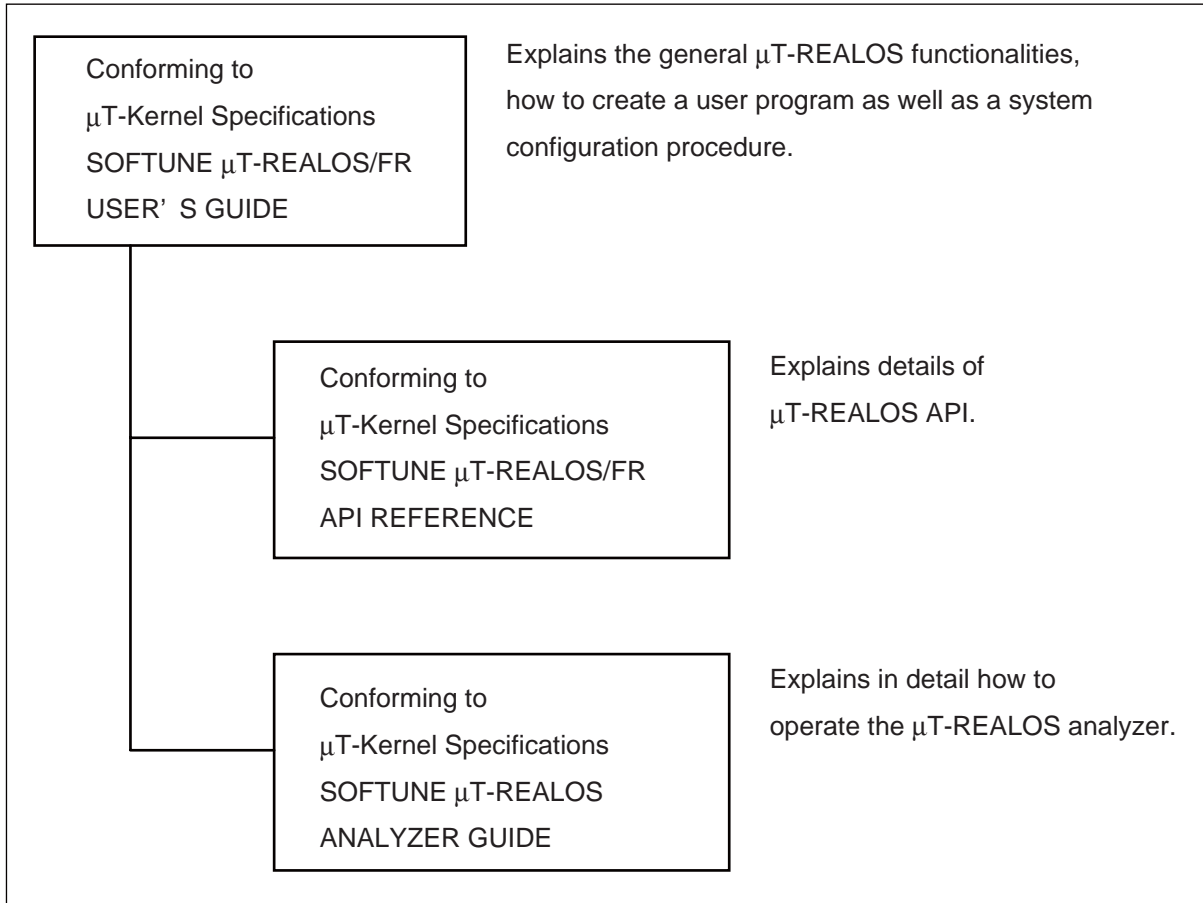
### APPENDIX

The appendix contains the analyzer processing time, error messages and restrictions.

- The contents of this document are subject to change without notice.  
Customers are advised to consult with sales representatives before ordering.
- The information, such as descriptions of function and application circuit examples, in this document are presented solely for the purpose of reference to show examples of operations and uses of FUJITSU MICROELECTRONICS device; FUJITSU MICROELECTRONICS does not warrant proper operation of the device with respect to use based on such information. When you develop equipment incorporating the device based on such information, you must assume any responsibility arising out of such use of the information. FUJITSU MICROELECTRONICS assumes no liability for any damages whatsoever arising out of the use of the information.
- Any information in this document, including descriptions of function and schematic diagrams, shall not be construed as license of the use or exercise of any intellectual property right, such as patent right or copyright, or any other right of FUJITSU MICROELECTRONICS or any third party or does FUJITSU MICROELECTRONICS warrant non-infringement of any third-party's intellectual property right or other right by using such information. FUJITSU MICROELECTRONICS assumes no liability for any infringement of the intellectual property rights or other rights of third parties which would result from the use of information contained herein.
- The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for use requiring extremely high reliability (i.e., submersible repeater and artificial satellite).  
Please note that FUJITSU MICROELECTRONICS will not be liable against you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products.
- Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.
- Exportation/release of any products described in this document may require necessary procedures in accordance with the regulations of the Foreign Exchange and Foreign Trade Control Law of Japan and/or US export control laws.
- The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

# μT-REALOS Manual Set

The μT-REALOS manual set consists of three volumes listed below. A first-time user of μT-REALOS should first read the SOFTUNE μT-REALOS/FR User's GUIDE.





# CONTENTS

<b>CHAPTER 1</b>	<b>GENERAL DESCRIPTION .....</b>	<b>1</b>
1.1	Precautions for Use .....	2
1.2	File Configuration .....	5
1.3	Features .....	6
1.4	Activation .....	7
1.5	General Description of the Functions .....	9
1.5.1	Object View .....	10
1.5.2	Setting OS Breaks .....	12
1.5.3	Log (Task Transition Diagram) .....	14
1.5.4	System Call Issue .....	17
1.5.5	Stack Information .....	18
1.5.6	Task Context Watch .....	19
<b>CHAPTER 2</b>	<b>EXPLANATION of FUNCTIONS .....</b>	<b>21</b>
2.1	RAnalyzer Window .....	22
2.2	Object Viewer .....	23
2.2.1	Explanation of the Window (Object Viewer) .....	24
2.2.2	Displaying Object Information .....	26
2.2.3	Contents of Each Objects Display .....	27
2.2.4	Changing Row Order .....	34
2.2.5	Source Jump Function .....	35
2.3	OS Break .....	36
2.3.1	Explanation of the Window (OS Break) .....	37
2.3.2	Setting Breaks .....	40
2.3.3	Deleting and Changing .....	41
2.4	Logs .....	42
2.4.1	Log Function Specification .....	43
2.4.2	Explanation of the Window (Log Setting) .....	45
2.4.3	Setting Logs (Dispatch Log) .....	47
2.4.4	Setting Logs (Module Use Log) .....	48
2.4.5	Setting Logs (Statistics Log) .....	50
2.4.6	Getting Logs .....	51
2.4.7	Log Data .....	52
2.4.8	Transition Diagram Window .....	55
2.4.9	Statistics Windows .....	59
2.4.10	Monitoring Window .....	61
2.4.11	Configuration Settings .....	63
2.4.12	Explanation of Realtime Statistics .....	66
2.4.13	Output of Log File .....	68
2.5	System Call Issue .....	69
2.5.1	System Call Issue Function .....	70
2.5.2	Explanation of the Window (System Call Issue) .....	72
2.5.3	Issuing System Call .....	74

2.6	Stack Information .....	75
2.7	Task Context Watch .....	81
<b>CHAPTER 3</b>	<b>TASK ANALYSIS MODULE .....</b>	<b>83</b>
3.1	Types of Task Analysis Modules .....	84
3.2	How to Use Comments .....	86
3.3	How to Customize .....	87
<b>APPENDIX</b>	<b>.....</b>	<b>89</b>
	APPENDIX A Speed of Information Acquisition .....	90
	APPENDIX B Symbols/Sections of the Task Analysis Module .....	91
	APPENDIX C Error Messages .....	93
	APPENDIX D Restrictions of Issuing System Call .....	97
	APPENDIX E Cautions when Using Stack Use Analysis .....	98
	APPENDIX F Restrictions with Debugger Type .....	99
<b>INDEX.....</b>	<b>.....</b>	<b>101</b>

# **CHAPTER 1**

---

## ***GENERAL DESCRIPTION***

**This chapter describes the precautions for using the SOFTUNE  $\mu$ T-REALOS Analyzer and its functions.**

- 1.1 Precautions for Use
- 1.2 File Configuration
- 1.3 Features
- 1.4 Activation
- 1.5 General Description of the Functions

## 1.1 Precautions for Use

---

**This lists precautions for using the SOFTUNE  $\mu$ T-REALOS Analyzer.**

---

### ■ Explanation of Terms

These terms are defined in the following way for use in this manual.

#### ● Objects

Resources managed by  $\mu$ T-REALOS.

Tasks, semaphores that manage synchronization and communication between tasks, event flags, mailboxes, mutexes, message buffers, rendezvous port, memory pools, cyclic handlers, and alarm handlers are all called  $\mu$ T-REALOS objects.

#### ● OS break

Breaks that watch  $\mu$ T-REALOS.

With ordinary Debuggers, it is not possible to set breaks that watch tasks in tasks assigned to the same area.

Example: When task 1 and task 2 are assigned to the following type of area.

```
void task_entry (VP_INT stacd) {
    funcA () ;
    ;
    ;
}
```

With ordinary Debuggers, it is possible to call out funcA() to set a break, but it is not possible to set a break only for task 1.

With OS Break, however, it is possible to set breaks for events like the ones in the above example. It is also possible to set breaks for OS behavior such as OS events (for example, issuing of a system call and starting of a dispatch).

#### ● Logs

Logs allow you to check the task transition state and the system calls that have been issued. In order to log behavior other than dispatches, it is possible to link programs called task analysis modules. (In actuality, this is only changing the configurator debugging type.)

Log time-stamp accuracy in type3 is the system clock counter order.

#### ● Task analysis module

This is an OS patch module to link to use the log function.

The task analysis module is called out each time a  $\mu$ T-REALOS event occurs and if that event is selected, a time-stamp is applied to that event information and is stored in an area ensured on the customer's memory. A task transition chart is displayed by the  $\mu$ T-REALOS Analyzer taking the data stored when execution is stopped. Also, an overhead is applied to enter the process described above.

- Statistics log

This log can be used to set an upper limit for task execution time (deadline setting) and record the number of times the limit is exceeded, the number of measurements performed, average execution time, minimum execution time, and maximum execution time.

## ■ Precautions for the Development Environment

- SOFTUNE  $\mu$ T-REALOS Analyzer (this product)

SOFTUNE  $\mu$ T-REALOS Analyzer (hereinafter referred to as  $\mu$ T-REALOS Analyzer) is a tool for analyzing user programs (user programs that use  $\mu$ T-REALOS are simply referred to as user programs in this manual) that use SOFTUNE  $\mu$ T-REALOS.

- SOFTUNE  $\mu$ T-REALOS (required software)

SOFTUNE  $\mu$ T-REALOS (hereinafter referred to as  $\mu$ T-REALOS) is a real-time OS for embedding which conforms to Fujitsu's  $\mu$ T-Kernel specifications. This product is an analysis tool for  $\mu$ T-REALOS, and cannot be used on other OS.

- Fujitsu's total development environment SOFTUNE Workbench (required software)

The  $\mu$ T-REALOS Analyzer is a tool that is used by being started from SOFTUNE Workbench. It cannot startup independently.

- Debugger link

The  $\mu$ T-REALOS Analyzer is linked to the SOFTUNE Workbench Debugger. Any Debugger type is applicable. However, you cannot use third party Debugger applications.

Also, the version that runs is prescribed. For the versions, see the installation manual.

- Debug environment for SOFTUNE Workbench

Before performing debug operations in SOFTUNE Workbench, set up the debug environment of SOFTUNE Workbench so that interrupts are enabled during general execution and disabled during step execution.

## ■ Precautions on the Use of an Emulator Debugger

- Notes regarding memory access

Immediately after the target file is loaded or reset,  $\mu$ T-REALOS Analyzer accesses the data area of  $\mu$ T-REALOS (including the stack area) and the task analysis module. If access fails,  $\mu$ T-REALOS Analyzer does not operate correctly.

Therefore, for writing the above data to memory, it is necessary to ensure that sufficient memory is available before loading the target file. This is done by, for example, executing a command procedure file that performs memory allocation.

### ● When the monitoring function is used

Since monitoring is implemented by using a function to read memory at runtime of the emulator debugger, an error occurs in the emulator debugger if it is used under conditions other than those listed below.

#### <Conditions required for use of the monitoring function>

- If the data area (including the stack area) of  $\mu$ T-REALOS is allocated in external memory.
  - The data area must be placed in emulation memory.
- If the data area (including the stack area) of  $\mu$ T-REALOS is allocated in internal memory
  - The internal memory area must be mirrored.

## ■ Precautions on the Use of an Monitor Debugger

### ● Monitoring function

Since there is no function to read memory at consecutive execution, the monitoring function cannot be used.

### ● Object display function

Since there is no function to read memory at consecutive execution, the object display cannot be updated.

## 1.2 File Configuration

---

**This section explains the configuration of files provided by the  $\mu$ T-REALOS Analyzer.**

---

### ■ File Configuration

Files provided by the  $\mu$ T-REALOS Analyzer are stored in the following manner after installation.

[Installation directory]

bin\	uTrim.dll	RTOS I/F DLL
	SiRan.dll	GUI DLL
	SiRanEnu.dll	English resources DLL
utkernel\911\dbg\	type1	Task analysis module type1 file set
	type2	Task analysis module type2 file set
	type3	Task analysis module type3 file set
	type5	Task analysis module type5 file set
	R_D_custom.h	Header file for log type [comment]

---

**Note:**

The task analysis module is a group of files needed for logs using the log function module.

---

## 1.3 Features

---

**This section explains the features of  $\mu$ T-REALOS Analyzer.**

---

### ■ Features of the $\mu$ T-REALOS Analyzer

$\mu$ T-REALOS Analyzer is a tool used for application debugging using  $\mu$ T-REALOS and for tuning.

It is possible to easily check the status of objects managed by  $\mu$ T-REALOS.

Also, it is possible to set breaks that watch tasks and to log  $\mu$ T-REALOS operating behavior.

Other than display a Task transition diagram of logged data, it analyzes each task execution time and makes it easy to determine if the system is maintaining real-time.

### ■ $\mu$ T-REALOS Analyzer Functions

- Objects list display

The  $\mu$ T-REALOS Analyzer [Object Viewer] displays the status of each object in list format.

- OS break

$\mu$ T-REALOS Analyzer [Setting] – [Break] enables you to set OS Break.

- Log (Task transition diagram)

$\mu$ T-REALOS Analyzer [Setting] – [Log] enables you to log  $\mu$ T-REALOS behavior.

- System Call issue

$\mu$ T-REALOS Analyzer [Setting] - [System Call] enables you to issue system calls when the Debugger is stopped.

- Stack information

Workbench menu [RAnalyzer] – [Stack information] makes it possible to check the current task's stack information.

- Task context display

Workbench menu [RAnalyzer] – [Task context watch] displays task contexts.



## 1.4 Activation

To activate the  $\mu$ T-REALOS Analyzer, start the Debugger and load the target before selecting [RAnalyzer] from the [RAnalyzer] menu of SOFTUNE Workbench.

### ■ Checking the Installation

If installed normally, the [RAnalyzer] menu will be added to the SOFTUNE Workbench menu.

### ■ Adding a Library

To use the dispatch break function and module use log function in  $\mu$ T-REALOS Analyzer, you must add the libraries provided with  $\mu$ T-REALOS Analyzer to your program.

To use the dispatch break function, add one of the four libraries provided.

When using the dispatch log function, add any one of the types from type1 to type3, and type5. For the module usage log function, add any one of the types from type1 to type3. To perform long time measurement and deadline measurement (statistics log) without using trace memory, add type5.

**Table 1.4-1 Task Analysis Module and Available Functions**

Task analysis module	Dispatch break	Dispatch logging	System call logging	System clock logging	Reload timer logging
type1	○	○	○	×	×
type2	○	○	○	○	×
type3	○	○	○	○	○
type5	○	○	×	×	×

○ : Enabled

×

Task analysis module	Interrupt logging	Statistics log
type1	○	×
type2	○	×
type3	○	×
type5	×	○

○ : Enabled

×

The above libraries use initialized variables. Accordingly, you need to initialize these variables in the initialization section of the user program. Refer to "5.9 ROM→ RAM Transfer Section" in the SOFTUNE Linkage Kit Manual for details of how to initialize these variables.

Note:

When installing, you must complete SOFTUNE Workbench once.

## ■ To Activate

Start the Debugger, load your target and execute the first task to operate up to the header once.

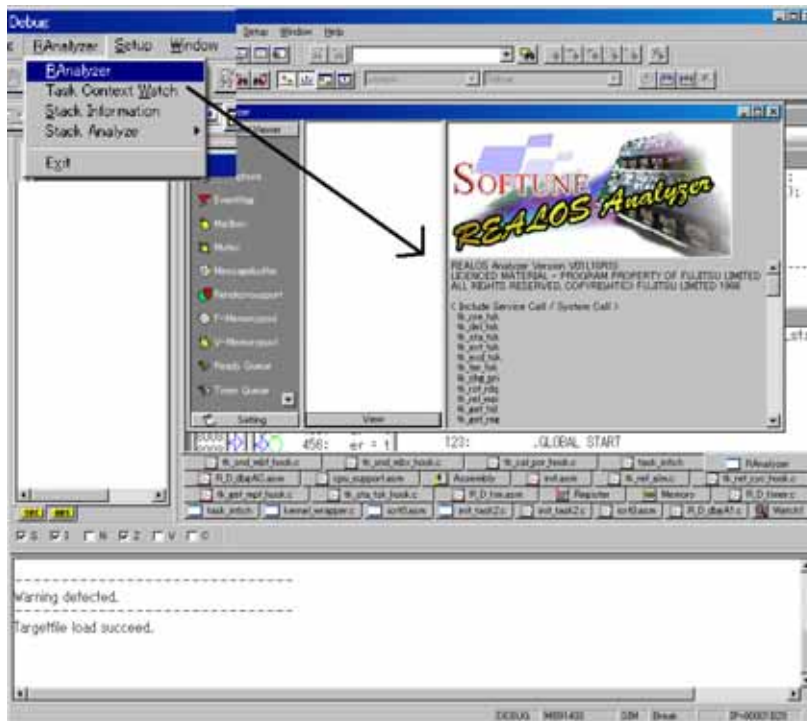
Select [RAnalyzer] from the [RAnalyzer] menu from the SOFTUNE Workbench menu.

**Check:**

Is the window shown in Figure 1.4-1 open? If it is open, the application has already started. If the menu is displayed in gray, it is possible that the version of Workbench is not compatible. Check your version of Workbench.

The corresponding Workbench version is described in the installation manual.

**Figure 1.4-1 Startup Screen**



## 1.5 General Description of the Functions

---

**This section describes each of the  $\mu$ T-REALOS Analyzer functions.**

---

- 1.5.1 Object View
- 1.5.2 Setting OS Breaks
- 1.5.3 Log (Task Transition Diagram)
- 1.5.4 System Call Issue
- 1.5.5 Stack Information
- 1.5.6 Task Context Watch

## 1.5.1 Object View

**Object view is a function to view the current state of a  $\mu$ T-REALOS object defined by the application when the Debugger is stopped.  $\mu$ T-REALOS Analyzer displays it by pressing on each object's switch.**

### ■ Object View

Object view is a window to see a list of the current state of each object when the Debugger is stopped.

Each object view shows a maximum of two screens of the information list display in list format and in a hierarchy to show the existence of waiting tasks, etc.

For the correspondence of object view for each product, see Table 1.5-1.

**Table 1.5-1 Object View Correspondence**

List	Product
	SOFTUNE $\mu$ T-REALOS
Task list	○
Semaphore list	○
Event flag list	○
Mailbox list	○
Mutex list	○
Message buffer list	○
Fixed-size memory pool list	○
Variable-size memory pool list	○
Ready queue	○
Timer queue	○
Cyclic startup handler list	○
Alarm handler list	○
Kernel information	○
Rendezvous port	○
Subsystem	○

○ : Enabled

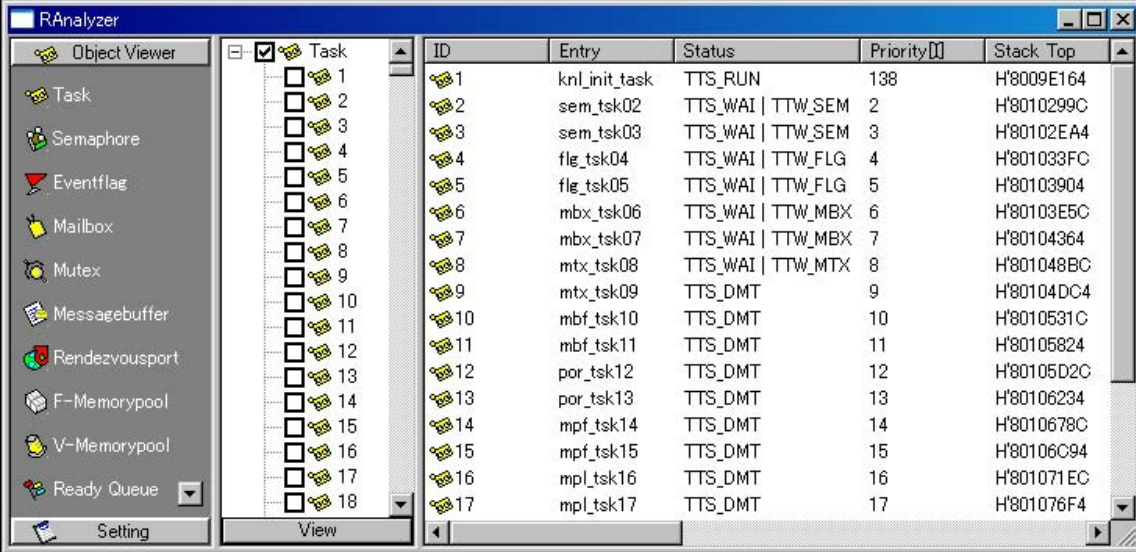
× : Disabled

## ■ Standard Functions of the Object Window

Each Object Window has the following standard functions.

- Save Column Order
- Source jump function when entry is solved

**Figure 1.5-1 Object View (Task)**



ID	Entry	Status	Priority	Stack Top
1	knl_init_task	TTS_RUN	138	H'8009E164
2	sem_tsk02	TTS_WAI   TTW_SEM	2	H'8010299C
3	sem_tsk03	TTS_WAI   TTW_SEM	3	H'80102EA4
4	flg_tsk04	TTS_WAI   TTW_FLG	4	H'801033FC
5	flg_tsk05	TTS_WAI   TTW_FLG	5	H'80103904
6	mbx_tsk06	TTS_WAI   TTW_MBX	6	H'80103E5C
7	mbx_tsk07	TTS_WAI   TTW_MBX	7	H'80104364
8	mtx_tsk08	TTS_WAI   TTW_MTX	8	H'801048BC
9	mtx_tsk09	TTS_DMT	9	H'80104DC4
10	mbf_tsk10	TTS_DMT	10	H'8010531C
11	mbf_tsk11	TTS_DMT	11	H'80105824
12	por_tsk12	TTS_DMT	12	H'80105D2C
13	por_tsk13	TTS_DMT	13	H'80106234
14	mpf_tsk14	TTS_DMT	14	H'8010678C
15	mpf_tsk15	TTS_DMT	15	H'80106C94
16	mpl_tsk16	TTS_DMT	16	H'801071EC
17	mpl_tsk17	TTS_DMT	17	H'801076F4

### Note:

When the PC is in the kernel, the contents displayed by the  $\mu$ T-REALOS Analyzer are not guaranteed because the kernel is still processing.

## 1.5.2 Setting OS Breaks

---

**The OS Break is a function for creating breaks that watch the  $\mu$ T-REALOS. Other than breaks that watch tasks, it is possible to set breaks for  $\mu$ T-REALOS behavior. Set for the OS break with [Break] in the [Setting] group.**

---

### ■ Setting OS Breaks

You can make settings or ascertain the current status of the execution of the settings using [Break] in the [Settings] group with OS break.  $\mu$ T-REALOS Analyzer displays the currently set breaks in the tree format window.

### ■ Types of OS Break

You can set the following 6 types of breaks with OS break.

- Execute break

Execute break can set a specific task ID to the function to break when executing the specified address.

- Access break

Access break can set a specific task ID to the function to break when a write occurred to the specified address (This does not operate when reading.).

- Dispatcher (ENTER) break

Dispatcher (ENTER) break can set breaks when the dispatcher has started. In this case, the task ID specified becomes the task to have its execution arbitration destroyed.

- Dispatcher (LEAVE) break

Dispatcher (LEAVE) break can set breaks when the dispatcher has ended. In this case, the task ID specified becomes the task to get execution arbitration.

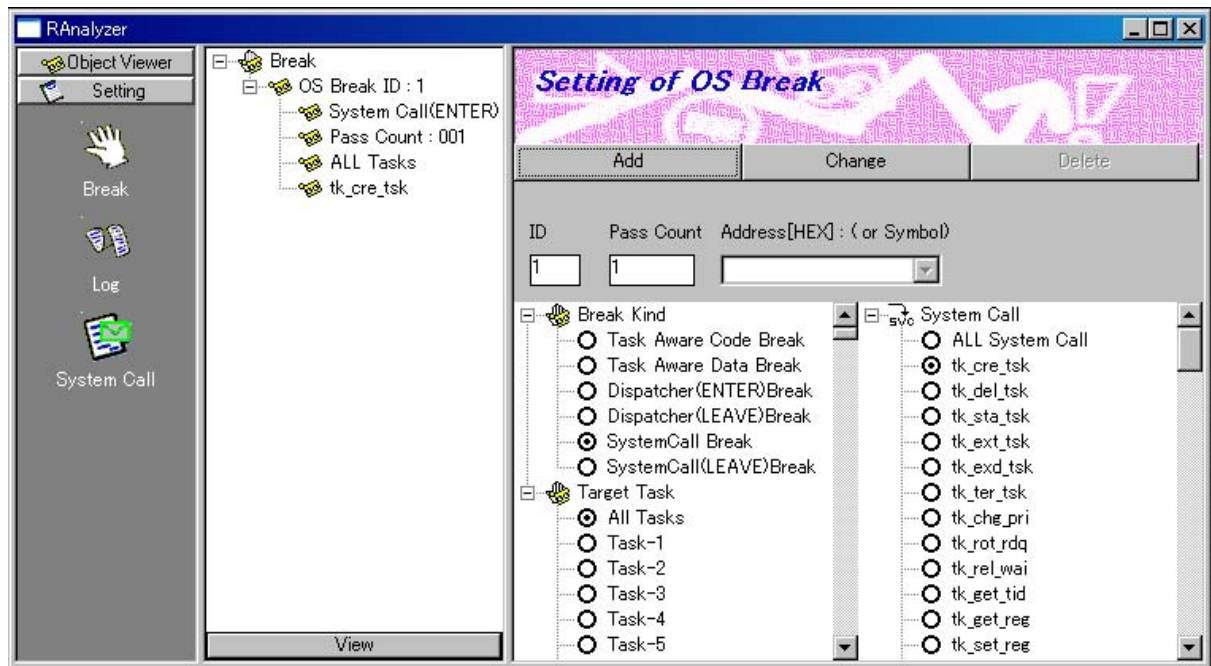
- System call (ENTER) break

System call (ENTER) break can set breaks when the system call issue has started. In this case, the task ID specified becomes the task to have its system call issued. It is also possible to specify system calls that started issuing.

- System call (LEAVE) break

System call (LEAVE) break can set breaks when the system call issue has ended. In this case, the task ID specified becomes the task to have its system call issued. It is also possible to specify system calls that ended issuing. When specifying system calls for issued tasks that are entering a wait state,  $\mu$ T-REALOS Analyzer sets a break with the timing to recover from the wait state.

Figure 1.5-2 Setting of OS Break

**Note:**

The OS break function set by the  $\mu$ T-REALOS Analyzer is not guaranteed, if the user uses the break setting function of SOFTUNE Workbench to cancel the break points which have been set by the OS break function of the  $\mu$ T-REALOS Analyzer.

### 1.5.3 Log (Task Transition Diagram)

---

**The log function gets the execution history that is dependent on  $\mu$ T-REALOS. It is possible to get the history of  $\mu$ T-REALOS behavior, such as task dispatch situations, system call issuer timing, and the point at which handlers are activated. Check the settings and acquired information related to the log with [Log] in the [Setting] group.**

---

#### ■ Setting the Log

Set the log using [Log] in the [Setting] group. Use the following procedures to make settings.

##### (1) Select the log to set.

There are 3 selections.

- Log [Dispatch]:
  - This log uses data breaks to watch only for the switching of execution tasks. While there are few types of events, this function does not require excessive operations such as the analysis module.
- Log [Module Use]:
  - This log allows you to use the task analysis module to select a variety of OS events. There is an increase in the number of selectable events, but it is necessary to ensure enough RAM size because the user memory is used as the buffering area. Also, it is necessary to select type1 to type3 in the  $\mu$ T-REALOS Configurator "Debug" definition. See the " $\mu$ T-REALOS/FR User's Guide" for details of the "debug" definitions. Cannot be used if using the type5 task analysis module (which includes statistics logging).
- Log [Statistic]:
  - The "debug" definitions in the  $\mu$ T-REALOS Configurator can only be used if using the type5 task analysis module (which includes statistics logging). See the " $\mu$ T-REALOS/FR User's Guide" for details of the "debug" definitions. This log records the execution time and deadline information for each task. As the log is buffered in each task analysis module, you do not need to allow for the additional RAM size in user memory.

##### (2) Select the event to get.

- Log [Dispatch]:
  - Currently, only [Run Task Switch] is selectable.
- Log [Module Use]:
  - You can select among the following events, the necessary items including "Dispatcher LEAVE". When Dispatcher LEAVE has not been selected,  $\mu$ T-REALOS Analyzer does not display Task Transition Diagram.
  - Interrupt handler (at call out, after processing is completed)
  - Timer handler (at call out, after processing is completed)
  - Dispatcher (at call out, after processing is completed)
  - System calls (at call out, when storing return value)
  - Customer specified event (called comments)



**(3) When selections have been completed, press the [Setting] button.**

This completes log settings.  $\mu$ T-REALOS Analyzer displays normally set events in the "Set Log Type" window.

## ■ Log Acquisition

Execute the system and cause a break at an appropriate place. At that point, press the [Get] button to start collecting data.

## ■ Setting the Debug Type

Setting the debug type allows you to switch the hook for debugging that is used in the  $\mu$ T-REALOS analyzer.

To switch the debugging hook, select a child item under the tree item "Debug" in the configurator project window by double-clicking it.

For the debugging hook, the file contained in the debug type subdirectory with the subdirectory name under the following path is used as the file for the debugging hook.

Directory in which the debugging hook is stored:

Directory in which  $\mu$ T-REALOS is installed \utkernel\911\dbg

When a debug type is selected,  $\mu$ T-REALOS Analyzer displays "Debug (debug type name)" for the tree item "Debug" in the configurator project window.

If a "make" or "build" operation is executed without setting the debug hook,  $\mu$ T-REALOS Analyzer displays the link error message "E4305L: External symbol is not defined" (\_kernel\_D\_xxx\_x/\_kernel\_D\_xxx\_x).

## ■ Function for Each Debug Type

The accuracy of the time stamp to be added during sampling varies, depending on the type selected. When a highly accurate time stamp is used, detailed data can be obtained. Consequently, however, the overhead becomes greater. The following table shows the accuracy level of each time stamp type.

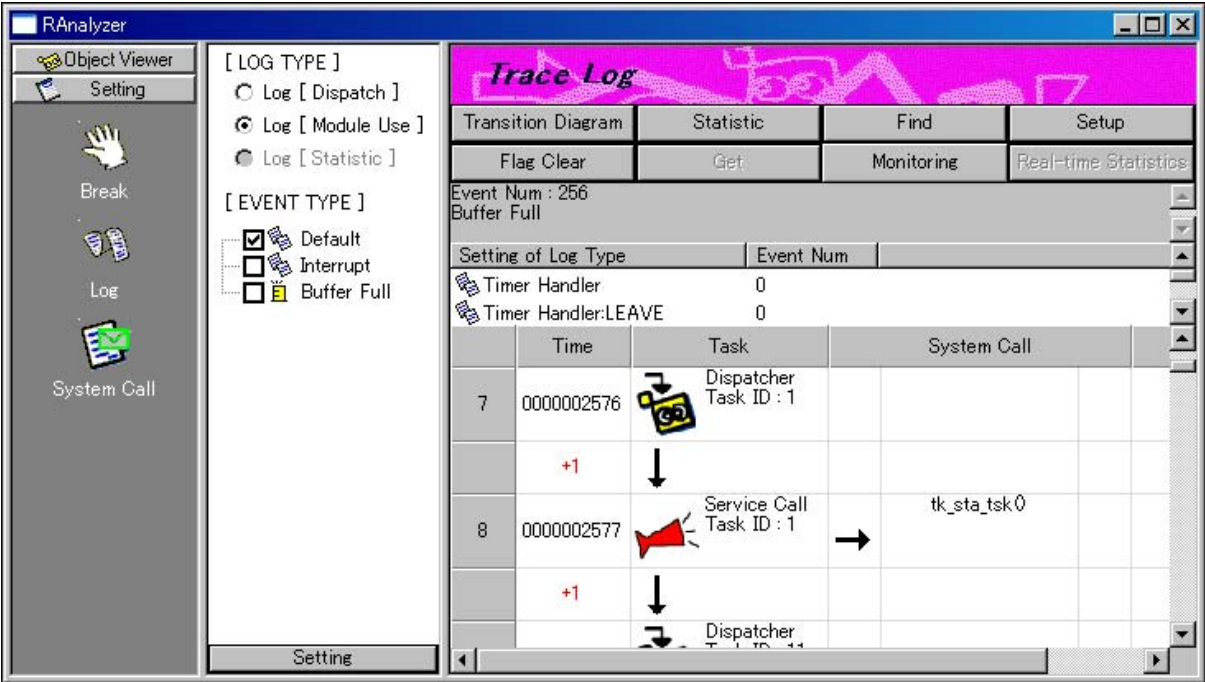
**Table 1.5-2 Time Stamp Accuracy of Each Type**

Type name	Library name	Explanation
type1	R_D_dbgA1.lib	Does not add time stamp
type2	R_D_dbgA2.lib	Adds system clock time
type3	R_D_dbgA3.lib	Adds count value of resource timer
type5	R_D_dbgA5.lib	Acquires Statistics log

■ Referencing Task Transition Diagram and Statistics Graphs

When a log has been acquired, press the [Transition Diagram/Statistics] button to open the RTOS Log (Transition Diagram) window and the RTOS Log (Statistics) window.

Figure 1.5-3 Log Settings and Display of Acquired Information



## 1.5.4 System Call Issue

The System Call issue function issues any system call from the Debugger during a break. It is possible to perform operations such as changing the task state and priorities by issuing system calls during a break. It is possible to operate with [System Call] in the [Setting] group.

### ■ System Call Issue

μT-REALOS Analyzer issues System calls using [System Call] in the [Setting] group.

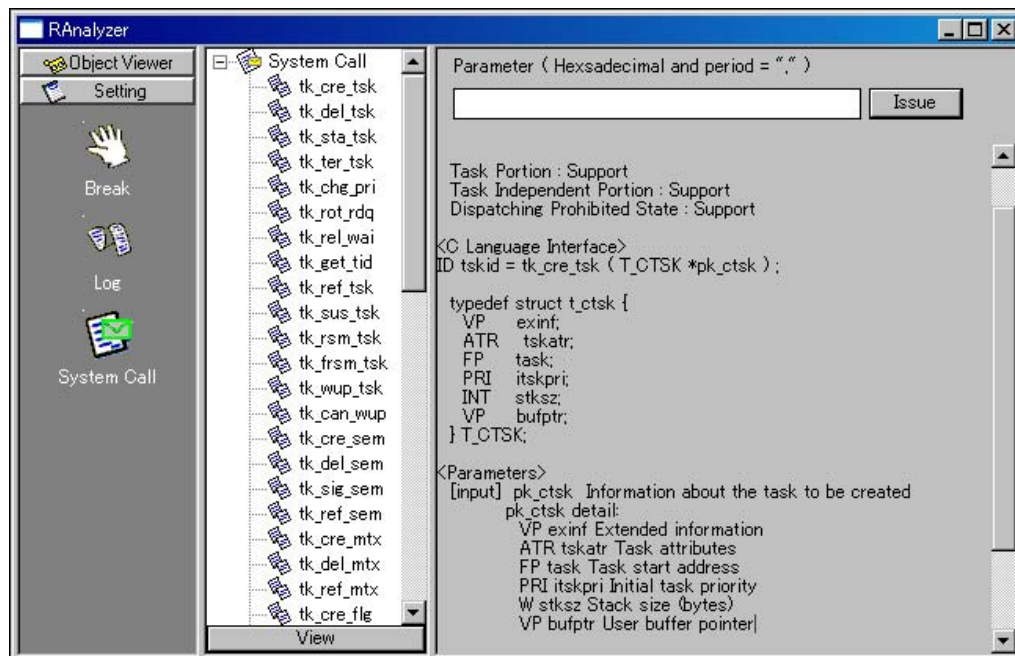
### ■ Issuing Procedures

μT-REALOS Analyzer displays system calls that can be issued on the system you are using. Select the system call you want to issue and press the [View] button. Input an argument and press the [Issue] button to complete a system call issue.

### ■ Issuing Results

After issuing, the μT-REALOS Analyzer displays the error code by a dialog box.

Figure 1.5-4 System Call Issue Screen



# 1.5.5 Stack Information

Stack information is a function to dynamically check the current usage state of the task's stack. It provides materials to customers to determine whether or not the stack is appropriately assigned. Open the window by [Stack Information] in the [RAnalyzer] menu of Workbench.

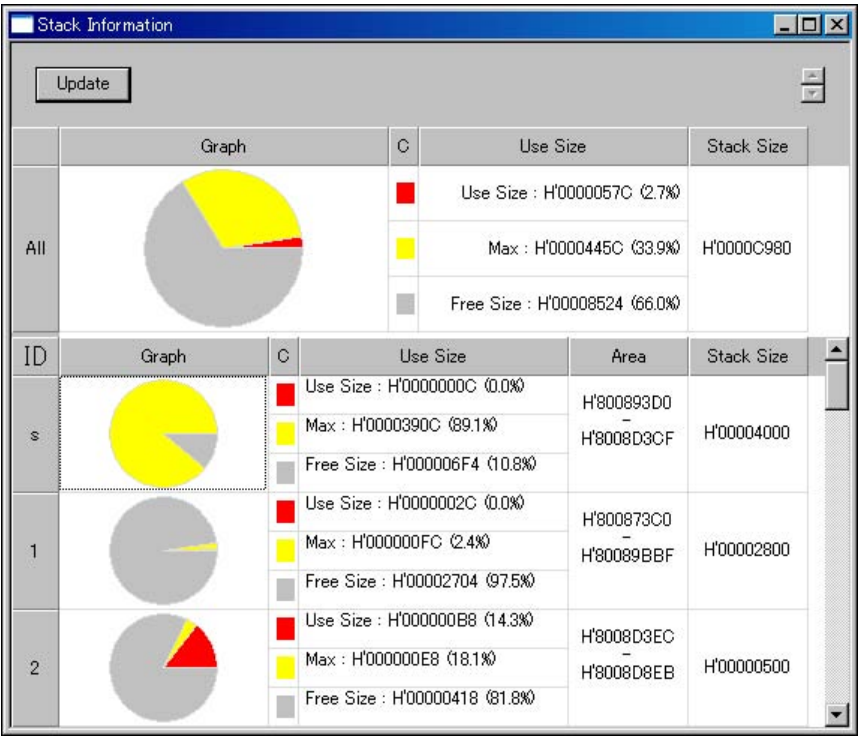
## ■ Stack Information

Open the window for Stack Information using [Stack Information] in the [RAnalyzer] menu of Workbench. When opening for the first time, data is automatically collected. so, you can re-acquire information by pressing the [Update] button.

## ■ Contents of Stack Information Display

The upper grid shows the each total value of "Use Size", "Max" value, and "Free Size" in entire system. The lower grid shows the information for system stack, idle task, and each task. μT-REALOS Analyzer displays only the created tasks. The following is an example of the stack information window.

Figure 1.5-5 Stack Information Window



Note:

The acquisition time of stack information depends upon the stack size.

## 1.5.6 Task Context Watch

Task context watch is a function for displaying task contexts. Open the window using [Task Context Watch] in the [RAnalyzer] menu of Workbench.

### ■ Task Context Watch

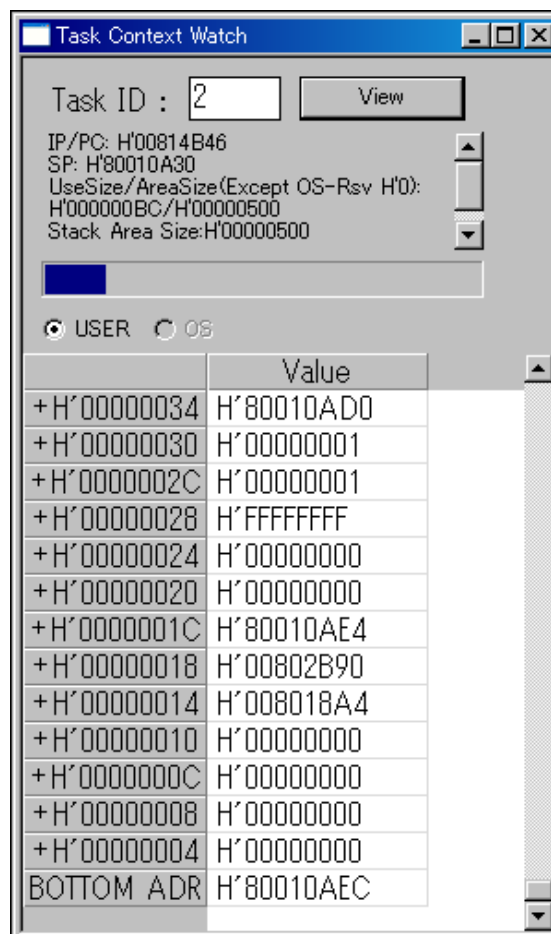
Open the window for task context watch using [Task Context Watch] in the [RAnalyzer] menu of Workbench.

### ■ Contents of Task Context Watch Display

Specify the ID of the task whose context information you want to display. The following contents are displayed by pressing the [View] button.

- IP/PC: Current IP/PC of the specified task
- SP: Current SP of the specified task
- Use Capacity/Size: Use Capacity/Stack Size calculated from the current SP
- Bar graph: Calculates the % from the above values and displays an image of the usage capacity below the above display.
- Context List: Displays the contents of the context used by the OS.

**Figure 1.5-6 Task Context Window**





# ***CHAPTER 2***

---

# ***EXPLANATION of FUNCTIONS***

**This chapter describes the each of the  $\mu$ T-REALOS Analyzer functions and windows.**

- 2.1 RAnalyzer Window
- 2.2 Object Viewer
- 2.3 OS Break
- 2.4 Logs
- 2.5 System Call Issue
- 2.6 Stack Information
- 2.7 Task Context Watch

## 2.1 RAnalyzer Window

**The RAnalyzer Window is the  $\mu$ T-REALOS Analyzer's main window. Almost all functions of the  $\mu$ T-REALOS Analyzer can be executed from the RAnalyzer Window.**

### ■ Opening the RAnalyzer Window

Open the window for RAnalyzer Window which is the main window of the  $\mu$ T-REALOS Analyzer using [RAnalyzer] below the [RAnalyzer] menu of Workbench.

### ■ Names of the Parts of the RAnalyzer Window

The RAnalyzer Window is composed of 3 panes. Each pane has the following role.

- Function Selection Pane (Left): Selects functions.
- Tree Format Pane (Middle): Selects each function setting status and targets for display.
- Information Display Pane (Right): Reflects settings and displays data acquired by each function.

Each function actually displays and sets the following.

**Table 2.1-1 Function Assignment List of Each Pane**

Function	Pane	Role
Object Viewer	Selects functions	Selects the object to display
Object Viewer-XXX*	Tree format	Selects ID to display and executes display
	Buttons in tree format	[View]: Executes object information display
	Information display	Displays object information
Setting	Selects functions	Selects from break, log and system call
Setting-Break	Tree format	Displays list of selected breaks
	Information display	Sets breaks
Setting-Log	Tree format	Selects log type
	Buttons in tree format	[Setting]: Determines and sets the log type
	Information display	Gets and displays log information
Setting-System Call	Tree format	Selects issued system calls
	Buttons in tree format	[View]: Determines issued system calls
	Information display	Argument input and issues

\*: XXX in Object-Viewer-XXX refers to OS objects.



## 2.2 Object Viewer

---

**The object viewer is a function for displaying current information relating to objects managed by  $\mu$ T-REALOS. The following describes the display items of the object viewer function and how to operate them.**

---

2.2.1 Explanation of the Window (Object Viewer)

2.2.2 Displaying Object Information

2.2.3 Contents of Each Objects Display

2.2.4 Changing Row Order

2.2.5 Source Jump Function

## 2.2.1 Explanation of the Window (Object Viewer)

This describes the Object Viewer window.

### ■ Opening the Object Viewer

Open the "RAnalyzer window" using SOFTUNE Workbench menu [RAnalyzer] – [RAnalyzer]. Press [Object Viewer] button in the function selection pane.

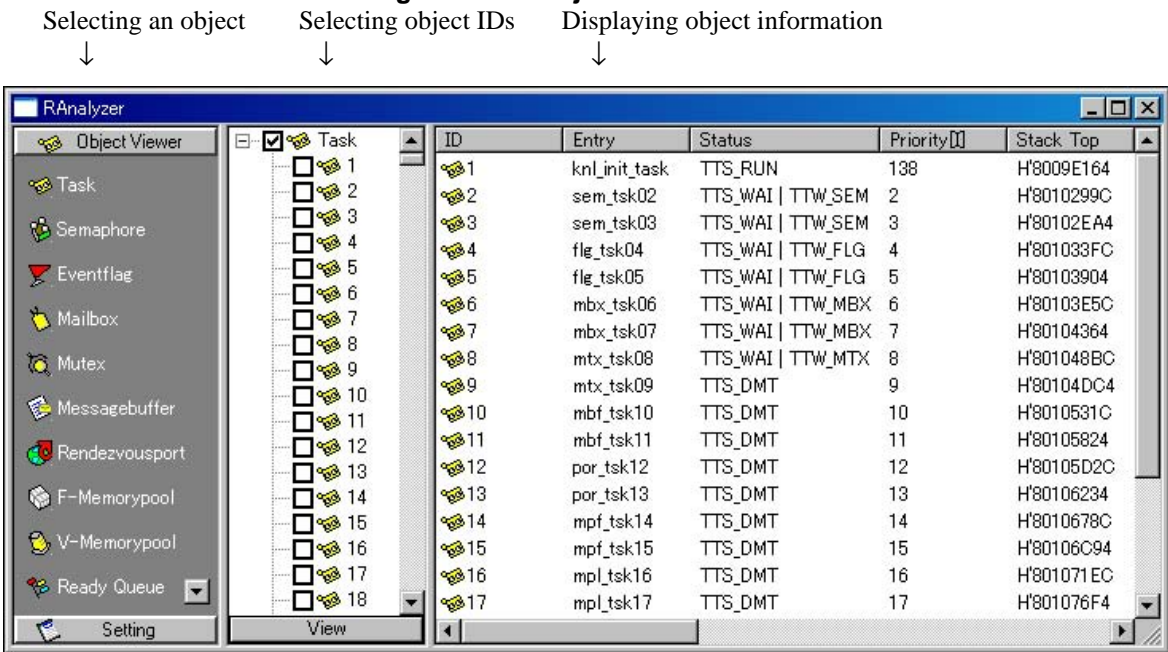
### ■ Functions of Each Pane when Object Viewer is Selected

When "Object Viewer" is selected, the following functions are assigned to each pane.

Functions assigned are as shown below beginning from the left pane.

- Selecting an object
- Selecting object IDs and [View] button
- Displaying object information

Figure 2.2-1 Object Viewer Window



### ■ Selecting an Object

Select the objects to display. See Section "2.2.3 Contents of Each Objects Display" for a list of objects that can be displayed.

## ■ Selecting Object IDs and [View] Button

Select the ID to display. Applying a check mark to each ID check box selects that item. You can select all items by applying a check mark to the object name selected (the highest level check box). To make it easier to switch between selecting all and specifying individual IDs, individual ID check boxes will not be removed even if you remove all check boxes that are displayed. Because any selective IDs do not exist in the Ready Queue or kernel information, there are no check boxes. After selecting, press the [View] button to start getting the information. The [View] button is disabled while information is being acquired.

## ■ Displaying Object Information

Press the [View] button to display a list of ID information selected for the selected objects. However,  $\mu$ T-REALOS Analyzer will not display the objects that have not been created even if they are selected. If there are waiting tasks or queued items, they will be displayed in their queued state in tree format. There are also other functions, described below.

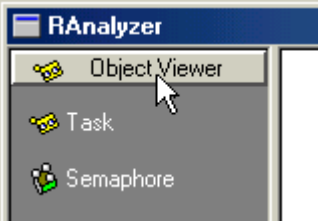
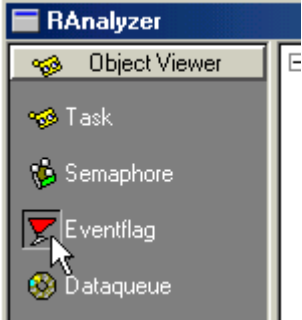
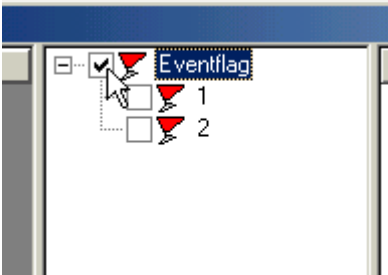
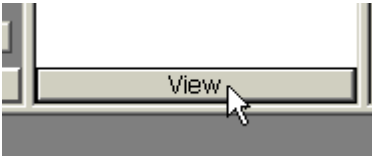
- Changing Row Order (Column drag-and-drop)  
Drag and drop the row to change the items.
- Saving Row Order (Right-click to save Row Order)  
It is possible to change the row order and save it with the information list display.  
Displaying list (excluding the columns), use a right-click to display a pop-up menu.  
When you select [Save Row Order],  $\mu$ T-REALOS Analyzer saves the row order.  
After that,  $\mu$ T-REALOS Analyzer displays the set row order even when you finish the operation.
- Source Jump (Right-click to Source Jump)  
Source windows will open at the position of the address of entries selected, if there is a source, when you select [Source Jump] on the pop-up menu that  $\mu$ T-REALOS Analyzer displays by making a right-click for objects having entries such as a task or handlers.

### 2.2.2 Displaying Object Information

To display information of objects, select the object to display and the ID, then press the [View] button.

■ Procedures to Display Object Information

Use the procedures below to display object information.

1.	Select Object Viewer with the Function Selection Pane.	
2.	Select the object to display.	
3.	Select the ID to display.	
4.	Press the [View] button.	

Procedures 2 and 3 do not occur for objects that do not have check boxes such as for the Ready queue.

## 2.2.3 Contents of Each Objects Display

This section describes the contents of each objects display.

### ■ Contents of Each Objects Display

The following shows the information that  $\mu$ T-REALOS Analyzer displays in the information display pane for each object.

**Table 2.2-1 Object View Item (List) (1 / 4)**

Objects Name	Item Name	Explanation
Task • Can Save Order or Row • Can Source Jump	Ex Info.	Extended information specified at creation.
	Entry	Task activate address With task exceptions, it is the handler activate address.
	Priority [I]	Priority when task is activated.
	Stack Top	Header address of stack area. Displays address specified at creation.
	Stack-size	Size of the stack Includes the size used by the OS.
	Status   Wait cause	Task state $\mu$ T-REALOS Analyzer displays cause with task exception.
	Priority	Current priority
	Priority [B]	Base priority
	Target ID	ID targeted when object is waiting.
	TMO	Indicates when the timer is waiting for a timeout. (Displays "TMO_FEVR" if "TMO_FEVR" was specified as the timeout.)
	Act Count	Activate request queuing number
	Wup Count	Wakeup request queuing number
	Sus Count	SUSPEND wait request nest number
Semaphore • Can Save Order or Row • Cannot Source Jump	Attribute	Semaphore attribute
	Count [I]	Initial count [Note] Currently, specifications do not allow for this display.
	Max. Count	Maximum count
	Count	Current count
	Wait Task Num	Number of tasks queued

**Table 2.2-1 Object View Item (List) (2 / 4)**

Objects Name	Item Name	Explanation
Event Flag • Can Save Order or Row • Cannot Source Jump	Attribute	Event flag attribute
	Pattern [I]	Initial pattern [Note] Currently, specifications do not allow for this display.
	Pattern	Current flag pattern
	Wait Task Num	Number of tasks queued
Mailbox • Can Save Order or Row • Cannot Source Jump	Attribute	Mailbox attribute
	Max Priority	Maximum message priority μT-REALOS Analyzer displays value by specifying message priority order in the attribute.
	Wait Task Num	Number of tasks queued
	Header Num	Number of message headers queued
Mutex • Can Save Order or Row • Cannot Source Jump	Attribute	Mutex attribute
	Max Priority	Enabled only when priority ceiling protocol is selected in the attribute.
	Hold Task ID	Locked task ID
	Lock Task Num	Number of tasks queued with lock wait
Message Buffer • Can Save Order or Row • Cannot Source Jump	Attribute	Message buffer attribute
	Max Size Message	Size of the maximum size message
	Buffer size	Entire size of the message buffer
	Free Size	Current empty size of the buffer
	Wait Task Num [S]	Number of tasks queued in send wait
	Wait Task Num [R]	Number of tasks queued in receive wait
	Message Num	Number of messages queued in buffer
Rendezvous port • Can Save Order or Row • Cannot Source Jump	Attribute	Attribute of rendezvous port
	maxcmsz	Maximum length of message when calling
	maxrmsz	Maximum length of message when responding
	ctskcnt	Number of tasks queued in call wait
	atskcnt	Number of tasks queued in accept wait
Fixed-size Memory Pool • Can Save Order or Row • Cannot Source Jump	Attribute	Variable-size memory pool area attribute
	Block Size	Memory pool area size
	Free Count	Current unused size
	Block Num	Total of block
	Waiting Task Num	Number of tasks queued

**Table 2.2-1 Object View Item (List) (3 / 4)**

Objects Name	Item Name	Explanation
Variable-size Memory Pool • Can Save Order or Row • Cannot Source Jump	Attribute	Variable-size memory pool area attribute
	Area Size	Memory pool area size
	Maximum Memory Block Size Available	Currently available, maximum unused size
	Waiting Task Num	Number of tasks queued
Ready Queue • Cannot Save Order • Cannot Source Jump	Queue Order	Header shows currently executing task in the () and 1/total count in parentheses. Below it is only the order.
	Task ID	Displays the task name and queued in each order. When stopped in the kernel, there are cases that the queue header and tasks being executed differ. In this case, the ID of the queue header is displayed in the parentheses.
Timer Queue • Cannot Save Row Order • Cannot Source Jump	Act Time	Displays relative time to activation after the current time in queued order in the parentheses. Removed from queue when $\mu$ T-REALOS Analyzer activates timer queue.
	Objects	Abbreviations of objects that are queued. tsk...task cyc...cyclic handlers/cyclic startup handlers alm...alarm handlers
	ID	ID of each Object
Cyclic Handler • Can Save Order or Row • Can Source Jump	Attribute	Cyclic handler/cyclic startup handler attribute
	Ex Info.	Extended information specified at creation
	Act Address	Handler activate address
	Cyc Time	Activate cycle that is set.
	Status	Displays the activate state. $\mu$ T-REALOS Analyzer displays "Activated" or "Stopped".
	Left Time	Time remaining until execution
Alarm Handler • Can Save Order or Row • Can Source Jump	Attribute	Alarm handler attribute
	Ex Info.	Extended information specified at creation.
	Act Address	Handler activate address
	Status	Displays the activate state. $\mu$ T-REALOS Analyzer displays "Activated" or "Stopped".
	Left Time	Time remaining until execution
Interrupt handler • Cannot Save Row Order • Can Source Jump	Handler Kind:	Displays whether either a CPU exception or user defined interrupt (displays interrupt) is defined.
	Act Address	Handler activate address

**Table 2.2-1 Object View Item (List) (4 / 4)**

Objects Name	Item Name	Explanation
Kernel information • Cannot Save Row Order • Cannot Source Jump	Context Status	Indicates whether or not non-task body is executing
	Lock Status of CPU	Displays whether or not CPU is in a locked state
	Disable Status of Dispatch	Displays whether or not it is in a dispatch disabled state.
	System time	Current system clock value
	System Operating Time	Duration from system activation to the present time
	Running status of OS	Displays whether or not OS execution
	Kernel Ver.	Kernel Version
Subsystem • Can Save Row Order • Cannot Source Jump	Extended SVC handler	Activate address of extended SVC handler
	Priority	Priority of subsystem
	Resource block size	Size of resource management block
	Break function address	Entry address of break function
	Startup function address	Entry address of startup function
	Cleanup function address	Entry address of cleanup function
	Event processing function address	Entry address of event processing function



**Table 2.2-2 Object Correspondence Table by Product (1 / 3)**

Object Name	Item Name	μT-REALOS
Task	Ex Info.	○
	Entry	○
	Priority [I]	○
	Stack Top	○
	Stack size	○
	Status   Wait Cause	○
	Priority	○
	Priority [B]	○
	Target ID	○
	TMO	○
	Act Count	×
	Wup Count	○
	Sus Count	○
Semaphore	Attribute	○
	Count [I]	×
	Max. Count	○
	Count	○
	Wait Task Num	○
Event flag	Attribute	○
	Pattern [I]	×
	Pattern	○
	Wait Task Num	○
Mailbox	Attribute	○
	Max Priority	×
	Wait Task Num	○
	Header Num	○
Mutex	Attribute	○
	Max Priority	○
	Hold Task ID	○
	Lock Task Num	○

**Table 2.2-2 Object Correspondence Table by Product (2 / 3)**

Object Name	Item Name	μT-REALOS
Message buffer	Attribute	○
	Max Size Message	○
	Buffer size	○
	Free Size	○
	Wait Task Num [S]	○
	Wait Task Num [R]	○
	Message Num	○
Rendezvous port	Attribute	○
	maxcmsz	○
	maxrmsz	○
	ctskcnt	○
	atskcnt	○
Fixed-size memory pool	Attribute	○
	Block size	○
	Free Size	○
	Block Num	○
	Wait Task Num	○
Variable-size memory pool	Attribute	○
	Area size	○
	Maximum Memory Block Size Available	○
	Wait Task Num	○
Ready queue	Queue Order	○
	Task ID	○
Timer queue	Act Time	○
	Object	○
	ID	○
Cyclic handler	Attribute	○
	Ex Info.	○
	Act Address	○
	Cyc Time	○
	Status	○
	Left Time	○

**Table 2.2-2 Object Correspondence Table by Product (3 / 3)**

Object Name	Item Name	μT-REALOS
Alarm handler	Attribute	○
	Ex Info.	○
	Act Address	○
	Status	○
	Left Time	○
Interrupt handler	Handler Kind	○
	Act Address	○
Kernel information	Context status	○
	Lock status of CPU	○
	Disable status of dispatch	○
	System time	○
	System operation time	○
	Running status of OS code	×
	Kernel Ver.	○
Subsystem	Extended SVC handler	○
	Priority	×
	Resource block size	×
	Break function address	×
	Startup function address	×
	Cleanup function address	×
	Event processing function address	×

○ : Enabled to display

× : Disabled to display

**Note:**

The value of each item is not displayed if it is determined from the OS operating state that it is disabled. However, when the OS initialization is not complete, such as at initial activation or at a reset, an undefined value will be entered to all items. Use at the stage executed up to the first task to start.

μT-REALOS Analyzer displays "E\_NOSPT" for items that are not supported.







## 2.2.4 Changing Row Order

You can change the display row order of each object display. Use the drag-and-drop method to operate.







### ■ Changing Row Order of the List Format of Each Object (Column Drag-and-Drop)

Excluding the ready queue, time queue, interrupt handler and kernel information display, you can change the row order of the list format for each object. To change the order, drag and drop the column to move into any desired position.

Figure 2.2-2 Example of Changed Row Order

ID	Entry	Attribute	Ex Info.	Entry	Priority[I]	Stack
 1		E_NOSPT	H'00000001	Task_1	1	H'000:
 2		E_NOSPT	H'00000002	Task_2	2	H'000:
 3		E_NOSPT	H'00000003	Task_3	3	H'000:
 4		E_NOSPT	H'00000004	Task_4	4	H'000:
 5		E_NOSPT	H'00000005	Task_5	5	H'000:
 6		E_NOSPT	H'00000006	Task_6	6	H'000:

↓

ID	Entry	Attribute	Ex Info.	Priority[I]	Stack
 1	Task_1	E_NOSPT	H'00000001	1	H'000
 2	Task_2	E_NOSPT	H'00000002	2	H'000
 3	Task_3	E_NOSPT	H'00000003	3	H'000
 4	Task_4	E_NOSPT	H'00000004	4	H'000
 5	Task_5	E_NOSPT	H'00000005	5	H'000
 6	Task_6	E_NOSPT	H'00000006	6	H'000

### ■ Saving Changed Row Orders (Right-click to Save Row Order)

To display the changed row order the next time (continuously even if the Debugger is quit), save the row order. Right-click on the list display field (excluding the column position) and select [Save Row Order] from the pop-up menu that μT-REALOS Analyzer displays to save the row order.

Note:

If reinstalled, the saved state is cleared.

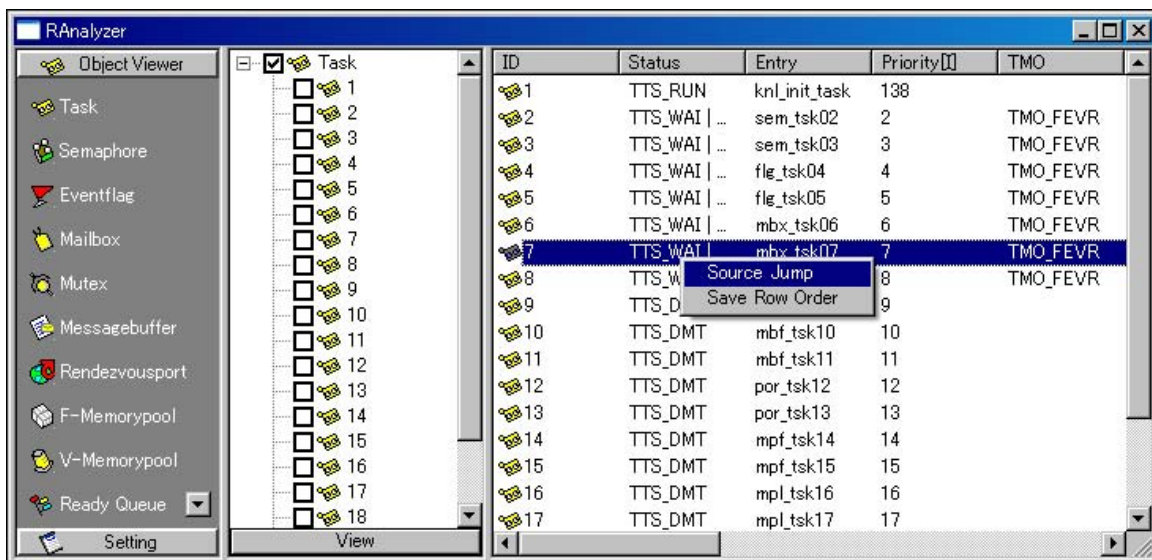
## 2.2.5 Source Jump Function

On object information displays that have information relating to the activate start address (entry), a function to open source codes exists.

### ■ Source Jump Function (Right-click to Source Jump)

On the task, cyclic handler, alarm handler, and interrupt handler displays, a jump function to the source code exists to indicate that activate address. To source jump, select [Source Jump] from the pop-up menu displayed by right-clicking on the ID row of the object to open.

Figure 2.2-3 Pop-up Menu [Source Jump] (CASE: SOFTUNE  $\mu$ T-REALOS)



#### Note:

If the targeted source is displayed as an icon, there is the possibility of mis-operation and jumping a file that is already open, same as when there is no source. Also, there are cases when this cannot be executed like, directory information to the targeted source is destroyed such as copying a project to a different directory.

## 2.3 OS Break

---

**OS break is a function that watches  $\mu$ T-REALOS. This section describes the contents of the OS break function display and how to operate it.**

---

- 2.3.1 Explanation of the Window (OS Break)
- 2.3.2 Setting Breaks
- 2.3.3 Deleting and Changing

## 2.3.1 Explanation of the Window (OS Break)

This describes the OS setting window.

### ■ Opening the OS Break Setting

Press [Setting] button in the function selection pane, then select [Break] icon.

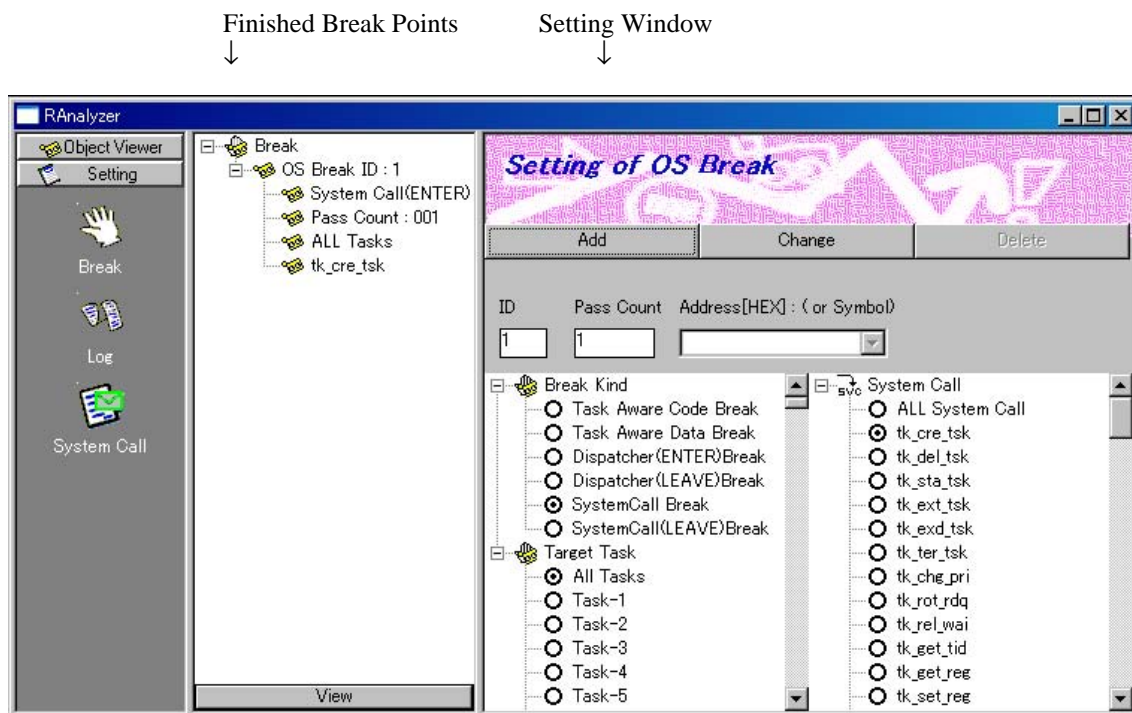
### ■ Functions of Each Pane when Object Break Setting Function is Selected

When "Break" is selected, the following functions are assigned to each pane.

Functions assigned are as shown below beginning from the left pane.

- "Break" Fixed
- Finished Break Points
- Setting Window

**Figure 2.3-1 OS Break Window**



### ■ Adding, Changing, Deleting Buttons (Setting Window)

It is possible to add, change or delete OS breaks using the panes of the setting window.

## ■ ID Pass Count, Address Input Form (Setting Window)

- [ID]:  
Box to input the OS break ID  
Clicking on [OS Break ID] for the finished settings will reflect them in this edit box.  
Input cardinal numbers using the decimal system.
- [Pass count]:  
Box to input the pass count of the break point.  
Setting it to "1" means stopping the target execution every break point passed, and setting it to "n" means stopping the target execution every "n" times of the break point passed.  
The setting range is 1 to 65535.  
Input cardinal numbers using the decimal system.
- [Address]:  
Box to input the address targeted for execution and an access break Symbols may be input. Input using hexadecimal (0x and h' descriptions are unnecessary).

## ■ Types of Breaks

The types of breaks that can be selected are following.

**Table 2.3-1 Break Types**

Break Type	Explanation
Execute break	Code break to specify a task Example: Break when calling a shared function func() from task 2.
Access break	Data break to specify a task Example: Break when large area variable ex_val value is changed by task 1.
Dispatcher (ENTER)	Break for dispatcher activation Example: Break when task 3 execution arbitration is taken away.
Dispatcher (LEAVE)	Break for dispatcher end Example: Break when task 4 execution arbitration is moved.
System Call (ENTER)	Break when a system call is issued Example: Break when task 6 issues tk_sig_sem().
System Call (LEAVE)	Break when system call return value is determined. Example: Break when task 7 tk_wai_sem() is canceled.



## ■ How Breaks are Implemented

Execution break ...

A SOFTUNE code breakpoint is used. When a break occurs, the task ID is determined and execution is restarted if the break conditions are not satisfied.

Access break ...

A SOFTUNE data breakpoint is used. When a break occurs, the task ID is determined and execution is restarted if the break conditions are not satisfied.

Dispatcher break ...

The break conditions are set in the user data area and evaluated in the  $\mu$ T-REALOS Analyzer library which is linked into the user program. The data area used is 10 bytes per task. Because this is performed while the conditions are being evaluated, realtime performance is excellent.

## ■ Target Task Break Types

Set the task ID for the break target. When you select "All Tasks", all tasks are targeted. When there is a dispatcher break, the targeted task image changes.

- Dispatcher (ENTER) Break: Task whose execution arbitration has been taken away.
- Dispatcher (LEAVE) Break: Task whose execution arbitration has been acquired.

## ■ System Calls

Set the system call for the break target. The difference between ENTER/LEAVE is "immediately after issue" and "before value to return after issue". Only for the tk\_wai\_xxx system calls is a break applied when the wait status is canceled. Also, recovery caused by a timeout is excluded.

---

Note:

When using ICE, use a hardware break for the OS breaks. It is impossible to set when there is no free space in the hardware break. Also, dispatcher breaks and system call breaks stop inside the kernel. We recommend that it is used like a trigger.

If you use the Dispatch breaks together with Monitoring or Dispatch log, debugger execution will not stop the breakpoint.

Unless the task analysis module is used, you cannot set dispatcher breaks or system call breaks.

---

## 2.3.2 Setting Breaks

This describes how to set OS breaks.

### ■ Resource Breaks

OS breaks are implemented using the Workbench break function. In case of access break, data breaks are used. For other break types, use an execution break (for ICE and hardware breaks).

It is impossible to set when these are not free.

### ■ How to Set ([Add] Button)

To set a new break, click on the [Add] button. Setting using the [Add] button, ignores the values in the ID setting box and automatically assigns break IDs.

A setting cannot be added when the configured break and pass count are the only differences of conditions. When using dispatcher break, make sure to set number of tasks for USER\_TASK\_NUM of R\_D\_siz.c in the task analysis module file set. For more information, see "3.3 How to Customize".

**Table 2.3-2 Input Items Required for Break Settings**

Break Type	Input Item			
	Address*	Pass Count	Target Task	System Call
Execution break	○	○	○	-
Access break	○	○	○	-
Dispatcher break	-	○	○	-
System call break	-	○	○	○

○ : Required

- : Not required

\* : If "Address" is a global symbol, it is also possible to enter symbols.

### ■ Setting Results

If the settings are successful, the set breaks are added to "Finished Break Points".

### ■ Break Occurs

If a break occurs,  $\mu$ T-REALOS Analyzer displays the following dialog box.

**Figure 2.3-2 Dialog Box at an OS Break**



## 2.3.3 Deleting and Changing

---

**This section describes how to delete and change OS breaks.**

---

### ■ Reflecting Finished Breaks in the Setting Window

Click on the "OS Break ID" item in the Finished Breaks to reflect the information of that break in the Setting Window.

### ■ Change

To change, change the address or targeted task ID and press the [Change] button.

### ■ Delete

To delete, select the Finished Break to reflect it in the Setting Window and click on the [Delete] button to delete it.

---

Note:

If the RAnalyzer window is closed, the  $\mu$ T-REALOS Analyzer is exited, or the Debugger is exited, all setting OS breaks are deleted.

---

## 2.4 Logs

---

**A log in the  $\mu$ T-REALOS Analyzer is a function that logs  $\mu$ T-REALOS behavior. This section describes how to set the logs and each window.**

---

- 2.4.1 Log Function Specification
- 2.4.2 Explanation of the Window (Log Setting)
- 2.4.3 Setting Logs (Dispatch Log)
- 2.4.4 Setting Logs (Module Use Log)
- 2.4.5 Setting Logs (Statistics Log)
- 2.4.6 Getting Logs
- 2.4.7 Log Data
- 2.4.8 Transition Diagram Window
- 2.4.9 Statistics Windows
- 2.4.10 Monitoring Window
- 2.4.11 Configuration Settings
- 2.4.12 Explanation of Realtime Statistics
- 2.4.13 Output of Log File

## 2.4.1 Log Function Specification

---

This section describes the log function specification in the  $\mu$ T-REALOS Analyzer.

---

### ■ Sampling Method

There are two types of sampling functions.

#### ● Sampling using the Debugger break (Dispatch log)

Sets the break inside the OS, and performs sampling using the tools when a dispatch occurs.

##### <Features>

- Enables easy acquisition of logs without linking to the task analysis module.
- Logs only task switches
- The maximum value of events that can be acquired on  $\mu$ T-REALOS is 32767.

#### ● Sampling using the task analysis module (Module use Log)

Through linkage to the task analysis module, OS behavior of a kind that cannot be acquired by the dispatch log can be sampled. Sampling uses user memory.

##### <Features>

- In contrast to sampling that uses debugger breaks, the system is never stopped.
- Enables logging after selection of a variety of events (maximum of nine types).
- The maximum number of events that can be acquired is 2048 (0x10000 bytes), which varies depending on the user memory.

### ■ Adding Time Stamps

This adds time stamps to each event that was sampled (excluding when type1 logs using the module are selected.). Time stamps are the normal  $\mu$ T-REALOS system clock value. Therefore, if the system clock has not been set, the time stamp will be 0 or an undefined value so when you are using the log function, always set the system clock.

### ■ Task Transition Diagram Display

Displays a transition table of the data that was sampled. The task transition diagram enables an easy grasp of the status of task switches.

### ■ Statistics Graph Display

Displays each task's CPU share time and the ratio in graph format. Maximum execution time is important data to verify if the processing was completed in the time restriction that considers the system time.

### ■ Monitoring Function

Monitors the executing task ID by using the function of pseudo on-the-fly.

### ■ Output of a Log File

The acquired log can be saved as data.

It is possible to see the log data in the "Log Viewer" in Product CD-ROM.

### ■ Configuration Settings - Interval Time Settings for Monitoring

You can set the sampling times for monitoring and for the realtime statistics. However, the sampling time setting for realtime statistics is only available if using task analysis module type5.

### ■ Configuration Settings - Deadline Settings

You can set an upper limit for task execution time. However, this is only available if using task analysis module type5.

## 2.4.2 Explanation of the Window (Log Setting)

This section describes the log setting window.

### ■ Opening the Log Settings

Press [Setting] button in the function selection pane, then select [Log] icon.

### ■ Functions of Each Pane when Log Function is Selected

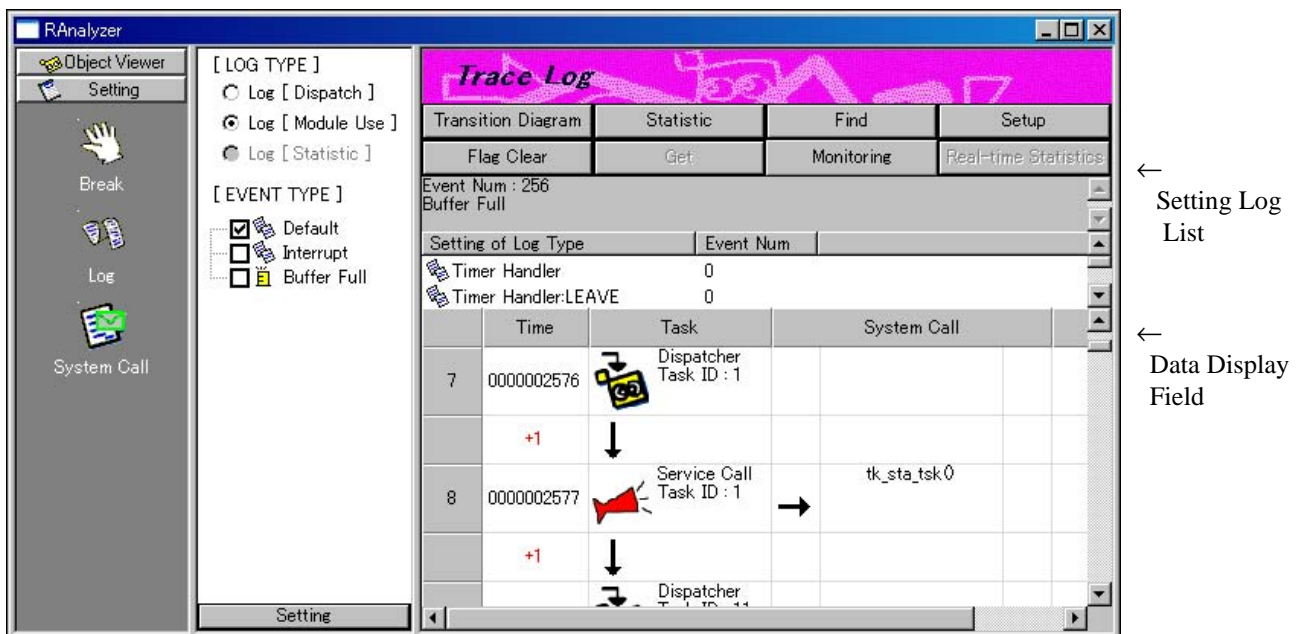
When "Log" is selected, the following functions are assigned to each pane. Functions assigned are as shown below beginning from the left pane.

- "Log" fixed.
- Log setting selection
- Setting state and data display window

**Figure 2.4-1 Log Setting Window**

Log Setting Selection  
↓

Setting State and Data Display Window  
↓



### ■ Log Setting Selection List (Setting Selection Window)

The log sampling method selection and event selection acquired using the selected sampling method are performed by switching the radio button and applying a check to the check box.

Also, with "Log [Module Use]", it is possible to check or uncheck the check box for buffer full/ring buffer selection.

The "statistics log" is only available if using task analysis module type5.

### ■ [Setting] Buttons (Log Setting Selection Window)

μT-REALOS Analyzer reflects the setting in the window by pressing the [Setting] button in the setting selection window.

Also, when setting log information, it is necessary to always enable access of all the memory (Particularly for Module Use Logs). When a task analysis module is allocated to an area that does not allow memory access, a memory access error will occur at flag control or sampling and it is assumed that normal operation will not occur. Therefore, we recommend that logs be set after executing up to the header of the first task to be activated.

### ■ Each Button of Display Window

- [Clear Flag] button: Initializes task analysis module variables.  
When setting Module Use Logs, use this when resetting the system and retrying. Clears the flag managing logs with the module when executed.
- [Get] button: Gets the log data that was sampled.  
This is not a static analysis, so always start the system after setting and get it when the system is stopped. We do not warranty getting or setting while the system is running. Also, in the space to get log information, just as with setting logs, it is necessary to always enable access of all the memory. (Particularly for logs that use the module) When a task analysis module is allocated to an area that does not allow memory access, a memory access error will occur at flag control or sampling and it is assumed that normal operation will not occur.
- [Transition Diagram], [Statistics Graph], [Monitoring], [Configuration Settings], and [Realtime Statistics] button  
This button is used to open the window for the transition diagram, statistics graph, monitoring, analyzer configuration settings, and realtime statistics.

### ■ Setting Log List (Display Window)

μT-REALOS Analyzer displays the set log type. Also, after getting, μT-REALOS Analyzer displays event counts that occurred for each type.

### ■ Data Display Field (Display Window)

Displays the data acquired. Note that μT-REALOS Analyzer does not display data if using task analysis module type5 because this module does not collect trace data.

See Section "2.4.7 Log Data" for details to reading the data.



## 2.4.3 Setting Logs (Dispatch Log)

This section describes the settings for the Dispatch Log.

### ■ Selectable Events

The following event is selectable.

**Table 2.4-1 Dispatch Log Event**

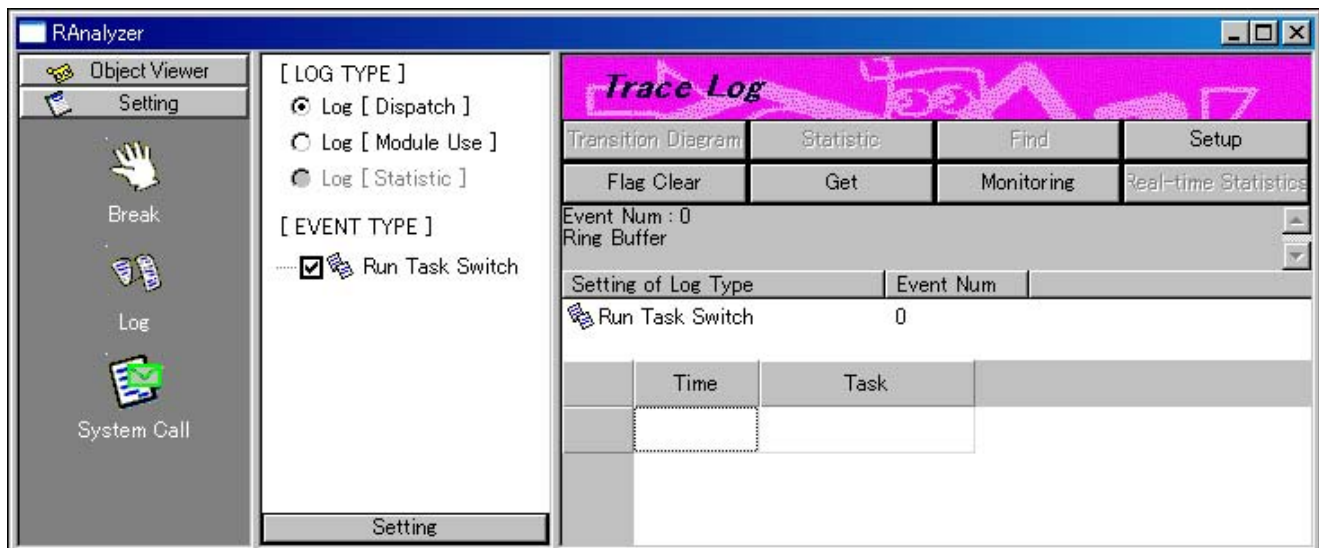
Event	Explanation
Run Task Switch	Logs events that are switched by the RUN task.

### ■ How to Set

1. Select Log [Dispatch] in the Log Setting Selection window.
2. Select the type of event.  
Select "Run Task Switch" in the dispatch log.
3. Press the [Setting] button.

Last, check that the set log type in the display window has been registered to complete the setup.

**Figure 2.4-2 Example of Setting Log List Registration when Setting the Dispatch Log**



## 2.4.4 Setting Logs (Module Use Log)

This section describes module use log settings.

### ■ Selectable Events

You can select a multiple of the following events.

**Table 2.4-2 Module Use Log Events**

Events	Explanation
Interrupt	Occurs when an interrupt handler starts
Interrupt: LEAVE	Occurs when an interrupt handler ends
Timer handler *	Occurs when a timer handler (cyclic and alarm) starts
Timer handler: LEAVE *	Occurs when a timer handler (cyclic and alarm) ends
Dispatcher	Occurs when a dispatch starts
Dispatcher: LEAVE *	Occurs when a dispatch ends
System Call *	Occurs when a system call is issued
System call: LEAVE	Occurs when a system call ends or when a return value is set
Comment *	Occurs when a user optional event occurs

\*: setup included in "default"

### ■ Setup to be Done in the Log Setting Selection Pane

More than one setup can be selected from the "Default" and "Interrupt" setups.

If the "Default" setup is selected, data required for displaying a task transition diagram can be acquired.

Events acquirable by selecting "Interrupt" setup are "interrupt" and "interrupt:LEAVE". This setup can be used to log the entrance and an exit of an interrupt.

It is also possible to set up "buffer full".

### ■ Buffer Full Settings

You can select the buffer type by changing the buffer full check. With no check, it uses the ring buffer.

- Ring buffer:

This mode does not stop even when the size of the buffer is exceeded. It deletes old events when the sampling events become larger than the size of the buffer and samples new events. Therefore, the event just prior to stopping is guaranteed.

● Buffer full:

This mode stops at the point where the buffer is full.

A break occurs when the sampling events become larger than the size of the buffer. This mode forcibly stops sampling over the size of the buffer.

Also, buffer type has no meaning independently. This just switches the operation mode of buffer.

## ■ Advanced Setup (Customize Setup)

A right click can open a pop-up menu and more than one can be chosen in a favorite combination by checking "customize setting"

However, in order to display a task changes figure, it is necessary to choose "dispatcher:LEAVE".

## ■ How to set

1. Select "Log [Module Use]" in the Log Setting Selection window.
2. Select the event.
3. Select the buffer type.
4. Press the [Setting] button.

Last, check that the set log list in the display window has been registered to complete the setup.

## ■ Recommended Combination of the Event

When you want to customize the combination of events, we recommend that you make your selection by adding other events to the two events ("dispatcher: LEAVE" and "system call") included in the "default" setup.

---

### Note:

Time stamp settings are not possible from the  $\mu$ T-REALOS Analyzer. To change the time stamp settings, select the module set required for the Configurator "Debugger" and rebuild it. See "CHAPTER 3 TASK ANALYSIS MODULE" for details regarding the module set.

---

## 2.4.5 Setting Logs (Statistics Log)

---

This section describes the statistics log.

---

### ■ Selectable Events

You can select the following events.

**Table 2.4-3 Statistics Log Events**

Event	Explanation
Time measurement	Dispatcher: Data is logged for each task execution from the time LEAVE occurs until one of the following system calls is executed. tk_ext_tsk(), tk_exd_tsk(), tk_ter_tsk()

### ■ Features of the Statistics Log

This log can only be used if using the type5 task analysis module. Because the maximum execution time, minimum execution time, total execution time, and number of executions are recorded inside the module without using trace memory, this log can be used over a long time period (Up to approximately 49 days).

Also, whether the task execution time has exceeded the specified upper limit for the execution time (deadline) is checked within the module, you can also display the number of times the task execution time exceeds the deadline.

### ■ Measurement of the Statistics Log

Dispatcher: The time from LEAVE until one of tk\_ext\_tsk(), tk\_exd\_tsk(), tk\_ter\_tsk() is executed is treated as one task execution. Always set the buffer size in TRC\_DATA\_NUM and the number of tasks in USER\_TASK\_NUM in R\_D\_siz.c, which is included in the file set for the type5 task analysis module you are using. See "3.3 How to Customize" for details on how to specify these settings.

### ■ How to Set

1. In the log debugger selection window, select "Statistics Log".
2. Select the "Time Measurement" checkbox.
3. Click the [Setting] button.

Finally, check that "Dispatcher: LEAVE" and "System call" appear in the selected log list in the display window.

## 2.4.6 Getting Logs

---

This section describes getting logs.

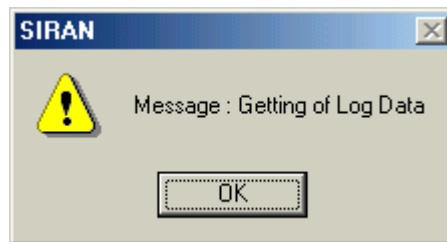
---

### ■ Getting Logs

Get logs using the [Get] button in the data display window.

When acquisition is complete,  $\mu$ T-REALOS Analyzer displays the following dialog box. Press "OK" button to display the data.

**Figure 2.4-3 Log Acquisition Complete Dialog Box**



---

Note:

Get and set logs during breaks. If done while the system is operating, the system will not operate normally.

---

## 2.4.7 Log Data

This section describes acquired log data.

### ■ Log Data Display

When the acquisition of logs is successful,  $\mu$ T-REALOS Analyzer will display in the data display field like the one below.

When  $\mu$ T-REALOS Analyzer displays the dispatch log, it displays only the items of "Time" and the "Task".

Figure 2.4-4 Example of Module use Log Display Data

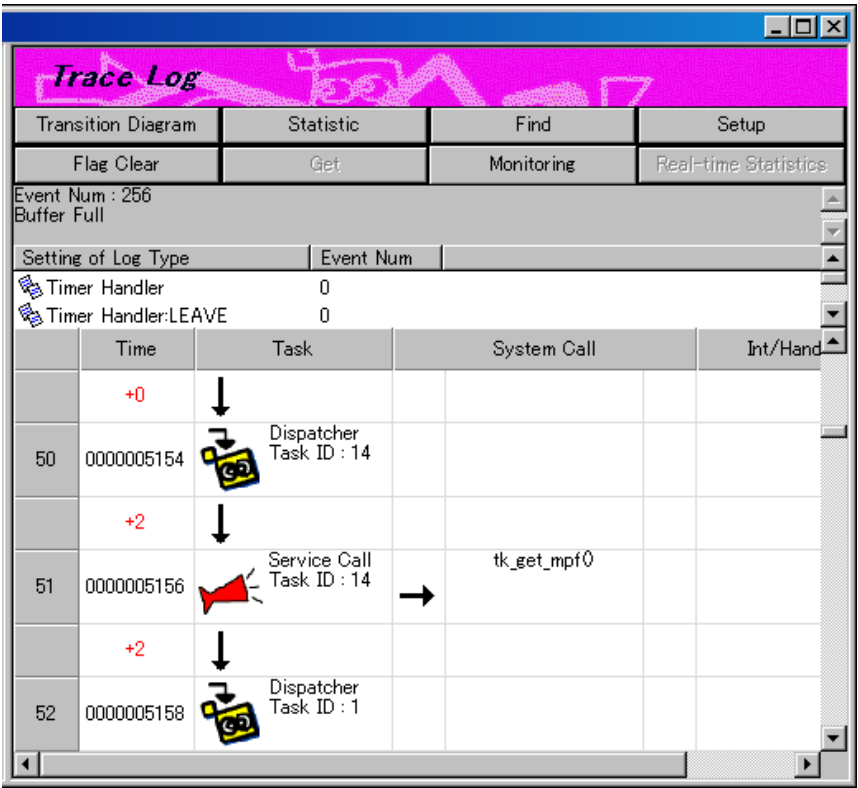
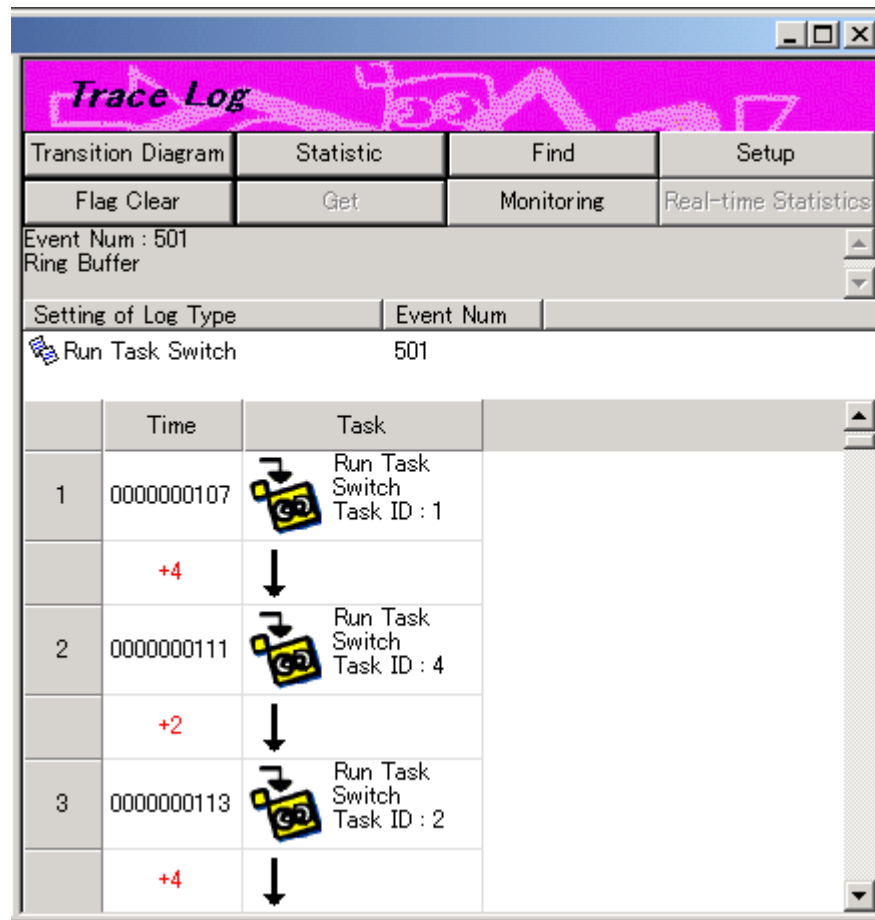












Figure 2.4-5 Example of Dispatch Log Display Data



## ■ Log Data Display Icons and Parameter List

The following shows the icons and parameters that  $\mu$ T-REALOS Analyzer displays in each item.

**Table 2.4-4 Log Data Display Icons and Parameters List**

Item	Icons	Contents of Display
Time	None	Black: Event time that occurred. Red: Difference of previous event.
TASK		Dispatch started. ID is the original task ID of the dispatcher.
		Dispatch stopped. ID is the destination task ID of the dispatcher.
		System calls issued from task are started. ID is the issued task ID
		System calls issued from task are stopped. ID is the issued task ID
		Mask space by the startup of the handler.
		Comment ID / comment character string
System Calls	None	System call name and parameter or return value.
Interrupt/handler		Interrupt and handler started. ID is the interrupt ID or the handler ID.
		Interrupt or handler stopped. ID is the interrupt ID or the handler ID.
		System calls issued from handler are started.
		System calls issued from handler are stopped.
Note	None	Not used.



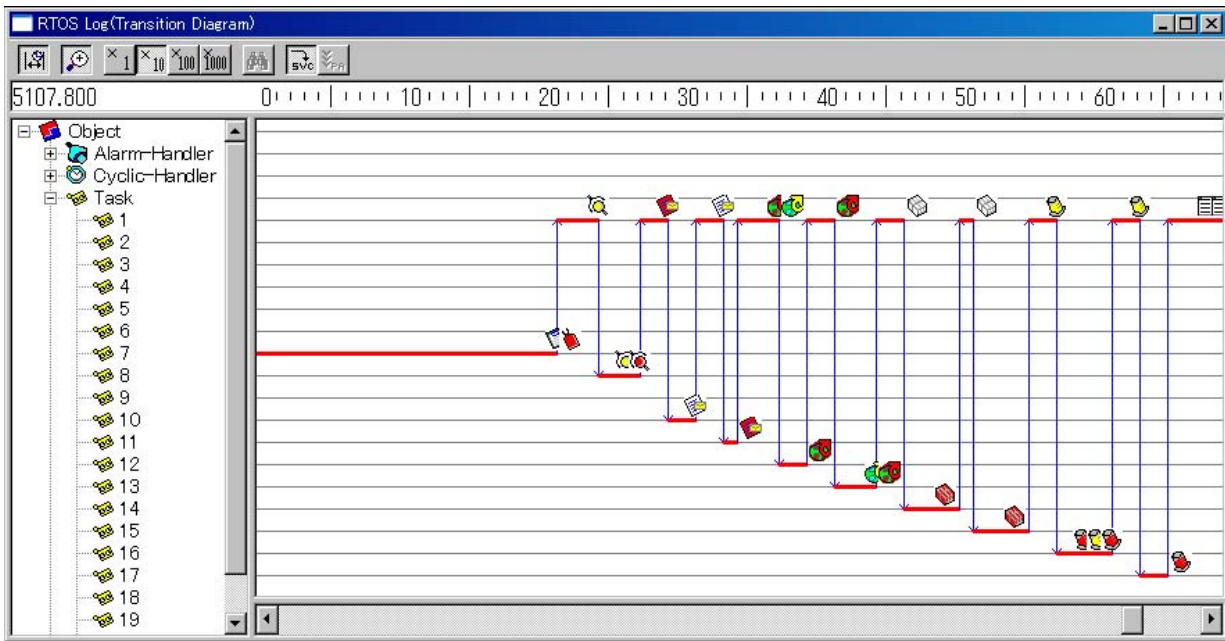
## 2.4.8 Transition Diagram Window

This describes the transition diagram window.

### ■ Opening the transition diagram window

Press the [Transition Diagram] button in the data display window to open that window.

Figure 2.4-6 Transition Diagram Window



↑ Object Index Window

↑ Transition Diagram Display Window



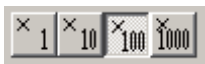
### ■ Object Index Window

Displays the tasks and timer handlers (cycle/cyclic startup and alarms) registered in the system, and the interrupt handlers in a tree format. However,  $\mu$ T-REALOS Analyzer will not display the objects that have not been selected in the log type.

## ■ Tool Bar

The display method can be changed by operating the button of tool bar.

**Table 2.4-5 Tool Bar of Transition Diagram Window**

Button	Explanation
	It changes whether it displays per time, or it displays per event. Push state: Display of a time unit (initial state) Pop state: Display in an event unit
	It changes whether it is made a whole display or a time unit is decided and displayed. In this case, it uses together with the magnifying-power button (refer to following) arranged on the right. Push state: Display of a time unit Pop state: Whole display
	Magnifying-power button It is 1 point per one system clock from the left. Hereafter, it is set as 10, 100, and 1000 point.


## ■ Line and Icon of Transition Diagram Display Window

To display the task transition lines with the module use log, it is necessary to select "Dispatcher: LEAVE" as the type of log to be acquired.  $\mu$ T-REALOS Analyzer displays the icons and lines shown in Table 2.4-6.

If you select "Run Task Switch" of "Dispatch log",  $\mu$ T-REALOS Analyzer displays only task transition lines.

The transition display window corresponds to the object index window, and  $\mu$ T-REALOS Analyzer displays events related to each ID of each object with the same height.






















**Table 2.4-6 Transition Diagram Window Lines and Icons**

Line/Icon	Explanation
Red Horizontal Line	Indicates RUN state
Blue Vertical Line (with arrow)	Indicates a dispatch. The direction of the arrow indicates a switch. Displayed based on the "Dispatcher: LEAVE" data.
	Indicates starting and stopping of the handlers. Displayed based on the "Timer handler", "Interrupt", and "Time out" data.

In the system call, in order to give intuition nature, the system call in which processing is related to some icons is assigned.

Assignment is shown in Table 2.4-7. If a red icon is displayed, the system call may enter a wait state.

Table 2.4-7 System Call Icon List

Icon	System Call
	tk_cre_tsk, tk_cre_sem, tk_cre_flg, tk_cre_mbx, tk_cre_mpf, tk_cre_cyc, tk_cre_mtx, tk_cre_mbf, tk_cre_mpl, tk_cre_alm, tk_cre_por, tk_def_dev, tk_def_int, tk_def_ssy, tk_opn_dev
	tk_del_tsk, tk_can_wup, tk_del_sem, tk_del_flg, tk_del_mbx, tk_del_mpf, tk_del_cyc, tk_del_mtx, tk_del_mbf, tk_del_mpl, tk_del_alm, tk_del_por, tk_cls_dev
	tk_sta_tsk, tk_sta_cyc, tk_sta_alm
	tk_ext_tsk, tk_exd_tsk, tk_stp_cyc, tk_stp_alm
	tk_ter_tsk, tk_sus_tsk, tk_sus_dev
	tk_rel_wai, tk_rsm_tsk, tk_frsm_tsk
	tk_chg_pri, tk_rot_rdq
	tk_ref_tsk, tk_ref_sem, tk_ref_flg, tk_ref_mbx, tk_ref_mpf, tk_set_tim, tk_get_tim, tk_ref_cyc, tk_get_tid, tk_dis_dsp, tk_get_reg, tk_ena_dsp, tk_ref_sys, tk_ref_ver, tk_ref_por, tk_get_dev, tk_get_otm, tk_ref_mtx, tk_ref_mbf, tk_ref_mpl, tk_ref_ssy, tk_ref_dev, tk_oref_dev, tk_set_reg, tk_lst_dev, tk_evt_dev, tk_ref_idv, tk_ref_alm
	tk_slp_tsk, tk_dly_tsk
	tk_wup_tsk
	tk_sig_sem, tk_wai_sem
	tk_set_flg, tk_clr_flg, tk_wai_flg
	tk_snd_mbx, tk_rcv_mbx
	tk_rel_mpf, tk_get_mpf
	tk_unl_mtx, tk_loc_mtx
	tk_snd_mbf, tk_rcv_mbf
	tk_rel_mpl, tk_get_mpl
	tk_acp_por, tk_fwd_por, tk_rpl_rdv, tk_cal_por
	tk_wri_dev, tk_swri_dev, tk_wai_dev, tk_rea_dev, tk_srea_dev
	Comment
	System call : LEAVE

## ■ Mouse Operations with the System Call Icons

The following functions are available the mouse on the system call icons.

- Hovering:

Stop the mouse on the system call icon for a determined amount of time.

μT-REALOS Analyzer displays a tool chip, the occurring event number and the occurring system call name.

- Double-click:

Double-click on the system call icon. Position on an event the data for which is specified in the data display window to scroll and activate the RAnalyzer window.

## 2.4.9 Statistics Windows

This describes the statistics window.

### ■ Statistics Window

The statistics window calculates the execution time between tasks and the number of dispatches from the sampled log data to display that information in graph format. It also displays the maximum consecutive executions and the minimum time.

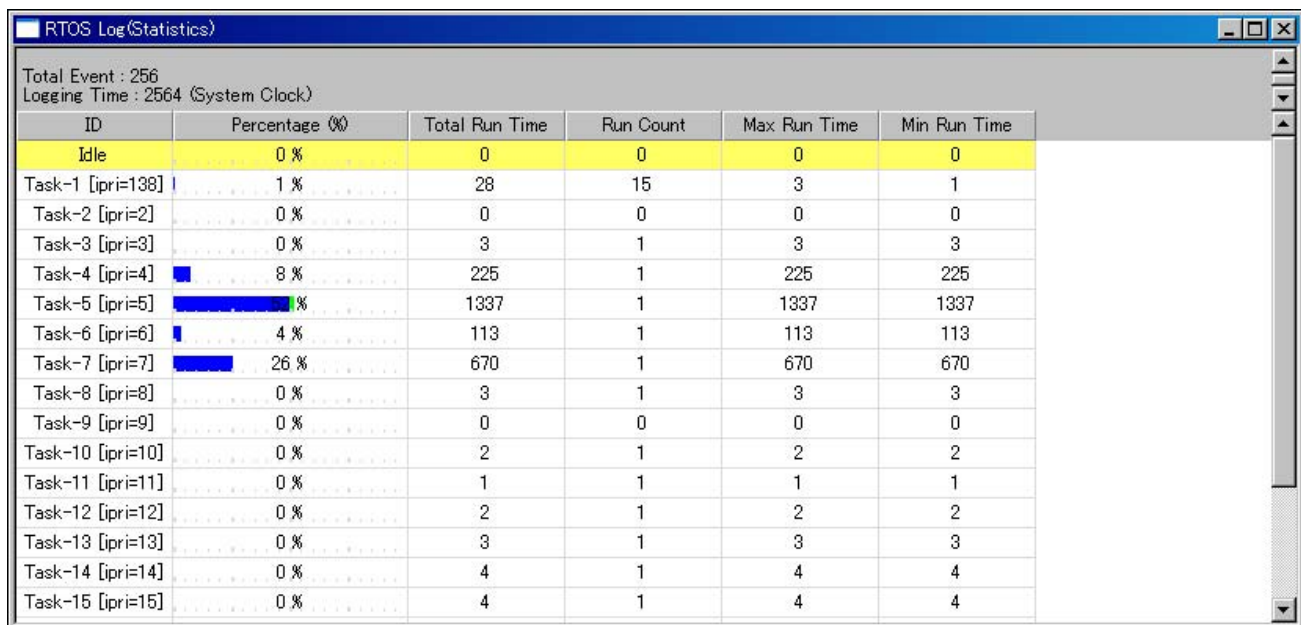
### ■ Opening the Statistics Window

You can open the statistics window when the following conditions are satisfied.

- The log is set to Log [Dispatch] or Log [Module use].
- Log acquisition has completed.

Clicking the [Statistics Graph] button in the data display window opens the window.

Figure 2.4-7 Statistics Window



## ■ Display Items in the Statistics Window

The following shows the items displayed in the Statistics Window.

**Table 2.4-8 Parameter List of Statistics Window**

Item	Explanation
Percentage (%)	Rate to entire sampling time of each task
Total Run Time	Total execution time of each task
Run Count	Dispatch count for each task
Max Run Time	Maximum consecutive execution time of each task
Min Run Time	Minimum consecutive execution time of each task

---

**Note:**

The maximum consecutive execution time of each task is important data to check the real-time capacity of the system. Be aware that it is the maximum time of the space that was sampled only.

---

## 2.4.10 Monitoring Window

This describes the Monitoring window.

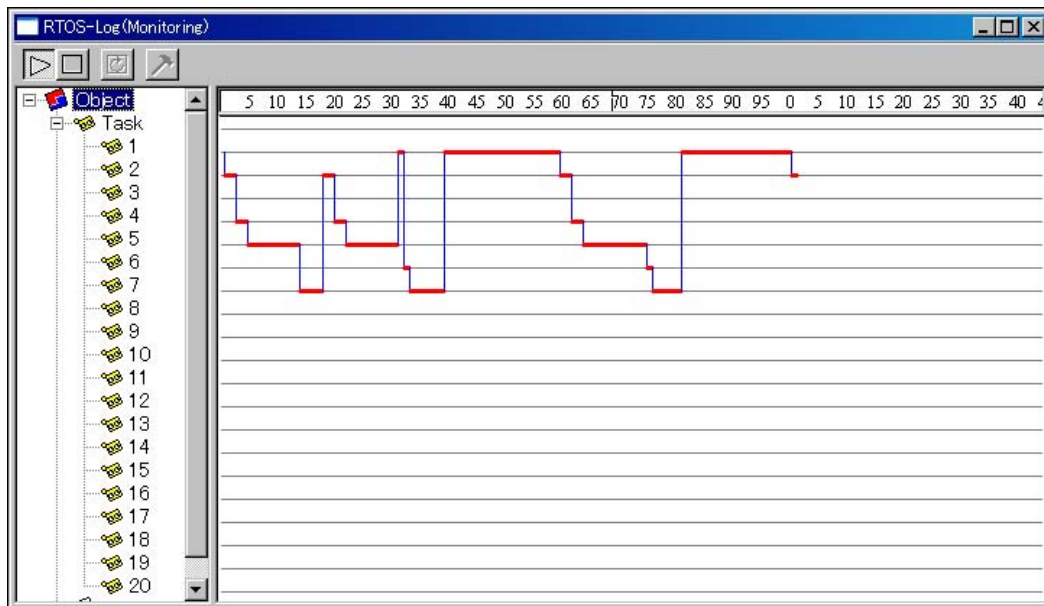
### ■ Monitoring Window

The monitoring window uses the debugger's pseudo on-the-fly function to obtain the ID of the currently executing task at fixed time intervals (the time interval can be modified in the "monitoring interval time" setting in the configuration settings). The window displays this data in a similar format to a task transition diagram. Execution switching to the idle task is shown by a transition to "Idle" at the bottom of the task tree.

### ■ Open the Monitoring Window

If the [Monitoring] button in the display window of a log setup is pushed, a window will open.

Figure 2.4-8 Monitoring Window







## ■ Operation of Monitoring (Tool Bar)

Each button of a tool bar performs operation of monitoring.

The function assigned to each button is shown in the following tables.

**Table 2.4-9 Operation of Monitoring**

Button	Explanation
	Monitoring is started. It asks whether to perform debugger and to start monitoring at the time of a click, when the debugger has stopped.
	Monitoring is stopped. It asks whether to stop debugger, when the debugger is executing at the time of a click.
	Display is cleared.
	This opens the configuration settings dialog box where you can set the sampling time for monitoring (See "2.4.11 Configuration Settings").

---

**Note:**

Monitoring cannot be used with a monitor debugger.

Also, the monitoring function cannot be used when using DSU3.

---



## 2.4.11 Configuration Settings

This section describes the configuration settings.

### ■ Configuration Settings Dialog Box

The configuration settings dialog box is used to set the sampling time and deadline settings. The conditions for setting a sampling time for realtime statistics and the deadline settings are as follows.

- Task analysis module type5 is used
- The log is set to "Statistics log - time measurement"

The sampling time for monitoring can be set regardless of the conditions.

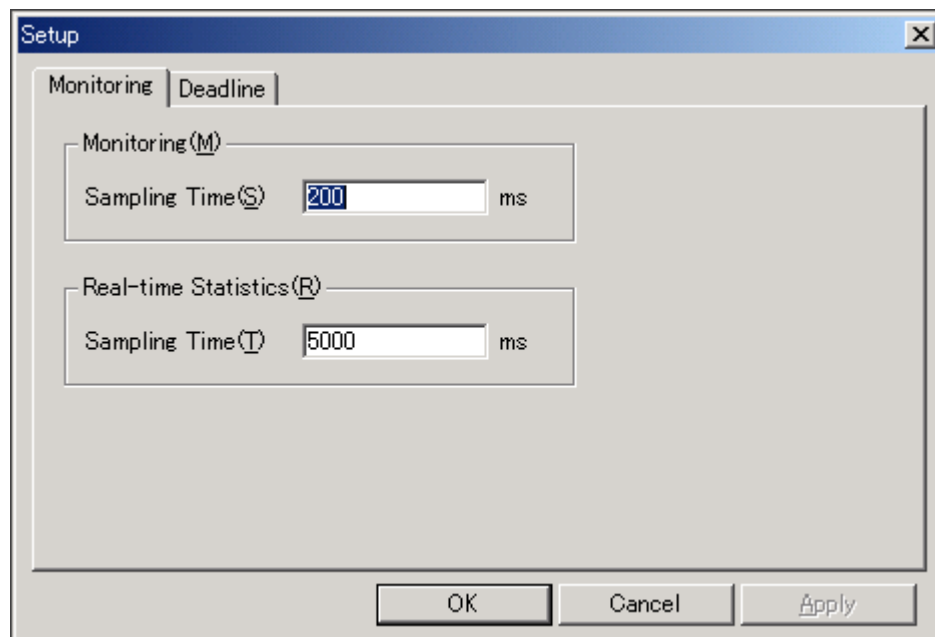
### ■ Opening the Configuration Settings Dialog Box

Click the [Configuration Settings] button in the data display window to open the dialog box.

### ■ Opening the Monitoring Settings Dialog Box

Click the "Monitoring" tab in the configuration settings dialog box to open the monitoring time settings dialog box.

**Figure 2.4-9 Monitoring Time Settings Dialog Box (When using R\_D\_dbgA5.lib)**



## ■ Operation of the Monitoring Time Settings Dialog Box

- Monitoring sampling time  
Enter the data update interval for the monitoring window and click the [OK] or [Apply] button.  
Setting range: 200 to 32767ms  
The default setting is 200ms.
- Realtime statistics sampling time  
Enter the data update interval for the realtime statistics window and click the [OK] or [Apply] button.  
Setting range: 200 to 32767ms  
The default setting is 5000ms.

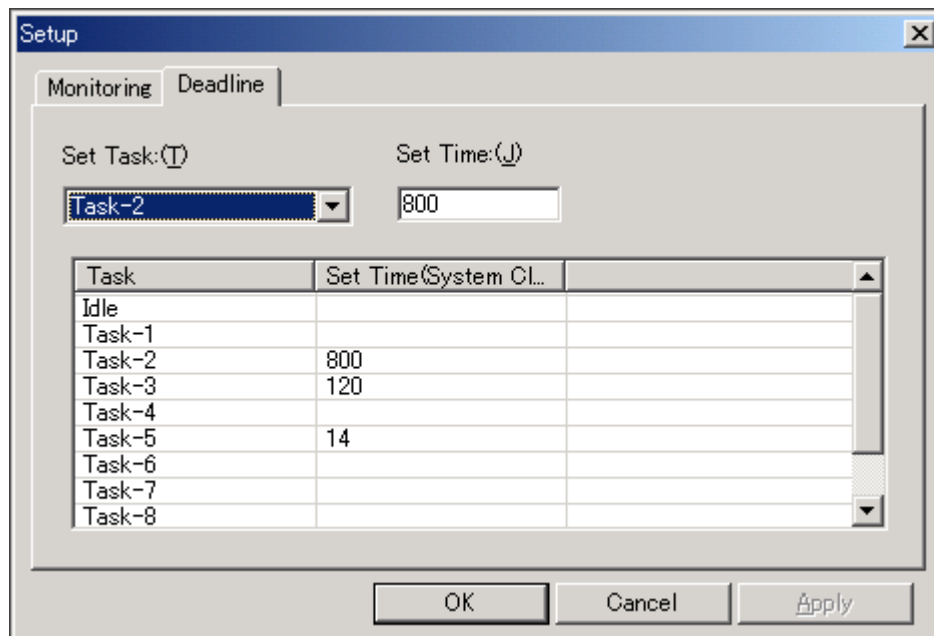
Note:

The realtime statistics window may not be able to display the statistics data for all tasks if the number of tasks increases and the specified sampling interval is too short.

## ■ Opening the Deadline Settings Dialog Box

Click the "Deadline" tab in the configuration settings dialog box to open the deadline settings dialog box.

**Figure 2.4-10 Deadline Settings Dialog Box**



### ■ Operation of the Deadline Settings Dialog Box

1. Select a task for which to specify a deadline time.
2. Enter the upper limit for the task execution time.  
Specify the time in milliseconds in the range 1 to 65535.
3. Click the [OK] or [Apply] button to apply your settings.  
 $\mu$ T-REALOS Analyzer displays the new setting in the deadline list.

### ■ Clearing Settings

1. Select the task in the deadline list from which to clear the deadline time.
2. From the right-click pop-up menu, select "Clear Setting".
3. Confirm that the time has been cleared from the deadline list.

## 2.4.12 Explanation of Realtime Statistics

This section describes the realtime statistics window.

### ■ Realtime Statistics

The realtime statistics window displays the results from the task analysis module in graphical form.

The window can display the percentage, total execution time, average execution time, maximum execution time, minimum execution time, number of executions, and times when the deadline time was exceeded for each task. The window can be monitored at periodic intervals.

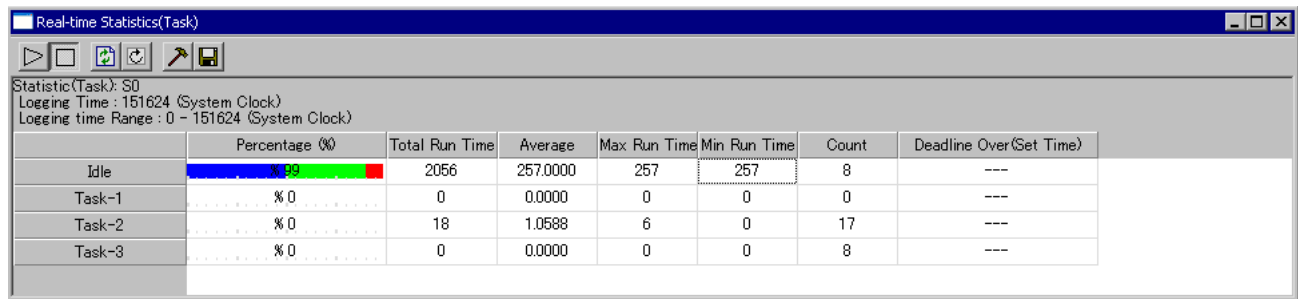
### ■ Opening the Realtime Statistics Window

You can open the realtime statistics window when the following conditions are satisfied.

- The type5 task analysis module is used
- The log is set to "Statistics log - time measurement".

Clicking the [Realtime Statistics] button in the data display window opens a window.

**Figure 2.4-11 Realtime Statistics Window**



## ■ Data Displayed in the Realtime Statistics Window

The realtime statistics window displays the following data.

**Table 2.4-10 Data Displayed in the Realtime Statistics Window**







Item	Description
Percentage	Task execution time as a percentage of the total logging time
Total Run Time	Total task execution time for task
Average	Average task execution time for task
Max. Run Time	Longest task execution time for task
Min. Run Time	Shortest task execution time for task
Count	Number of times task executed
Deadline Over (Set Time)	Number of times task exceeded the upper limit for the execution time (time setting)

## ■ Operation of the Realtime Statistics Window (Tool Bar)

Use the tool bar buttons to operate the realtime statistics window.

The table below lists the function of each button.

**Table 2.4-11 Operation of the Realtime Statistics Window**

Button	Function
	Start monitoring If the debugger is halted when the button is clicked, a prompt asks you whether to start the debugger so monitoring can start.
	Stop monitoring If the debugger is running when the button is clicked, a prompt asks you whether to stop the debugger. If you leave the debugger running, data continues to be accumulated but the screen is not updated.
	Clicking this button while monitoring is in progress updates the screen.
	Clears the display. Clears the accumulated data.
	Opens the configuration settings dialog box.
	Saves the displayed data log. The data is saved in csv format.

## 2.4.13 Output of Log File

The acquired log data can be output to the file. The extension of the data file is (\*.log). This section explains how to output log file.

### ■ Output of Log File

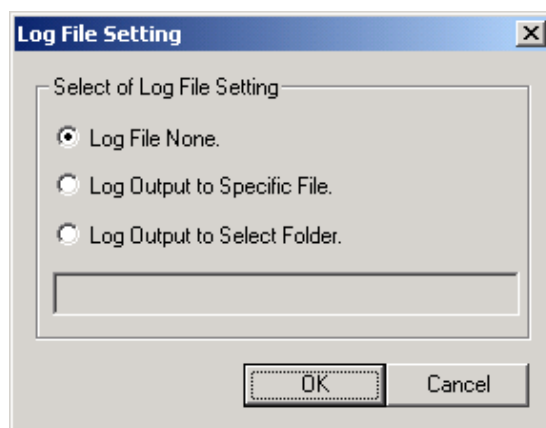
The acquired log data can be outputted to the file of text form. The data can also be viewed with "Log Viewer" located on the product CD-ROM.

The extension of the data file is (\*.log).

In order to set up a file output, a pop-up menu opens by carrying out a right click by the log setting selection list (setting selection window).

If "Log File Setting..." of a pop-up menu is clicked, the following "Log File Setting" dialog will open.

**Figure 2.4-12 "Log File Setting" dialog**



### ■ Selection of Output Method

You can select one of the following three output methods:

- Log File None  
μT-REALOS Analyzer does not output a log file.
- Log Output to specific File  
μT-REALOS Analyzer outputs the log to the file entered in the edit box. μT-REALOS Analyzer updates its contents each time new data is acquired. This method is effective if you need only the latest data. You can view the data with "Log Viewer" by dragging and dropping the file.  
Edit box input example: C:\log\logfile.log
- Log Output to Select Folder  
The file will be output to the folder entered in the edit box. The file will be saved with the date data as its file name. μT-REALOS Analyzer will create a new file each time new data is acquired. This method is effective if you accumulate data. You can view the data with "Log Viewer" by simply activating to read registry information and selecting data to be viewed in data list form.

## 2.5 System Call Issue

---

**The system call issue function issues system calls from the  $\mu$ T-REALOS Analyzer during breaks. This section describes the window issuing system calls and how to operate it.**

---

2.5.1 System Call Issue Function

2.5.2 Explanation of the Window (System Call Issue)

2.5.3 Issuing System Call

## 2.5.1 System Call Issue Function

This section describes system call issue function specifications.

### ■ Precautions for Use

When you use this function, execute only from task body.

### ■ Call Command

System calls in the  $\mu$ T-REALOS Analyzer use the Debugger call command.

Therefore, restrictions of the Debugger call command exist with this function, so we recommend reading the "Workbench Command Reference Manual".

### ■ List of System Calls that can be Issued

For system calls that can be issued, see Table 2.5-1 below.

**Table 2.5-1 List of System Calls that can be Issued**

System call function classification	System calls that can be issued
Task management function	tk_cre_tsk, tk_del_tsk, tk_sta_tsk, tk_ter_tsk, tk_chg_pri, tk_ref_tsk
Task-attached synchronization function	tk_wup_tsk, tk_can_wup, tk_rel_wai, tk_sus_tsk, tk_rsm_tsk, tk_frsm_tsk
Synchronization/communication function	tk_cre_sem, tk_del_sem, tk_sig_sem, tk_ref_sem, tk_cre_flg, tk_del_flg, tk_set_flg, tk_clr_flg, tk_ref_flg, tk_cre_mbx, tk_del_mbx, tk_snd_mbx, tk_ref_mbx
Extended synchronization/communication function	tk_cre_mtx, tk_del_mtx, tk_ref_mtx, tk_cre_mbf, tk_del_mbf, tk_ref_mbf, tk_ref_por
Memory pool management function	None
Time management function	tk_set_tim, tk_get_tim, tk_get_otm, tk_cre_cyc, tk_del_cyc, tk_sta_cyc, tk_stp_cyc, tk_ref_cyc, tk_cre_alm, tk_del_alm, tk_sta_alm, tk_stp_alm, tk_ref_alm
Interrupt management function	tk_def_int
System state management function	tk_rot_rdq, tk_get_tid, tk_ref_sys, tk_ref_ver
Subsystem function	tk_def_ssy, tk_ref_ssy
Device management function	None

The system call issue function of the  $\mu$ T-REALOS Analyzer cannot issue system calls that cannot be issued from a non-task block.

Example:

tk\_ext\_tsk and other system calls that change their own task state.



### ■ System Calls that is not Linked

On  $\mu$ T-REALOS, system calls that are not used for design are not linked. Because the  $\mu$ T-REALOS Analyzer specifications specify that entries on the target are to be called, system calls that are not linked cannot be issued.

### ■ Pointer Type Arguments

Pointer type arguments require that that state is ensured on the target. The Debugger and  $\mu$ T-REALOS Analyzer do not have specifications to ensure it.

---

Note:

For restrictions of the system call issue function, see "APPENDIX D Restrictions of Issuing System Call".

---

## 2.5.2 Explanation of the Window (System Call Issue)

This section describes the System Call issue window.

### ■ Opening the System Call Issue

Press [Setting] button in the function selection pane, then select [System Call] icon.

### ■ Functions of Each Pane when System Call Issue Function is Selected

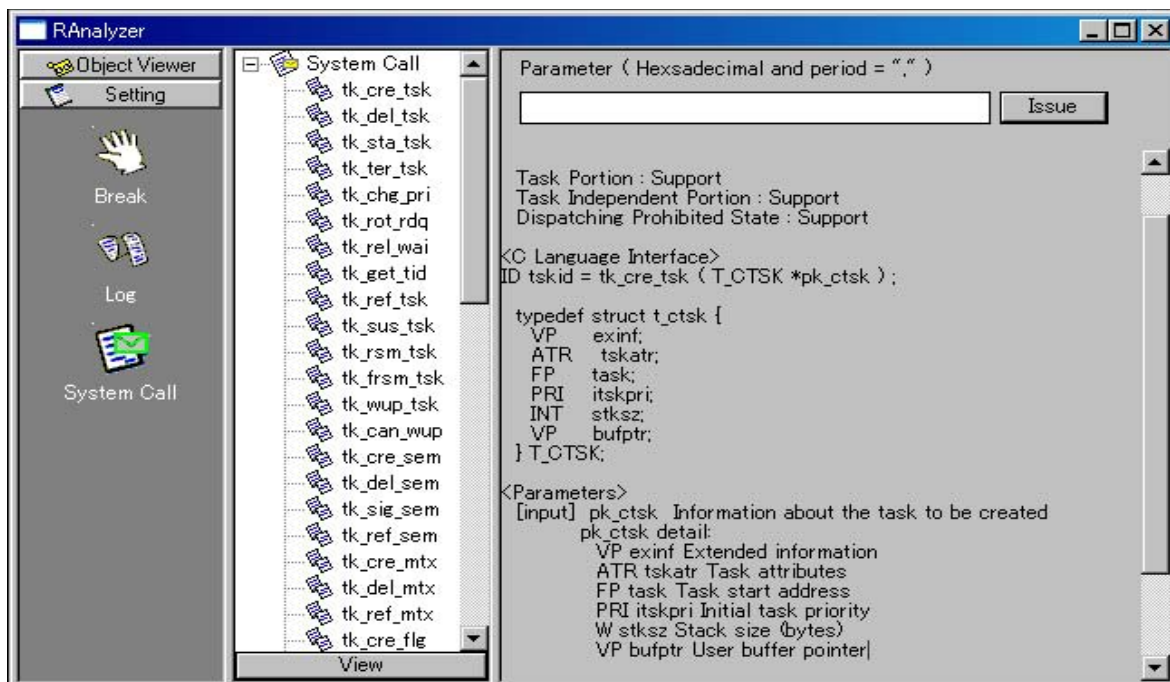
When "System Call" is selected, the following functions are assigned to each pane. Functions assigned are as shown below beginning from the left pane.

- "System Call" fixed
- System Call selection window
- Argument setting window

**Figure 2.5-1 System Call Issue Window**

Issuing System Call Selection

Argument Setting Window



### ■ System Call Selection List (System Call Selection Window)

Displays the system calls that can be issued. Select the system call to issue and press the [View] button to display the system call specifications selected in the argument setting window.

**■ System Call Specifications Display Field (Argument Setting Window)**

μT-REALOS Analyzer displays the selected system call specifications.

**■ Argument Input Box (Argument Setting window)**

Separate arguments with a comma (,).

**■ [Issue] Button (Argument Setting Window)**

Issues system calls with the value set by each control. μT-REALOS Analyzer displays the return value in a dialog box.

## 2.5.3 Issuing System Call

---

**This section describes the procedures to issue a system call.**

---

### ■ How to Issue

1. Select the "System Call" to issue in the System Call Selection window.
2. Press the [View] button.
3. Set the argument.
4. Press the [Issue] button.
5. The return value is displayed in a dialog box.

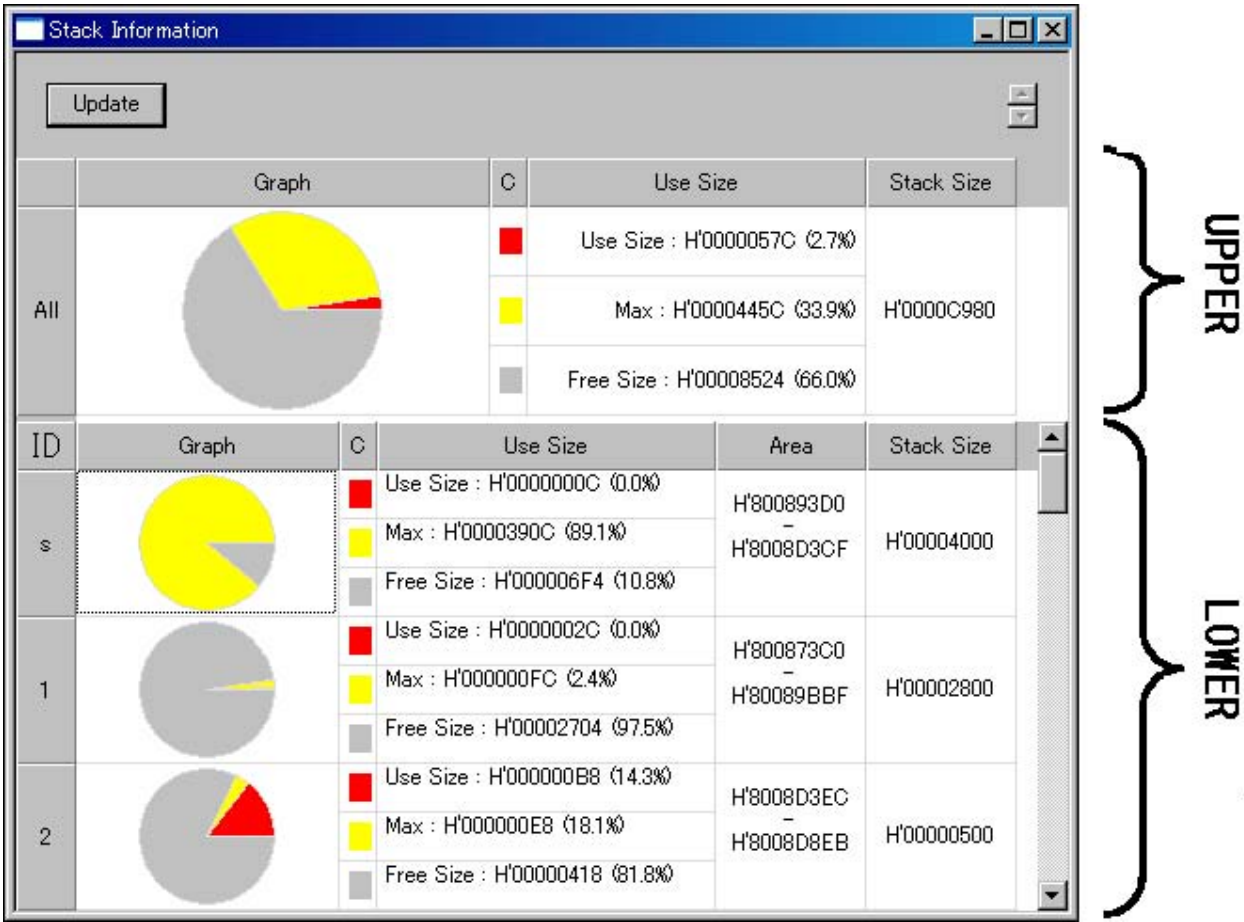
## 2.6 Stack Information

Stack information displays the use conditions of each task's stack in graph format. The Stack Information window is opened by using [Stack Information] in the [RAnalyzer] menu. The stack "Max" value is displayed by selecting [Start] command of [Stack Used Amount Analysis] in the [RAnalyzer] menu. The stack "Max" value is not displayed by selecting [End] command.

### ■ Opening the Stack Information Window

Open the "Stack Information Window" using [Stack Information] in the Workbench [RAnalyzer] menu. Because information is collected when opening, it is expected that the amount of time to open will be approximately 1 second for each registered task. The following shows the Stack Information window.

Figure 2.6-1 Stack Information Window



## ■ Meaning of Terms

The table below lists the meanings of the terms used in the stack information window.

Term	Meaning
Use Size	Currently used stack area
Max	Largest stack area usage since stack usage analysis was started.
Free Size	Unused stack area
Area	Stack top address - stack bottom address
Stack Size	Size of stack area
ID	Task IDs (s indicates the system stack and i is the idle stack)
Graph	Pie chart showing the current usage, maximum usage, and unused area.
C	Colors used in the pie chart Used : Red Maximum : Yellow Unused : Grey
All	Stack information totals for each task, system stack, and idle stack.

## ■ [Update] Button

This button updates the information. The information while Kernel system is running is not guaranteed.

Open after getting information when opening for the first time. After that, press the update button to get the latest information.

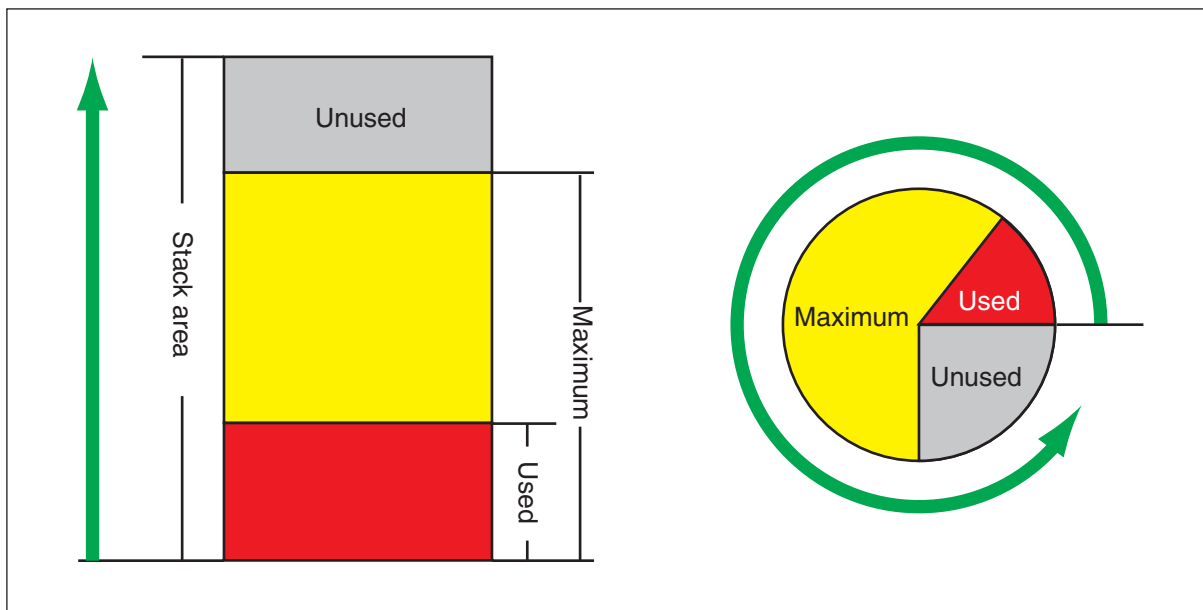
## ■ Graph

Two different graphs are displayed (see Figure 2.6-1).

- Upper graph: Graph of overall system stack area  
Shows the used and unused size of the system stack, idle stack, and all task stacks.  
 $\mu$ T-REALOS Analyzer does not add the data of tasks for which tk\_cre\_tsk has not been called at the time of data acquisition.
- Lower graph: Graph of each task stack  
Shows the used and unused stack size for each task ID.  
 $\mu$ T-REALOS Analyzer does not display tasks for which tk\_cre\_tsk has not been called at the time of data acquisition.

The graph shows the area for each task anti-clockwise starting from 3 o'clock (see Figure 2.6-2).

**Figure 2.6-2 Display of Pie Chart**



## ■ Stack Usage Analysis

You can display the largest amount of stack area used up to date.

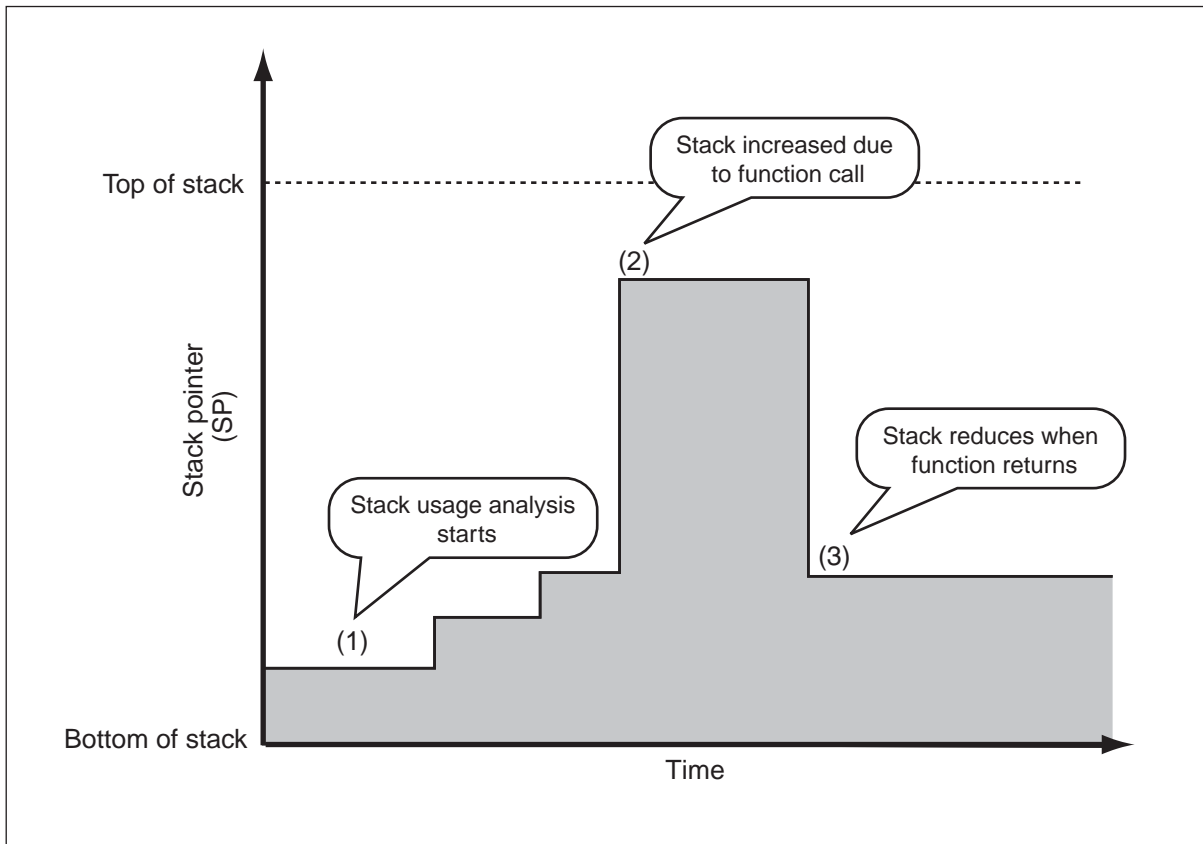
- Menu command [RAnalyzer] - [Stack Usage Analysis] - [Start]

The [Maximum] is calculated from the unused area and stack area.

The [Maximum] is calculated as follows.

Example:

**Figure 2.6-3 Stack Pointer Trend Graph**

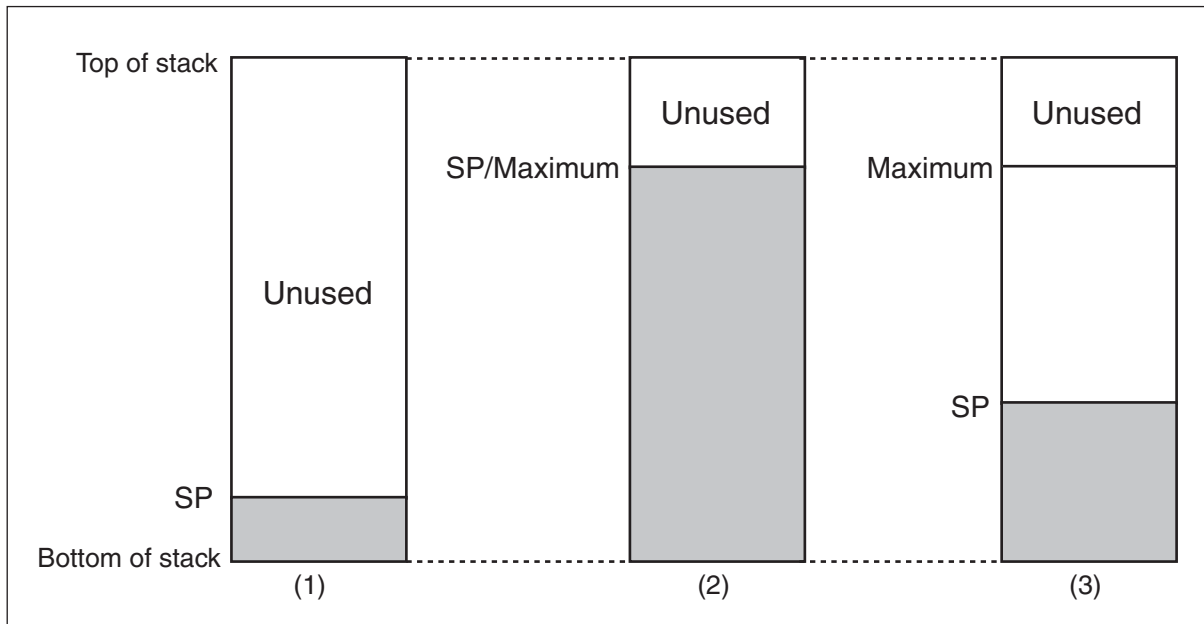


When the relationship between stack pointer and time is as shown in Figure 2.6-3, sections [1] to [3] in the graph indicate the following.

- (1) Stack usage analysis starts
- (2) Stack area usage increases due to function call
- (3) Stack area usage reduces when function returns

The stack areas for cases (1) to (3) in the figure are shown in Figure 2.6-4.

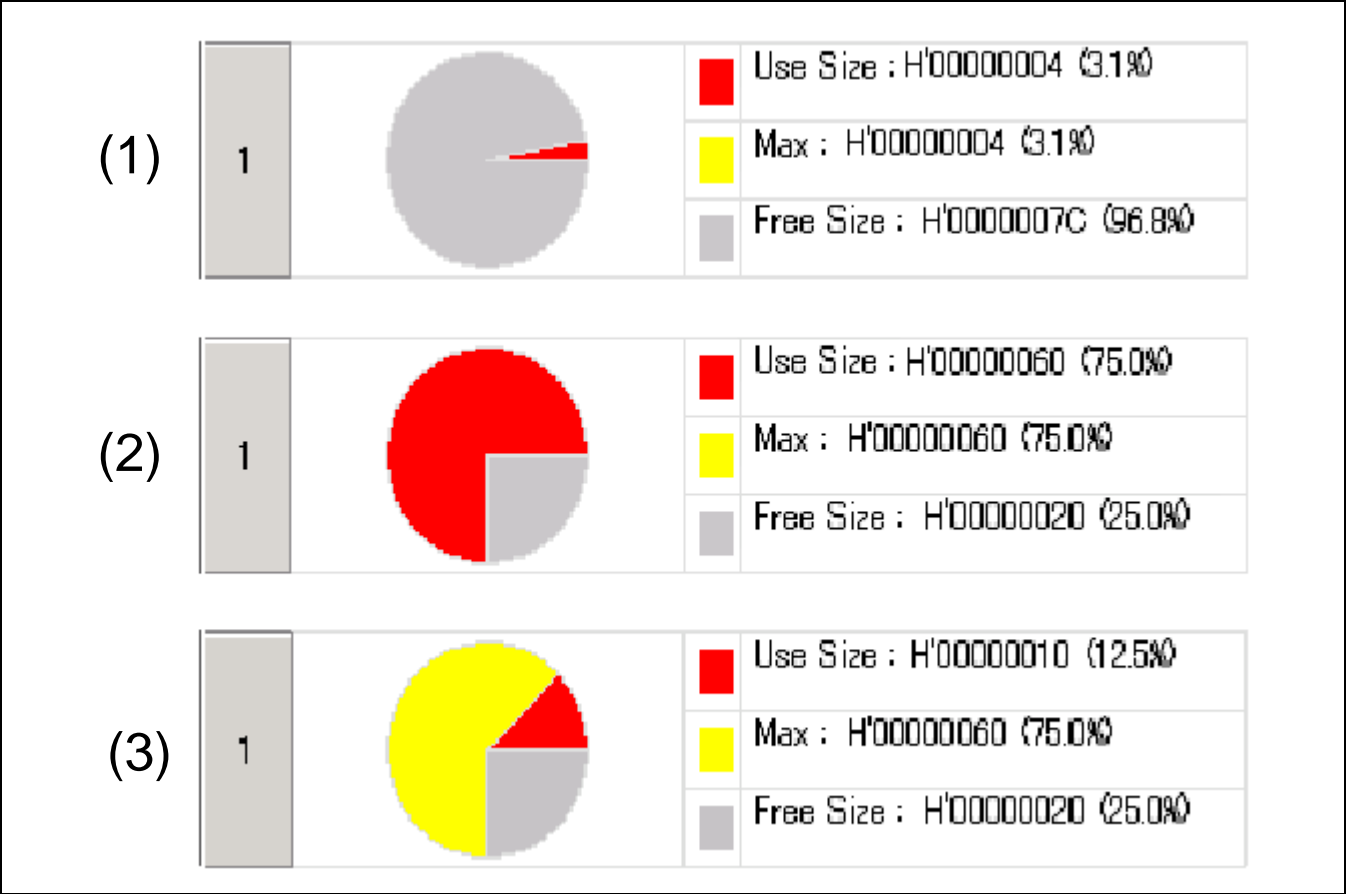


**Figure 2.6-4 Stack Areas for Cases (1) to (3)**

- (1) The [Used] stack area is the size from the bottom of the stack to the SP (stack pointer) and the [Maximum] is the same as the [Used] value.
- (2) The [Used] stack area is the size from the bottom of the stack to the SP. At this time, [Used] = [Maximum].
- (3) The [Used] stack area is the size from the bottom of the stack to the SP. The value reached at (2) remains as the [Maximum].

μT-REALOS Analyzer displays the actual pie chart as Figure 2.6-5.

Figure 2.6-5 Actual Display for (1) to (3)



● Menu command [RAnalyzer] - [Stack Usage Analysis] - [Stop]  
Hides the [Maximum] value.

Note:

Do not start stack usage analysis until  $\mu$ T-REALOS has initialized. Also, provide a sufficiently large stack area to avoid stack overflows (See "APPENDIX E Cautions when Using Stack Use Analysis").

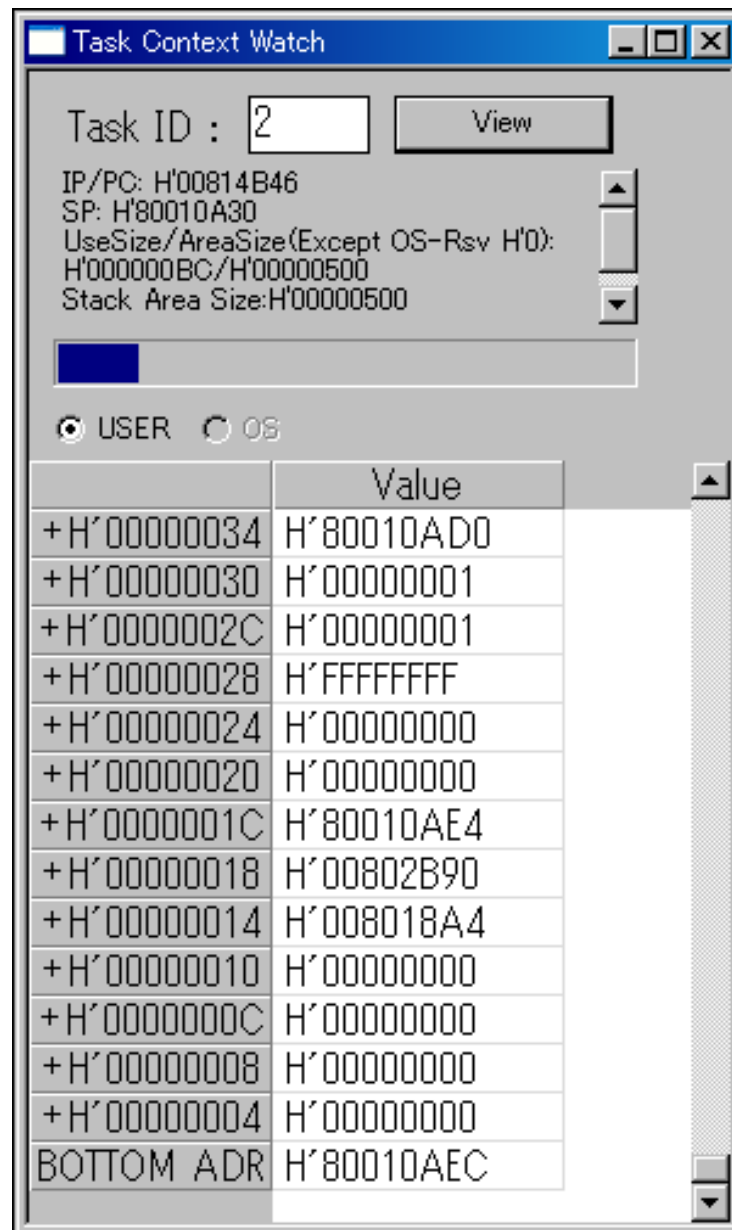
## 2.7 Task Context Watch

Task context watch is to display the contents of each task context. The Task Context Watch window can be opened using [Task Context Watch] in the [RAnalyzer] menu.

### ■ Opening the Task Context Watch

Open the "Task Context Watch" using [Task Context Watch] in the Workbench [RAnalyzer] menu.

Figure 2.7-1 Task Context Watch



## ■ Watching the Specified Task Context

Specify the task ID to display the task context.

The following contents are displayed by pressing the [View] button.

- IP/PC: Current IP/PC of the specified task
- SP: Current SP of the specified task
- Use Size/Area Size:  
Usage amount calculated from the current SP, Stack size
- Bar graph: Calculates the % from the above values and displays an image of the usage capacity under the above display.
- Context List: Switches display of the data in OS reserve area and the contents of the user stack.

# **CHAPTER 3**

---

## ***TASK ANALYSIS MODULE***

**This chapter describes the "task analysis module" used in sampling by the module use logs.**

- 3.1 Types of Task Analysis Modules
- 3.2 How to Use Comments
- 3.3 How to Customize

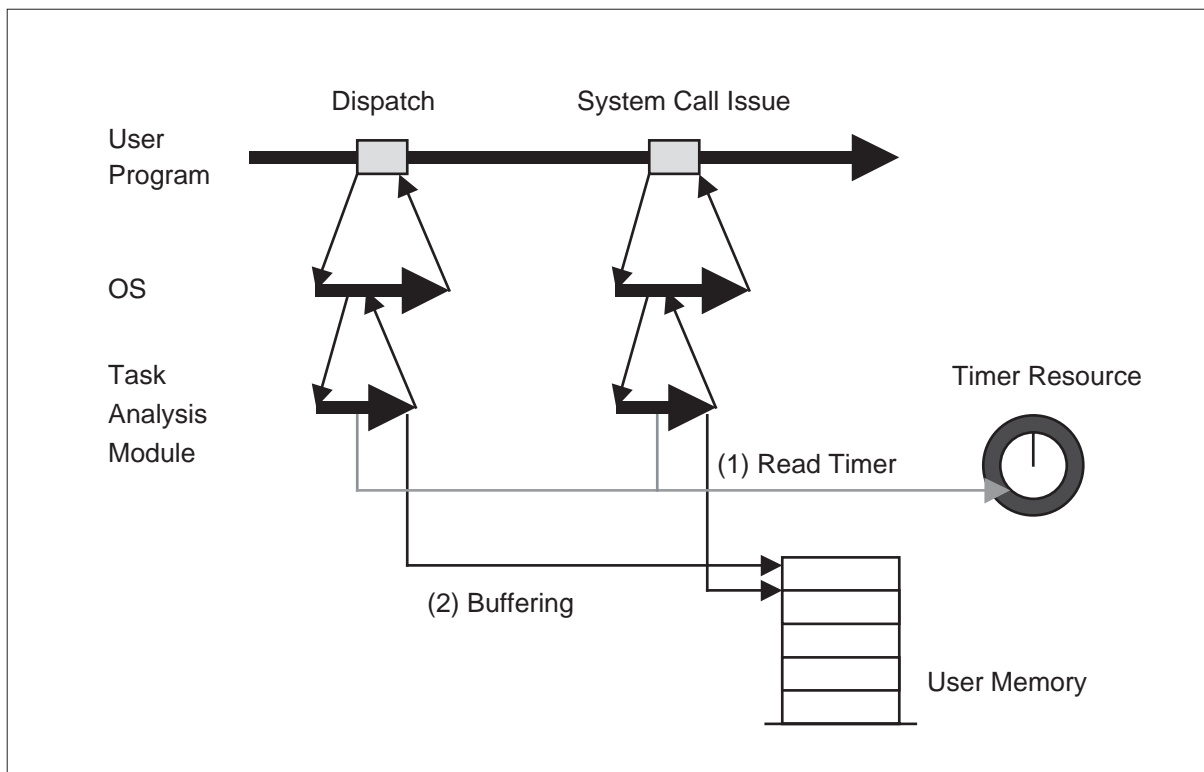
## 3.1 Types of Task Analysis Modules

This section describes the types of task analysis modules.

### ■ Task Analysis Module

The task analysis module is a program that adds processes to the  $\mu$ T-REALOS kernel program task dispatcher and system call issuing timing and acquires the RTOS operating log. It is necessary to realize the log function of the "module use log". There are three types of task analysis modules provided by the  $\mu$ T-REALOS Analyzer. Also, to realize a RTOS log with a time stamp, use the  $\mu$ T-REALOS system clock in the time stamp. The following describes the operation of the task analysis module.

**Figure 3.1-1 General Description of the Operation of the Task Analysis Module**



### ■ Selecting the Task Analysis Module to Use

It is possible to select using the Configurator's "Debug Definition". The default setting, provided by the kernel, is a program that does nothing.

## ■ Function of Each Type

The accuracy of the time stamps recorded during sampling is different depending on which type you select. However, the more accurate the time stamp, the greater the overhead. The table below lists the time stamp accuracy for each type.

**Table 3.1-1 Time Stamp Accuracy for Each Type**

Type Name	Library Name	Description
type1	R_D_dbgA1.lib	Does not record time stamps.
type2	R_D_dbgA2.lib	Records the time from the system clock.
type3	R_D_dbgA3.lib	Records the count from a resource timer.
type5	R_D_dbgA5.lib	This type is used for the statistics log.

## ■ Customization File for Specifying the Buffer Size and Number of Tasks

R\_D\_siz.c is the customization file used to specify the buffer and number of task when using the task analysis module. The default buffer size is 128 and the default number of tasks is 1. See Section "3.3 How to Customize" for details on how to customize these settings.

## ■ File for Timer Customize

If you use SOFTUNE  $\mu$ T-REALOS/FR, R\_D\_cnt.c is a file for reload timer customizing. The default setting for the timer is 12500. See Section "3.3 How to Customize" for details on customizing.

## 3.2 How to Use Comments

---

**This section describes how to use comment event in the module use log.**

---

### ■ Comment Event

Comment events are used when setting the module use log.

Normal events sample only the behavior dependent on the  $\mu$ T-REALOS that occurs in the  $\mu$ T-REALOS. Comment events, occur when a user describes a hook for a comment event at any location.

However, comment events cannot be used when a task analysis module is not set.

### ■ Header Files Needed for Include

Include the "R\_D\_custom.h" in the utkernel\xxx\dbg below the directory where  $\mu$ T-REALOS is installed, as a comment in the source whose event you want to register.

### ■ C Language Interface of the Hook for a Comment Event

The following shows the C language interface of the hook for a comment event.

C Language Interface	void _R_D_func_Custom(ID id, char* strbuf)
Mounting Position	User set
Argument	ID id : Any user number char* strbuf : Buffer header address of the comment character string (up to 20 characters)

### ■ How to Describe Comment Events

By describing using the method shown below, a comment event occurs and if the comment is selected in the log setting, a sampling occurs.

```
#include "R_D_custom.h"

:
char chrbuf [20] = {'R', 'e', 'a', 'l', 'o', 's'};
funcA ()
{
    ID id = 0;
:
/* Calling USER COMMENT */
_R_D_func_Custom (id, chrbuf) ;
:
}
```



### 3.3 How to Customize

To change the buffer size, number of tasks and the timer resource, it is necessary to change the contents of the file for customizing the task analysis module.

#### ■ Changing the Buffer Size

Customize R\_D\_siz.c as shown below and rebuild.

You can rewrite the "TRC\_DATA\_NUM" value defining the buffer size to "0 to 2048". The default setting is "128". Set to 0 if using the type5 task analysis module.

```

/*****
/* [Enable Custom ! ] : DEFINE DATA NUM                                     */
/*****
#define TRC_DATA_NUM 128
#define USER_TASK_NUM 1

```

#### ■ Changing the Number of Tasks

The buffer size of number of tasks must be changed to use statistics log and dispatcher break. Make sure to provide buffer size to cover total number of tasks of dynamic creation and statistic creation.

Customize R\_D\_siz.c as shown below and rebuild.

To define the buffer size, set the USER\_TASK\_NUM value in the range 1 to 32767. The default is 1.

```

/*****
/* [Enable Custom ! ] : DEFINE DATA NUM                                     */
/*****
#define TRC_DATA_NUM 128
#define USER_TASK_NUM 1

```

#### ■ Changing the Timer Resource

Rebuild after customizing R\_D\_cnt.c.

You can rewrite the value of reload timer value which defines "R\_D\_RELOAD\_TIMER".

The default setting is "12500".

This reload timer value is setup in accordance with the sample system of  $\mu$ T-REALOS.

```

/*   Reload Timer Initial Value   */
extern const int R_D_RELOAD_TIMER;
const int R_D_RELOAD_TIMER = 12500;

```



# ***APPENDIX***

---

**The appendix contains the analyzer processing time, error messages and restrictions.**

APPENDIX A Speed of Information Acquisition

APPENDIX B Symbols/Sections of the Task Analysis Module

APPENDIX C Error Messages

APPENDIX D Restrictions of Issuing System Call

APPENDIX E Cautions when Using Stack Use Analysis

APPENDIX F Restrictions with Debugger Type

# APPENDIX A    Speed of Information Acquisition

This section describes the speed of information acquisition.

■ **Measurement Results when Acquiring Information with ICE.**

The amount of time required for acquiring information is described below. As the number of objects increases, the time required for acquisition increases relatively. Also, objects that have not been created are accessed to check the creation information.

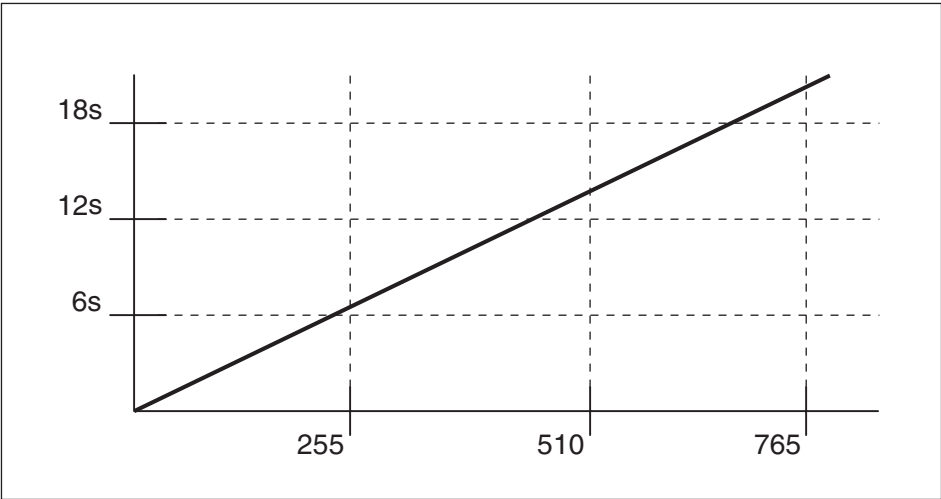
**Example:**

If only one task has been created out of 10 tasks, it will require the amount of time for 10.

CASE : DSU3

- The time required for information acquisition for 255 tasks is 8.0 seconds.
- The time required for information acquisition for 255 semaphore is 3.0 seconds.
- The time required for information acquisition for 255 tasks when waiting on one event flag is 2.0 seconds.
- The time required for information acquisition for the READY queue is 2.0 seconds.

**Figure A-1 Task Information Acquisition Time Requirement Calculation Table**



## APPENDIX B Symbols/Sections of the Task Analysis Module

This section describes the symbols and sections used by the module "task analysis module" used by the  $\mu$ T-REALOS Analyzer for acquiring logs.

Module	Section	Symbol
R_D_dbgA1.c	_kernel_dbginit_sc	R_D_ptrctop R_D_trctop2 R_D_trccnt R_D_loopcnt R_D_log_bufful_set R_D_log_bufful_evt R_D_log_flag_sta R_D_log_type R_D_log_id R_D_log_scadr R_D_log_prm0 R_D_log_prm1 R_D_log_prm2 R_D_log_ip R_D_log_timL R_D_log_timR R_D_log_ercd R_D_stackflag R_D_stkwpat R_D_stktskid R_D_tsk_all
R_D_dbgA5.c	_kernel_dbginit_sc	R_D_ptrctop R_D_trctop2 R_D_trccnt R_D_loopcnt R_D_log_bufful_set R_D_log_bufful_evt R_D_log_flag_sta R_D_log_type R_D_log_id R_D_log_scadr R_D_log_prm0 R_D_log_prm1 R_D_log_prm2 R_D_log_ip R_D_log_timL R_D_log_timR R_D_log_ercd R_D_stackflag R_D_stkwpat R_D_stktskid R_D_tsk_all R_D_dead_line_buf R_D_PA_func

(Continued)

## APPENDIX

(Continued)

Module	Section	Symbol
R_D_cnt.c	_kernel_dbginit_sc	R_D_tc_channel R_D_tc_mode R_D_tc_rlcmd R_D_tc_val
R_D_siz.c	_kernel_dbgdata_sc	R_D_trcbuf
	_kernel_dbginit_sc	R_D_trcend R_D_trc_data_num R_D_bufsiz R_D_task_break R_D_Pre_break_flg R_D_Post_break_flg R_D_task_num R_D_Pre_break_all R_D_Post_break_all R_D_Pre_break_all_pass R_D_Post_break_all_pass R_D_Pre_break_all_pass_ini R_D_Post_break_all_pass_ini

## APPENDIX C Error Messages

This section describes the error messages in  $\mu$ T-REALOS Analyzer.

### ■ Error Messages (E48xxM)

E4801M	Invalid ID. ( by Debugger )
[ Content ]	The ID requested to the $\mu$ T-REALOS Analyzer does not exist.
[ Handling ]	The requested ID does not exist. Input the correct ID and retry.

E4802M	Memory Access Error.
[ Content ]	READ/WRITE to an address that cannot be accessed was specified.
[ Handling ]	<ol style="list-style-type: none"> <li>1. The area where the management area of OS is arranged is in the state where it cannot be accessed. Please do not use <math>\mu</math>T-REALOS analyzer's function until access becomes possible.</li> <li>2. Since the management information on OS was destroyed, it failed in analysis processing.</li> </ol>

E4803M	Out of ID.
[ Content ]	The outside of the range or 0 were specified for specified ID.
[ Handling ]	Please reinput specification of ID to fit in the range.

E4804M	Memory Allocation Error ( by Target )
[ Content ]	In the debugger, since a memory was insufficient, processing has not been performed.
[ Handling ]	Please extend the memory mounted in the personal computer or end the other applications which are currently activated.

E4807M	System Error.
[ Content ]	Return of a debugger became impossible when accessing the target.
[ Handling ]	A debugger may once be ended.

E4808M	Not Support Function.
[ Content ]	The specified function is not supported.
[ Handling ]	A part of function is not supported by this analyzer.

E4809M	Parameter Error.
[ Content ]	parameter is inaccurate.
[ Handling ]	Please reinput the right parameter.

E4810M	Invalid ID.
[ Content ]	The specified ID is invalid.
[ Handling ]	Please re-perform after inputting the right ID.

E4811M	Memory Allocation Error.
[ Content ]	Since the memories for program execution were insufficient, it has not performed.
[ Handling ]	Please extend the memory mounted in the personal computer or end the other applications which are activated currently.

E4812M	ID cannot be Assigned.
[ Content ]	Since ID was insufficient, ID was not able to be assigned.
[ Handling ]	It generates, when the restriction 255 of OS break setup at the time of simulator use is exceeded. An unnecessary break should be deleted and please reset up.

E4815M	Already Carried Out.
[ Content ]	Already issued the system call is shown in system call issue.
[ Handling ]	Please do not issue the next system call until the current system call in processing ends.

E4816M	Warning. ( Re Execution is Possible )
[ Content ]	The error occurred at the time of functional execution of $\mu$ T-REALOS Analyzer.
[ Handling ]	Although re-execution is possible, it outputs, when a phenomenon cannot be specified. When this error occurs frequently, please contact our sales representatives.



## ■ Warning Messages

	Warning: Task ID = x Invalid arrange Stack
[ Content ]	The stack of the described ID is not allocated in the 8 bytes boundary.
[ Handling ]	It is possible that the $\mu$ T-REALOS is not operating normally because the stack was not correctly allocated in the boundary. Allocate the stack of the described ID in the 8 bytes boundary.

	Message: Clear Module's Flags. Are You OK?
[ Content ]	The task analysis module flag (variable) should be initialized.
[ Handling ]	Checks when initializing the task analysis module flag (variable) such as when resetting. When doing so, it clears the logs currently being acquired.

## ■ Fatal Errors (F95xxM)

F9503M	File Not Found. Installation may not have been performed normally.
[ Content ]	Cannot execute the function because the $\mu$ T-REALOS Analyzer installation information is illegal.
[ Handling ]	<p>The followings are possible.</p> <ol style="list-style-type: none"> <li>1. <math>\mu</math>T-REALOS Analyzer installation information is corrupt.</li> <li>2. It was not installed normally.</li> <li>3. The file was moved from the installation directory.</li> </ol> <p>In the case of 3, move back the file that was moved. For others, reinstall the program.</p>

F9504M	RIM Not Found. Installation may not have been performed normally.
[ Content ]	RTOS I/F Module (RIM) for $\mu$ T-REALOS were not found. $\mu$ T-REALOS Analyzer cannot Start.
[ Handling ]	<p>The followings are possible.</p> <ol style="list-style-type: none"> <li>1. <math>\mu</math>T-REALOS Analyzer installation information is corrupt.</li> <li>2. It was not installed normally.</li> <li>3. The file was moved from the installation directory.</li> </ol> <p>In the case of 3, move back the file that was moved. For others, reinstall the program.</p>

F9505M	Output File Not Opened
[ Content ]	The error occurred with the memory check before information getting.
[ Handling ]	<p>There are the following possibilities.</p> <ol style="list-style-type: none"> <li>1. Be in the state where a memory cannot be read.</li> <li>2. ICE or target is running away.</li> </ol> <p>Please Terminate <math>\mu</math>T-REALOS Analyzer and Re-Start debugger.</p>

F9506M	Target or Debugger Irregular Action. Please Check your environment.
[ Content ]	Failed in creation or the beginning of an output file.
[ Handling ]	File is respecified or specification folder or File is changed into the state in which the writing is possible.

## APPENDIX D Restrictions of Issuing System Call

---

This section explains the restrictions of issuing System Call.

---

### ■ System Call Issue Function Specifications and Restrictions

Because the system call issue (API issue hereafter) function is implemented by the Debugger call command, restrictions on the Debugger call command and those on the  $\mu$ T-REALOS Analyzer API issue function are the same. The following lists the restrictions:

- While the call command is being executed, it is not possible to execute all steps even though the Debugger is stopped with the abort command.
- While the call command is being executed, it is not possible to operate the call command in multiplex even though the Debugger is stopped with the abort command.
- This function cannot issue system calls that cannot be issued from non-task context.
- The address of the system call to be used must be physically solved beforehand. (Only the system calls that are linked through system configuration can be issued.)

### ■ Notes on Using the System Call Issue Function

After the specified function is issued, processing of the call command is completed when control is returned to the current PC.

When a  $\mu$ T-REALOS system call is issued, tasks may be switched in each system call. If tasks are switched, system call processing is completed. However, because control is not returned to the current PC by task switching, the SVC issue function of  $\mu$ T-REALOS is not completed.

#### **Example:**

The above restrictions apply when the system call function of  $\mu$ T-REALOS is used and a task with a higher priority than the current task is started.

## APPENDIX E Cautions when Using Stack Use Analysis

---

This section explains the points to note when using stack use analysis.

---

### ■ Restrictions when a Stack Overflow Occurs

Stack use analysis monitors data changes in the stack area allocated to each task.

Accordingly, the stack size usage is not shown correctly if an overflow occurs in a task's stack.

### ■ Stack Usage and Displayed Values

Movement of the stack pointer and the use of the memory addresses indicated by the stack pointer do not always match.

The stack pointer may be moved to allocate space, but this space may not necessarily be used.

Because stack use analysis does not monitor movement of the stack pointer, it only displays the amount of stack actually used, in case that the stack pointer is not used but only moves.

## APPENDIX F Restrictions with Debugger Type

This section explains restrictions with debugger type.

### ■ Functional Restrictions for Debugger Type

The table below lists available functions for each debugger type.

**Table F-1 Debugger Type and Available Functions List**

		Simulator	Emulator		Monitor Debugger
			DSU3	DSU4	
Object	Object View	○	○	○	○
Break	Execute Break	○	○	○	○
	Access Break	○	×	○	○
	Dispatch Break (ENTER/LEAVE)	○ *1	○ *1	○ *1	○ *1
	System Call break (ENTER/LEAVE)	○ *1	○ *1	○ *1	○ *1
Log	Dispatch Log	○	×	○	×
	Module use Log	○ *2	○ *2	○ *2	○ *2
	Monitoring	○	×	○	×
	Statistics Log	○ *3	○ *3	○ *3	○ *3
Task Context Watch	Task Context Watch	○	○	○	○
Stack Information	Stack Information	○	○	○	○

○ : Enabled

× : Disabled

\*1 : The "Dispatch Break" and the "System Call break" are available if setting task analysis module type1 to type3, and type5.

\*2 : The "module use log" is available if setting task analysis module type1 to type3.

\*3 : The "statistics log" is available if setting task analysis module type5.



# INDEX

---

The index follows on the next page.  
This is listed in alphabetic order.

---

# Index

<b>A</b>	
Activate	
To Activate.....	8
Add	
How to Set ([Add] Button) .....	40
Adding Button	
Adding,Changing,Deleting Buttons (Setting Window) .....	37
Address Input Form	
ID Pass Count, Address Input Form (Setting Window) .....	38
Advanced Setup	
Advanced Setup (Customize Setup) .....	49
Analyzer	
Features of the $\mu$ T-REALOS Analyzer .....	6
$\mu$ T-REALOS Analyzer Functions .....	6
Argument Input Box	
Argument Input Box (Argument Setting Window) .....	73
Argument Setting	
System Call Specifications Display Field (Argument Setting Window) .....	73
Argument Setting Window	
[Issue] Button (Argument Setting Window) .....	73
Argument Input Box (Argument Setting Window) .....	73
<b>B</b>	
Break	
Break Occurs .....	40
Functions of Each Pane when Object Break Setting Function is Selected.....	37
How Breaks are Implemented.....	39
Opening the OS Break Setting .....	37
Reflecting Finished Breaks in the Setting Window .....	41
Resource Breaks.....	40
Setting OS Breaks .....	12
Target Task Break Types .....	39
Types of Breaks .....	38
Types of OS Break .....	12
Buffer Full	
Buffer Full Settings .....	48
Buffer Size	
Changing the Buffer Size .....	87
Customization File for Specifying the Buffer Size and Number of Tasks .....	85
Button	
[Issue] Button (Argument Setting Window) .....	73
[Setting] Buttons (Log Setting Selection Window).....	46
[Update] Button .....	76
Adding,Changing,Deleting Buttons (Setting Window).....	37
How to Set ([Add] Button).....	40
<b>C</b>	
C Language	
C Language Interface of the Hook for a Comment Event .....	86
Call	
Call Command.....	70
Mouse Operations with the System Call Icons .....	58
Notes on using the System Call Issue Function .....	97
System Call Issue Function Specifications and Restrictions .....	97
System Calls.....	39
System Calls that can be Issued .....	70
System Calls that is not Linked .....	71
Change	
Change .....	41
Changing Button	
Adding,Changing,Deleting Buttons (Setting Window).....	37
Clearing	
Clearing Settings.....	65
Column	
Changing Row Order of the List Format of Each Object (Column Drag-and-Drop) .....	34
Column Drag-and-Drop	
Changing Row Order of the List Format of Each Object (Column Drag-and-Drop) .....	34
Comment Event	
C Language Interface of the Hook for a Comment Event .....	86
Comment Event .....	86
How to Describe Comment Events .....	86
Configuration Settings Dialog Box	
Configuration Settings Dialog Box .....	63
Opening the Configuration Settings Dialog Box .....	63
Customization File	
Customization File for Specifying the Buffer Size and Number of Tasks .....	85
Customize	
File for Timer Customize.....	85



Customize Setup		
Advanced Setup (Customize Setup) .....	49	
<b>D</b>		
Data Display Field		
Data Display Field (Display Window).....	46	
Deadline		
Configuration Settings - Deadline Settings .....	44	
Deadline Settings Dialog Box		
Opening the Deadline Settings Dialog Box .....	64	
Operation of the Deadline Settings Dialog Box		
.....	65	
Debug Type		
Function for Each Debug Type .....	15	
Setting the Debug Type .....	15	
Debugger		
Functional Restrictions for Debugger Type.....	99	
Precautions on the use of an Emulator Debugger		
.....	3	
Precautions on the use of an Monitor Debugger		
.....	4	
Delete		
Delete .....	41	
Deleting Button		
Adding,Changing,Deleting Buttons		
(Setting Window).....	37	
Development Environment		
Precautions for the Development Environment .....	3	
Dialog Box		
Configuration Settings Dialog Box .....	63	
Opening the Configuration Settings Dialog Box		
.....	63	
Opening the Deadline Settings Dialog Box .....	64	
Opening the Monitoring Settings Dialog Box .....	63	
Operation of the Deadline Settings Dialog Box		
.....	65	
Operation of the Monitoring Time Settings Dialog		
Box.....	64	
Display Field		
System Call Specifications Display Field		
(Argument Setting Window) .....	73	
Display Icon		
Log Data Display Icons and Parameter List .....	54	
Display Window		
Data Display Field (Display Window).....	46	
Each Button of Display Window .....	46	
Line and Icon of Transition Diagram Display		
Window .....	56	
Setting Log List (Display Window) .....	46	
<b>E</b>		
Emulator Debugger		
Precautions on the use of an Emulator		
Debugger .....	3	
Error		
Fatal Errors (F95xxM) .....	96	
Error Message		
Error Messages (E48xxM) .....	93	
Event		
C Language Interface of the Hook for a Comment		
Event.....	86	
Comment Event.....	86	
How to Describe Comment Events .....	86	
Recommended Combination of the Event .....	49	
Selectable Events.....	47, 48, 50	
<b>F</b>		
Fatal Error		
Fatal Errors (F95xxM) .....	96	
Field		
Data Display Field (Display Window) .....	46	
File Configuration		
File Configuration .....	5	
Finished Breaks		
Reflecting Finished Breaks in the Setting Window		
.....	41	
Function		
Function of Each Type .....	85	
<b>G</b>		
Graph		
Graph .....	77	
Statistics Graph Display .....	44	
<b>H</b>		
Header Files		
Header Files Needed for Include .....	86	
<b>I</b>		
ICE		
Measurement Results when Acquiring Information		
with ICE. ....	90	
Icon		
Line and Icon of Transition Diagram Display		
Window .....	56	
Log Data Display Icons and Parameter List.....	54	
Mouse Operations with the System Call Icons .....	58	
ID Pass Count		
ID Pass Count, Address Input Form (Setting Window)		
.....	38	
Installation		
Checking the Installation .....	7	
Interval Time		
Configuration Settings - Interval Time Settings for		
Monitoring .....	44	

## INDEX

Issue	
[Issue] Button (Argument Setting Window) .....	73
How to Issue .....	74
Issuing	
Issuing Procedures.....	17
Issuing Results .....	17
<b>L</b>	
Library	
Adding a Library .....	7
Line	
Line and Icon of Transition Diagram Display Window .....	56
List Format	
Changing Row Order of the List Format of Each Object (Column Drag-and-Drop) .....	34
Log	
[Setting] Buttons (Log Setting Selection Window) .....	46
Features of the Statistics Log .....	50
Functions of Each Pane when Log Function is Selected.....	45
Getting Logs .....	51
Log Acquisition .....	15
Log Data Display .....	52
Log Data Display Icons and Parameter List .....	54
Log Setting Selection List (Setting Selection Window).....	45
Measurement of the Statistics Log .....	50
Opening the Log Settings .....	45
Output of Log File.....	68
Setting Log List (Display Window).....	46
Setting the Log.....	14
Setup to be Done in the Log Setting Selection Pane .....	48
Log Data	
Log Data Display .....	52
Log Data Display Icons and Parameter List .....	54
Log File	
Output of a Log File .....	44
Log Setting Selection	
[Setting] Buttons (Log Setting Selection Window) .....	46
Setup to be Done in the Log Setting Selection Pane .....	48
Log Setting Selection List	
Log Setting Selection List (Setting Selection Window).....	45
<b>M</b>	
Measurement	
Measurement Results when Acquiring Information with ICE. ....	90
Message	
Error Messages (E48xxM) .....	93
Warning Messages .....	95
Monitor Debugger	
Precautions on the use of an Monitor Debugger .....	4
Monitoring	
Configuration Settings - Interval Time Settings for Monitoring .....	44
Monitoring Function .....	44
Operation of Monitoring (Tool Bar).....	62
Monitoring Settings Dialog Box	
Opening the Monitoring Settings Dialog Box .....	63
Monitoring Time Settings Dialog Box	
Operation of the Monitoring Time Settings Dialog Box .....	64
Monitoring Window	
Monitoring Window .....	61
Open the Monitoring Window.....	61
Mouse	
Mouse Operations with the System Call Icons .....	58
μT-REALOS Analyzer	
Features of the μT-REALOS Analyzer .....	6
μT-REALOS Analyzer Functions .....	6
<b>O</b>	
Object	
Changing Row Order of the List Format of Each Object (Column Drag-and-Drop) .....	34
Contents of Each Objects Display .....	27
Displaying Object Information .....	25
Functions of Each Pane when Object Break Setting Function is Selected .....	37
Functions of Each Pane when Object Viewer is Selected .....	24
Object Index Window .....	55
Object View .....	10
Opening the Object Viewer.....	24
Procedures to Display Object Information .....	26
Selecting an Object .....	24
Selecting Object IDs and [View] Button.....	25
Standard Functions of the Object Window.....	11
Object IDs	
Selecting Object IDs and [View] Button.....	25
Object Viewer	
Functions of Each Pane when Object Viewer is Selected .....	24
Opening the Object Viewer.....	24
OS	
Opening the OS Break Setting.....	37
Setting OS Breaks .....	12
Types of OS Break .....	12

Output Method	
Selection of Output Method .....	68
Overflow	
Restrictions when a Stack Overflow Occurs .....	98
<b>P</b>	
Parameter	
Log Data Display Icons and Parameter List .....	54
Pointer Type	
Pointer Type Arguments .....	71
Precaution	
Precautions for the Development Environment .....	3
Precautions for Use .....	70
Precautions on the use of an Emulator Debugger .....	3
Precautions on the use of an Monitor Debugger .....	4
<b>R</b>	
RAnalyzer	
Names of the Parts of the RAnalyzer Window .....	22
Opening the RAnalyzer Window .....	22
Realtime Statistics	
Realtime Statistics .....	66
Realtime Statistics Window	
Data Displayed in the Realtime Statistics Window .....	67
Opening the Realtime Statistics Window .....	66
Operation of the Realtime Statistics Window (Tool Bar) .....	67
Recommended Combination	
Recommended Combination of the Event .....	49
Resource	
Resource Breaks .....	40
Restriction	
Functional Restrictions for Debugger Type .....	99
Result	
Setting Results .....	40
Row	
Changing Row Order of the List Format of Each Object (Column Drag-and-Drop) .....	34
Saving Changed Row Orders (Right-click to Save Row Order) .....	34
Row Order	
Changing Row Order of the List Format of Each Object (Column Drag-and-Drop) .....	34
Saving Changed Row Orders (Right-click to Save Row Order) .....	34
<b>S</b>	
Sampling Method	
Sampling Method .....	43
Set	
How to Set .....	47, 49, 50
Setting	
[Setting] Buttons (Log Setting Selection Window) .....	46
Setting Results .....	40
Setup	
Advanced Setup (Customize Setup) .....	49
Source Jump	
Source Jump Function (Right-click to Source Jump) .....	35
Stack	
Contents of Stack Information Display .....	18
Opening the Stack Information Window .....	75
Restrictions when a Stack Overflow Occurs .....	98
Stack Information .....	18
Stack Usage Analysis .....	78
Stack Usage and Displayed Values .....	98
Standard Functions	
Standard Functions of the Object Window .....	11
Statistics Graph	
Referencing Task Transition Diagram and Statistics Graphs .....	16
Statistics Graph Display .....	44
Statistics Log	
Features of the Statistics Log .....	50
Measurement of the Statistics Log .....	50
Statistics Window	
Data Displayed in the Realtime Statistics Window .....	67
Display Items in the Statistics Window .....	60
Opening the Realtime Statistics Window .....	66
Opening the Statistics Window .....	59
Operation of the Realtime Statistics Window (Tool Bar) .....	67
Statistics Window .....	59
System Call	
List of System Calls that can be Issued .....	70
Mouse Operations with the System Call Icons .....	58
Notes on using the System Call Issue Function .....	97
System Call Issue .....	17
System Call Issue Function Specifications and Restrictions .....	97
System Call Selection List (System Call Selection Window) .....	72
System Call Specifications Display Field (Argument Setting Window) .....	73
System Calls .....	39
System Calls that is not Linked .....	71

## INDEX

System Call Issue	
Functions of Each Pane when System Call Issue	
Function is Selected.....	72
Opening the System Call Issue .....	72
<b>T</b>	
Target Task	
Target Task Break Types .....	39
Task	
Changing the Number of Tasks.....	87
Contents of Task Context Watch Display.....	19
Customization File for Specifying the Buffer Size and	
Number of Tasks .....	85
Opening the Task Context Watch.....	81
Referencing Task Transition Diagram and Statistics	
Graphs .....	16
Selecting the Task Analysis Module to Use .....	84
Target Task Break Types .....	39
Task Analysis Module .....	84
Task Context Watch .....	19
Task Transition Diagram Display.....	43
Watching the Specified Task Context.....	82
Task Analysis Module	
Selecting the Task Analysis Module to Use .....	84
Task Analysis Module .....	84
Task Context	
Watching the Specified Task Context.....	82
Task Context Watch	
Contents of Task Context Watch Display.....	19
Opening the Task Context Watch.....	81
Task Context Watch .....	19
Terms	
Explanation of Terms .....	2
Meaning of Terms .....	76
Time Stamps	
Adding Time Stamps .....	43
Timer Customize	
File for Timer Customize .....	85
Timer Resource	
Changing the Timer Resource .....	87
Tool Bar	
Operation of Monitoring (Tool Bar).....	62
Operation of the Realtime Statistics Window	
(Tool Bar) .....	67
Tool Bar.....	56
Transition Diagram	
Line and Icon of Transition Diagram Display Window	
.....	56
Referencing Task Transition Diagram and Statistics	
Graphs .....	16
Task Transition Diagram Display .....	43
<b>U</b>	
Update	
[Update] Button .....	76
Usage Analysis	
Stack Usage Analysis .....	78
<b>V</b>	
View Button	
Selecting Object IDs and [View] Button.....	25
Viewer	
Functions of Each Pane when Object Viewer is	
Selected .....	24
Opening the Object Viewer.....	24
<b>W</b>	
Warning Message	
Warning Messages .....	95
Watch	
Contents of Task Context Watch Display .....	19
Opening the Task Context Watch .....	81
Task Context Watch .....	19

CM81-00320-1E

---

**FUJITSU MICROELECTRONICS • CONTROLLER MANUAL**

FR Family

CONFORMING TO  $\mu$ T-Kernel SPECIFICATION

SO FUTURE™  $\mu$ T-REALOS/FR

ANALYZER GUIDE

---

June 2008 the first edition

Published **FUJITSU MICROELECTRONICS LIMITED**

Edited Business & Media Promotion Dept.

---

