

FR ファミリ

μT-Kernel 仕様準拠

SOFTUNE™ μT-REALOS/FR

アナライザガイド

FR ファミリ
μT-Kernel 仕様準拠
SOFTUNE™ μT-REALOS/FR
アナライザガイド

富士通マイクロエレクトロニクス株式会社

はじめに

■ 本書の目的と対象読者

本書は、SOFTUNE μ T-REALOS/FR (以降、 μ T-REALOS とよびます) のアプリケーションプログラムを開発する方を対象としており、 μ T-REALOS アナライザについて記述しています。 μ T-REALOS アナライザの操作方法について確認したいときに、適宜本書を参照してください。また、本書をご覧になる前に、「SOFTUNE μ T-REALOS/FR ユーザーズガイド」(以降、「ユーザーズガイド」とよびます)を一読されることをお勧めします。

μ T-REALOSは、32ビットRISCコントローラFRファミリで動作する μ T-Kernel仕様のリアルタイムOSです。

μ T-Kernel仕様は、T-Engineフォーラムが策定したオープンなリアルタイムOSの仕様です。 μ T-Kernelの仕様書は、T-Engineフォーラムのホームページ(<http://www.t-engine.org/>)から入手できます。 μ T-Kernelの著作権は、坂村健氏に属しています。 μ T-Kernel仕様の著作権は、T-Engineフォーラムに属しています。

FRは、FUJITSU RISC controllerの略で、富士通マイクロエレクトロニクス株式会社の製品です。

■ 商標

SOFTUNEは、富士通マイクロエレクトロニクス株式会社の商標です。

REALOSは、富士通マイクロエレクトロニクス株式会社の商標です。

TRONは、「The Real-time Operating system Nucleus」の略称です。

ITRONは、「Industrial TRON」の略称です。

μ ITRONは、「Micro Industrial TRON」の略称です。

T-Kernelおよび μ T-Kernelは、コンピュータの仕様に対する名称であり、特定の商品ないしは商品群を指すものではありません。

その他の記載されている社名および製品名などの固有名詞は、各社の商標または登録商標です。

■ 本書の全体構成

本書は、以下に示す 3 つの章と付録から構成されています。

第 1 章 概要

この章では、SOFTUNE μ T-REALOS アナライザを使用する上での注意事項や、機能概要について説明します。

第 2 章 機能説明

この章では μ T-REALOS アナライザの各機能とウィンドウについて説明します。

第 3 章 タスク解析モジュール

この章では、モジュール使用ログによるサンプリングで使用する「タスク解析モジュール」について説明します。

付録

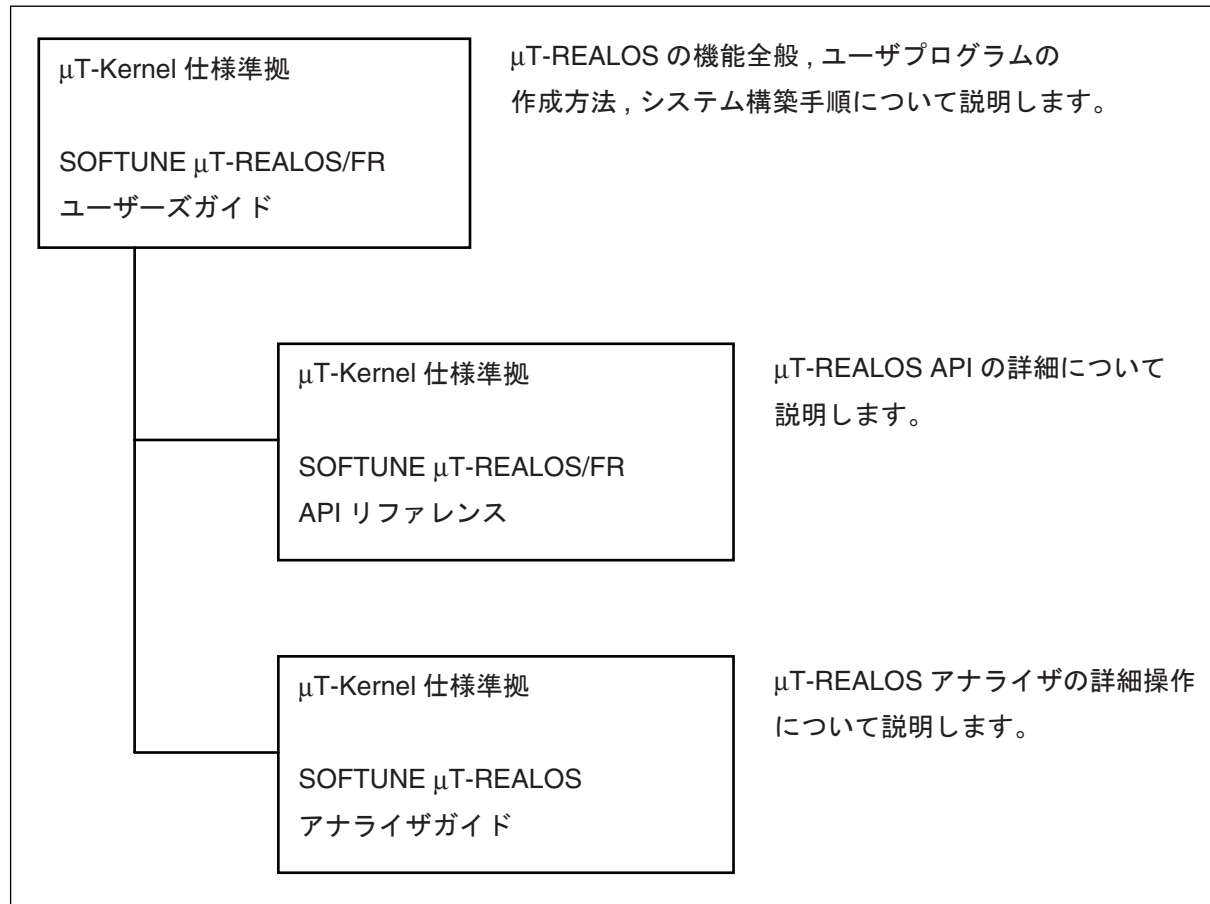
付録では、アナライザの処理時間やエラーメッセージ、制限事項を掲載します。

- 本資料の記載内容は、予告なしに変更することがありますので、ご用命の際は営業部門にご確認ください。
- 本資料に記載された動作概要や応用回路例は、半導体デバイスの標準的な動作や使い方を示したもので、実際に使用する機器での動作を保証するものではありません。したがって、これらを使用するにあたってはお客様の責任において機器の設計を行ってください。これらの使用に起因する損害などについては、当社はその責任を負いません。
- 本資料に記載された動作概要・回路図を含む技術情報は、当社もしくは第三者の特許権、著作権等の知的財産権やその他の権利の使用権または実施権の許諾を意味するものではありません。また、これらの使用について、第三者の知的財産権やその他の権利の実施ができることの保証を行うものではありません。したがって、これらの使用に起因する第三者の知的財産権やその他の権利の侵害について、当社はその責任を負いません。
- 本資料に記載された製品は、通常の産業用、一般事務用、パーソナル用、家庭用などの一般的用途に使用されることを意図して設計・製造されています。極めて高度な安全性が要求され、仮に当該安全性が確保されない場合、社会的に重大な影響を与えかつ直接生命・身体に対する重大な危険性を伴う用途（原子力施設における核反応制御、航空機自動飛行制御、航空交通管制、大量輸送システムにおける運行制御、生命維持のための医療機器、兵器システムにおけるミサイル発射制御をいう）、ならびに極めて高い信頼性が要求される用途（海底中継器、宇宙衛星をいう）に使用されるよう設計・製造されたものではありません。したがって、これらの用途にご使用をお考えのお客様は、必ず事前に営業部門までご相談ください。ご相談なく使用されたことにより発生した損害などについては、責任を負いかねますのでご了承ください。
- 半導体デバイスはある確率で故障が発生します。当社半導体デバイスが故障しても、結果的に人身事故、火災事故、社会的な損害を生じさせないように、お客様は、装置の冗長設計、延焼対策設計、過電流防止対策設計、誤動作防止設計などの安全設計をお願いします。
- 本資料に記載された製品を輸出または提供する場合は、外国為替及び外国貿易法および米国輸出管理関連法規等の規制をご確認の上、必要な手続きをおとりください。
- 本書に記載されている社名および製品名などの固有名詞は、各社の商標または登録商標です。

μT-REALOS のマニュアル体系

μT-REALOS のマニュアルは、以下の 3 分冊になっています。

μT-REALOS を初めてお使いになる方は、最初に『SOFTUNE μT-REALOS/FR ユーザーズガイド』をお読みください。



目次

第 1 章	概要	1
1.1	使用にあたっての注意事項	2
1.2	提供ファイル構成	5
1.3	特長	6
1.4	起動方法	7
1.5	機能	9
1.5.1	オブジェクト表示	10
1.5.2	OS ブレークの設定	12
1.5.3	ログ (タスク遷移図)	14
1.5.4	システムコール発行	17
1.5.5	スタック情報	18
1.5.6	タスクコンテキストウォッチ	19
第 2 章	機能説明	21
2.1	R アナライザウィンドウ	22
2.2	オブジェクト表示 (Object Viewer)	23
2.2.1	ウィンドウの説明 (オブジェクト表示)	24
2.2.2	オブジェクト情報を表示させる	26
2.2.3	各オブジェクトの表示内容	27
2.2.4	列順を変更する	34
2.2.5	ソースジャンプ機能	35
2.3	OS ブレーク	36
2.3.1	ウィンドウの説明 (OS ブレーク)	37
2.3.2	ブレークを設定する	41
2.3.3	削除と変更	42
2.4	ログ	43
2.4.1	ログ機能の仕様について	44
2.4.2	ウィンドウの説明 (ログ設定)	46
2.4.3	ログを設定する (ディスパッチログ)	48
2.4.4	ログを設定する (モジュール使用ログ)	49
2.4.5	ログを設定する (統計ログ)	51
2.4.6	ログの取得	52
2.4.7	ログデータの説明	53
2.4.8	遷移図ウィンドウの説明	56
2.4.9	統計ウィンドウの説明	60
2.4.10	モニタリングウィンドウの説明	62
2.4.11	環境設定	64
2.4.12	リアルタイム統計の説明	67
2.4.13	ログファイルの出力	69
2.5	システムコール発行	70
2.5.1	システムコール発行について	71
2.5.2	ウィンドウの説明 (システムコール発行)	73
2.5.3	システムコールを発行する	75
2.6	スタック情報	76
2.7	タスクコンテキストウォッチ	82

第 3 章	タスク解析モジュール	85
3.1	タスク解析モジュールの種類	86
3.2	コメントの使用方法	88
3.3	カスタマイズの方法	89
付録	91
付録 A	情報取得の速度について	92
付録 B	タスク解析モジュールのシンボル / セクション	93
付録 C	エラーメッセージ	95
付録 D	システムコール発行時の制限について	99
付録 E	スタック使用量解析時の注意事項について	100
付録 F	デバッガ種別による制限について	101
索引	103

第1章

概要

この章では、SOFTUNE μ T-REALOS アナライザを使用する上での注意事項や、機能概要について説明します。

- 1.1 使用にあたっての注意事項
- 1.2 提供ファイル構成
- 1.3 特長
- 1.4 起動方法
- 1.5 機能

1.1 使用にあたっての注意事項

SOFTUNE μ T-REALOS アナライザを使用する上での注意事項について記述しています。

■ 用語の説明

本書では以下の用語を以下のように定義して使用しています。

● オブジェクト

μ T-REALOS が管理するリソースです。

タスクをはじめ、タスク間の同期・通信を司るセマフォ、イベントフラグ、メールボックス、メッセージバッファ、ミューテックス、ランデブポートのほか、メモリプール、周期ハンドラ / 周期起動ハンドラ、アラームハンドラを μ T-REALOS オブジェクトとよんでいます。

● OS ブレーク

μ T-REALOS を意識したブレークです。

通常のデバッガでは、同一エントリに割り付けられたタスクにおいて、タスクを意識したブレークを設定することはできません。

例) タスク 1 とタスク 2 が以下のようなエントリに割り付けられている場合、

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

通常のデバッガではfuncA () の呼び出しアドレスにブレークを設定できますが、タスク 1 の場合だけブレークをさせることはできません。

OS ブレークでは、例に挙げた事象にブレークを設定できます。また、OS のイベント（システムコール発行時、ディスパッチ開始時など）といった、OS のふるまいにもブレークを設定できます。

● ログ

タスクの遷移状態や、発行されたシステムコールを確認できます。ディスパッチ以外のふるまいをロギングするためには、タスク解析モジュールとよんでいるプログラムをリンクする必要があります（実際はコンフィギュレータのデバッグタイプを変更するだけです）。type3 におけるログのタイムスタンプの精度は、システムクロックのカウンタオーダーになります。

● タスク解析モジュール

ログ機能を実現する際にリンクする OS のパッチモジュールです。

タスク解析モジュールは μ T-REALOS のイベントが発生するたびに呼び出され、そのイベントが選択されたものである場合、そのイベントの情報にタイムスタンプを付加し、お客様のメモリ上に確保した領域に格納していきます。実行停止時に格納されたデータを μ T-REALOS アナライザが吸い上げることで、タスク遷移図を表示します。なお、

前述したような処理が入るためオーバーヘッドがかかります。

- 統計ログ

タスクの実行時間の上限を設定（デッドライン設定）し、その上限を超えた回数、測定回数、平均実行時間、最長実行時間を測定します。

■ 開発環境における注意点

- SOFTUNE μ T-REALOS アナライザ（本製品）

SOFTUNE μ T-REALOS アナライザ（以降、 μ T-REALOS アナライザと記載します）は、SOFTUNE μ T-REALOS を使用したユーザプログラム（以下本書では、 μ T-REALOS を使用したユーザプログラムを単にユーザプログラムとよびます）を解析するツールです。

- SOFTUNE μ T-REALOS（必須ソフトウェア）

SOFTUNE μ T-REALOS（以降、 μ T-REALOS と記載します）は、富士通製の μ TRON 仕様に準拠した組込み向けリアルタイム OS のことです。本製品は、 μ T-REALOS 用の解析ツールであり、他 OS での使用はできません。

- 富士通統合開発環境 SOFTUNE Workbench（必須ソフトウェア）

μ T-REALOS アナライザは、SOFTUNE Workbench から起動して使用するツールです。単独では起動しません。

- デバッグ連携

μ T-REALOS アナライザは SOFTUNE Workbench のデバッグと連携して動作します。デバッグの種類は問いません。ただし、サードパーティ製のデバッグでは使用できません。

また動作するバージョンが定められています。

- SOFTUNE Workbench のデバッグ環境

SOFTUNE Workbench 上でデバッグする際には、SOFTUNE Workbench のデバッグ環境の設定で実行時は割り込み許可、ステップ実行時は割り込み禁止に設定してください。

■ エミュレータデバッグ使用時における注意点

- メモリアクセス時の注意

μ T-REALOS アナライザは、ターゲットファイルのロードおよびリセットが実行された直後に μ T-REALOS（スタック領域を含む）およびタスク解析モジュールのデータ領域にアクセスを行います。アクセスに失敗すると μ T-REALOS アナライザは正常に動作しません。

したがって、メモリに前述のデータを配置する場合は、ターゲットファイルのロード前にメモリに対しアクセス可能となる処理を記述したコマンドプロシージャファイルを実行させるなどしてメモリをアクセスできる状態にしておく必要があります。

- モニタリング機能を使用する場合

モニタリングは、エミュレータデバッグのデバッグ実行中にメモリをリードする機能を使用して実現しているため、以下の条件以外で使用了場合はエミュレータデバッグ側でエラーが発生します。

< モニタリング機能が使用できる条件 >

- 外部メモリに μ T-REALOS のデータ領域 (スタック領域を含む) を配置している場合
エミュレーションメモリ上に配置する必要があります。
- 内部メモリに μ T-REALOS のデータ領域 (スタック領域を含む) を配置している場合
内部メモリ領域をミラーする必要があります。

■ モニタデバッガ使用時の注意点

● モニタリング機能

連続実行中にメモリをリードする機能がないため、モニタリング機能は使用できません。

● オブジェクト表示機能

連続実行中にメモリをリードする機能がないため、オブジェクト表示を更新できません。

1.2 提供ファイル構成

μT-REALOS アナライザの提供ファイル構成について説明します。

■ ファイル構成

μT-REALOS アナライザの提供するファイルはインストール後、以下のように格納されています。

【インストールディレクトリ】

bin¥	uTrim.dll	RTOS I/F DLL
	SiRan.dll	GUI DLL
	SiRanEnu.dll	英語用 リソース DLL
utkernel¥911¥dbg¥		
	type1	タスク解析モジュール type1 ファイルセット
	type2	タスク解析モジュール type2 ファイルセット
	type3	タスク解析モジュール type3 ファイルセット
	type5	タスク解析モジュール type5 ファイルセット
	R_D_custom.h	ログタイプ [コメント] 用ヘッダーファイル

< 注意事項 >

タスク解析モジュールは、ログ機能のモジュール使用ログを実現するために必要なファイル群です。

1.3 特長

μT-REALOS アナライザの特長について説明します。

■ μT-REALOS アナライザの特長

μT-REALOS アナライザは、μT-REALOS を使用したアプリケーションのデバッグおよびチューニング時に使用するツールです。

μT-REALOS の管理するオブジェクトの状態を容易に確認できます。

また、タスクを意識したブレークの設定や、μT-REALOS のふるまいをロギングできます。

ロギングしたデータをタスク遷移図にして表示するほか、各タスクの実行時間を解析して、システムのリアルタイム性が確保されているか容易に判断できます。

■ μT-REALOS アナライザの機能

- オブジェクト一覧表示

μT-REALOS アナライザの「Object viewer」により、各オブジェクトの状態を一覧形式で表示します。

- OS ブレーク

μT-REALOS アナライザの「Setting」-「Break」により、OS ブレークを設定できます。

- ログ (タスク遷移図)

μT-REALOS アナライザの「Setting」-「Log」により、μT-REALOS のふるまいをロギングできます。

- システムコール発行

μT-REALOS アナライザの「Setting」-「System Call」により、システムコールをデバッグ停止中に発行できます。

- スタック情報

Workbench のメニュー [R アナライザ] - [スタック情報] により、現在のタスクのスタック情報を確認できます。

- タスクコンテキスト表示

Workbench のメニュー [R アナライザ] - [タスクコンテキストウォッチ] により、タスクのコンテキストを表示します。

1.4 起動方法

μT-REALOS アナライザを起動するには、デバッガを起動しターゲットをロード後、SOFTUNE Workbench の [R アナライザ] メニューから [R アナライザ] を選択してください。

■ インストールの確認

正常にインストールされた場合、SOFTUNE Workbench のメニューに [R アナライザ] のメニューが追加されます。

■ ライブラリの追加

μT-REALOS アナライザのディスパッチブレイク機能、モジュール使用ログ機能を使用するためにユーザプログラムに μT-REALOS アナライザで提供しているライブラリを追加します。

ディスパッチブレイク機能を使用する場合は 4 種類のライブラリのいずれかを追加してください。

ディスパッチログ機能を使用する場合は type1 ~ type3, type5 のいずれかを追加します。モジュール使用ログ機能については type1 ~ type3 のいずれかを追加します。トレースメモリを使用せずに長時間の測定とデッドライン測定 (統計ログ) を行うには type5 を追加してください。

表 1.4-1 タスク解析モジュールと使用可能な機能

タスク解析 モジュール	ディスパッチ ブレイク	ディスパッチ 取得	システムコール 取得	システムクロック 取得	リロードタイム 取得
type1				×	×
type2					×
type3					
type5			×	×	×

タスク解析 モジュール	割込み取得	統計ログ
type1		×
type2		×
type3		×
type5	×	

: 使用可

× : 使用不可

また、上記ライブラリは初期値付き変数を使用しています。そのためユーザプログラムの初期化処理に初期値付き変数の初期化が必要になります。初期値付き変数の初期化方法については SOFTUNE リンケージキットマニュアルの「5.9 ROM RAM 転送セクション」を参照してください。

< 注意事項 >

インストールする際には、一度 SOFTUNE Workbench を終了してください。

■ 起動方法

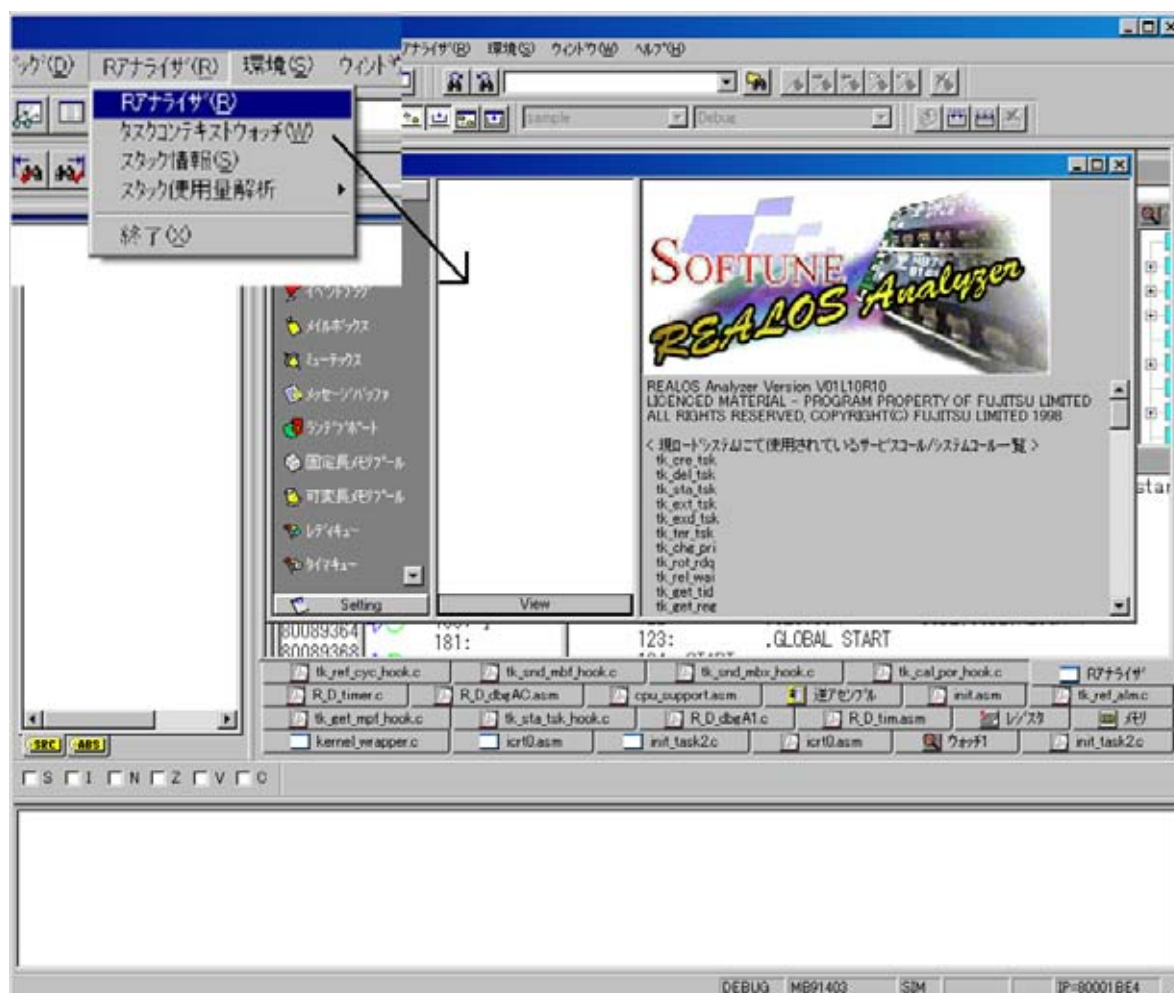
デバッガを起動し、ターゲットをロードし、最初に動作するタスクの先頭まで一度実行してください。

SOFTUNE Workbench のメニューから、[R アナライザ] メニューの [R アナライザ] を選択します。

< 確認 >

「図 1.4-1 起動画面」にあるウィンドウが開きましたか?開いていれば起動完了です。メニューがグレー表示になっている場合、Workbenchの版数に対応していない可能性があります。Workbench の版数を確認してください。インストール説明書に対応する Workbench の版数が記載されています。

図 1.4-1 起動画面



1.5 機能

μT-REALOS アナライザの各機能について説明します。

- 1.5.1 オブジェクト表示
- 1.5.2 OS ブレークの設定
- 1.5.3 ログ (タスク遷移図)
- 1.5.4 システムコール発行
- 1.5.5 スタック情報
- 1.5.6 タスクコンテキストウォッチ

1.5.1 オブジェクト表示

オブジェクト表示は、デバッガ停止時にアプリケーションで定義されている μ T-REALOS オブジェクトの現在の状態を表示する機能です。各オブジェクトのスイッチを押すことで表示します。

■ オブジェクト表示

オブジェクト表示は、デバッガ停止時に現在の各オブジェクトの状態を一覧で表示するウィンドウです。各オブジェクト表示は、リスト形式の情報一覧表示と、待ちタスクなどの存在を表示する階層表示の最大2画面による表示を行います。各製品のオブジェクト表示対応状況は、表 1.5-1 を参照してください。

表 1.5-1 オブジェクト表示の対応表

一覧	製品
	SOFTUNE μ T-REALOS
タスク一覧	
セマフォ一覧	
イベントフラグ一覧	
メールボックス一覧	
ミューテックス一覧	
メッセージバッファ一覧	
固定長メモリプール一覧	
可変長メモリプール一覧	
レディキュー	
タイマキュー	
周期 / 周期起動ハンドラー一覧	
アラームハンドラー一覧	
カーネル情報	
ランデブポート	
サブシステム	

: 表示可

× : 表示不可

■ オブジェクトウィンドウの標準機能

各オブジェクトウィンドウには、標準で以下の機能があります。

- 列順保存
- エントリが解決する場合のソースジャンプ機能

図 1.5-1 オブジェクト表示 (タスク)

ID	エントリ	状態/待ち要因	優先度	スタック領域先
1	kn1_init_task	TTS_RUN	138	H'800873C0
2	sem_tsk02	TTS_WAI TTW_SEM	2	H'80089BF4
3	sem_tsk03	TTS_WAI TTW_SEM	3	H'8008A0FC
4	flg_tsk04	TTS_WAI TTW_FLG	4	H'8008A654
5	flg_tsk05	TTS_WAI TTW_FLG	5	H'8008AB5C
6	mbx_tsk06	TTS_WAI TTW_MBX	6	H'8008B0B4
7	mbx_tsk07	TTS_WAI TTW_MBX	7	H'8008B5BC
8	mtx_tsk08	TTS_WAI TTW_MTX	8	H'8008BB14
9	mtx_tsk09	TTS_DMT	9	H'8008C01C
10	mbf_tsk10	TTS_DMT	10	H'8008C574
11	mbf_tsk11	TTS_DMT	11	H'8008CA7C
12	por_tsk12	TTS_DMT	12	H'8008CF84
13	por_tsk13	TTS_DMT	13	H'8008D48C
14	mpf_tsk14	TTS_DMT	14	H'8008D9E4
15	mpf_tsk15	TTS_DMT	15	H'8008DEEC
16	mpl_tsk16	TTS_DMT	16	H'8008E444
17	mpl_tsk17	TTS_DMT	17	H'8008E94C
18				

< 注意事項 >

PC がカーネル内にある場合、カーネル処理の途中であるため μ T-REALOS アナライザの表示する内容は保証されません。

1.5.2 OS ブレークの設定

OS ブレークは、 μ T-REALOS を意識したブレークを実現する機能です。タスクを意識したブレークのほか、 μ T-REALOS のふるまいにブレークを設定できます。

[Setting] グループ内の [Break] で OS ブレークに関する設定を行ってください。

■ OS ブレークの設定

OS ブレークは、[Setting] グループの [Break] で、設定の実行や、現在の設定状態を知ることができます。

ツリー形式のウィンドウには、現在設定されているブレークを表示します。

■ OS ブレークの種類

OS ブレークには、以下の 6 種類のブレークを設定できます。

● 実行ブレーク

実行ブレークは、指定したアドレスを実行する場合にブレークさせる機能に、タスク ID を特定して設定できます。

● アクセスブレーク

アクセスブレークは、指定するアドレスに書込みが発生した場合にブレークさせる機能に、タスク ID を特定して設定できます（読み出し時は動作しません）。

● ディスパッチャ (ENTER) ブレーク

ディスパッチャ (ENTER) ブレークは、ディスパッチャが動作を開始した段階にブレークを設定できます。この場合、指定するタスク ID は、実行権を放棄するタスクとなります。

● ディスパッチャ (LEAVE) ブレーク

ディスパッチャ (LEAVE) ブレークは、ディスパッチャが動作を完了した段階にブレークを設定できます。この場合、指定するタスク ID は、実行権を取得するタスクとなります。

● システムコール (ENTER) ブレーク

システムコール (ENTER) ブレークは、システムコール発行が開始した段階にブレークを設定できます。この場合、指定するタスク ID は、システムコールを発行したタスクとなります。また、発行を開始したシステムコールを指定できます。

● システムコール (LEAVE) ブレーク

システムコール (LEAVE) ブレークは、システムコール発行が完了した段階にブレークを設定できます。この場合、指定するタスク ID は、システムコールを発行したタスクとなります。また、発行を完了したシステムコールを指定できます。発行したタスクが待ち状態になるようなシステムコールを指定した場合、待ちから復帰するタイミングでブレークを設定します。

図 1.5-2 OS ブレークの設定



< 注意事項 >

μT-REALOS アナライザの OS ブレーク機能で設定されたブレークポイントをユーザが SOFTUNE Workbench のブレーク設定機能を用いて解除した場合、μT-REALOS アナライザで設定した OS ブレーク機能は保証されません。

1.5.3 ログ (タスク遷移図)

ログ機能は、 μ T-REALOS に依存した実行履歴を取得する機能です。タスクのディスパッチ状況や、システムコールの発行タイミング、ハンドラの起動時といった μ T-REALOS のふるまいに関する履歴を取得できます。[Setting] グループ内の [Log] でログに関する設定や取得した情報を確認してください。

■ ログの設定

ログの設定は、[Setting] グループの [Log] で行います。設定は以下の手順で行います。

設定するログを選択します。

選択肢は 3 つです。

- ディスパッチログ
ブレークを利用し、実行タスクの切替わりだけに注目したログです。イベントの種類が少ない代わりに解析モジュールなど、面倒な操作を必要としない機能です。
- モジュール使用ログ
タスク解析モジュールを利用し、様々な OS イベントをお客様が選択できるログです。選択できるイベントは増えますが、ユーザメモリをバッファリング領域にしているため、RAM サイズを確保する必要があります。また、 μ T-REALOS コンフィギュレータの「デバッグ」定義において type1 ~ type3 を選択する必要があります。「デバッグ」定義の詳細は「 μ T-REALOS/FR ユーザーズガイド」を参照してください。タスク解析モジュールの type5 (統計ログ用モジュール) を利用している場合は使用できません。
- 統計ログ
 μ T-REALOS コンフィギュレータの「デバッグ」定義においてタスク解析モジュールの type5 (統計ログ用モジュール) を利用している場合に限り使用できます。「デバッグ」定義の詳細は「 μ T-REALOS/FR ユーザーズガイド」を参照してください。各タスクの実行時間、デッドラインを測定するためのログです。タスク解析モジュール内でバッファリングを行うためユーザメモリに RAM サイズを確保する必要はありません。

取得するイベントを選択します。

- ディスパッチログ
現在、「実行タスク切替時」だけ選択可能です。
- モジュール使用ログ
以下のイベントのうち必須となる「ディスパッチャ LEAVE」を含めて、選択できます (ディスパッチャ LEAVE を選択しなかった場合、タスク遷移図を表示しません)。
 - 割り込みハンドラ (呼出し時, 処理終了時)
 - タイマハンドラ (呼出し時, 処理終了時)
 - ディスパッチャ (呼出し時, 処理終了時)
 - システムコール (呼出し時, 復帰値格納時)
 - お客様の指定したイベント (コメントとよびます)

選択が完了した時点で [Setting] ボタンを押してください。

以上でログの設定は完了です。正常に設定されたイベントはウィンドウの「設定ログタイプ」を表示します。

■ ログの取得

システムを実行し、適当な場所でブレークさせます。その時点で [取得] ボタンを押すとデータの収集を開始します。

■ デバッグタイプの設定

デバッグタイプの設定では、 μ T-REALOS アナライザで使用するデバッグ用のフックの切換えを行います。

デバッグ用フックの切換えはコンフィギュレータプロジェクトウィンドウのツリーアイテム "デバッグ" の子アイテムをダブルクリックして選択します。

デバッグ用フックは次のパスにあるサブディレクトリ名をデバッグタイプ、サブディレクトリ内に含まれているファイルをデバッグ用フックのファイルとして使用します。

デバッグ用フックが格納されているディレクトリ

μ T-REALOS をインストールしたディレクトリ ¥utkernel¥911¥dbg

デバッグタイプが選択されているときはコンフィギュレータプロジェクトウィンドウのツリーアイテム "デバッグ" に "デバッグ (デバッグタイプ名)" と表示します。

また、デバッグ用フックを設定しないで、メイクおよびビルドを実行した場合は、
"E4305L: 外部シンボルが定義されていません" (`_kernel_D_xxx_x/_kernel_D_xxx_x`) というリンクエラーを表示します。

■ デバッグタイプごとの機能

選択するタイプによって、サンプリングする際に付加するタイムスタンプの精度が異なります。精度の高いタイムスタンプを使用する場合、詳細なデータを取得できますが、その分オーバーヘッドが大きくなります。以下、タイプごとのタイムスタンプの精度を示します。

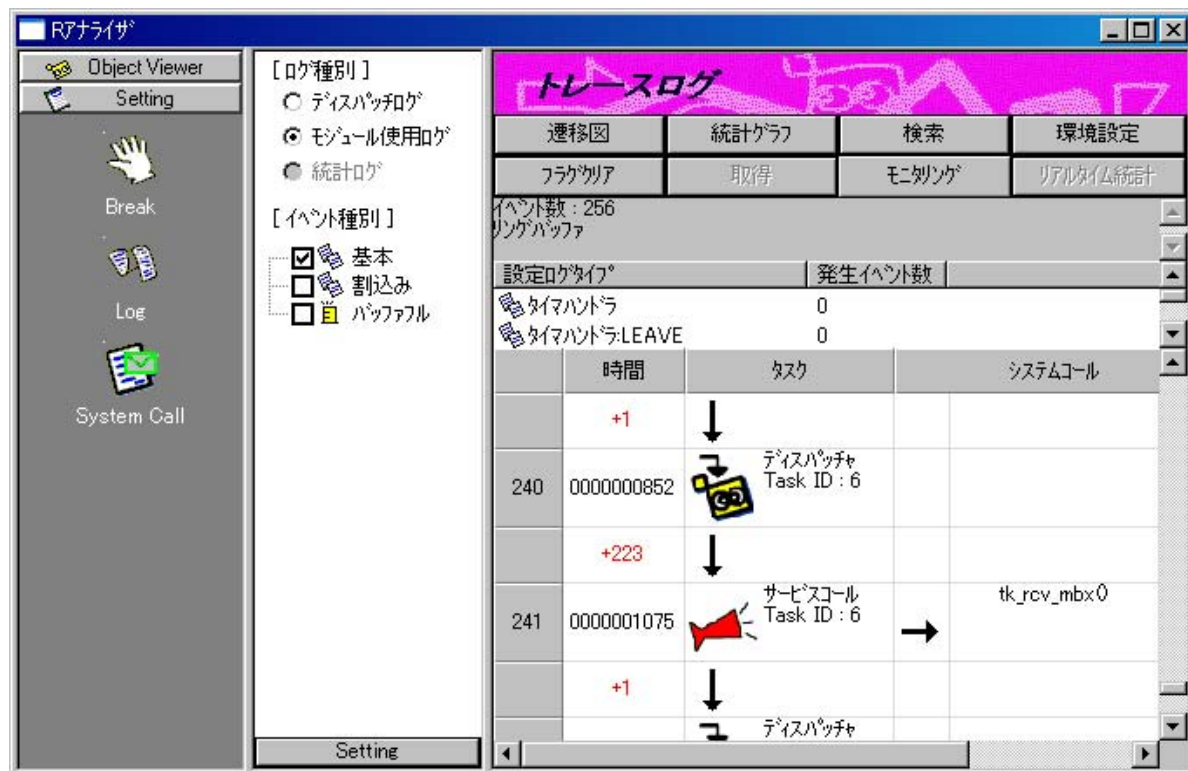
表 1.5-2 タイプ別タイムスタンプ精度

タイプ名	ライブラリ名	説明
type1	R_D_dbgA1.lib	タイムスタンプを付加しないタイプです。
type2	R_D_dbgA2.lib	システムクロック時間を付加するタイプです。
type3	R_D_dbgA3.lib	リソースタイマのカウント値を付加するタイプです。
type5	R_D_dbgA5.lib	統計ログを取得するタイプです。

■ 遷移図および統計グラフの参照

ログが取得されている状態で、[遷移図] や [統計グラフ] ボタンを押すことで、RTOS ログ (遷移図) ウィンドウと RTOS ログ (統計) ウィンドウが開きます。

図 1.5-3 ログ設定および取得情報の表示



1.5.4 システムコール発行

システムコール発行機能は、ブレーク中にデバッガから任意のシステムコール発行を行う機能です。ブレーク中にシステムコールを発行することにより、タスクの状態を変更したり、優先度の変更などの操作が可能です。[Setting] グループ内の[System Call] で操作できます。

■ システムコール発行

システムコール発行は、[Setting] グループの[System Call] で行います。

■ 発行手順

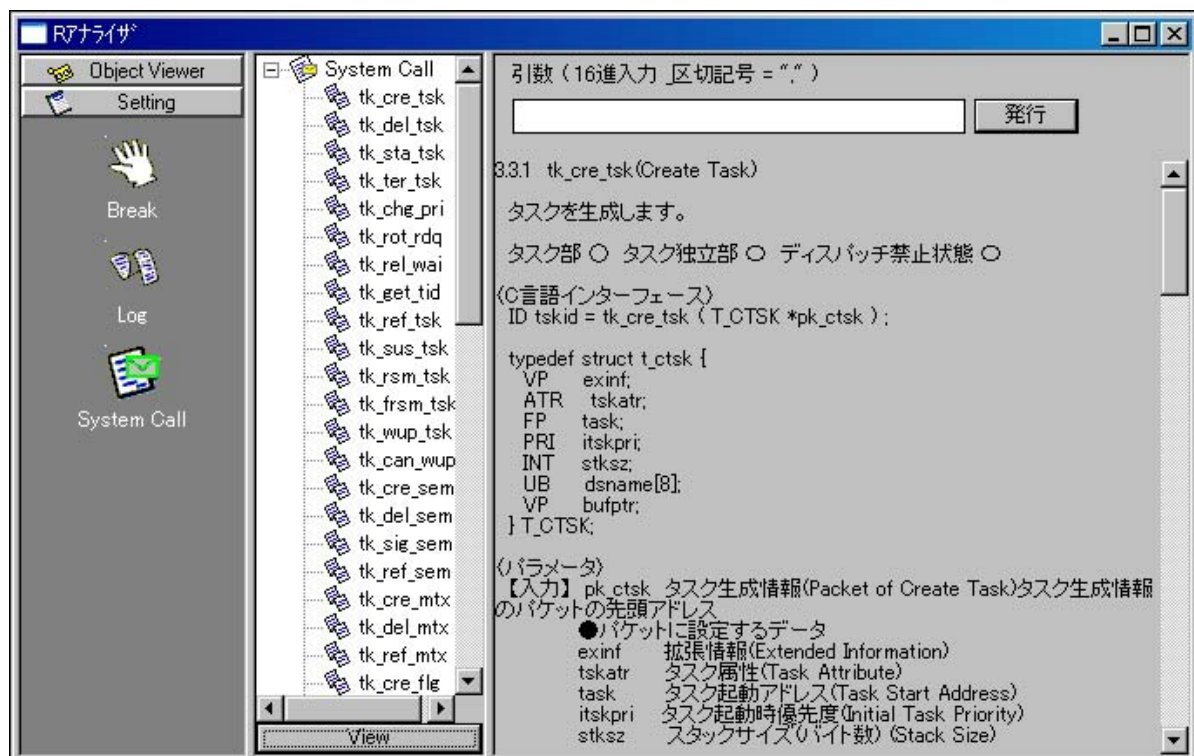
使用しているシステムで発行可能なシステムコールを表示します。発行したいシステムコールを選択して[View] ボタンを押してください。

引数を入力し、[発行] ボタンを押すとシステムコール発行が完了します。

■ 発行の結果

発行後、エラーコードをダイアログで表示します。

図 1.5-4 システムコール発行画面



1.5.5 スタック情報

スタック情報は、タスクのスタックの現在の使用状況を確認する機能です。スタックの割当てが適切であるかどうかお客様が判断するための材料を提供いたします。Workbench の [R アナライザ] メニューの [スタック情報] でウィンドウを開きます。

■ スタック情報

スタック情報は Workbench の [R アナライザ] メニュー下にある [スタック情報] でウィンドウを開きます。最初に関く場合は自動でデータの収集を行います。それ以降は、[更新] ボタンを押すことで情報を再取得します。

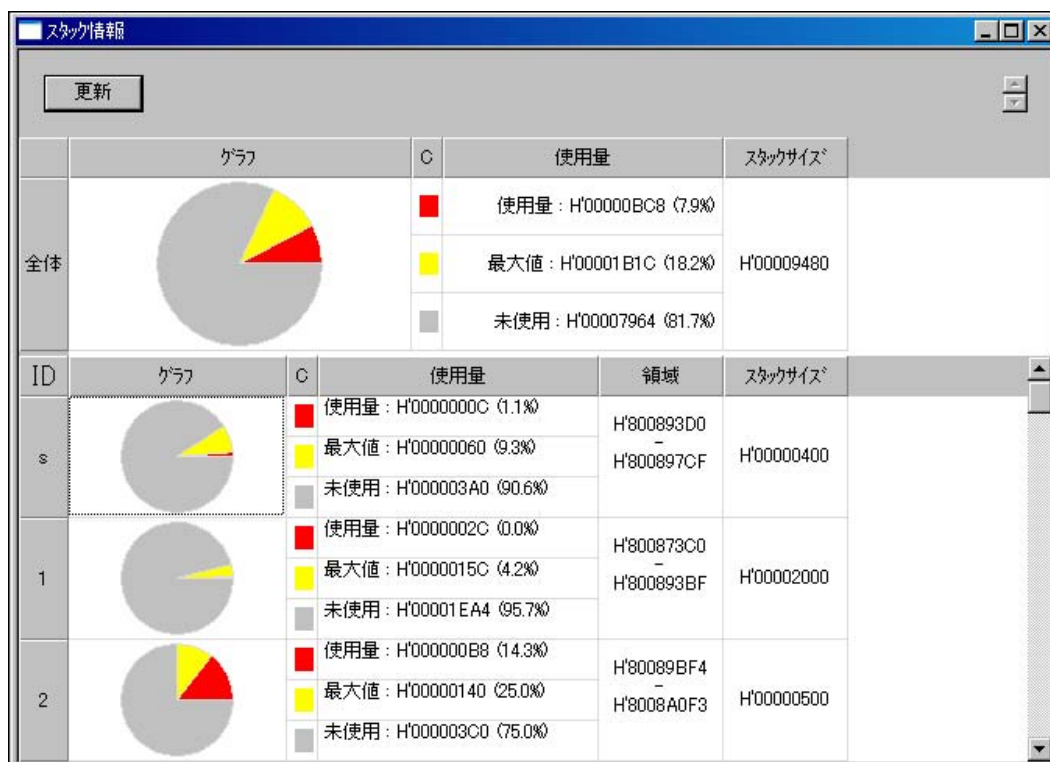
■ スタック情報の表示内容

上段のグリッドは、システム全体の使用量・最大値・未使用量の合計を示します。

下段のグリッドは、システムスタック、アイドルタスク、各タスクごとの情報です。生成されているタスクだけ表示します。

以下にスタック情報ウィンドウの表示例を示します。

図 1.5-5 スタック情報ウィンドウ



< 注意事項 >

スタック情報の取得時間は、スタックサイズに依存します。

1.5.6 タスクコンテキストウォッチ

タスクコンテキストウォッチは、タスクのコンテキストを表示する機能です。Workbench の [R アナライザ] メニューの [タスクコンテキストウォッチ] でウィンドウを開きます。

■ タスクコンテキストウォッチ

タスクコンテキストウォッチは Workbench の [R アナライザ] メニュー下にある [タスクコンテキストウォッチ] でウィンドウを開きます。

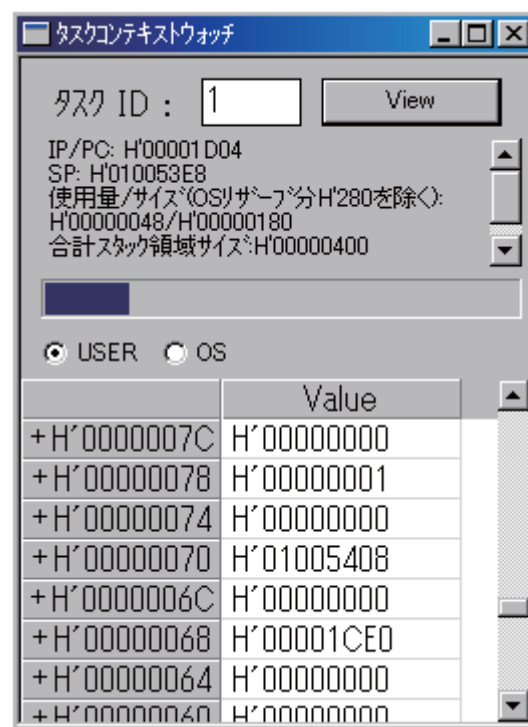
■ タスクコンテキストウォッチの表示内容

コンテキスト情報を表示したいタスクの ID を指定してください。

[View] ボタンにより、以下の内容を表示します。

- IP/PC : 指定タスクの現在の IP/PC
- SP : 指定タスクの現在の SP
- 使用量 / サイズ : 現在の SP から計算した使用量 / スタックサイズ
- バー : 上記の値から % を計算し、上記の表示の下に使用量のイメージを表示します。
- コンテキストリスト : OS の使用しているコンテキストの内容を表示します。

図 1.5-6 タスクコンテキストウィンドウ



第2章

機能説明

この章では μ T-REALOS アナライザの各機能とウィンドウについて説明します。

- 2.1 R アナライザウィンドウ
- 2.2 オブジェクト表示 (Object Viewer)
- 2.3 OS ブレーク
- 2.4 ログ
- 2.5 システムコール発行
- 2.6 スタック情報
- 2.7 タスクコンテキストウォッチ

2.1 R アナライザウィンドウ

R アナライザウィンドウは、μT-REALOS アナライザのメインウィンドウです。μT-REALOS アナライザのほとんどの機能をRアナライザ ウィンドウから実行できます。

■ R アナライザウィンドウを開く

μT-REALOS アナライザのメインウィンドウとなる「R アナライザウィンドウ」は Workbench の [R アナライザ] メニュー下にある [R アナライザ] でウィンドウを開きます。

■ R アナライザウィンドウの各部名称

R アナライザウィンドウは 3 ペイン型のウィンドウです。各ペインの役割は以下のようになります。

- 機能選択ペイン（左）：機能選択
 - ツリー形式ペイン（中）：各機能の設定状態や、表示対象を選択
 - 情報表示ペイン（右）：設定の反映や、各機能により取得されたデータの表示
- 実際には各機能において、以下の表示や設定を行います。

表 2.1-1 各ペインごとの機能割当て一覧

機能	ペイン	役割
Object Viewer	機能選択	表示するオブジェクトの選択
ObjectViewer-XXX *	ツリー形式	表示する ID の選択と表示実行
	ツリー形式内ボタン	[View]：オブジェクト情報表示実行
	情報表示	オブジェクト情報の表示
Setting	機能選択	Break, Log, SystemCall を選択
Setting-Break	ツリー形式	設定されたブレーク一覧表示
	情報表示	ブレークの設定
Setting-Log	ツリー形式	ログタイプの選択
	ツリー形式内ボタン	[Setting]：ログタイプの決定と変更
	情報表示	ログ情報の取得と表示
Setting-SystemCall	ツリー形式	発行システムコールの選択
	ツリー形式内ボタン	[View]：発行システムコールの決定
	情報表示	引数入力と発行の実施

*: ObjectViewer-XXX の XXX は、OS のオブジェクトを指しています。

2.2 オブジェクト表示 (Object Viewer)

オブジェクト表示は、 μ T-REALOS の管理するオブジェクトに関する現在の情報を表示する機能です。ここではオブジェクト表示機能の表示内容や操作方法について説明します。

2.2.1 ウィンドウの説明 (オブジェクト表示)

2.2.2 オブジェクト情報を表示させる

2.2.3 各オブジェクトの表示内容

2.2.4 列順を変更する

2.2.5 ソースジャンプ機能

2.2.1 ウィンドウの説明 (オブジェクト表示)

オブジェクト表示のウィンドウについて説明します。

■ オブジェクト表示 (Object Viewer) の開き方

SOFTUNE Workbench メニュー - [R アナライザ] - [R アナライザ] で「R アナライザウィンドウ」を開きます。機能選択ペインで「Object Viewer」を選択してください。

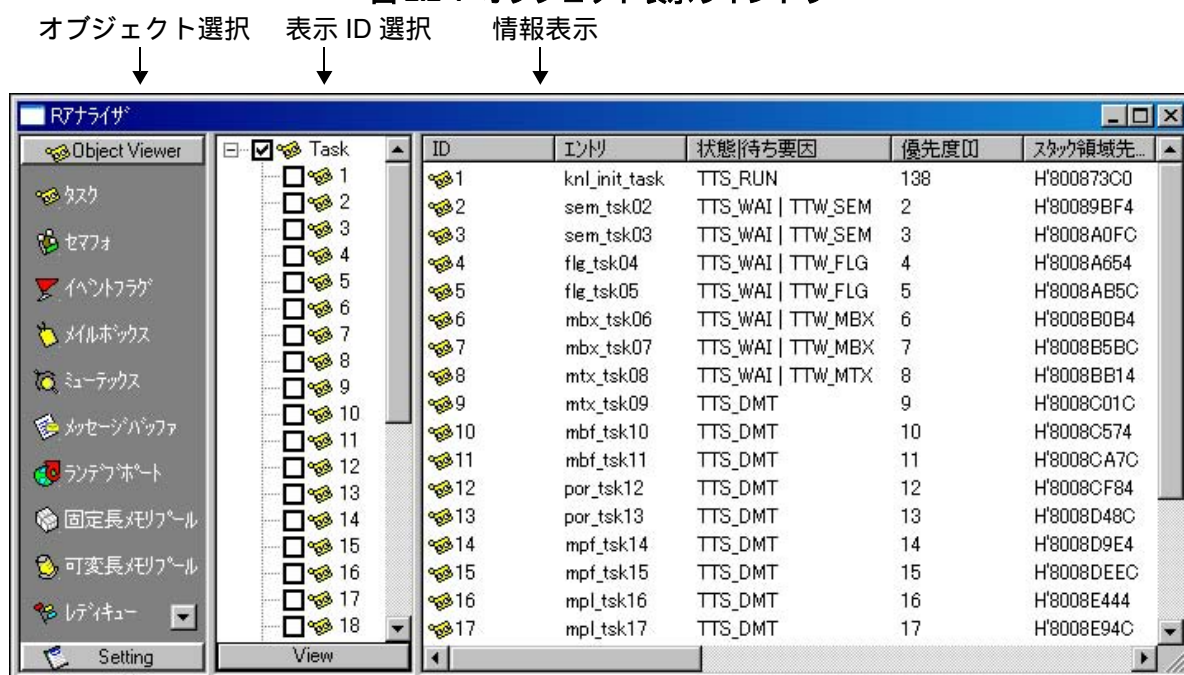
■ オブジェクト表示選択時の各ペインの機能

「Object Viewer」選択時、各ペインに以下の機能を割り当てます。

割当てられる機能は左のペインから以下ようになります。

- オブジェクト選択
- 表示 ID 選択および表示実施ボタン ([View] ボタン)
- 情報表示

図 2.2-1 オブジェクト表示ウィンドウ



■ オブジェクト選択

表示したいオブジェクトを選択してください。表示できるオブジェクトの一覧は「2.2.3 各オブジェクトの表示内容」を参照してください。

■ 表示 ID 選択および表示実施ボタン ([View] ボタン)

表示したい ID を選択してください。各 ID のチェックボックスにチェックをつけると、選択していることになります。

選択したオブジェクト名 (最上位レベルのチェックボックス) にチェックすると、すべて選択したことになります。全選択と ID 単位での指定を切り換えやすくするため、すべて表示するチェックボックスを外しても ID 単位のチェックボックスは外れません。レディキューやカーネル情報には選択する ID などが存在しないため、チェックボックスはありません。

選択後、[View] ボタンを押すことで、情報の取得を開始します。

情報取得中 [View] ボタンは、無効状態になります。

■ 情報表示

[View] ボタンを押すことで、選択されたオブジェクトの選択された ID の情報をリスト形式で表示します。

ただし、選択していても生成されていないオブジェクトは表示しません。

待ちタスクや、キューイングされているアイテムがある場合、ツリー形式でキュー状態を表示します。

その他に以下の機能があります。

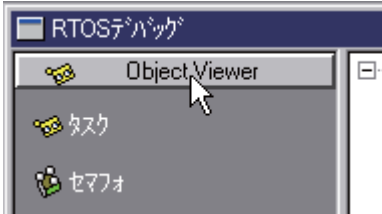


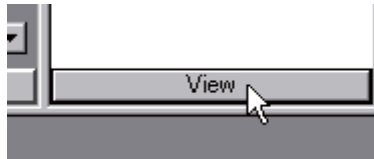
- 列順の変更【カラムのドラッグ移動】
変更したい列のカラムをドラッグし移動します。
- 列順の保存【右クリック 順列保存】
情報のリスト表示は、列順を変更し保存できます。
リスト表示 (カラム部分を除く) において、右クリックによりポップアップメニューを表示します。[列順保存] を選択すると、列順を保存します。以降、終了しても設定した列順で表示します。
- ソースジャンプ【右クリック ソースジャンプ】
タスクやハンドラのようにエントリのあるオブジェクトは、右クリックにより表示するポップアップメニューで [ソースジャンプ] を選択すると、ソースがある場合に限り指定したエントリのアドレスにあたる位置でソースウィンドウを開きます。

2.2.2 オブジェクト情報を表示させる

オブジェクトの情報を表示させる場合，表示させるオブジェクトと ID を選択し，[View] ボタンを押します。

■ オブジェクト情報の表示手順

以下にオブジェクト情報を表示させるための手順を示します。

1.	機能選択ペインで Object Viewer を選択してください。	
2.	表示させるオブジェクトを選択してください。	
3.	表示させる ID を選択してください。	
4.	[View] ボタンを押してください。	

レディキューなどのチェックボックスがないオブジェクトについては，2 および 3 の操作は発生しません。

2.2.3 各オブジェクトの表示内容

各オブジェクトの表示内容について説明します。

■ 各オブジェクトの表示内容

各オブジェクトごとの情報表示ペインに表示する情報は以下の内容になります。

表 2.2-1 オブジェクト表示の項目一覧 (1 / 4)

オブジェクト名	項目名	説明
タスク ・列順保存 可 ・ソースジャンプ可	拡張情報	生成時に指定した拡張情報です。
	エントリ	タスクの起動アドレスです。 タスク例外の場合はそのハンドラの起動アドレスとなります。
	優先度 [I]	タスク起動時の優先度です。
	スタック領域先頭アドレス	スタック領域の先頭アドレスです。 生成時に指定したアドレスを表示します。
	スタックサイズ	スタックのサイズです。 OS が使用するサイズも含まれます。
	状態 待ち要因	タスクの状態です。 タスク例外の場合は要因を表示します。
	優先度	現在の優先度です。
	優先度 [B]	ベース優先度です。
	対象 ID	オブジェクト待ちの際に対象となる ID です。
	TMO	タスクがタイムアウト待ちであることを知らせます。 (タイムアウトに「TMO_FEVR」を指定している場合は「TMO_FEVR」と表示します)
	起動カウント	起動要求キューイング数
	起床カウント	起床要求キューイング数
	強制待ちカウント	強制待ち要求ネスト数
セマフォ ・列順保存 可 ・ソースジャンプ不可	属性	セマフォの属性です。
	カウント [I]	初期カウントです。
	最大カウント	最大カウントです。
	現カウント	現在のカウントです。
	待ちタスク数	キューイングされているタスクの数です。

表 2.2-1 オブジェクト表示の項目一覧 (2 / 4)

オブジェクト名	項目名	説明
イベントフラグ ・列順保存 可 ・ソースジャンプ不可	属性	イベントフラグの属性です。
	パターン [I]	初期パターンです。
	パターン	現在のフラグパターンです。
	待ちタスク数	キューイングされているタスクの数です。
メールボックス ・列順保存 可 ・ソースジャンプ不可	属性	メールボックスの属性です。
	最大優先度	メッセージ優先度の最大です。 属性にメッセージ優先度順を指定することで値を表示します。
	待ちタスク数	キューイングされているタスクの数です。
	メッセージヘッダー数	キューイングされているメッセージヘッダーの数です。
ミューテックス ・列順保存 可 ・ソースジャンプ不可	属性	ミューテックスの属性です。
	上限優先度	属性で優先度上限プロトコルを選択した場合だけ有効です。
	ロック中のタスク ID	ロック中のタスク ID です。
	ロック待ちタスク数	ロック待ちでキューイングされているタスク数です。
メッセージバッファ ・列順保存 可 ・ソースジャンプ不可	属性	メッセージバッファの属性です。
	最大長メッセージ	最大長メッセージのサイズです。
	バッファサイズ	メッセージバッファ全体のサイズです。
	空きサイズ	現在のバッファの空きサイズです。
	送信待ちタスク数	送信待ちにキューイングされているタスク数です。
	受信待ちタスク数	受信待ちにキューイングされているタスク数です。
	メッセージ数	バッファにキューイングされているメッセージ数です。
ランデブポート ・列順保存 可 ・ソースジャンプ 不可	属性	ランデブポートの属性です。
	呼出しメッセージ 最大長	呼出し時のメッセージ最大長です。
	返答メッセージ最大長	返答時のメッセージ最大長です。
	呼出し待ちタスク数	呼出し待ちタスクの数です。
	受付け待ちタスク数	受付け待ちタスクの数です。

表 2.2-1 オブジェクト表示の項目一覧 (3 / 4)

オブジェクト名	項目名	説明
固定長 メモリプール ・列順保存 可 ・ソースジャンプ不可	属性	固定長メモリプールの属性です。
	ブロックサイズ	メモリブロック 1 つあたりのサイズです。
	残りブロック数	現在の未使用ブロック数です。
	全ブロック数	ブロックの総数です。
	待ちタスク数	キューイングされているタスク数です。
可変長 メモリプール ・列順保存 可 ・ソースジャンプ不可	属性	可変長メモリプールの属性です。
	領域サイズ	メモリプール領域のサイズです。
	獲得可能な最大サイズ	現在獲得可能な最大未使用サイズです。
	待ちタスク数	キューイングされているタスク数です。
レディキュー ・列順保存 不可 ・ソースジャンプ不可	起動順位	先頭は現在実行中のタスクと () 内に 1 / キューイング総数を表示します。以降は順位だけです。
	タスク ID	各順位にキューイングされているタスク ID を表示します。カーネル内で停止した場合は、キューの先頭と実行中タスクが異なる場合があります。この場合キューの先頭の ID を () 内に表示します。
タイマキュー ・列順保存 不可 ・ソースジャンプ不可	起動時間	現在時間の後ろに起動までの相対時間を () 内にキューイング順で表示します。起動するとキューから外れます。
	オブジェクト	キューイングされているオブジェクトの略称が入ります。 tsk...タスク cyc...周期ハンドラ / 周期起動ハンドラ alm...アラームハンドラ
	ID	各オブジェクトの ID です。
周期ハンドラ 周期起動ハンドラ ・列順保存 可 ・ソースジャンプ可	属性	周期ハンドラ / 周期起動ハンドラの属性です。
	拡張情報	生成時に指定した拡張情報です。
	起動番地	ハンドラの起動アドレスです。
	周期	設定されている起動周期です。
	起動状態	起動状態を示します。起動中 / 停止中を表示します。
	残り時間	実行までの残り時間です。
アラームハンドラ ・列順保存 可 ・ソースジャンプ可	属性	アラームハンドラの属性です。
	拡張情報	生成時に指定した拡張情報です。
	起動番地	ハンドラの起動アドレスです。
	起動状態	起動状態を示します。起動中 / 停止中を表示します。
	残り時間	実行までの残り時間です。

表 2.2-1 オブジェクト表示の項目一覧 (4 / 4)

オブジェクト名	項目名	説明
割込みハンドラ ・列順保存 不可 ・ソースジャンプ可	ハンドラ種別	CPU 例外または、ユーザ定義の割込み (割込と表示する) のどちらが定義されているか表示します。
	ハンドラ起動番地	ハンドラの起動アドレスです。
カーネル情報 ・列順保存 不可 ・ソースジャンプ不可	コンテキスト状態	非タスク部が実行中かどうか示します。
	CPU ロック状態	CPU ロック状態かどうか表示します。
	ディスパッチ禁止状態	ディスパッチ禁止状態かどうか表示します。
	システム時刻	現在のシステムクロックの値です。
	システム稼働時間	システムが起動してから現在までの時間です。
	OS コード実行状態	OS コードを実行中かどうかを示します (タスク解析モジュールがリンクされている必要があります) 。
	Kernel Ver	カーネルのバージョンです。
サブシステム ・列順保存 可 ・ソースジャンプ 不可	拡張 SVC ハンドラ	拡張 SVC ハンドラの起動アドレスです。
	優先度	サブシステムの優先度です。
	リソース ブロックサイズ	リソース管理ブロックのサイズです。
	ブレイク関数アドレス	ブレイク関数のエントリアドレスです。
	スタートアップ 関数アドレス	スタートアップ関数のエントリアドレスです。
	クリーンアップ 関数アドレス	クリーンアップ関数のエントリアドレスです。
	イベント処理 関数アドレス	イベント処理関数のエントリアドレスです。

表 2.2-2 オブジェクト表示の項目一覧 (1 / 3)

オブジェクト名	項目名	μT-REALOS
タスク	拡張情報	
	エントリ	
	優先度 [I]	
	スタック領域先頭アドレス	
	スタックサイズ	
	状態 待ち要因	
	優先度	
	優先度 [B]	
	対象 ID	
	TMO	
	起動カウント	×
	起床カウント	
	強制待ちカウント	
セマフォ	属性	
	カウント [I]	×
	最大カウント	
	現カウント	
	待ちタスク数	
イベントフラグ	属性	
	パターン [I]	×
	パターン	
	待ちタスク数	
メールボックス	属性	
	最大優先度	×
	待ちタスク数	
	メッセージヘッダー数	
ミューテックス	属性	
	上限優先度	
	ロック中のタスク ID	
	ロック待ちタスク数	

表 2.2-2 オブジェクト表示の項目一覧 (2 / 3)

オブジェクト名	項目名	μT-REALOS
メッセージ バッファ	属性	
	最大長メッセージ	
	バッファサイズ	
	空きサイズ	
	送信待ちタスク数	
	受信待ちタスク数	
	メッセージ数	
ランデブポート	属性	
	呼出しメッセージ最大長	
	返答メッセージ最大長	
	呼出し待ちタスク数	
	受付け待ちタスク数	
固定長 メモリプール	属性	
	ブロックサイズ	
	残りブロック数	
	全ブロック数	
	待ちタスク数	
可変長 メモリプール	属性	
	領域サイズ	
	獲得可能な最大サイズ	
	待ちタスク数	
レディキュー	起動順位	
	タスク ID	
タイマキュー	起動時間	
	オブジェクト	
	ID	
周期ハンドラ 周期起動 ハンドラ	属性	
	拡張情報	
	起動番地	
	周期	
	起動状態	
	残り時間	

表 2.2-2 オブジェクト表示の項目一覧 (3 / 3)

オブジェクト名	項目名	μT-REALOS
アラーム ハンドラ	属性	
	拡張情報	
	起動番地	
	起動状態	
	残り時間	
割込みハンドラ	ハンドラ種別	
	ハンドラ起動番地	
カーネル情報	コンテキスト状態	
	CPU ロック状態	
	ディスパッチ禁止状態	
	システム時刻	
	システム稼動時刻	
	OS コード実行状態	×
	Kernel Ver	
サブシステム	拡張 SVC ハンドラ	
	優先度	×
	リソースブロックサイズ	×
	ブレーク関数アドレス	×
	スタートアップ関数アドレス	×
	クリーンアップ関数アドレス	×
	イベント処理関数アドレス	×
: 表示可 × : 表示不可		

< 注意事項 >

各項目の値は、OS の動作状態から判断し無効の場合は表示しません。

ただし、初期起動時およびリセット実行時といった、OS の初期化が完了していない場合はすべての項目に不定の値が入ります。最初に起動するタスクまで実行した段階で利用してください。

また、対応していない項目については "E_NOSPT" を表示します。

2.2.4 列順を変更する

各オブジェクトの表示列順を変更できます。操作はカラムのドラッグアンドドロップにより行います。

■ 各オブジェクトのリスト形式の列順を変更する【カラムのドラッグアンドドロップ】

レディキュー、タイマキュー、割込みハンドラ、カーネル情報の表示を除き、各オブジェクトごとにリスト形式の列順を変更できます。

列順を変更するには、移動したいカラムをドラッグして、希望の位置にドロップすることで行います。

図 2.2-2 列順変更の例

ID	エントリ	属性	拡張情報	エントリ	優先度	スタック
1	E_NOSPT	H'00000001	Task_1	1	H'00	
1(スクリプト例外)			cpuexcrt1			
2	E_NOSPT	H'00000002	Task_2	2	H'00	
3	E_NOSPT	H'00000003	Task_3	3	H'00	
4	E_NOSPT	H'00000004	Task_4	4	H'00	
5	E_NOSPT	H'00000005	Task_5	5	H'00	

ID	エントリ	属性	拡張情報	優先度	スタック
1	Task_1	E_NOSPT	H'00000001	1	H'00
1(スクリプト例外)	cpuexcrt1				
2	Task_2	E_NOSPT	H'00000002	2	H'00
3	Task_3	E_NOSPT	H'00000003	3	H'00
4	Task_4	E_NOSPT	H'00000004	4	H'00
5	Task_5	E_NOSPT	H'00000005	5	H'00

■ 変更した列順を保存する【右クリック [列順保存]】

変更した列順で次回から（デバッグを終了しても継続的に）表示したい場合は、列順の保存を行ってください。

列順の保存は、リスト表示部（カラム位置を除く）で右クリックすると表示するポップアップメニューから、[列順保存]を選択することで行われます。

< 注意事項 >

再インストールを実行した場合は、保存状態はクリアされます。

2.2.5 ソースジャンプ機能

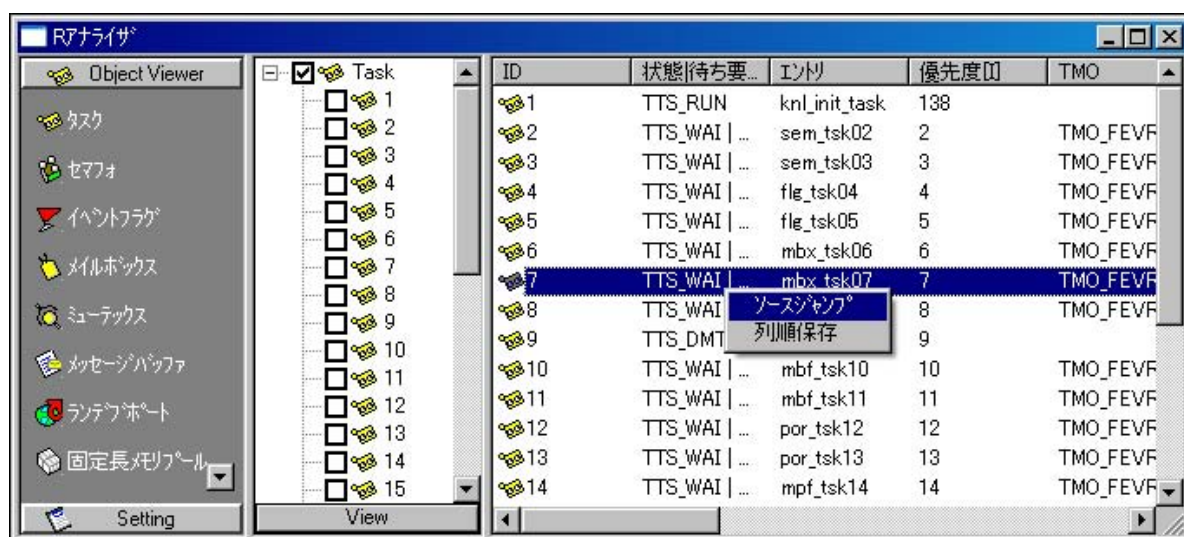
起動開始アドレス (エントリ) に関する情報を持つオブジェクト情報表示では , ソースコードを開く機能があります。

■ ソースジャンプ機能【右クリック [ソースジャンプ]】

タスク、周期ハンドラ / 周期起動ハンドラ、アラームハンドラ、割込みハンドラの表示では、その起動アドレスを示すソースコードへのジャンプ機能があります。

ソースジャンプを行うには、開きたいオブジェクトの ID の列において、右クリックしポップアップメニューから [ソースジャンプ] を選択します。

図 2.2-3 ポップアップメニュー [ソースジャンプ] (SOFTUNE μ T-REALOS の場合)



< 注意事項 >

対象のソースがアイコン化されている場合は、すでに開いているファイルへジャンプするなどの誤動作を起こす場合があります。ソースがない場合も同様です。

また、プロジェクトを別ディレクトリにコピーするなど、対象のソースへのディレクトリ情報が破壊されている場合についても実行できない場合があります。

2.3 OS ブレーク

OS ブレークは、 μ T-REALOS を意識したブレーク機能です。以降、OS ブレーク機能の表示内容や操作方法について説明します。

2.3.1 ウィンドウの説明 (OS ブレーク)

2.3.2 ブレークを設定する

2.3.3 削除と変更

2.3.1 ウィンドウの説明 (OS ブレーク)

OS ブレークの設定ウィンドウについて説明します。

■ OS ブレーク設定の開き方

機能選択ボタンで [Setting] を選択し, [Break] で開きます。

■ OS ブレーク設定機能選択時の各ペインの機能

「Break」選択時, 各ペインには以下の機能を割り当てます。

割当てられる機能は左のペインから以下ようになります。

- 「Break」固定
- 設定済ブレークポイント
- 設定ウィンドウ

図 2.3-1 OS ブレークウィンドウ

設定済ブレークポイント 設定ウィンドウ



■ 追加・変更・削除ボタン (設定ウィンドウ)

設定ウィンドウの各ボタンにより, OS ブレークの追加・変更・削除が可能です。

■ ID・パスカウント・アドレス入力フォーム (設定ウィンドウ)

- [ID]: OS ブレーク ID を入力するボックスです。
設定済の [OS Break ID] をクリックすることで、本エディットボックスに反映します。
入力基数は 10 進数です。
- [パスカウント]: パスカウントを入力するボックスです。
「1」で、事象が発生するたびに停止します。「n」の場合、「n」回の事象発生ごとに停止します。
- 設定範囲は「1 ~ 65535」です。
入力基数は 10 進数です。
- [アドレス]: 実行・アクセスブレークの対象となるアドレスを入力するボックスです。
シンボルでの入力も可能です。
入力は 16 進数 (0x や h' の記述は不要) です。

■ ブレーク種別

各ブレーク種別を選択してください。選択可能なブレーク種別は以下のとおりです。

表 2.3-1 ブレーク種別

ブレーク種別	説明
実行ブレーク	タスクを特定するコードブレーク 例) 共通関数 func () をタスク 2 からよんだときにブレーク
アクセスブレーク	タスクを特定するデータブレーク 例) 大域変数 ex_val の値をタスク 1 が変更したときにブレーク
ディスパッチャ (ENTER)	ディスパッチャ起動時に対するブレーク 例) タスク 3 の実行権が奪われるときにブレーク
ディスパッチャ (LEAVE)	ディスパッチャ終了時に対するブレーク 例) タスク 4 に実行権が移るときにブレーク
システムコール (ENTER)	システムコール発行時に対するブレーク 例) タスク 6 が tk_sig_sem () を発行するときにブレーク
システムコール (LEAVE)	システムコールの復帰値決定時に対するブレーク 例) タスク 7 の tk_wai_sem () を解除されるときにブレーク

■ ブレークの実現方法

実行ブレーク ...

SOFTUNE のコードブレークポイントを使用しブレークするたびにタスク ID を判定し、ブレーク条件を満たさない場合は再度実行を行います。

アクセスブレーク ...

SOFTUNE のデータブレークポイントを使用しブレークするたびにタスク ID を判定し、ブレーク条件を満たさない場合は再度実行を行います。

ディスパッチャブレーク ...

ユーザのデータ領域にブレーク設定条件を書き込み、ユーザプログラムに組み込んだ μ T-REALOS アナライザが提供するライブラリ中で判定を行います。ただし、タスク 1 個につき 10 バイトのデータ領域を使用します。条件の判定を実行しながら行うためリアルタイム性に優れています。

■ 対象タスクブレーク種別

ブレークの対象となるタスク ID を選択してください。「All Tasks」を選択した場合、すべてのタスクが対象となります。ディスパッチャブレークの場合、対象となるタスクイメージが変わります。

- ディスパッチャ (ENTER) ブレーク : 実行権を奪われる方のタスク
- ディスパッチャ (LEAVE) ブレーク : 実行権を取得する方のタスク

■ システムコール

ブレークの対象となるシステムコールを選択してください。ENTER/LEAVE の相違点は、「発行直後」と「発行から戻る直前」です。tk_wai_xxx のシステムコールの場合、Wait が解除されたときにブレークがかかります。また、タイムアウトによる復帰は、対象外となります。

< 注意事項 >

ICE 使用時、OS ブレークはハードウェアブレークを使用します。ハードウェアブレークに空きがない場合、設定できません。また、ディスパッチャブレークおよびシステムコールブレークは、カーネル内部で停止します。よってトリガーとして使用することをお勧めします。

なお、ディスパッチャ (LEAVE) のブレークは、モニタリング機能やディスパッチログ実施時に併用するとブレークできません。

また、タスク解析モジュールを使用していない場合は、ディスパッチャブレーク、システムコールブレークは設定できません。

2.3.2 ブレークを設定する

OS ブレークの設定方法について説明します。

■ リソースとなるブレーク

OS ブレークは Workbench のブレーク機能を使用して実現しています。アクセスブレークの場合はデータブレークを使用します。それ以外のブレーク種別では、実行ブレーク (ICE の場合、ハードウェアブレーク) を使用します。これらに空きがない場合設定できません。

■ 設定方法【[追加] ボタン】

新たにブレークを設定する場合は、[追加] ボタンを使用します。

[追加] ボタンによる設定では、ID の設定ボックスに入っている値を無視し、ブレーク ID を自動的に割り当てます。

設定済みのブレークとパスカウントだけが異なる条件での設定追加はできません。

ディスパッチャブレークを使用する場合は必ずタスク解析モジュールファイルセット内の R_D_siz.c の USER_TASK_NUM にタスク数を設定してください。設定の詳細は「3.3 カスタマイズの方法」を参照してください。

表 2.3-2 各ブレーク種別使用時に入力が必要な項目

ブレーク種別	入力項目			
	アドレス *	パスカウント	対象タスク	システムコール
実行ブレーク				-
アクセスブレーク				-
ディスパッチャブレーク	-			-
システムコールブレーク	-			

* : 「アドレス」がグローバルシンボルの場合、シンボル入力も可能です。

: 必要

- : 不要

■ 設定結果

設定が成功した場合は、「設定済ブレークポイント」に設定したブレークが追加されます。

■ ブレーク発生

ブレークが発生した場合、以下のダイアログを表示します。

図 2.3-2 OS ブレークヒット時のダイアログ



2.3.3 削除と変更

OS ブレークの「削除」と「変更」方法について説明します。

■ 設定済のブレークを設定ウィンドウに反映させる

設定済のブレークの「OS Break ID」アイテムをクリックすることで、設定ウィンドウにクリックしたブレークの情報を反映します。

■ 変更

「変更」する場合は、アドレスや対象タスクIDなどを変更して[変更]ボタンを押します。

■ 削除

「削除」する場合は、設定済ブレークを選択し、設定ウィンドウに反映させて [削除] ボタンにより削除できます。

< 注意事項 >

R アナライザウィンドウを閉じた場合や、μT-REALOS アナライザ終了した場合、デバッグを終了した場合、設定された OS ブレークはすべて削除します。

2.4 ログ

μT-REALOS アナライザにおけるログとは、μT-REALOS のふるまいをロギングした機能のことです。以降、ログの設定方法と各ウィンドウについて説明します。

- 2.4.1 ログ機能の仕様について
- 2.4.2 ウィンドウの説明 (ログ設定)
- 2.4.3 ログを設定する (ディスパッチログ)
- 2.4.4 ログを設定する (モジュール使用ログ)
- 2.4.5 ログを設定する (統計ログ)
- 2.4.6 ログの取得
- 2.4.7 ログデータの説明
- 2.4.8 遷移図ウィンドウの説明
- 2.4.9 統計ウィンドウの説明
- 2.4.10 モニタリングウィンドウの説明
- 2.4.11 環境設定
- 2.4.12 リアルタイム統計の説明
- 2.4.13 ログファイルの出力

2.4.1 ログ機能の仕様について

μT-REALOS アナライザにおけるログ機能の仕様について説明します。

■ サンプルング手法

2 種類のサンプルング機能を用意しています。

● デバッガのブレークを利用したサンプルング (ディスパッチログ)

OS 内部にブレークを設定し、ディスパッチが発生したときにツール側でサンプルングを行います。

< 特長 >

- タスク解析モジュールをリンクすることなくログの取得を行うことができます。
- タスク切換えだけロギングします。
- 取得できるイベント数は最大 32767 です。

● タスク解析モジュールを使用したサンプルング (モジュール使用ログ)

タスク解析モジュールをリンクすることにより、ディスパッチログでは取得できない種類の OS のふるまいをサンプルングすることが可能になります。サンプルングはユーザメモリに行います。

< 特長 >

- ブレークを使用したサンプルングと異なり、システムをまったく停止させません。
- 様々なイベントを選択してロギングできます (最大 9 個)。
- 取得できるイベント数は最大 2048 (0x10000 バイト) です。ユーザメモリ量により変動します。

■ タイムスタンプの付加

サンプルングした各イベントには、タイムスタンプを付加します (モジュール使用ログの type1 を選択した場合を除きます)。

タイムスタンプは、通常 μT-REALOS のシステムクロックの値となります。

よって、システムクロックを設定していない場合、タイムスタンプは 0 または不定値になりますので、ログ機能を使用する場合は必ずシステムクロックを設定してください。

■ タスク遷移図の表示

サンプルングしたデータを遷移図として表示します。タスク遷移図は、タスクの切換え状態を容易に把握できます。

■ 統計グラフの表示

各タスクの CPU 占有時間とその割合をグラフで表示します。最大実行時間は、システム設計時に考慮した制限時間内で処理が完了しているか確認する上で、重要なデータとなります。

■ モニタリング機能

デバッガの擬似オンザフライ機能を使用して、実行中のタスク ID をモニタリングします。

■ ログファイルの出力

取得したログは、データとして保存できます。製品ディスクに添付されているログビューアで、データを見ることが可能です。

■ 環境設定 - モニタリング間隔時間設定

モニタリングのサンプリング時間設定とリアルタイム統計のサンプリング時間設定ができます。ただし、リアルタイム統計のサンプリング時間設定はタスク解析モジュールに type5 を使用している場合に限りします。

■ 環境設定 - デッドライン設定

タスクの実行時間の上限を設定できます。ただし、タスク解析モジュールに type5 を使用している場合に限りします。

2.4.2 ウィンドウの説明 (ログ設定)

ログの設定ウィンドウについて説明します。

■ ログ設定の開き方

機能選択ボタンで [Setting] を選択し, [Log] で開きます。

■ ログ機能選択時の各ペインの機能

「Log」選択時, 各ペインには以下の機能を割り当てます。

割当てられる機能は左のペインから以下ようになります。

- 「Log」固定
- ログ設定選択
- 設定状態およびデータ表示ウィンドウ



■ ログ設定選択リスト (ログ設定選択ウィンドウ)

ログのサンプリング手法の選択と, 選択したサンプリング手法における取得イベントの選択をラジオボタンの切換えとチェックボックスのチェックにより行います。

また, 「モジュール使用ログ」ではバッファフル/リングバッファもチェックボックスの ON/OFF により可能です。

「統計ログ」はタスク解析モジュールに type5 を使用している場合に限り使用可能になります。

■ [Setting] ボタン (ログ設定選択ウィンドウ)

設定選択ウィンドウの [Setting] ボタンを押すことで、設定を反映します。

なお、ログを設定するときは、必ずすべてのメモリにアクセスが可能な状態である必要があります (特にモジュール使用ログ)。

メモリアクセスが許されない領域にタスク解析モジュールが配置されている場合、フラグの制御やサンプリング時にメモリアクセスエラーが発生し、正常に動作しないことが予測されます。

よってログの設定は、最初に起動するタスクの先頭まで実行した後に行うことをお勧めします。

■ フラグクリア・取得・遷移図・統計グラフ・モニタリングボタン (データ表示ウィンドウ)

- [フラグクリア] ボタン : タスク解析モジュールの変数を初期化します。
モジュール使用ログを設定している場合に、システムのリセット時や再設定を行いたい場合に使用します。
実行するとモジュールでログを管理しているフラグをクリアします。
- [取得] ボタン : サンプリングしたログデータを取得します。
静的解析ではありませんので、設定後にシステムを必ず動作させ、かつ停止させたときに取得する必要があります。システム動作中の取得や設定に関する保証はできません。
なお、ログ情報を取得する区間では、ログの設定同様に、必ずすべてのメモリにアクセスが可能な状態である必要があります (特にモジュール使用ログ)。
メモリアクセスが許されない領域にタスク解析モジュールが配置されている場合、フラグの制御やサンプリング時にメモリアクセスエラーが発生し、正常に動作しない場合があります。
- [遷移図 / 統計グラフ / モニタリング / 環境設定 / リアルタイム統計] ボタン
遷移図、統計グラフ、モニタリング、アナライザの環境設定、リアルタイム統計のウィンドウを開くためのボタンです。

■ 設定ログリスト (データ表示ウィンドウ)

設定されているログタイプを表示します。また、取得後、各タイプの発生イベント数を表示します。

■ データ表示部 (データ表示ウィンドウ)

取得したデータを表示します。ただしタスク解析モジュールに type5 を使用している場合はトレースデータを取得しないため、データの表示はしません。

データの読み方については、「2.4.7 ログデータの説明」を参照してください。

2.4.3 ログを設定する (ディスパッチログ)

ディスパッチログの設定について説明します。

■ 選択可能なイベント

以下のイベントを選択できます。

表 2.4-1 ディスパッチログのイベント

イベント	説明
実行タスク切替時	実行 (RUN 状態) タスクが切り替わるイベントをロギングします。

■ 設定方法

ログ設定選択ウィンドウで、「ディスパッチログ」を選択します。

イベントの種類を選択します。

ディスパッチログでは「実行タスク切替時」を選択します。

[Setting] ボタンを押します。

最後に、表示ウィンドウの設定ログタイプに登録されていることを確認して完了です。

図 2.4-2 ディスパッチログ設定時の設定ログリスト登録例



2.4.4 ログを設定する (モジュール使用ログ)

モジュール使用ログの設定について説明します。

■ 選択可能なイベント

ログに取得可能なイベントは、以下のイベントから複数選択できます。

表 2.4-2 モジュール使用ログのイベント

イベント	説明
割込み	割込みハンドラ開始時に発生
割込み :LEAVE	割込みハンドラ終了時に発生
タイマハンドラ *	タイマハンドラ (周期, アラーム) 開始時に発生
タイマハンドラ :LEAVE *	タイマハンドラ (周期, アラーム) 終了時に発生
ディスパッチャ	ディスパッチ開始時に発生
ディスパッチャ :LEAVE *	ディスパッチ終了時に発生
システムコール *	システムコール発行時に発生
システムコール :LEAVE	システムコールの終了時または復帰値確定時に発生
コメント *	ユーザ任意のイベント発生時

※: 「基本」に含まれる設定

■ ログ設定選択ペインで行う設定

「基本」、「割込み」の設定から複数選択できます。

「基本」を設定することで、タスク遷移図を表示させるのに必要なデータを取得します。

「割込み」で取得できるイベントは「割込み」、「割込み :LEAVE」で、いわゆる割込みの入口と出口を押さえた場合に選択してください。

「バッファフル」の設定を行うことも可能です。

■ バッファフルの設定

バッファフルのチェックを変更することでバッファのタイプを選択できます。

チェックしない場合はリングバッファとなります

- リングバッファ: バッファサイズを超えても停止しないモードです。
サンプリングイベントがバッファサイズより大きくなると古いイベントを削除して新しいイベントをサンプリングします。よって停止させたときの直前のイベントが保証されます。

- バッファフル: バッファが一杯になった時点で停止するモードです。
サンプリングイベントがバッファサイズより大きくなるとブレイクします。バッファサイズ以上のサンプリングを強制的に停止するモードです。
なお、バッファタイプは単独では意味をなしません。バッファの動作モードを切り換えるだけです。

■ 高度な設定 (カスタマイズ設定)

右クリックによりポップアップメニューを開き、「カスタマイズ設定」をチェックすることにより、好きな組み合わせで複数選択できます。

ただしタスク遷移図を表示させるには、「ディスパッチャ:LEAVE」を選択する必要があります。

■ 設定方法

ログ設定選択ウィンドウで、「モジュール使用ログ」を選択します。

イベントを選択します。

バッファタイプを選択します。

[Setting] ボタンを押します。

最後に、表示ウィンドウの設定ログリストに登録されていることを確認して完了です。

■ 推奨するイベントの組み合わせ

お勧めするイベントの組み合わせは、「基本」です。

「カスタマイズ設定」を行う場合も、「基本」に含まれる設定のうち、「ディスパッチャ:LEAVE」|「システムコール」の2点に、ほかのイベントを追加する方向で選択することを推奨します。

< 注意事項 >

タイムスタンプの設定は、 μ T-REALOS アナライザからはできません。タイムスタンプの設定を変更するには、コンフィギュレータの「デバッグ」で必要なモジュールセットを選択し、ビルドし直すことで行います。モジュールセットに関する説明は「第3章 タスク解析モジュール」を参照してください。

コメントの使用方法については「第3章 タスク解析モジュール」を参照してください。

2.4.5 ログを設定する (統計ログ)

統計ログについて説明します。

■ 選択可能なイベント

以下のイベントを選択できます。

表 2.4-3 統計ログ時のイベント

イベント	説明
時間測定	ディスパッチャ：LEAVE の発生時から以下のシステムコールの実行までをタスクの実行単位として測定します。 tk_ext_tsk(), tk_exd_tsk(), tk_ter_tsk()

■ 統計ログの特長

タスク解析モジュールに type5 を使用している場合に限り使用できます。トレースメモリを使用せずにモジュール内で最長実行時間、最短実行時間、実行時間合計、処理回数を測定しているので、長時間の測定が可能です (最大約 49 日)。

また、設定したタスク実行時間の上限値 (デッドライン) の判定をモジュール内で行っているためタスクの実行時間がデッドラインをオーバーした回数を表示できます。

■ 統計ログの測定方法

ディスパッチャ：LEAVE の発生から tk_ext_tsk(), tk_exd_tsk(), tk_ter_tsk() までを 1 回の実行として測定します。必ずタスク解析モジュール type5 ファイルセット内の R_D_siz.c の TRC_DATA_NUM にバッファサイズを、USER_TASK_NUM にタスク数をそれぞれ設定してください。設定の詳細は「3.3 カスタマイズの方法」を参照してください。

■ 設定方法

ログ設定選択ウィンドウで、「統計ログ」を選択します。

「時間測定」をチェックします。

[Setting] ボタンを押します。

最後に、表示ウィンドウの設定ログリストに「ディスパッチャ：LEAVE」「システムコール」が登録されていることを確認して完了です。

2.4.6 ログの取得

ログの取得について説明します。

■ ログの取得

データ表示ウィンドウの [取得] ボタンによりログを取得します。

取得が完了すると以下のダイアログが表示され、[OK] ボタンを押すとデータを表示します。

図 2.4-3 ログ取得完了ダイアログ



< 注意事項 >

ログの設定および取得は、ブレーク中に実施してください。システム動作中に実施した場合、正常に動作しない場合があります。

2.4.7 ログデータの説明

取得したログデータについて説明します。

■ ログデータの表示

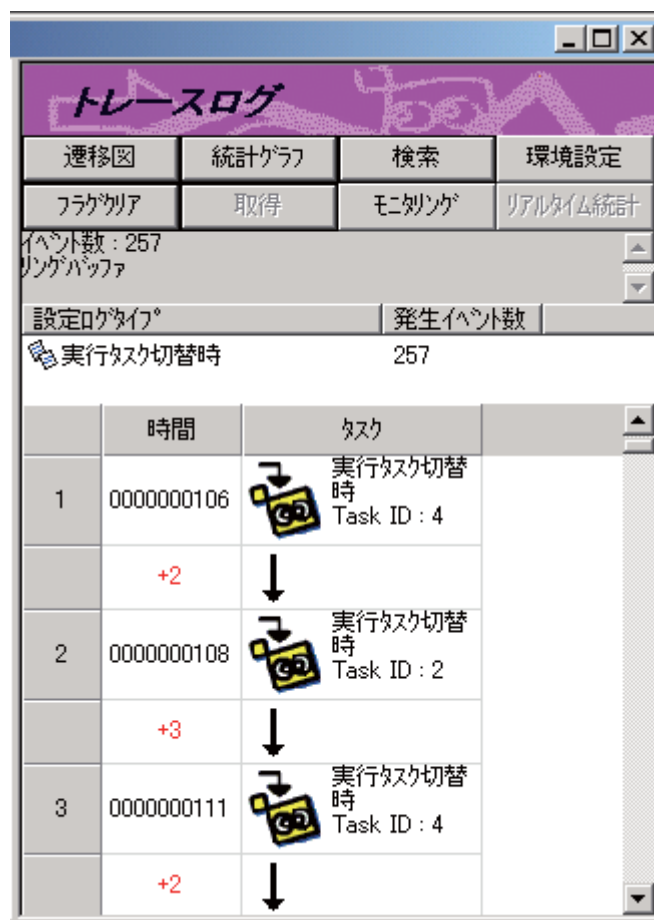
ログの取得に成功するとデータ表示部に以下のように表示します。

ディスパッチログの場合、項目は時間とタスクだけになります。

図 2.4-4 モジュール使用ログの表示データ例

トレースログ					
遷移図		統計グラフ		検索	
フラグクリア		取得		モニタリング	
				環境設定	
				リアルタイム統計	
イベント数 : 256					
バッファフル					
設定ログタイプ		発生イベント数			
タイマハンドラ		1			
タイマハンドラ:LEAVE		1			
	時間	タスク	システムコール		割り込み/ハンドラ
	+0	↓			↓
3	0000014742	 割り込み終了			 タスクタイムアウト ID : 1 終了
	+1	↓			
4	0000014743	 デイスパッチャ Task ID : 1			
	+1	↓			
5	0000014744	 サービスコール Task ID : 1	tk_sta_alm0		
	+1	↓			

図 2.4-5 ディスパッチログの表示データ例



■ ログデータの表示アイコンとパラメーター一覧

以下、各項目に表示するアイコンと表示パラメーターの一覧を示します。

表 2.4-4 ログデータの表示アイコンとパラメーター一覧

項目	アイコン	表示内容
時間	なし	黒：発生イベント時間 赤：前のイベントとの時間差
タスク		ディスパッチの開始 ID はディスパッチ元のタスク ID
		ディスパッチの終了 ID はディスパッチ先のタスク ID
		タスクからのシステムコール発行の開始 ID は発行タスクの ID
		タスクからのシステムコール発行の終了 ID は発行タスクの ID
		ハンドラ起動によるマスク区間
		コメント ID, コメント文字列
システムコール	なし	システムコール名および復帰値
割込み / ハンドラ		割込みまたはハンドラの開始 ID は割込みまたはハンドラの ID
		割込みまたはハンドラの終了 ID は割込みまたはハンドラの ID
		ハンドラからのシステムコール発行の開始
		ハンドラからのシステムコール発行の終了
備考	なし	使用していません。

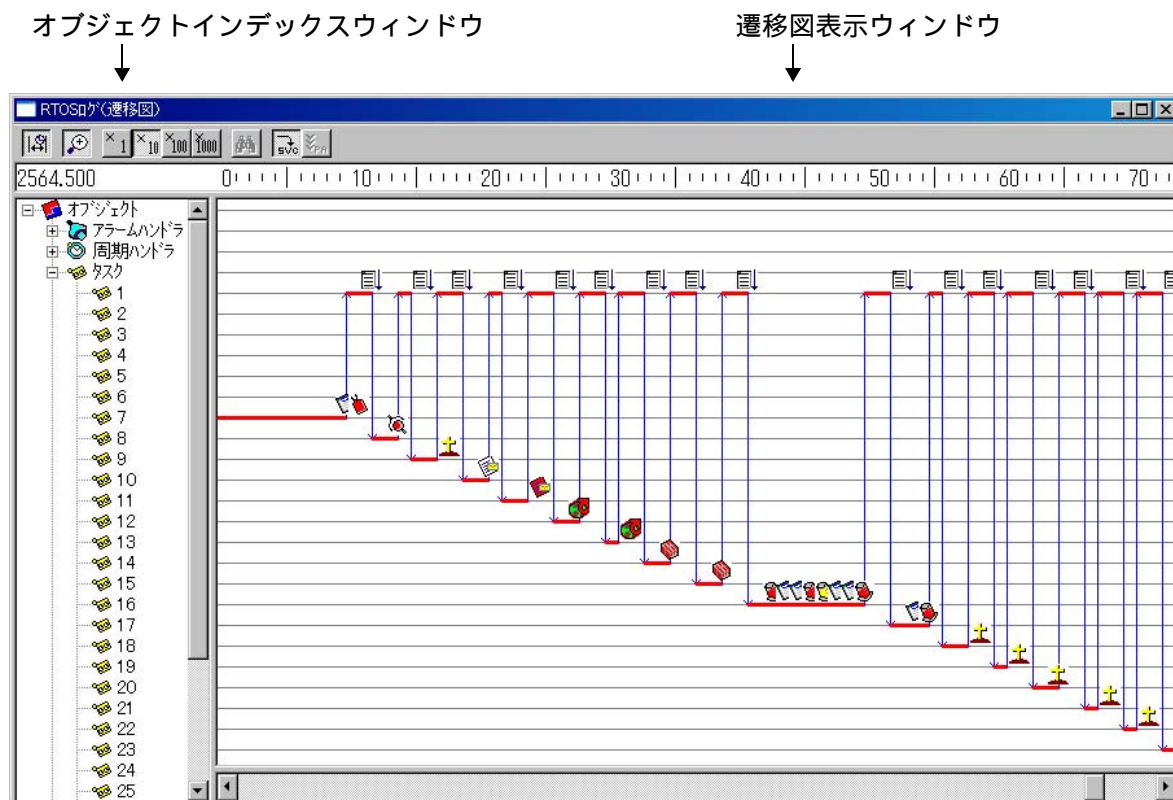
2.4.8 遷移図ウィンドウの説明

遷移図ウィンドウについて説明します。

■ 遷移図ウィンドウを開く

データ表示ウィンドウ内にある「遷移図」ボタンを押すとウィンドウが開きます。

図 2.4-6 遷移図ウィンドウ





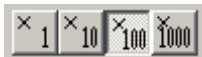
■ オブジェクトインデックスウィンドウ

システムに登録されているタスクおよびタイマハンドラ（周期 / 周期起動，アラーム），割込みハンドラをツリー形式で表示します。ただし，ログタイプに選択しなかったオブジェクトについては表示しません。

■ ツールバー

ツールバーのボタンを操作することで、表示方法を切り換えることができます。

表 2.4-5 遷移図ウィンドウのツールバー

ボタン	機能説明
	時間単位で表示するか、イベント単位で表示するか切り換えます。 Push 状態：時間単位の表示（初期状態） Pop 状態：イベント単位での表示
	全体表示にするか、時間単位を決めて表示するか切り換えます。 この場合、右に配置された拡大率ボタン（以下参照）と併用します。 Push 状態：時間単位の表示 Pop 状態：全体表示
	拡大率ボタン 左から 1 システムクロックあたり 1point。 以下、10, 100, 1000 point に設定。


■ 遷移図表示ウィンドウの線とアイコン

タスク遷移の線を表示するにはモジュール使用ログの場合、「ディスパッチャ:LEAVE」を取得ログタイプに選択する必要があります。表 2.4-6 に示すアイコンと線を表示します。

「ディスパッチログ」の「実行タスク切替時」選択時は、タスク遷移の線だけを表示します。






















遷移図表示ウィンドウは、オブジェクトインデックスウィンドウに対応しており、各オブジェクトの各 ID に関するイベントは同じ高さに表示します。

表 2.4-6 遷移図ウィンドウの線とアイコン

線 / アイコン	説明
赤横線	RUN 状態を示します。
青縦線 (矢印付き)	ディスパッチを示します。矢印の方向に切り換わったことを示します。 「ディスパッチャ:LEAVE」のデータを元に表示します。
	ハンドラや割込みの開始と終了を示します。 「タイマハンドラ」、「割込み」、「タイムアウト」のデータを元に表示します。

システムコールの場合は、直感性を持たせるため、数個のアイコンに処理が関係しているシステムコールを割り当ててあります。割り当てを表 2.4-7 に示します。また、赤いアイコンで表示された場合は待ち状態になる可能性のあるシステムコールです。

表 2.4-7 システムコールのアイコン

アイコン	システムコール
	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
	コメント
	システムコール ILEAVE 設定時

■ システムコールアイコン上でのマウス操作

システムコールアイコン上でマウス操作を行うことで、以下の機能が動作します。

- ホバリング : マウスをシステムコールアイコン上で一定時間停止させる動作です。
ツールチップが表示され、発生イベント番号と発生したシステムコール名を表示します。
- ダブルクリック : システムコールアイコン上でダブルクリックさせる動作です。
データ表示ウィンドウのデータ表示部を指定したイベントに合わせてスクロールさせ、R アナライザ ウィンドウをアクティブにします。

2.4.9 統計ウィンドウの説明

統計ウィンドウについて説明します。

■ 統計ウィンドウ

統計ウィンドウは、サンプリングしたログデータから、タスクごとの実行時間やディスパッチ回数を計算し、グラフで表示します。このほか、連続実行の最大 / 最小時間について表示します。

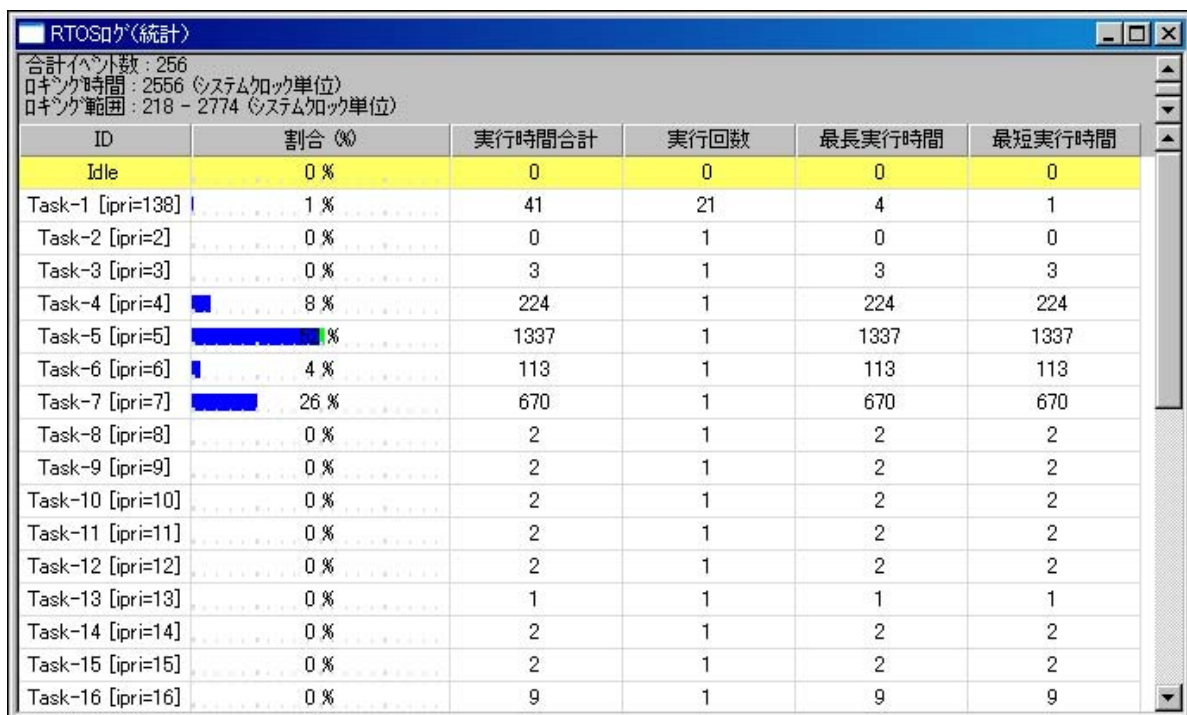
■ 統計ウィンドウを開く

以下の条件を満足している場合、統計ウィンドウを開くことが可能です。

- ログを「ディスパッチログ」または「モジュール使用ログ」で設定している。
- ログの取得が完了している。

データ表示ウィンドウ内にある「統計グラフ」ボタンを押すとウィンドウが開きます。

図 2.4-7 統計ウィンドウ



■ 統計ウィンドウの表示項目

統計ウィンドウの表示項目は、以下のとおりです。

表 2.4-8 統計ウィンドウの表示項目

項目	説明
割合 [%]	各タスクのサンプリング時間全体に対する割合
実行時間合計	各タスクの実行時間の合計
実行回数	各タスクのディスパッチ回数
最長実行時間	各タスクの連続実行時間の最大
最短実行時間	各タスクの連続実行時間の最小

< 注意事項 >

各タスクの連続実行時間の最大は、システムのリアルタイム性を確認する上で重要なデータですが、サンプリングした区間での最大時間であることにご注意ください。

2.4.10 モニタリングウィンドウの説明

モニタリングウィンドウについて説明します。

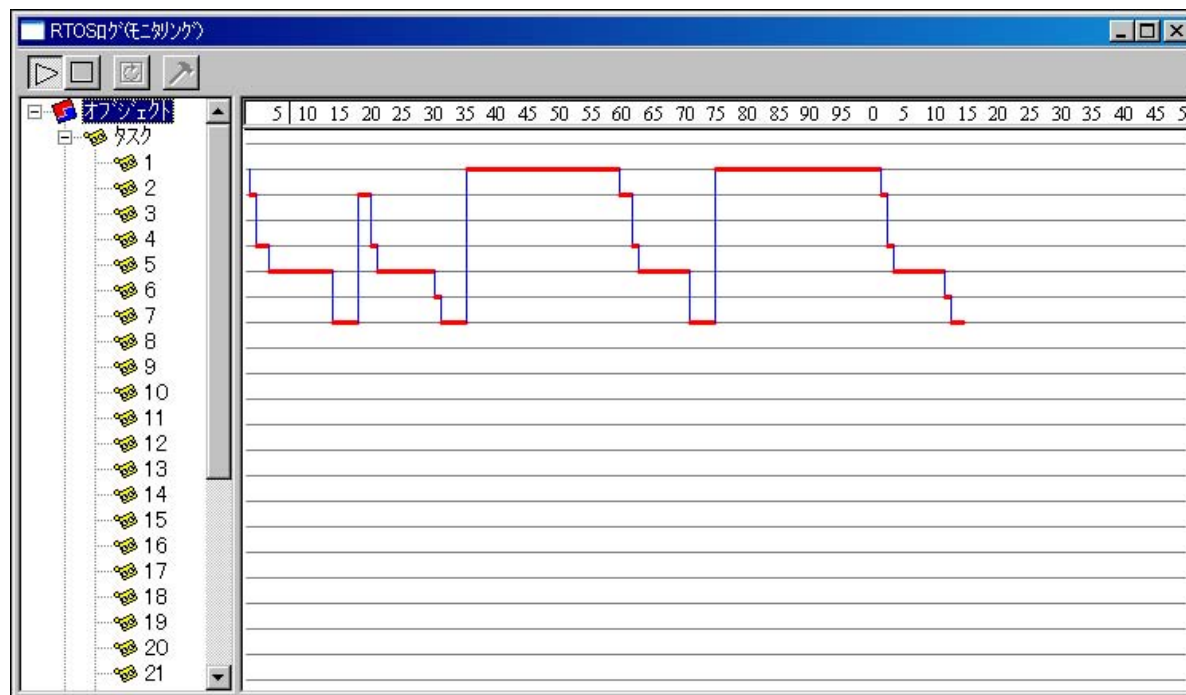
■ モニタリングウィンドウ

モニタリングウィンドウは、デバッガの擬似オンザフライ機能を使用し、一定間隔（サンプリング時間間隔は環境設定の「モニタリング間隔設定」で変更できます）で現在の実行タスク ID を取得します。このデータをタスク遷移図ライクに表示するウィンドウです。アイドルタスクに遷移した場合はタスクツリーの一番下の「アイドル」に遷移します。

■ モニタリングウィンドウを開く

データ表示ウィンドウ内にある[モニタリング]ボタンを押すとウィンドウが開きます。





図 2.4-8 モニタリングウィンドウ



■ モニタリングの操作 (ツールバー)

モニタリングの操作はツールバーの各ボタンで行います。
各ボタンに割り当てられた機能を以下の表に示します。

表 2.4-9 モニタリングの操作

ボタン	機能説明
	モニタリングを開始します。 クリック時に、デバッガが停止している場合は、デバッガを実行させてモニタリングを開始するか問い合わせます。
	モニタリングを停止します。 クリック時に、デバッガが実行している場合は、デバッガを停止させるか問い合わせます。
	表示をクリアします。
	環境設定ダイアログを表示し、モニタリングのサンプリング時間を設定できます (「 2.4.11 環境設定 」 を参照) 。

< 注意事項 >

モニタリング機能はモニタデバッガでは使用できません。
また、DSU3 を使用している場合もモニタリング機能は使用できません。

2.4.11 環境設定

環境設定について説明します。

■ 環境設定ダイアログ

環境設定ダイアログは、サンプリング時間設定、デッドライン設定を行うダイアログです。リアルタイム統計のサンプリング時間、デッドラインを設定するための条件は以下のとおりです。

- タスク解析モジュールに type5 を使用している。
- ログを「統計ログ - 時間測定」で設定している。

モニタリングのサンプリング時間は条件によらずに設定できます。

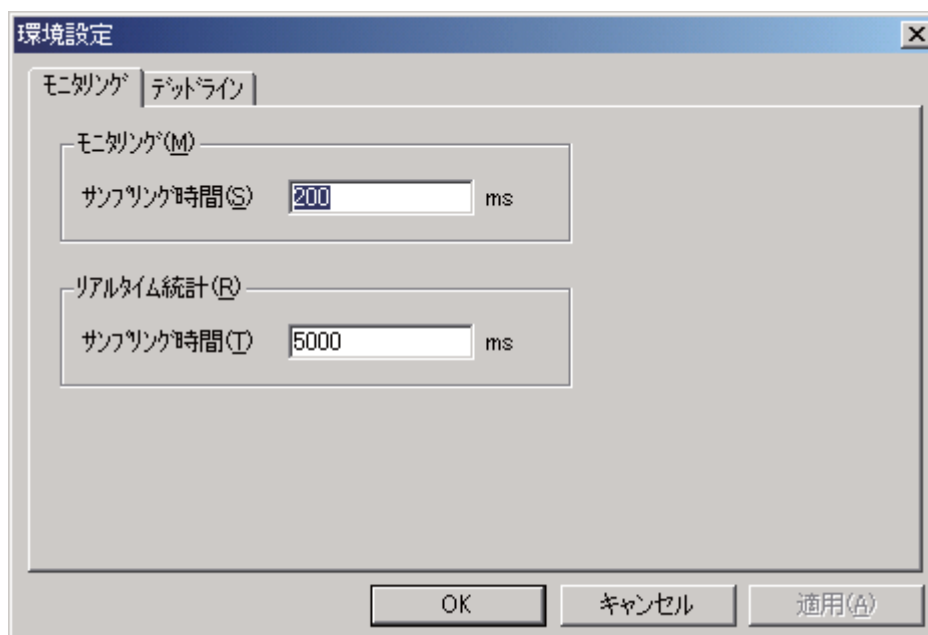
■ 環境設定ダイアログを開く

データ表示ウィンドウ内にある「環境設定」ボタンを押すと環境設定ダイアログが開きます。

■ モニタリング設定ダイアログを開く

環境設定ダイアログのモニタリングタブを押すとモニタリング間隔設定ダイアログが開きます。

図 2.4-9 モニタリング間隔設定ダイアログ (R_D_dbgA5.lib 使用時)



■ モニタリング間隔設定ダイアログの操作

- モニタリング - サンプル時間
モニタリングウィンドウのデータ更新間隔を入力しOK, または適用ボタンを押します。
設定範囲は 200ms ~ 32767ms です。
初期設定は 200ms です。
- リアルタイム統計 - サンプル時間
リアルタイム統計ウィンドウのデータ更新間隔を入力し, OKまたは適用ボタンを押します。
設定範囲は 200ms ~ 32767ms です。
初期設定は 5000ms です。

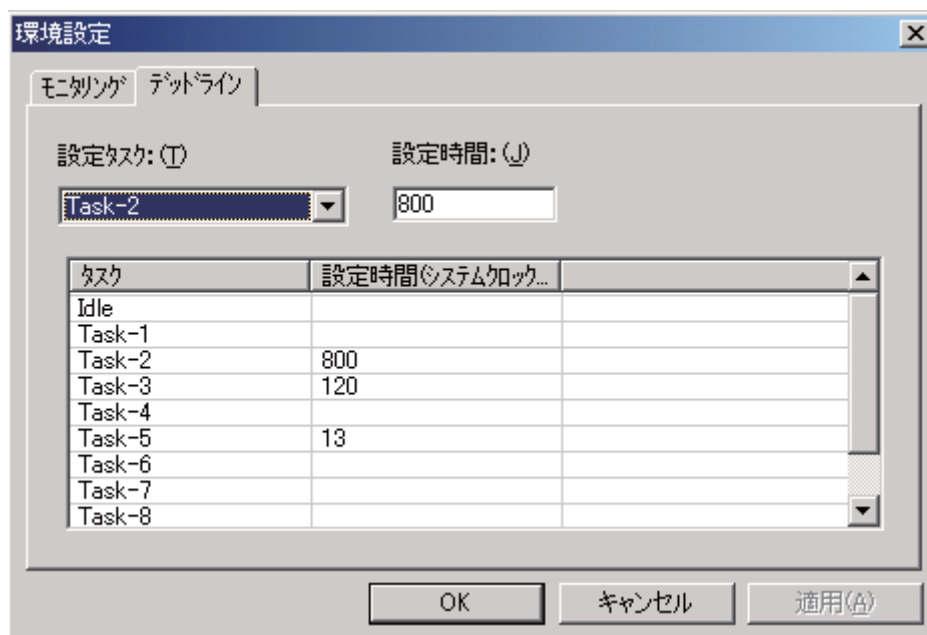
< 注意事項 >

リアルタイム統計ウィンドウはタスク数が増加した場合, 設定したサンプル時間が短すぎると各タスクの統計情報が表示できない場合があります。

■ デッドライン設定ダイアログを開く

環境設定ダイアログのデッドラインタブを押すとデッドライン設定ダイアログが開きます。

図 2.4-10 デッドライン設定ダイアログ



■ デッドライン設定ダイアログの操作

デッドラインを設定するタスクを選択します。

タスクの実行時間の上限となる制限時間を入力します。

設定時間は 1ms 単位で 1 ~ 65535 まで指定できます。

OK または適用ボタンにより設定します。

設定された内容は設定済みデッドラインリストに表示します。

■ 設定の消去

設定済みデッドラインリストから設定を消去するタスクを選択します。

右クリックのポップアップメニューから「設定消去」を選択します。

設定済みデッドラインリストから設定が消去されていることを確認します。

2.4.12 リアルタイム統計の説明

リアルタイム統計ウィンドウについて説明します。

■ リアルタイム統計

リアルタイム統計ウィンドウはタスク解析モジュール内で測定した結果をグラフ表示します。

タスクごとの割合、実行時間合計、平均実行時間、最長実行時間、最短実行時間、処理回数、デッドラインオーバを表示し、結果を一定間隔でモニタリングできます。

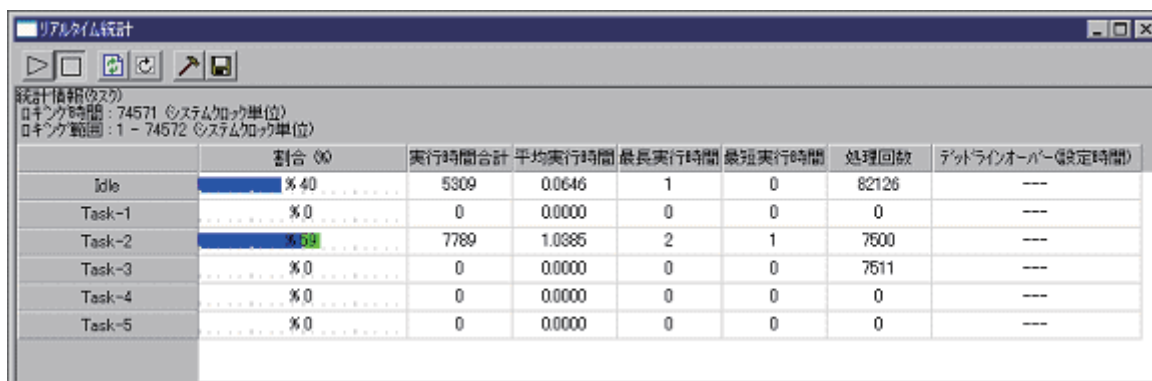
■ リアルタイム統計ウィンドウを開く

以下の条件を満足している場合、リアルタイム統計ウィンドウを開くことが可能です。

- ・ タスク解析モジュールに type5 を使用している。
- ・ ログを「統計ログ - 時間測定」で設定している。

データ表示ウィンドウ内にある[リアルタイム統計]ボタンを押すとウィンドウが開きます。

図 2.4-11 リアルタイム統計ウィンドウ



■ リアルタイム統計ウィンドウの表示項目

リアルタイム統計ウィンドウの表示項目は、以下のとおりです。

表 2.4-10 リアルタイム統計ウィンドウの表示項目







項目	説明
割合 [%]	ロギング時間全体に対する各タスクの実行時間の割合
実行時間合計	各タスクの実行時間の合計
平均実行時間	各タスクの実行時間の平均
最長実行時間	各タスクの最長実行時間
最短実行時間	各タスクの最短実行時間
処理回数	各タスクの処理回数
デッドラインオーバ (設定時間)	各タスクの制限時間を超えた回数 (設定時間)

■ リアルタイム統計ウィンドウの操作 (ツールバー)

リアルタイム統計ウィンドウの操作はツールバーの各ボタンで行います。

各ボタンに割り当てられた機能を以下の表に示します。

表 2.4-11 リアルタイム統計ウィンドウの操作

ボタン	機能説明
	モニタリングを開始します。 クリック時に、デバッガが停止している場合は、デバッガを実行させてモニタリングを開始するか問い合わせます。
	モニタリングを停止します。 クリック時に、デバッガが実行している場合は、デバッガを停止させるか問い合わせます。デバッガを停止させない場合は、表示の更新はせず情報を蓄積します。
	モニタリングを停止中に、ボタンを押すと表示を更新します。
	表示をクリアします。 蓄積されている情報をクリアします。
	環境設定ダイアログを開きます。
	表示している情報のログを保存します。 csv 形式で保存します。

2.4.13 ログファイルの出力

取得したログデータをファイルに出力できます。ファイルの拡張子は (*.log) です。ここでは、ログファイルの出力方法について説明します。

■ ログファイルの出力

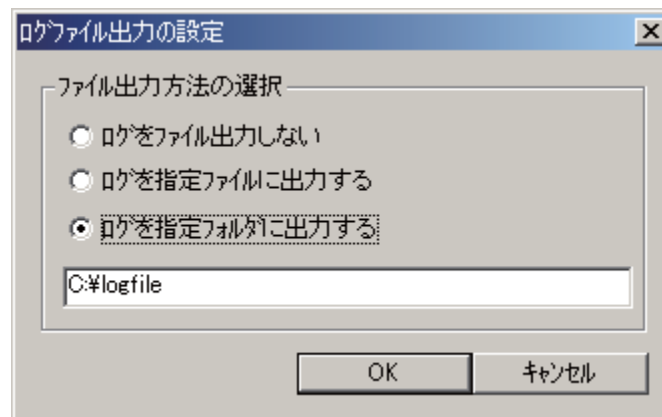
取得したログデータをテキスト形式のファイルに出力できます。また、製品ディスクに添付された「Log Viewer」で見ることができます。

データファイルの拡張子は (*.log) になります。

ファイル出力の設定を行うには、ログ設定選択リスト（設定選択ウィンドウ）で、右クリックすることでポップアップメニューが開きます。

ポップアップメニューの、「ログファイルの設定 ...」をクリックすると以下の「ログファイル出力の設定」ダイアログが開きます。

図 2.4-12 ログファイル出力設定ダイアログ



■ 出力方法の選択

出力方法には以下の3種類から選択できます。

- ログをファイル出力しない
ログファイルの出力を行いません。
- ログを指定ファイルに出力する
エディットボックス入力したファイルに出力します。データ取得のたびに内容を更新します。最新のデータだけ必要な場合に有効です。「Log Viewer」で見える場合は、ファイルをドラッグアンドドロップすることで見ることができます。
エディットボックス入力例 :C:\log¥logfile.log
- ログを指定フォルダに出力する
エディットボックス入力したフォルダにファイルを出力します。日時データをファイル名にして保存します。データ取得のたびにファイルを作成します。データを収集しておく場合に有効です。「Log Viewer」で見える場合は、単に起動するだけでレジストリ情報を読み出し、データリストの形で見るデータを選択できます。

2.5 システムコール発行

システムコール発行機能は、ブレイク中に μ T-REALOS アナライザからシステムコールを発行する機能です。この項ではシステムコール発行のウィンドウや操作方法について説明します。

2.5.1 システムコール発行について

2.5.2 ウィンドウの説明 (システムコール発行)

2.5.3 システムコールを発行する

2.5.1 システムコール発行について

システムコール発行機能の仕様について説明します。

■ 使用上の注意

本機能を使用する場合は、タスク部からだけ実行してください。

■ call コマンド

μT-REALOS アナライザのシステムコール発行は、デバッガの call コマンドを利用しています。よってデバッガの call コマンドの制限が、本機能の制限になりますので「Workbench コマンドリファレンス」を併せて一読されることをお勧めします。

■ 発行可能なシステムコール一覧

発行可能なシステムコールは以下の表 2.5-1 を参照してください。

表 2.5-1 発行可能なシステムコール一覧表

システムコール機能分類	発行可能なシステムコール
タスク管理機能	tk_cre_tsk, tk_del_tsk, tk_sta_tsk, tk_ter_tsk, tk_chg_pri, tk_ref_tsk
タスク付属同期機能	tk_wup_tsk, tk_can_wup, tk_rel_wai, tk_sus_tsk, tk_rsm_tsk, tk_frsm_tsk
同期・通信機能	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
拡張同期・通信機能	tk_cre_mtx, tk_del_mtx, tk_ref_mtx, tk_cre_mbf, tk_del_mbf, tk_ref_mbf, tk_ref_por
メモリプール管理機能	なし
時間管理機能	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
割込み管理機能	tk_def_int
システム状態管理機能	tk_rot_rdq, tk_get_tid, tk_ref_sys, tk_ref_ver
サブシステム機能	tk_def_ssy, tk_ref_ssy
デバイス管理機能	なし

μT-REALOS アナライザのシステムコール発行は、非タスク部からの発行ができないシステムコールは発行できません。

例)

tk_ext_tsk など自タスクの状態を変更するようなシステムコール

■ リンクされていないシステムコールについて

μT-REALOS では設計上、使用していないシステムコールはリンクされません。
μT-REALOS アナライザでは、ターゲット上にあるエントリを call する仕様であるため、
リンクされていないシステムコールの発行はできません。

■ ポインタ型の引数について

ポインタ型の引数は、その実体がターゲット上に確保されていることが必要となります。
デバッガおよび μT-REALOS アナライザ側で確保する仕様ではありません。

< 注意事項 >

システムコール発行機能の制限については、「付録 D システムコール発行時の制限について」を参照してください。

2.5.2 ウィンドウの説明 (システムコール発行)

システムコール発行のウィンドウについて説明します。

■ システムコール発行の開き方

機能選択ボタンで [Setting] を選択し, [System Call] で開きます。

■ システムコール発行機能選択時の各ペインの機能

[System Call] 選択時, 各ペインには以下の機能を割り当てます。

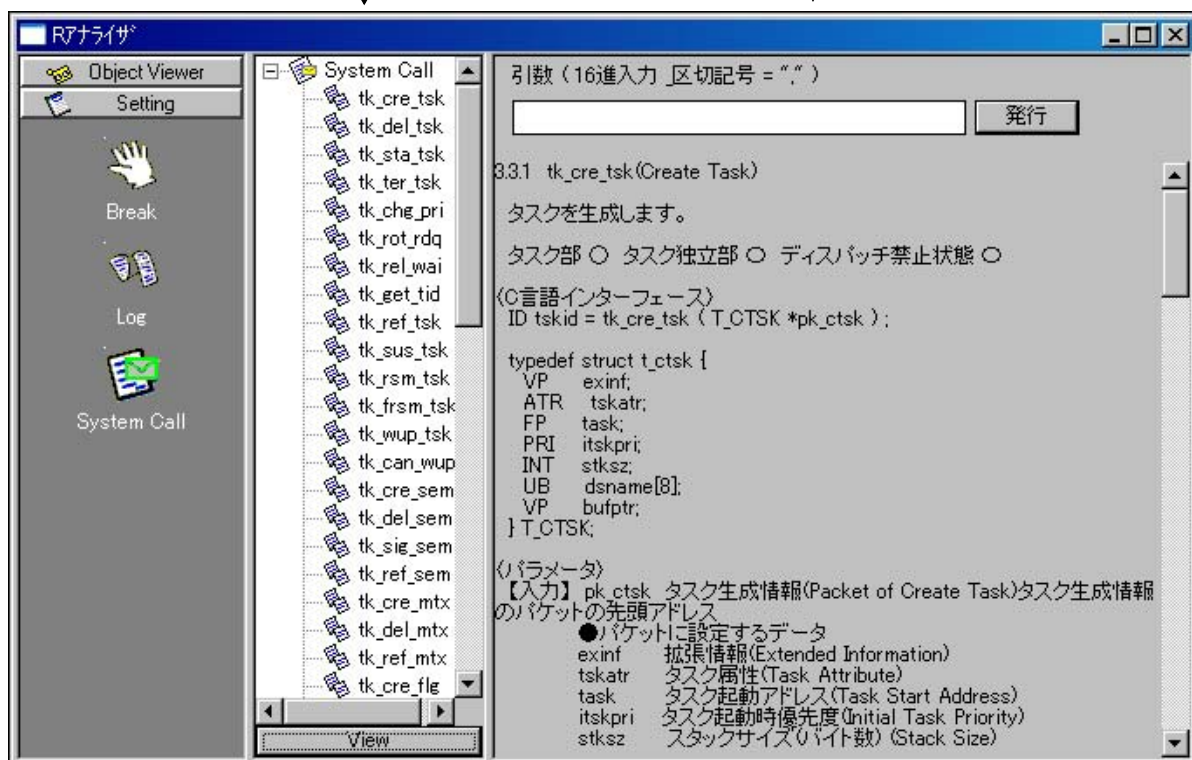
割当てられる機能は左のペインから以下ようになります。

- [System Call] 固定
- システムコール選択ウィンドウ
- 引数設定ウィンドウ

図 2.5-1 システムコール発行ウィンドウ

システムコール選択

引数設定ウィンドウ



■ システムコール選択リスト (システムコール選択ウィンドウ)

発行可能なシステムコールを表示します。発行するシステムコールを選択して [View] ボタンを押すことで引数設定ウィンドウに選択したシステムコールの仕様を表示します。

■ システムコール仕様表示部 (引数設定ウィンドウ)

選択したシステムコールの仕様を表示します。

■ 引数入力ボックス (引数設定ウィンドウ)

引数は, " で区切って入力してください。

■ [発行] ボタン (引数設定ウィンドウ)

各コントロールで設定された値でシステムコールを発行します。復帰値はダイアログで表示します。

2.5.3 システムコールを発行する

システムコール発行の手順について説明します。

■ 発行方法

システムコール選択ウィンドウで、発行する「システムコール」を選択します。

[View] ボタンで設定を決定します。

引数を設定します。

[発行] ボタンを押します。

復帰値をダイアログで表示します。

2.6 スタック情報

スタック情報は、各タスクのスタック使用状況をグラフで表示します。スタック情報ウィンドウは、[R アナライザ] メニューの [スタック情報] により、開くことができます。また、スタック最大使用量を表示するには、[R アナライザ] メニューの [スタック使用量解析] から [開始] を選択します。スタック最大使用量を非表示にするには、[終了] を選択します。

■ スタック情報ウィンドウを開く

Workbench メニュー [R アナライザ] - [スタック情報] により、「スタック情報ウィンドウ」が開きます。

開く際に情報収集を行うため、登録タスク × 1 秒程度の時間がかかります。

以下、スタック情報ウィンドウの表示例です。

図 2.6-1 スタック情報ウィンドウ



■ 用語説明

スタック情報ウィンドウで使われている用語の説明は以下のとおりです。

用語	説明
使用量	現在, 使用しているスタックの量
最大値	スタック使用量解析を開始してから現在までに最も使用したスタックの量
未使用	使用していないスタックの量
領域	スタック先頭アドレス - スタック末尾アドレス
スタックサイズ	スタック領域の大きさ
ID	各タスクの ID(s はシステムスタック, i はアイドルスタック)
グラフ	使用量, 最大値, 未使用の関係を示す円グラフ
C	円グラフを構成する色 使用量: 赤 最大値: 黄 未使用: 灰
全体	各タスク, システムスタック, アイドルスタックのスタック情報の合計

■ [更新] ボタン

情報を更新するボタンです。カーネル実行中は情報を保証しません。

最初に開いたときは情報取得を行ってから開きます。それ以降は, [更新] ボタンを押して最新の情報取得を行ってください。

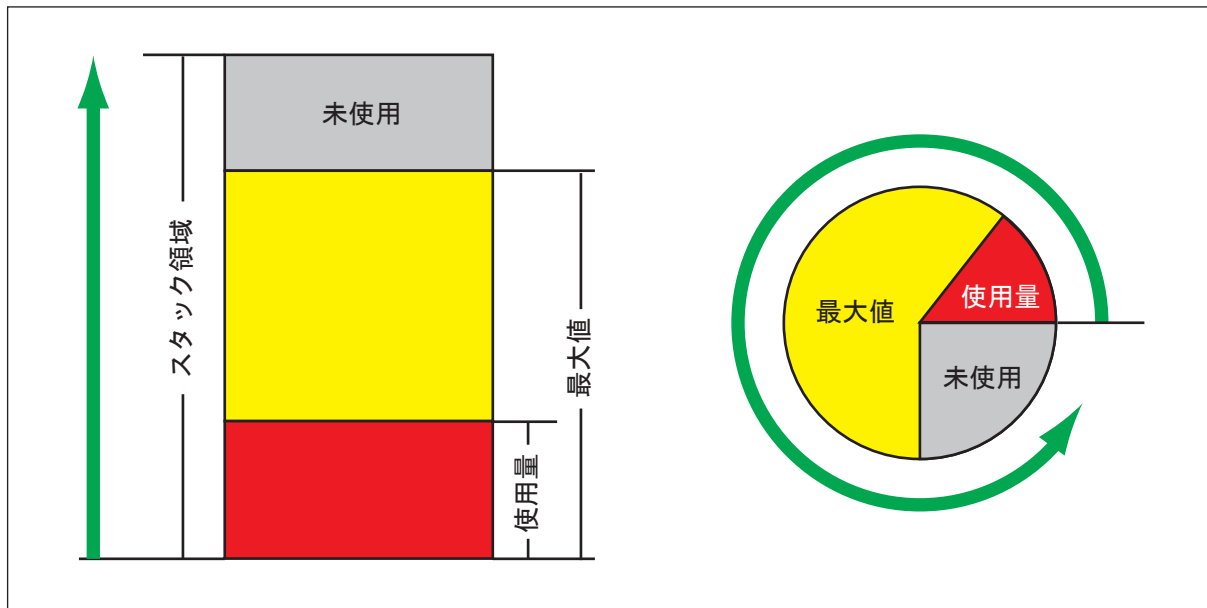
■ グラフ

表示するグラフは上段と下段の 2 種類あります (図 2.6-1 参照)。

- 上段のグラフ : システム全体のスタック領域に関するグラフです。
システムスタック , アイドルタスク , 全タスクのスタックの使用サイズおよび未使用サイズを示します。
データ取得時に tk_cre_tsk されていないタスクについては , 加算しません。
- 下段のグラフ : タスクごとのスタックに関するグラフです。
タスク ID ごとのスタックの使用サイズおよび未使用サイズを示します。
データ取得時に tk_cre_tsk されていないタスクについては , 表示しません。

グラフは各スタック領域を 3 時方向から反時計回りに表示しています (図 2.6-2 参照)。

図 2.6-2 円グラフ表示



■ スタック使用量解析

現在までに最も使用したスタックの量を表示できます。

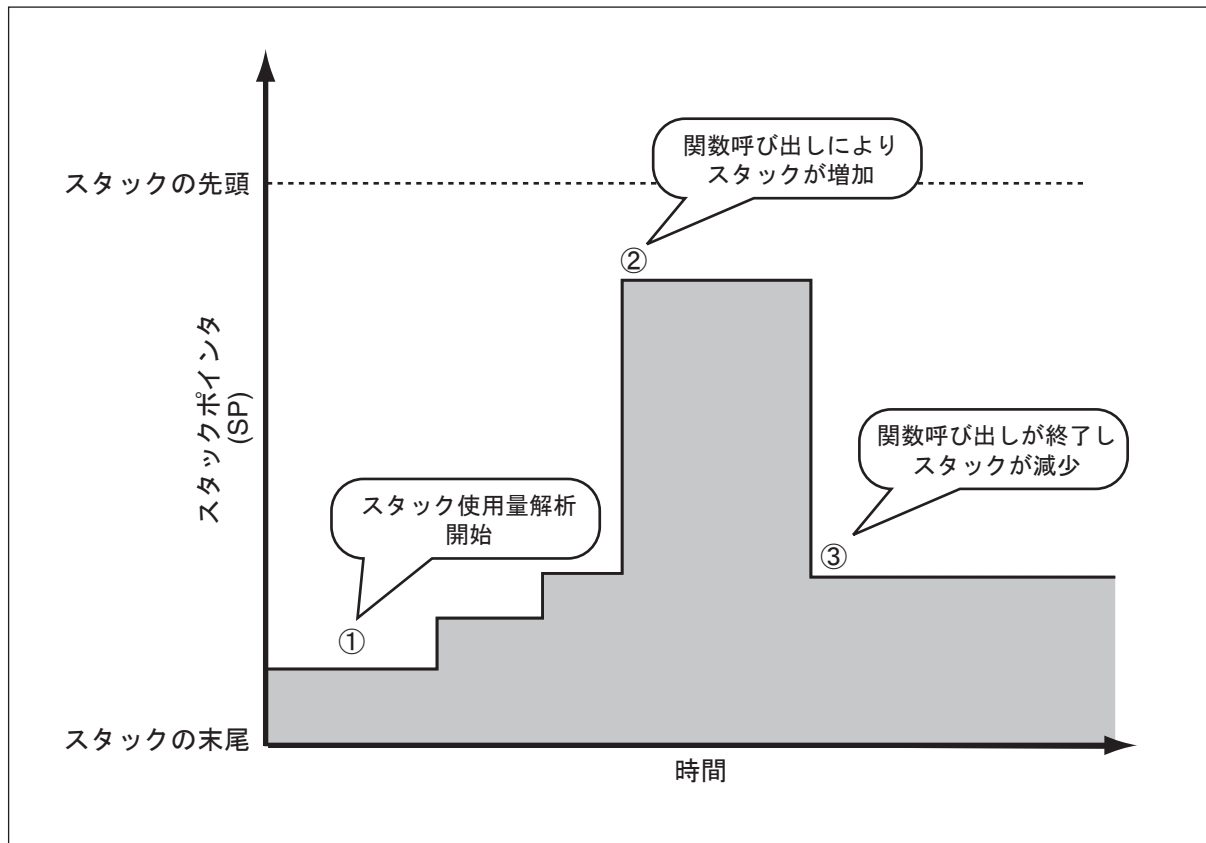
● メニュー [R アナライザ]-[スタック使用量解析]-[開始]

未使用領域, スタック領域から [最大値] を計測します。

[最大値] の計測方法を以下に示します。

例)

図 2.6-3 スタックポインタ - 時間グラフ



スタックポインタと時間の関係が図 2.6-3 のようになっている場合、図中の ~ はそれぞれ以下の状態を示します。

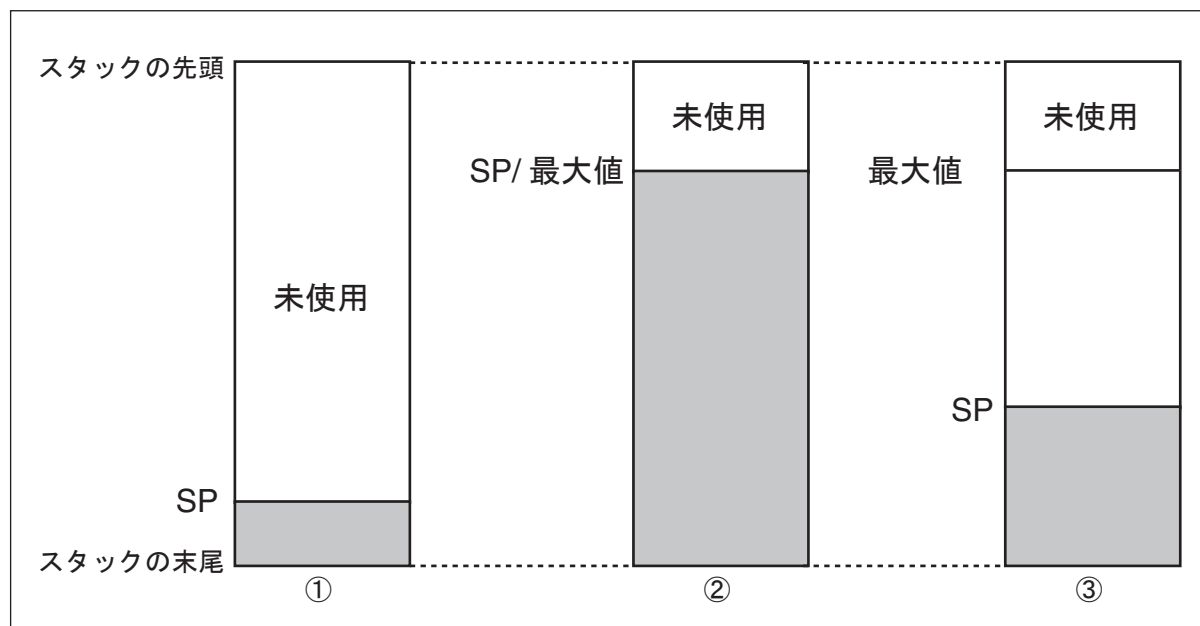
スタック使用量解析を開始

関数呼出しによりスタック使用量が増加

関数呼出しが終了しスタック使用量が減少

図中の ~ の場合のスタック領域内はそれぞれ図 2.6-4 のようになります。

図 2.6-4 ~ のスタック領域



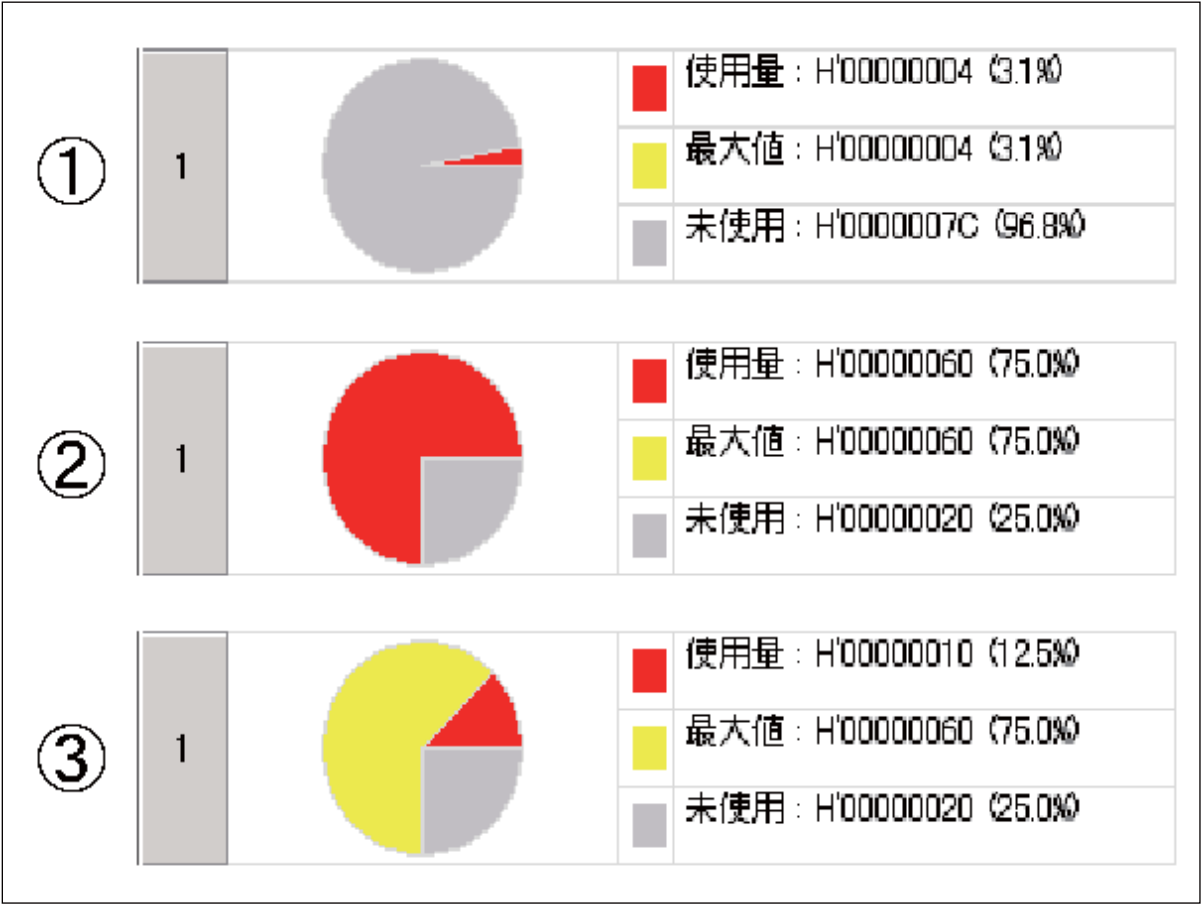
スタック末尾から SP(スタックポインタ) の値までが [使用量] となり , [最大值] は使用量と同値を示します。

スタック末尾から SP までが [使用量] になります。 [使用量] = [最大值] になります。

スタック末尾から SP までが [使用量] になります。 で達した値が [最大值] となります。

実際の円グラフではそれぞれ図 2.6-5 のように表示します。

図 2.6-5 ~ の実際の表示



● メニュー [R アナライザ]-[スタック使用量解析]-[終了]

[最大値] を非表示にします。

< 注意事項 >

スタック使用量解析は μ T-REALOS の初期化が終了した後に開始してください。またスタックがオーバーフローすることがないようにスタック領域は十分な大きさを確保してください(「付録 E スタック使用量解析時の注意事項について」参照)。

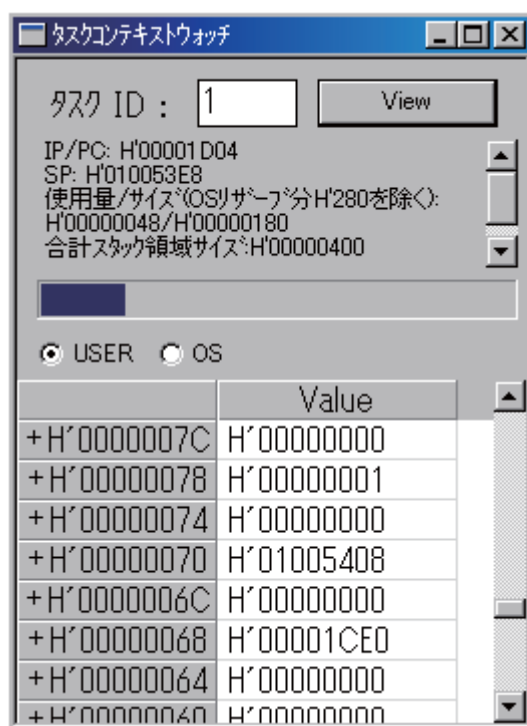
2.7 タスクコンテキストウォッチ

タスクコンテキストウォッチは、各タスクのコンテキストの内容を表示するものです。タスクコンテキストウォッチは、[R アナライザ] メニューの [タスクコンテキストウォッチ] により、開くことができます。

■ タスクコンテキストウォッチを開く

Workbench メニュー [R アナライザ] - [タスクコンテキストウォッチ] により、「タスクコンテキストウォッチ」が開きます。

図 2.7-1 タスクコンテキストウォッチ



■ 指定タスクのコンテキストをウォッチする

タスクコンテキストを表示するタスク ID を指定してください。

[View] ボタンにより、以下の内容を表示します。

- IP/PC : 指定タスクの現在の IP/PC
- SP : 指定タスクの現在の SP
- 使用量 / サイズ : 現在の SP から計算した使用量 / スタックサイズ
- バー : 上記の値から % を計算し、上記の表示の下に使用量のイメージを表示します。
- コンテキストリスト : OS のリザーブ領域内のデータとユーザスタックの内容を切り換えて表示します。

第3章

タスク解析モジュール

この章では、モジュール使用ログによるサンプリングで使用する「タスク解析モジュール」について説明します。

3.1 タスク解析モジュールの種類

3.2 コメントの使用方法

3.3 カスタマイズの方法

3.1 タスク解析モジュールの種類

タスク解析モジュールの種類について説明します。

■ タスク解析モジュール

タスク解析モジュールは、 μ T-REALOS カーネルプログラムのタスクディスパッチやシステムコール発行のタイミングに処理を追加して、RTOS 動作ログの取得を実現するプログラムです。

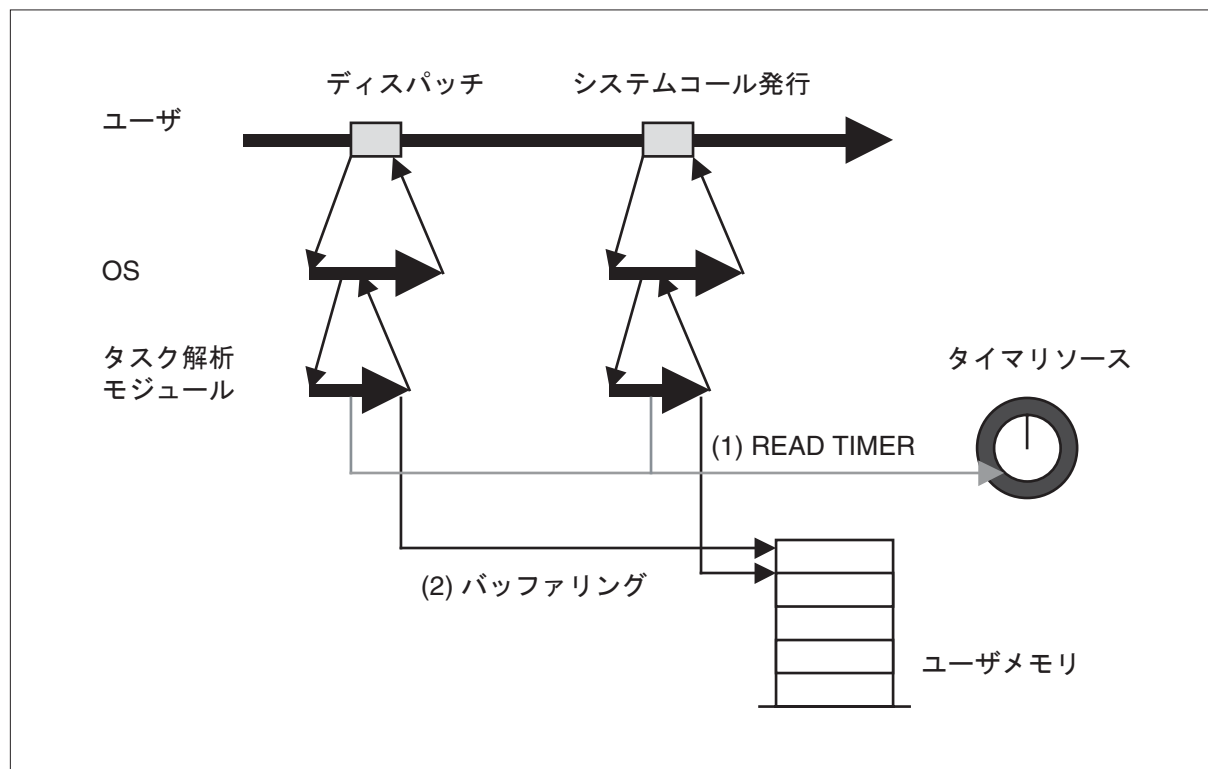
ログ機能の「モジュール使用ログ」を実現するために必要となります。

μ T-REALOS アナライザが提供するタスク解析モジュールは、3 種類です。

また、タイムスタンプ付の RTOS ログを実現するため、 μ T-REALOS のシステムクロックをタイムスタンプに使用します。

以下にタスク解析モジュールの動作概要を示します。

図 3.1-1 タスク解析モジュール動作概要



■ 使用するタスク解析モジュールの選択

コンフィギュレータの「デバッグ定義」で選択できます。デフォルトの設定は、カーネルが提供する、何も処理を行わないプログラムです。

■ タイプごとの機能

選択するタイプによって、サンプリングする際に付加するタイムスタンプの精度が異なります。精度の高いタイムスタンプを使用する場合、デメリットとしてオーバーヘッドは大きくなります。以下、各タイプごとのタイムスタンプの精度を示します。

表 3.1-1 タイプ別タイムスタンプ精度

タイプ名	ライブラリ名	説明
type1	R_D_dbgA1.lib	タイムスタンプを付加しないタイプです。
type2	R_D_dbgA2.lib	システムクロック時間を付加するタイプです。
type3	R_D_dbgA3.lib	リソースタイマのカウント値を付加するタイプです。
type5	R_D_dbgA5.lib	統計ログ用のタイプです。

■ バッファサイズ、タスク数カスタマイズ用ファイル

タスク解析モジュールを使用している場合、R_D_siz.c はバッファ、タスク数カスタマイズ用のファイルです。バッファサイズのデフォルトは 128 に設定されています。タスク数のデフォルトは 1 に設定されています。カスタマイズの詳細は「3.3 カスタマイズの方法」を参照してください。

■ リソースタイマカスタマイズ用ファイル

SOFTUNE μ T-REALOS/FR をご使用の場合、R_D_cnt.c はリロードタイマカスタマイズ用のファイルです。デフォルトは 12500 に設定されています。カスタマイズの詳細は「3.3 カスタマイズの方法」を参照してください。

3.2 コメントの使用方法

モジュール使用ログにおいて、コメントイベントを使用する方法について説明します。

■ コメントイベント

コメントイベントはモジュール使用ログ設定時に利用可能です。

通常のイベントは μ T-REALOS 内部で発生する、 μ T-REALOS 依存のふるまいだけをサンプリングします。

それに対し、コメントイベントはユーザが任意の個所へコメントイベント用のフックを記述していた場合に発生します。ただし、タスク解析モジュールを設定していないときは、コメントイベントを使用できません。

■ インクルードが必要なヘッダーファイル

μ T-REALOS インストールディレクトリ下にある `utkernel¥xxx¥dbg` 内の `"R_D_custom.h"` をコメントとしてイベント登録したいソース中でインクルードしてください。

■ コメントイベント用フックのC言語用インタフェース

以下にコメントイベント用フックのC言語用インタフェースを示します。

C 言語インタフェース	<code>void _R_D_func_Custom(ID id, char* strbuf)</code>
実装位置	ユーザ任意
引数	ID id: ユーザ任意の番号 char* strbuf: コメント文字列のバッファ先頭アドレス (20 文字まで)

■ コメントイベントの記述方法

以下のように記述することで、コメントイベントが発生し、ログの設定にコメントを選択している場合、サンプリングします。

```
#include "R_D_custom.h"

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

3.3 カスタマイズの方法

バッファのサイズやタスク数，タイマリソースを変更する場合には，タスク解析モジュールのカスタマイズ用ファイルの中身を変更する必要があります。

■ バッファサイズの変更

R_D_siz.c を以下の方法でカスタマイズして再ビルドしてください。

バッファサイズを定義している「TRC_DATA_NUM」の値を「0 ~ 2048」に書き換えることができます。デフォルトは 128 です。タスク解析モジュールに type5 を使用している場合は，0 に設定してください。

```

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

```

■ タスク数の変更

統計ログやディスパッチブレークを使用するためにタスク数のバッファサイズを変更する必要があります。バッファサイズは動的生成と静的生成の両方を合わせたタスク数分を必ず用意してください。

R_D_siz.c を以下の方法でカスタマイズして再ビルドしてください。

バッファサイズを定義している「USER_TASK_NUM」の値を「1 ~ 32767」の範囲で設定してください。デフォルトは 1 です。

```

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

```

■ タイマリソースの変更

R_D_cnt.c をカスタマイズして再ビルドしてください。

リロードタイマ値を定義している「R_D_RELOAD_TIMER」の値を書き換えることができます。

デフォルトは「12500」です。

このタイマ設定値は，μT-REALOS のサンプルシステム使用時に適応するように，作成されています。

```

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

```


付録では、アナライザの処理時間やエラーメッセージ、制限事項を掲載します。

付録 A 情報取得の速度について

付録 B タスク解析モジュールのシンボル / セクション

付録 C エラーメッセージ

付録 D システムコール発行時の制限について

付録 E スタック使用量解析時の注意事項について

付録 F デバッガ種別による制限について

付録 A 情報取得の速度について

情報取得の速度について説明します。

■ ICE での各情報取得時における測定結果

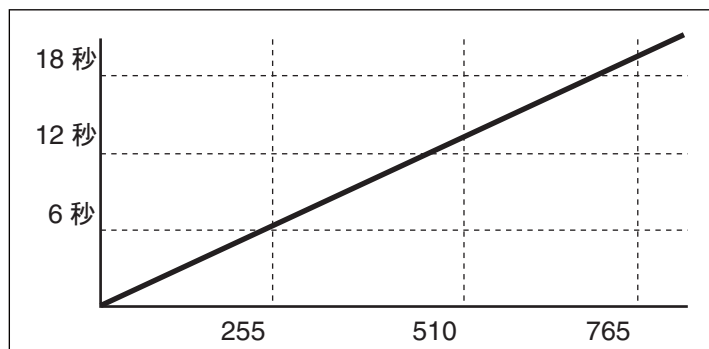
以下、各情報取得にかかる時間を記載します。相対的に各オブジェクトの個数が増えると取得時間は長くなります。また、生成されていないオブジェクトについても、生成情報をチェックするためにアクセスを行います。

例) タスク 10 個のうち 1 つしか生成されていない場合でも 10 個分の時間がかかります。

【DSU3 の場合】

- タスク 255 個分の情報取得に要する時間 : 8 秒
- セマフォ 255 個分の情報取得に要する時間 : 3 秒
- タスク 255 個を 1 つのイベントフラグで待たせた場合の情報取得に要する時間 : 2 秒
- レディキュー情報取得に要する時間 : 2 秒

付図 A-1 タスク情報取得所要時間計算グラフ



付録 B タスク解析モジュールのシンボル / セクション

μT-REALOS アナライザがログを取得する際に使用しているモジュール「タスク解析モジュール」で使用するシンボルやセクションについて説明します。

モジュール	セクション	シンボル
R_D_dbgA1.c	_kernel_dbginit_sc	R_D_ptrctop R_D_trctop2 R_D_trcnt 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_trcnt 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

(続く)

付録

(続き)

モジュール	セクション	シンボル
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

付録 C エラーメッセージ

μT-REALOS アナライザのエラーメッセージについて説明します。

■ エラーメッセージ (E48xxM)

E4801M	指定された ID は無効でした。
[内容]	μT-REALOS アナライザに要求した ID は無効です。
[対処]	要求した ID は無効です。正しい ID を入力後、再実行してください。

E4802M	不正なメモリ領域に対するアクセスを行いました。
[内容]	アクセスできないアドレスへの READ/WRITE が指定されました。
[対処]	1.OS の管理領域が配置されている領域などが、アクセスできない状態にあります。 アクセスが可能になるまで、μT-REALOS アナライザの機能を使用しないでください。 2.OS の管理情報が破壊されているため、解析処理に失敗しました。

E4803M	指定された ID が範囲外でした。
[内容]	指定された ID が範囲外か 0 が指定されました。
[対処]	ID の指定を範囲に収まるよう、入力しなおしてください。

E4804M	ターゲット上のメモリが不足のため、実行できませんでした。
[内容]	デバッガにおいてメモリ不足のため、処理が実行できませんでした。
[対処]	パソコンに実装されているメモリを増設するか、現在起動中のほかのアプリケーションを終了してください。

E4807M	復帰できないエラーが発生しました。
[内容]	ターゲットにアクセスした際に、デバッガが復帰不能になりました。
[対処]	デバッガを一度終了する可能性があります。

E4808M	サポートしていない機能です。
[内容]	指定された機能はサポートされていません。
[対処]	一部の機能は、本アナライザではサポートされていません。

E4809M	不正なパラメータです。
[内容]	指定されたパラメータが不正です。
[対処]	正しいパラメータを入力しなおしてください。

E4810M	指定された ID は無効です。
[内容]	指定された ID は無効です。
[対処]	正しい ID を入力後、再実行してください。

E4811M	ホスト上のメモリが不足のため、実行できませんでした。
[内容]	プログラム実行のためのメモリが不足しているため、実行できませんでした。
[対処]	パソコンに実装されているメモリを増設するか、現在起動中のほかのアプリケーションを終了してください。

E4812M	ID を割り当てられませんでした。
[内容]	ID が不足しているため、ID を割り当てるできませんでした。
[対処]	シミュレータ使用時の OS ブレーク設定の制限 255 を超えた場合に発生します。不要なブレークを削除して設定し直してください。

E4815M	要求された操作はすでに実施されています。
[内容]	システムコール発行において、すでにシステムコールを発行していることを示します。
[対処]	既に処理を行っているシステムコールが完了するまでは、次のシステムコールを発行しないでください。

E4816M	再実行は可能ですが、何らかの原因によりエラーが発生しました。
[内容]	アナライザの機能実行時にエラーが発生しました。
[対処]	再実行は可能ですが、現象が特定できない場合に出力します。頻繁に発生する場合は弊社までご連絡ください。

■ 警告メッセージ

	タスク ID = x のスタックが 8Byte 境界に配置されていません。
[内容]	記載された ID のスタックが 8Byte 境界に配置されていません。
[対処]	スタックが正しい境界上に配置されていないため、 μ T-REALOS が正常動作しない可能性があります。記載された ID のスタックを 8Byte 境界に配置してください。

	モジュールのフラグを初期化します。
[内容]	タスク解析モジュールのフラグ (変数) を初期化します。
[対処]	リセットする場合など、タスク解析モジュールのフラグ (変数) を初期化したい場合に確認を行います。この際、現在取得中のログはクリアします。

■ 致命的エラー (F95xxM)

F9503M	必要なファイルが見つかりませんでした。インストールが正常におこなわれていない可能性があります。
[内容]	μT-REALOS アナライザのインストール情報が不正であるため、機能を実行できません。
[対処]	以下の可能性があります。 1. μT-REALOS アナライザのインストール情報が破損しています。 2. 正常にインストールされていません。 3. インストール先からファイルが移動しています。 3 の場合は移動したファイルを元に戻してください。それ以外の場合は、再度インストールを行ってください。

F9504M	RTOS I/F Module が見つかりませんでした。インストールが正常に行われていない可能性があります。
[内容]	μT-REALOS 用の I/F モジュールが見つかりませんでした。μT-REALOS アナライザは起動できません。
[対処]	以下の可能性があります。 1. μT-REALOS アナライザのインストール情報が破損しています。 2. 正常にインストールされていません。 3. インストール先からファイルが移動しています。 3 の場合は移動したファイルを元に戻してください。それ以外の場合は、再度インストールを行ってください。

F9505M	ターゲットが暴走しているかデバッガに問題が発生している可能性があります。ターゲットまたはデバッガに問題はないか確認してください。
[内容]	情報取得前のメモリチェックでエラーが発生しました。
[対処]	以下の可能性があります。 1. メモリが読めない状態にあります。 2. ICE またはターゲットが暴走しています。 アナライザを終了し、デバッガを再起動してください。

F9506M	出力ファイルが開けませんでした。
[内容]	出力ファイルの作成または書き出しに失敗しました。
[対処]	ファイルを指定し直すか、指定フォルダまたはファイルを書込み可能な状態にしてください。

付録 D システムコール発行時の制限について

システムコール発行時に生じる制限事項について記述します。

■ システムコール発行機能の仕様と制限事項

システムコール発行（以下、API 発行）機能はデバッガの call コマンドにより実現しています。よって、デバッガの call コマンドと μ T-REALOS アナライザの API 発行機能の制限事項は同じです。以下に制限事項を示します。

- call コマンド実行中は、デバッガを abort コマンドで停止させても、すべてのステップ実行を行うことができません。
- call コマンド実行中は、デバッガを abort コマンドで停止させても、call コマンドを多重に動作させることはできません。
- 非タスクコンテキストから発行できないシステムコールは、本機能により発行することはできません。
- 使用するシステムコールのアドレスが物理的に解決している必要があります（システムのコンフィギュレーションでリンクされたシステムコールだけが発行可能です）。

■ システムコール発行機能使用時の注意事項

call コマンドは、指定された関数発行後、カレント PC に戻ってきた時点で処理が完了します。

μ T-REALOS のシステムコール発行の場合、各システムコール内においてタスクを切り換える可能性があります。切り換えが発生した場合は、システムコールの処理は完了しますが、タスクが切り換わることでカレント PC に復帰しないため、 μ T-REALOS の SVC 発行機能は完了しません。

例)

μ T-REALOS のシステムコール発行機能により、現在のタスクより優先度の高いタスクを起動した場合など

このような場合、上に記載した制限事項が適用されます。

付録 E スタック使用量解析時の注意事項について

スタック使用量解析時の注意事項について記述します。

■ スタックオーバーフロー発生時の制限

スタック使用量解析は、各タスクに割り当てられたスタック領域内のデータ変化を監視しています。

よって、各タスクのスタック領域を越えた（オーバーフローした）場合は正しい使用量を表示できません。

■ 使用状況と表示内容

スタックポインタの移動とスタックポインタの指し示すアドレスのメモリの利用は必ずしも一致しません。

スタックポインタを移動して領域を確保しても使用されない場合があります。

スタック使用量解析はスタックポインタの移動を監視していないため、利用されずにポインタだけ移動するような場合でも、実際に使用された分だけが使用量として表示します。

付録 F デバッガ種別による制限について

デバッガ種別によって生じる制限事項について記述します。

■ デバッガ種別による機能制限

各デバッガ種別で使用可能な機能の一覧は以下のとおりです。

付表 F-1 デバッガ種別と使用可能な機能一覧

		シミュレータ	エミュレータ		モニタデバッガ
			DSU3	DSU4	
オブジェクト	オブジェクト表示				
ブレーク	実行ブレーク				
	アクセスブレーク		×		
	ディスパッチブレーク (ENTER/LEAVE)	*1	*1	*1	*1
	システムコールブレーク (ENTER/LEAVE)	*1	*1	*1	*1
ログ	ディスパッチログ		×		×
	モジュール使用ログ	*2	*2	*2	*2
	モニタリング		×		×
	統計ログ	*3	*3	*3	*3
タスクコンテキスト ウォッチ	タスクコンテキスト ウォッチ				
スタック情報	スタック情報				

：表示可

×：表示不可

*1：タスク解析モジュール Type1 ~ Type3, Type5 を設定している場合に使用可能

*2：タスク解析モジュール Type1 ~ Type3 を設定している場合に使用可能

*3：タスク解析モジュール Type5 を設定している場合に使用可能

索引

C

call	
call コマンド	71
C 言語	
コメントイベント用フックの C 言語用インタフェース	88

E

E48xxM	
エラーメッセージ (E48xxM)	95

F

F95xxM	
致命的エラー (F95xxM)	98

I

ICE	
ICE での各情報取得時における測定結果	92
ID	
ID・パスカウント・アドレス入力フォーム (設定ウィンドウ)	38
表示 ID 選択および表示実施ボタン ([View] ボタン)	25

M

μT-REALOS	
μT-REALOS アナライザの機能	6
μT-REALOS アナライザの特長	6

O

Object	
オブジェクト表示 (Object Viewer) の開き方	24
OS ブレーク	
OS ブレーク設定機能選択時の各ペインの機能	37
OS ブレーク設定の開き方	37
OS ブレークの種類	12
OS ブレークの設定	12

R

R アナライザ	
R アナライザウィンドウの各部名称	22
R アナライザウィンドウを開く	22

S

Setting	
[Setting] ボタン (ログ設定選択ウィンドウ)	47

V

View	
表示 ID 選択および表示実施ボタン ([View] ボタン)	25

あ

アイコン

遷移図表示ウィンドウの線とアイコン	57
ログデータの表示アイコンとパラメーター一覧	55

アドレス

ID・パスカウント・アドレス入力フォーム (設定ウィンドウ)	38
---	----

アナライザ

μT-REALOS アナライザの機能	6
μT-REALOS アナライザの特長	6

い

イベント

推奨するイベントの組み合わせ	50
選択可能なイベント	48, 49, 51

インクルード

インクルードが必要なヘッダーファイル	88
--------------------------	----

インストール

インストールの確認	7
-----------------	---

インタフェース

コメントイベント用フックの C 言語用インタ フェース	88
--------------------------------------	----

う

ウィンドウ

システムコール仕様表示部 (引数設定ウィンドウ)	74
スタック情報ウィンドウを開く	76
[発行] ボタン (引数設定ウィンドウ)	74
引数入力ボックス (引数設定ウィンドウ)	74
リアルタイム統計ウィンドウの操作 (ツールバー)	68
リアルタイム統計ウィンドウの表示項目	68
リアルタイム統計ウィンドウを開く	67

ウォッチ

指定タスクのコンテキストをウォッチする	83
---------------------------	----

え

エミュレータデバッグ

エミュレータデバッグ使用時におけ注意点	3
---------------------------	---

エラー

致命的エラー (F95xxM)	98
-----------------------	----

エラーメッセージ

エラーメッセージ (E48xxM)	95
-------------------------	----

お

オブジェクト

オブジェクト情報の表示手順	26
オブジェクト選択	24
オブジェクト表示	10
オブジェクト表示 (Object Viewer) の開き方	24
オブジェクト表示選択時の各ペインの機能	24

各オブジェクトの表示内容	27
各オブジェクトのリスト形式の列順を変更する 【カラムのドラッグアンドドロップ】	34
オブジェクトインデックスウィンドウ	56
オブジェクトインデックスウィンドウ	56
オブジェクトウィンドウ	11
オブジェクトウィンドウの標準機能	11

か

開発環境

開発環境における注意点	3
-------------------	---

カスタマイズ

高度な設定 (カスタマイズ設定)	50
バッファサイズ, タスク数カスタマイズ用ファイ ル	87

カラム

各オブジェクトのリスト形式の列順を変更する 【カラムのドラッグアンドドロップ】	34
--	----

間隔時間

環境設定 - モニタリング間隔時間設定	45
---------------------------	----

環境設定

環境設定ダイアログ	64
環境設定ダイアログを開く	64
環境設定 - デッドライン設定	45
環境設定 - モニタリング間隔時間設定	45

き

記述方法

コメントイベントの記述方法	88
---------------------	----

起動方法

起動方法	8
------------	---

く

組み合わせ

推奨するイベントの組み合わせ	50
----------------------	----

グラフ

グラフ	78
-----------	----

け

警告

警告メッセージ	97
---------------	----

こ

更新

[更新] ボタン	77
----------------	----

コマンド

call コマンド	71
-----------------	----

コメントイベント

コメントイベント	88
コメントイベントの記述方法	88
コメントイベント用フックの C 言語用インタ フェース	88

コンテキスト

指定タスクのコンテキストをウォッチする	83
---------------------------	----

さ

削除

削除	42
追加・変更・削除ボタン (設定ウィンドウ)	37

サンプリング

サンプリング手法	44
----------	----

し

システムコール

システムコール	40
システムコール仕様表示部 (引数設定ウィンドウ)	74
システムコール選択リスト (システムコール選択ウィンドウ)	74
システムコール発行	17
システムコール発行機能使用時の注意事項	99
システムコール発行機能選択時の各ペインの機能	73
システムコール発行機能の仕様と制限事項	99
システムコール発行の開き方	73
発行可能なシステムコール一覧	71
リンクされていないシステムコールについて	72

システムコールアイコン

システムコールアイコン上でのマウス操作	59
---------------------	----

出力方法

出力方法の選択	69
---------	----

取得

フラグクリア・取得・遷移図・統計グラフ・ モニタリングボタン (データ表示ウィンドウ)	47
--	----

消去

設定の消去	66
-------	----

使用状況

使用状況と表示内容	100
-----------	-----

情報表示

情報表示	25
------	----

使用量解析

スタック使用量解析	79
-----------	----

す

スタック

スタック情報	18
スタック情報ウィンドウを開く	76
スタック情報の表示内容	18
スタック使用量解析	79
スタックオーバーフロー	
スタックオーバーフロー発生時の制限	100

せ

制限事項

システムコール発行機能の仕様と制限事項	99
---------------------	----

設定

高度な設定 (カスタマイズ設定)	50
設定結果	41
設定の消去	66
設定方法	48, 50, 51

設定ウィンドウ

設定済のブレークを設定ウィンドウに反映させる	42
------------------------	----

設定ログ

設定ログリスト (データ表示ウィンドウ)	47
----------------------	----

遷移図

遷移図および統計グラフの参照	16
タスク遷移図の表示	44
フラグクリア・取得・遷移図・統計グラフ・ モニタリングボタン (データ表示ウィンドウ)	47

遷移図ウィンドウ

遷移図ウィンドウを開く	56
-------------	----

遷移図表示

遷移図表示ウィンドウの線とアイコン	57
-------------------	----

選択ウィンドウ

[Setting] ボタン (ログ設定選択ウィンドウ)	47
-----------------------------	----

そ

ソースジャンプ

ソースジャンプ機能 【右クリック [ソースジャンプ]】	35
--------------------------------	----

た

ダイアログ

環境設定ダイアログ	64
環境設定ダイアログを開く	64
デッドライン設定ダイアログの操作	66
デッドライン設定ダイアログを開く	65
モニタリング間隔設定ダイアログの操作	65
モニタリング設定ダイアログを開く	64

タイプ

タイプごとの機能	87
----------	----

タイマ

タイマリソースの変更	89
------------	----

タイマカスタマイズ

リソースタイマカスタマイズ用ファイル	87
--------------------	----

タイムスタンプ

タイムスタンプの付加	44
------------	----

タスク

指定タスクのコンテキストをウォッチする	83
---------------------	----

タスク数の変更	89
---------	----

タスク遷移図の表示	44
-----------	----

タスク解析

使用するタスク解析モジュールの選択	86
タスク解析モジュール	86

タスクコンテキストウォッチ

タスクコンテキストウォッチ	19
タスクコンテキストウォッチの表示内容	19
タスクコンテキストウォッチを開く	82

タスク数

バッファサイズ、タスク数カスタマイズ用ファイル	87
-------------------------	----

タスクブレイク	
対象タスクブレイク種別	39

ち

注意	
使用上の注意	71

つ

追加	
設定方法【[追加]ボタン】	41
追加・変更・削除ボタン(設定ウィンドウ)	37
ツールバー	
ツールバー	57
モニタリングの操作(ツールバー)	63
リアルタイム統計ウィンドウの操作 (ツールバー)	68

て

データ	
データ表示部(データ表示ウィンドウ)	47
デッドライン設定	
環境設定 - デッドライン設定	45
デッドライン設定ダイアログの操作	66
デッドライン設定ダイアログを開く	65
デバッグ	
デバッグ種別による機能制限	101
デバッグタイプ	
デバッグタイプごとの機能	15
デバッグタイプの設定	15

と

統計	
リアルタイム統計	67
リアルタイム統計ウィンドウの操作 (ツールバー)	68
リアルタイム統計ウィンドウの表示項目	68
リアルタイム統計ウィンドウを開く	67
統計ウィンドウ	
統計ウィンドウ	60
統計ウィンドウの表示項目	61
統計ウィンドウを開く	60
統計グラフ	
遷移図および統計グラフの参照	16
統計グラフの表示	44
フラグクリア・取得・遷移図・統計グラフ・ モニタリングボタン(データ表示ウィンドウ)	47
統計ログ	
統計ログの測定方法	51
統計ログの特長	51
ドラッグアンドドロップ	
各オブジェクトのリスト形式の列順を変更する 【カラムのドラッグアンドドロップ】	34

に

入力フォーム	
ID・パスカウント・アドレス入力フォーム (設定ウィンドウ)	38

は

パスカウント	
ID・パスカウント・アドレス入力フォーム (設定ウィンドウ)	38
発行	
システムコール発行機能使用時の注意事項	99
システムコール発行機能の仕様と制限事項	99
発行手順	17
発行の結果	17
発行方法	75
[発行]ボタン(引数設定ウィンドウ)	74
バッファサイズ	
バッファサイズ, タスク数カスタマイズ用ファ イル	87
バッファサイズの変更	89
バッファフル	
バッファフルの設定	49
パラメータ	
ログデータの表示アイコンとパラメーター一覧	55

ひ

引数	
システムコール仕様表示部 (引数設定ウィンドウ)	74
[発行]ボタン(引数設定ウィンドウ)	74
引数入力ボックス(引数設定ウィンドウ)	74
ポインタ型の引数について	72
表示項目	
統計ウィンドウの表示項目	61
表示内容	
使用状況と表示内容	100

ふ

ファイル	
ファイル構成	5
フック	
コメントイベント用フックのC言語用インタ フェース	88
フラグクリア	
フラグクリア・取得・遷移図・統計グラフ・ モニタリングボタン(データ表示ウィンドウ)	47
ブレイク	
OS ブレイクの種類	12
OS ブレイクの設定	12
設定済のブレイクを設定ウィンドウに反映させる	42
ブレイク種別	39
ブレイクの実現方法	39

ブレイク発生	41
リソースとなるブレイク	41
へ	
ペイン	
OS ブレイク設定機能選択時の各ペインの機能	37
オブジェクト表示選択時の各ペインの機能	24
システムコール発行機能選択時の各ペインの機能	73
ログ機能選択時の各ペインの機能	46
ログ設定選択ペインで行う設定	49
ヘッダーファイル	
インクルードが必要なヘッダーファイル	88
変更	
追加・変更・削除ボタン (設定ウィンドウ)	37
変更	42
ほ	
ポインタ型	
ポインタ型の引数について	72
ボタン	
[更新] ボタン	77
設定方法【[追加] ボタン】	41
追加・変更・削除ボタン (設定ウィンドウ)	37
[発行] ボタン (引数設定ウィンドウ)	74
表示 ID 選択および表示実施ボタン ([View] ボタン)	25
ボックス	
引数入力ボックス (引数設定ウィンドウ)	74
ま	
マウス	
システムコールアイコン上でのマウス操作	59
め	
メッセージ	
警告メッセージ	97
も	
モジュール	
使用するタスク解析モジュールの選択	86
タスク解析モジュール	86
モニタデバッグ	
モニタデバッグ使用時の注意点	4
モニタリング	
環境設定 - モニタリング間隔時間設定	45
モニタリング機能	45
モニタリングの操作 (ツールバー)	63
モニタリングウィンドウ	
モニタリングウィンドウ	62
モニタリングウィンドウを開く	62

モニタリング間隔	
モニタリング間隔設定ダイアログの操作	65
モニタリング設定	
モニタリング設定ダイアログを開く	64
モニタリングボタン	
フラグクリア・取得・遷移図・統計グラフ・モニタリングボタン (データ表示ウィンドウ)	47
よ	
用語	
用語説明	77
用語の説明	2
ら	
ライブラリ	
ライブラリの追加	7
り	
リアルタイム統計	
リアルタイム統計	67
リアルタイム統計ウィンドウの操作 (ツールバー)	68
リアルタイム統計ウィンドウの表示項目	68
リアルタイム統計ウィンドウを開く	67
リスト	
各オブジェクトのリスト形式の列順を変更する【カラムのドラッグアンドドロップ】	34
システムコール選択リスト (システムコール選択ウィンドウ)	74
ログ設定選択リスト (ログ設定選択ウィンドウ)	46
リソース	
タイマリソースの変更	89
リソースタイムカスタマイズ用ファイル	87
リソースとなるブレイク	41
リンク	
リンクされていないシステムコールについて	72
れ	
列順	
各オブジェクトのリスト形式の列順を変更する【カラムのドラッグアンドドロップ】	34
変更した列順を保存する【右クリック [列順保存]】	34
ろ	
ログ	
ログの取得	15, 52
ログの設定	14
ログ機能	
ログ機能選択時の各ペインの機能	46
ログ設定	
[Setting] ボタン (ログ設定選択ウィンドウ)	47

ログ設定選択ペインで行う設定	49	ログデータの表示アイコンとパラメーター一覧	55
ログ設定選択リスト (ログ設定選択ウィンドウ)	46	ログファイル	
ログ設定の開き方	46	ログファイルの出力	45, 69
ログデータ			
ログデータの表示	53		

CM81-00320-1

富士通マイクロエレクトロニクス・CONTROLLER MANUAL

FR ファミリ

μT-Kernel 仕様準拠

SOFTUNE™ μT-REALOS/FR

アナライザガイド

2008 年 6 月 初版発行

発行 **富士通マイクロエレクトロニクス株式会社**

編集 マーケティング統括部 ビジネス推進部
