

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日

ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

デバッグングエクステンション

ユーザーズマニュアル

HS6400IWIN1SJ

ルネサスマイクロコンピュータ開発環境システム

ご注意

安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご注意ください。

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりますは、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ(<http://www.renesas.com>)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したものです。万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジは、適用可否に対する責任を負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気付きの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。

【商標等について】

- μ ITRON は、Micro Industrial TRON の略称です。TRON は The Realtime Operating System Nucleus の略称です。
- Microsoft® Windows® 98 operating system, Microsoft® Windows® Millennium Edition(Windows® Me) operating system, Microsoft® Windows NT® operating system, Microsoft® Windows® 2000 operating system, Microsoft® Windows® XP operating system は、米国 Microsoft Corporation の米国およびその他の国における登録商標です。
- SuperH™ は、株式会社ルネサス テクノロジーの商標です。
- その他、本書で登場するシステム名、製品名は各社の登録商標または商標です。

はじめに

デバッグエクステンション（以下、DX と略します）は、「High-performance Embedded Workshop」（以下、HEW と略します）に、リアルタイムオペレーティングシステム（以下、RTOS と略します）用のマルチタスクデバッグ機能を付加するソフトウェアです。

本マニュアルでは、DX の使用方法について説明します。

ご使用になる前に本マニュアルを良く読んで理解してください。また、下記の関連マニュアルもお読みの上、理解してください。

マニュアルの構成

本マニュアルは、以下の章と付録から構成されています。

- 第 1 章では、DX の概説を記載しています。
- 第 2 章では、例を用いて DX の操作方法を説明しています。
- 第 3 章では、DX の機能概略を説明しています。
- 第 4 章では、留意・制限事項を説明しています。
- 付録では、トラブルシューティングを記載しています。

特に第 4 章の「留意・制限事項」は、DX を使用する前に必ずお読みください。

なお、DX の機能の詳細については、オンラインヘルプを参照してください。

関連マニュアル

本 DX に関連する以下の製品のマニュアルも、あわせて参照してください。

- High-performance Embedded Workshop (HEW) ユーザーズマニュアル
- 使用する RTOS のユーザーズマニュアル
- 使用するコンパイラのユーザーズマニュアル
- 使用するマイコンのハードウェアマニュアル、プログラミングマニュアル

本マニュアルで使用する記号などの意味

[メニュー -> メニュー オプション]	'->' はメニューオプションを示します (例 [ファイル->新規作成])
H' と D'	"H" は 16 進数、"D" は 10 進数を意味するプリフィックスです。プリフィックスの無い場合は 10 進数です。

なお、日本語環境でないホストマシンでは、ディレクトリ区切り記号「¥」が「\」と表示される場合があります。読み替えてご使用ください。

目次

1.	概説	1
1.1	概要	1
1.2	特長	1
1.3	使用前の準備	2
2.	チュートリアル	3
2.1	チュートリアルの構築	3
2.1.1	コンフィギュレータ	3
2.1.2	ビルド	3
2.2	例題プログラムの内容	4
2.3	例題プログラムの実行	5
3.	機能	19
3.1	オブジェクト状態の参照	19
3.2	オブジェクトの操作	19
3.3	サービスコールトレースの表示	21
3.4	機能一覧	22
3.4.1	メニュー	22
3.4.2	ウィンドウとダイアログボックス	23
4.	留意・制限事項	27
4.1	ターゲットシステムのリアルタイム性と周期ハンドラ	27
4.2	表示内容の整合性	27
付録A	トラブルシューティング	29
A.1	[Action Result]ウィンドウの結果表示	29

図目次

図1.1	DX概要	1
図2.1	Debug Settingsダイアログ	5
図2.2	Component Galleryダイアログ	5
図2.3	Simulator SystemダイアログのSystemページ	6
図2.4	Simulator SystemダイアログのMemoryページ	6
図2.5	ロードモジュールのダウンロード	7
図2.6	RTOSトレースの設定	8
図2.7	Trace Optionsダイアログ	8
図2.8	プログラムの実行	9
図2.9	プログラムの停止	10
図2.10	プログラム停止時のソースウィンドウ	11
図2.11	[Select Diagram Object]ダイアログボックス	11
図2.12	[RTOS Trace]ウィンドウ	12
図2.13	[Task]ウィンドウ	13
図2.14	[Task]ウィンドウのポップアップメニュー	14
図2.15	[Delete Task]ダイアログボックス	14
図2.16	[Terminate Task]ダイアログボックス	15
図2.17	オブジェクト操作結果	15
図2.18	[Create Task]ダイアログボックス	16
図2.19	ブレークポイントの設定	16

図2.20	[RTOS Trace Diagram]ウィンドウと[RTOS Trace Text]ウィンドウ	17
図3.1	タスク状態の表示例	19
図3.2	オブジェクト操作の要求例	19
図3.3	オブジェクト操作の結果	20
図3.4	[RTOS Trace Diagram]ウィンドウと[RTOS Trace Text]ウィンドウ	21

表目次

表 2.1	シミュレータ・デバッガのメモリリソース	6
表3.1	HEWの [View]メニューに追加される内容	22
表3.2	ウィンドウとダイアログボックス一覧	23

1. 概 説

1.1 概 要

DX は RTOS 用に作成されたアプリケーションのデバッグを行うためのソフトウェアです。HEW および RTOS のシステムに組み込んで使用します。

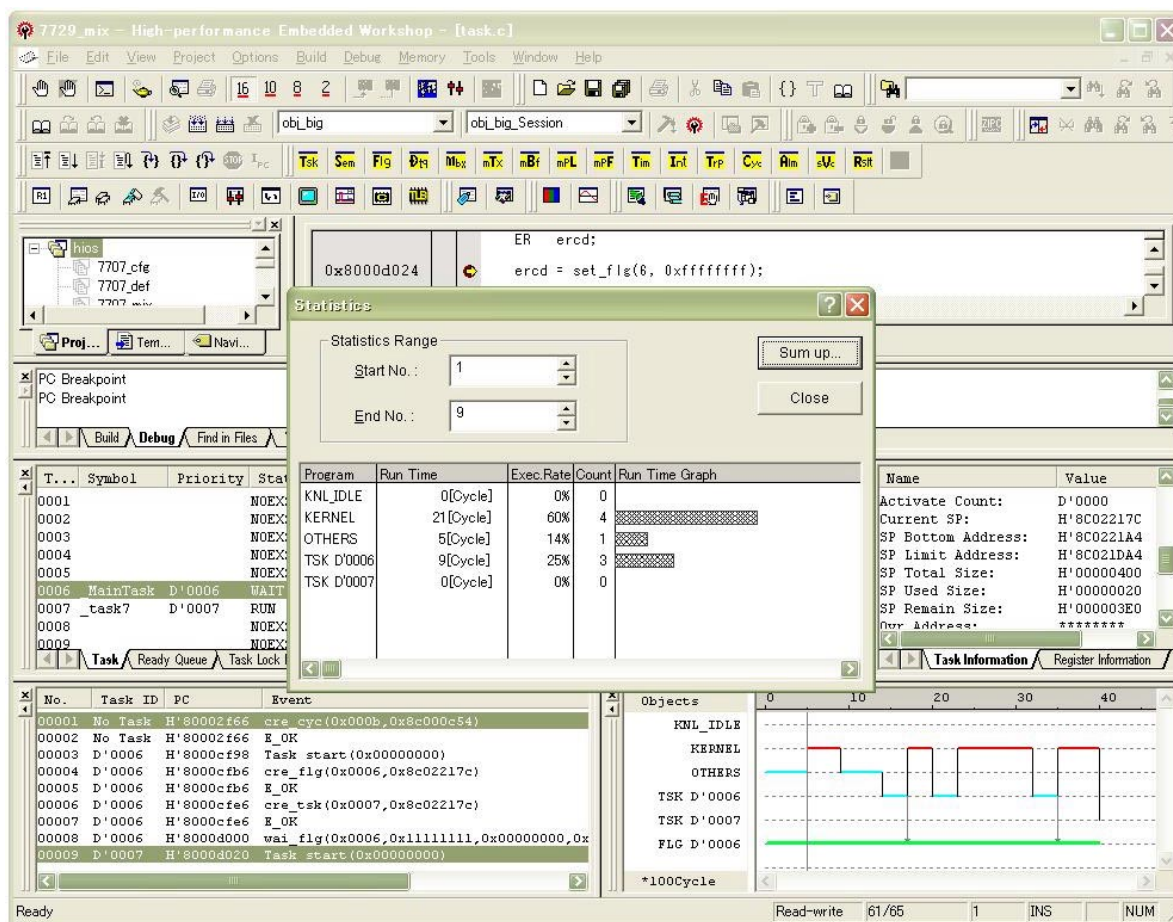


図1.1 DX 概要

1.2 特 長

(1) グラフィカルユーザインタフェース

タスクなどのオブジェクト状態の参照、変更が、ウィンドウやダイアログボックスを通して行え、HEW の環境でマルチタスクアプリケーションをデバッグできます。

(2) サービスコールトレース情報の表示

RTOS のサービスコール履歴をグラフィカルに表示できます。また、その履歴から必要な項目だけを選択して表示することも可能です。

1.3 使用前の準備

以下を行うことにより DX を使用したデバッグが可能となります。

(1) インストール

インストーラに従って DX をインストールします。

(2) RTOS のコンフィギュレーション

使用する RTOS のシステムを DX 用に構築します。構築例は「2. チュートリアル」を参照してください。

(3) HEW の起動

HEW を起動します。

プログラムをビルドした後、マルチタスクデバッグが可能となります。DX の使用例は「2. チュートリアル」を参照してください。また、機能については「3. 機能」およびオンラインヘルプを参照してください。

2. チュートリアル

本章では、RTOS として HI7000/4 シリーズを例に用い、DX の操作例について説明します。詳しい操作方法は、オンラインヘルプを参照してください。

ここでは、以下の環境を前提にしています。

- RTOS : HI7700/4
- マイコン : SH7729
- デバッガ : SuperH RISC engine シミュレータ・デバッガ

2.1 チュートリアルの構築

HI7700/4 が提供するサンプルを使用します。構築に関する詳細は、HI7000/4 シリーズ ユーザーズマニュアルの「5. コンフィギュレーション」を参照してください。

2.1.1 コンフィギュレータ

HI7700/4 が提供するサンプル HCF ファイル(hiuser¥sh7729¥7729.hcf)を使って構築ファイルを生成します。コンフィギュレータの設定項目の中で、特に以下の項目を設定してください。

- CFG_ACTION をチェックしてください。
- CFG_TRACE をチェックしてください。
- CFG_TRCTYPE を「エミュレータトレース」に設定してください。（本設定によりシミュレータにおいても DX の RTOS トレースが使用できます）
- CFG_MAXTSKID は 7 以上を設定してください。
- CFG_STSTKID は 5 以下を設定してください。
- CFG_MAXTSKPRI は 7 以上を設定してください。
- CFG_MAXFLGID は 6 以上を設定してください。
- 以下のサービスコールを組み込んでください。
cre_tsk, ext_tsk, cre_flg, wai_flg, set_flg

2.1.2 ビルド

HI7700/4 の一括リンクによりビルドします。

サンプルの hios.hws を HEW で開き、プロジェクトとして 7729_mix を選択してください。[Options->SuperH RISC engine Standard Toolchain...]の[C/C++]ページの[Optimize]カテゴリの[Gbr relative logic operation]を[User]に設定してください。[Build->Build]によりビルドを行ってください。

2.2 例題プログラムの内容

例題プログラムのソースは、hiuser¥tutorial¥task.c です。ここには、MainTask と Task7 の 2 つのタスクがあります。

MainTask は、以下の条件でコンフィギュレータによって生成・起動が指定されているので、システムを起動すると自動的に MainTask が実行されます。

- タスク ID : 6
- 初期優先度 : 6
- 開始アドレス : MainTask()
- 初期状態 : READY (TA_ACT 属性指定)

一方、Task7 は以下の条件で MainTask によって動的に生成されます。

- タスク ID : 7
- 初期優先度 : 7
- 開始アドレス : Task7()
- 初期状態 : READY (TA_ACT 属性指定)

優先度は、MainTask の方が高い設定になっています。

MainTask は、ID 6 のイベントフラグと Task7 を生成し、wai_flg サービスコールで ID 6 のイベントフラグがセットされるのを待ちます。その後、del_flg サービスコールで ID 6 のイベントフラグを削除し、ext_tsk サービスコールで終了します。

Task7 は、set_flg サービスコールで ID 6 のイベントフラグをセットし、その後 while(1);文で無限ループに入ります。

なお、本例題プログラムはカーネルの動作および DX の操作を習得するためだけに作られたプログラムであり、処理内容には何の意味もありません。

2.3 例題プログラムの実行

- (1) シミュレータ・デバッガを起動します。HEW の[Options>Debug Settings...]の左のリストから 7729_mix を選択し、[Target]ページで[SH3-DSP Simulator]、[Add...]ボタンにより HI7000/4 の hiuser¥obj_big¥7729_mix.abs を選択してください。

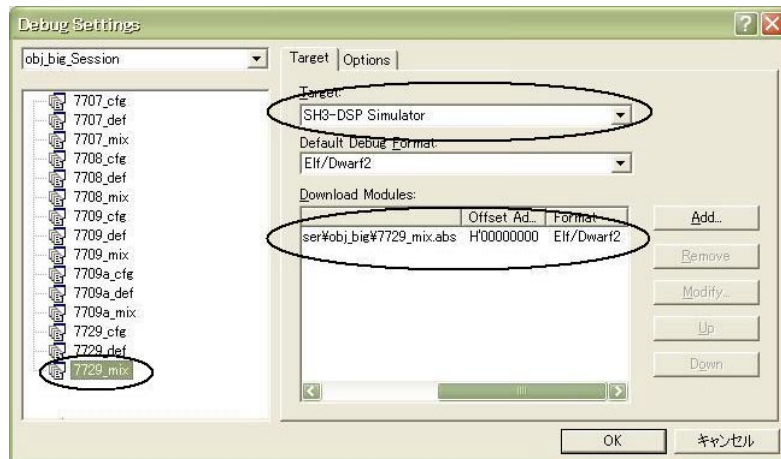


図2.1 Debug Settings ダイアログ

- (2) DX を起動します。[Project->Components...]の[RTOS HI7000_4DX ECX]を選択し、[Load]ボタンおよび[OK]ボタンを押してください。

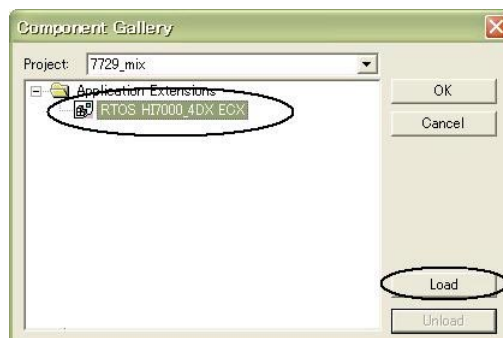


図2.2 Component Gallery ダイアログ

- (3) シミュレータ・デバッガの設定を行います。[Options->Simulator->System...][Enable Timer]をチェックし、[Execution Mode]を[Continue]に設定してください。

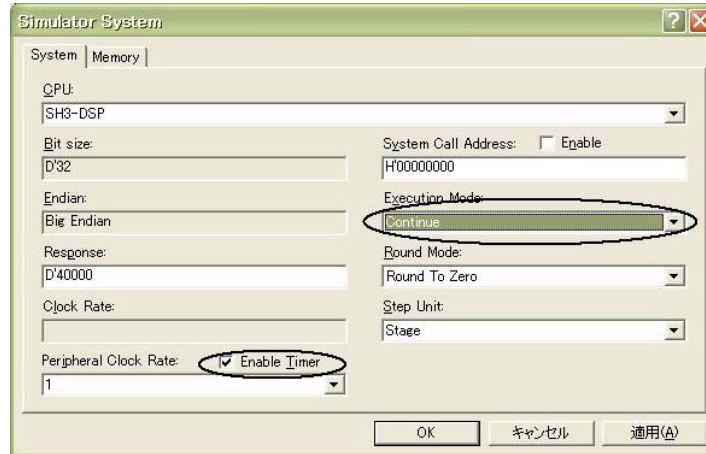


図2.3 Simulator System ダイアログの System ページ

- (4) Simulator System ダイアログの[Memory]ページにおいて、表 2.1に示すメモリリソースを確保してください。

表2.1 シミュレータ・デバッガのメモリリソース

開始アドレス	終了アドレス	属性
H'00000000	H'0000FFFF	Read
H'0C000000	H'0C00FFFF	Read/Write

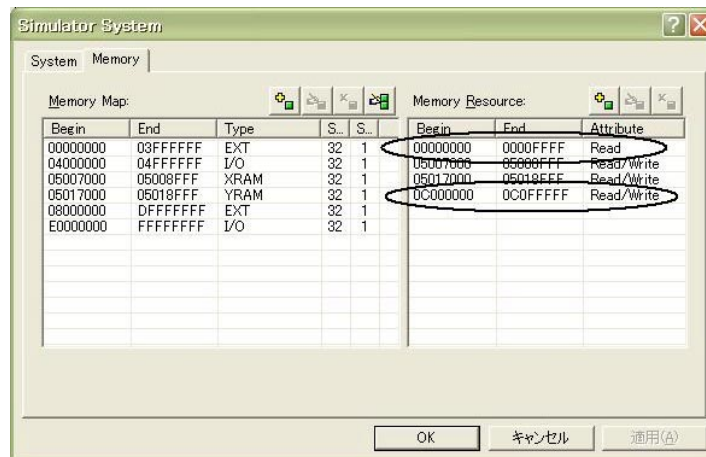


図2.4 Simulator System ダイアログの Memory ページ

- (5) ロードモジュールをダウンロードします。[Debug->Download Modules]の `hiuser#obj_big#7729_mix.abs` を選択してください。

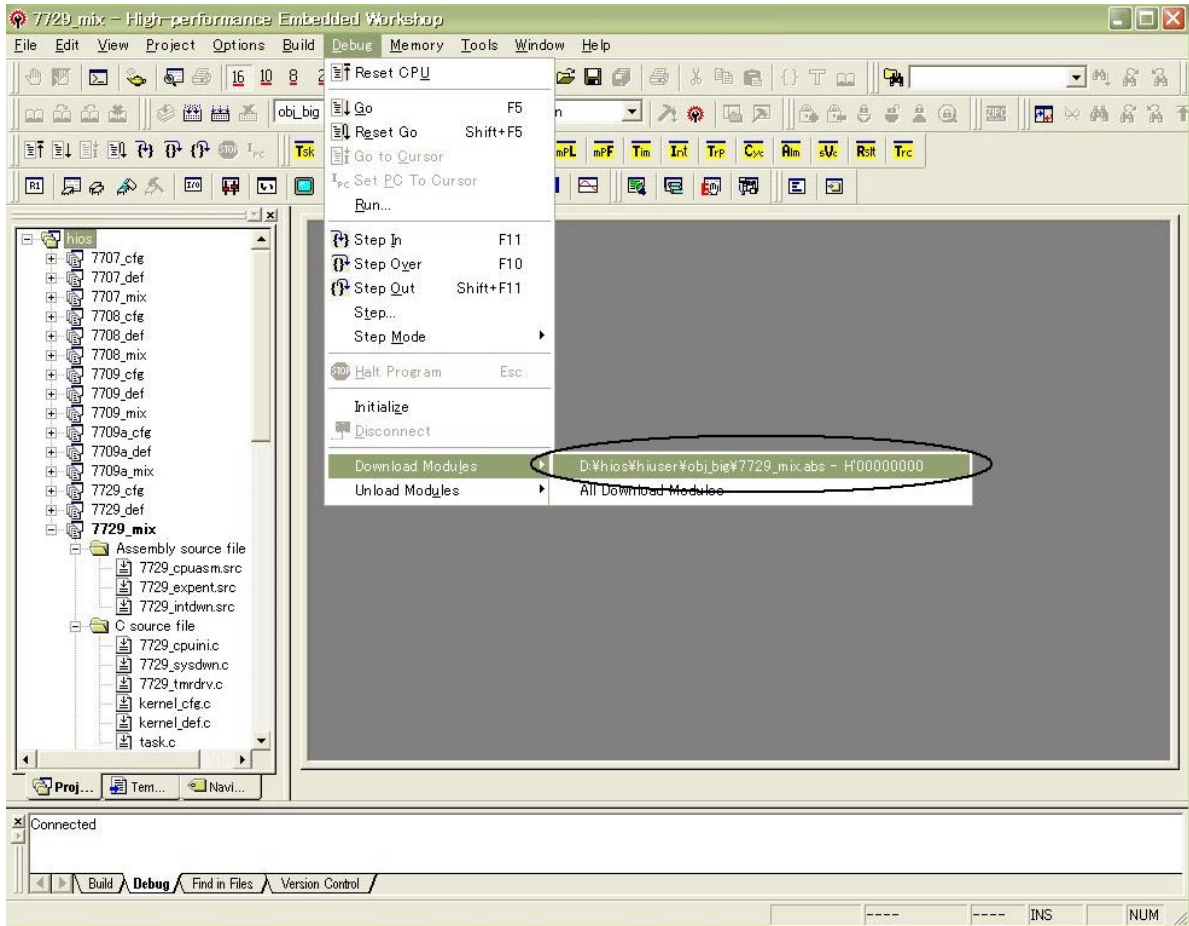


図2.5 ロードモジュールのダウンロード

2. チュートリアル

- (6) RTOS トレースの設定を行います。[View->RTOS->Trace]を選択してください。[RTOS Trace]ウィンドウで右クリックし、ポップアップメニューから[Set trace options...]を選択してください。

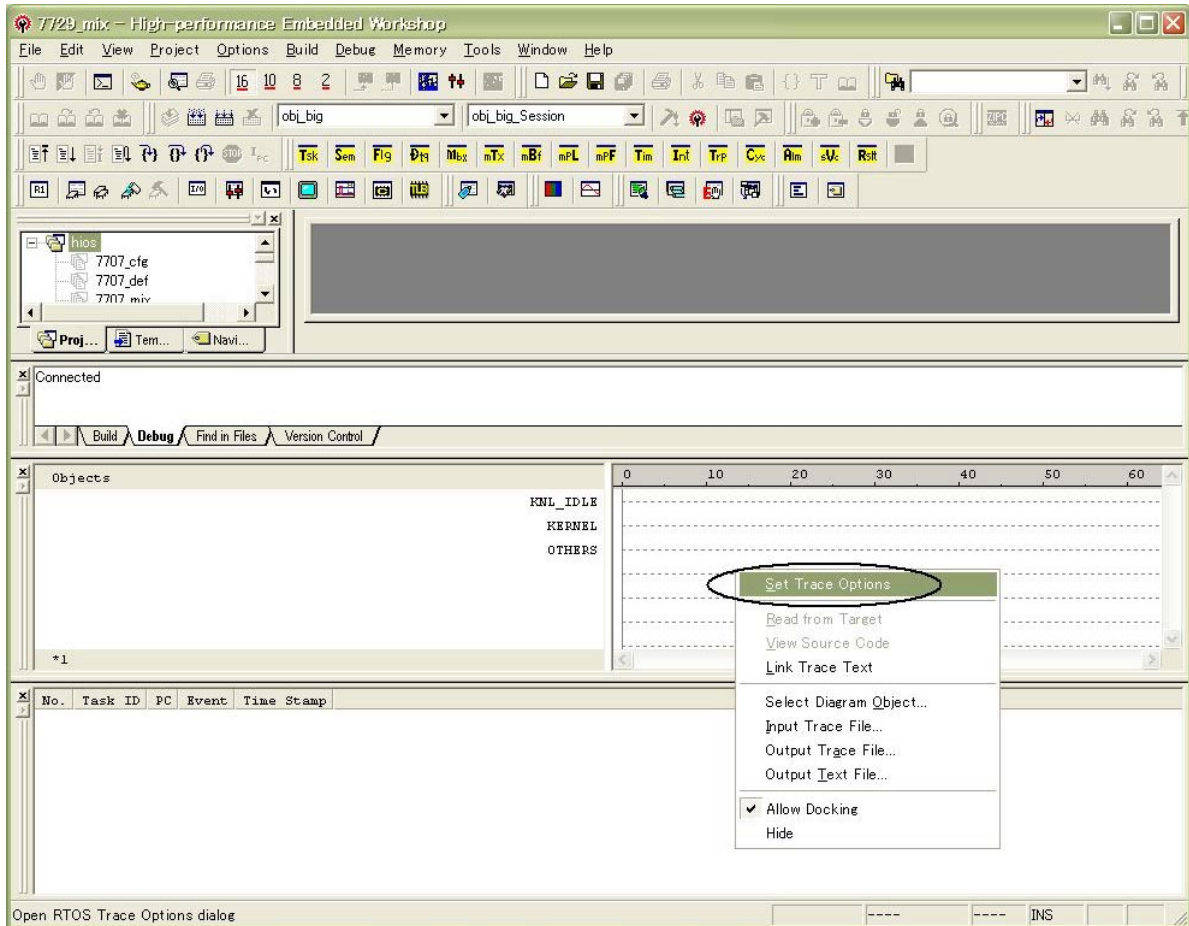


図2.6 RTOS トレースの設定

- (7) [Scale Type]を[Cycle]に、[Scale List]を[*100 Cycle]に設定してください。

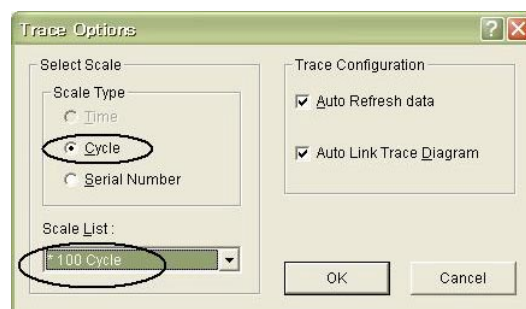


図2.7 Trace Options ダイアログ

- (8) プログラムを実行します。[Debug->Reset Go]を選択してください。

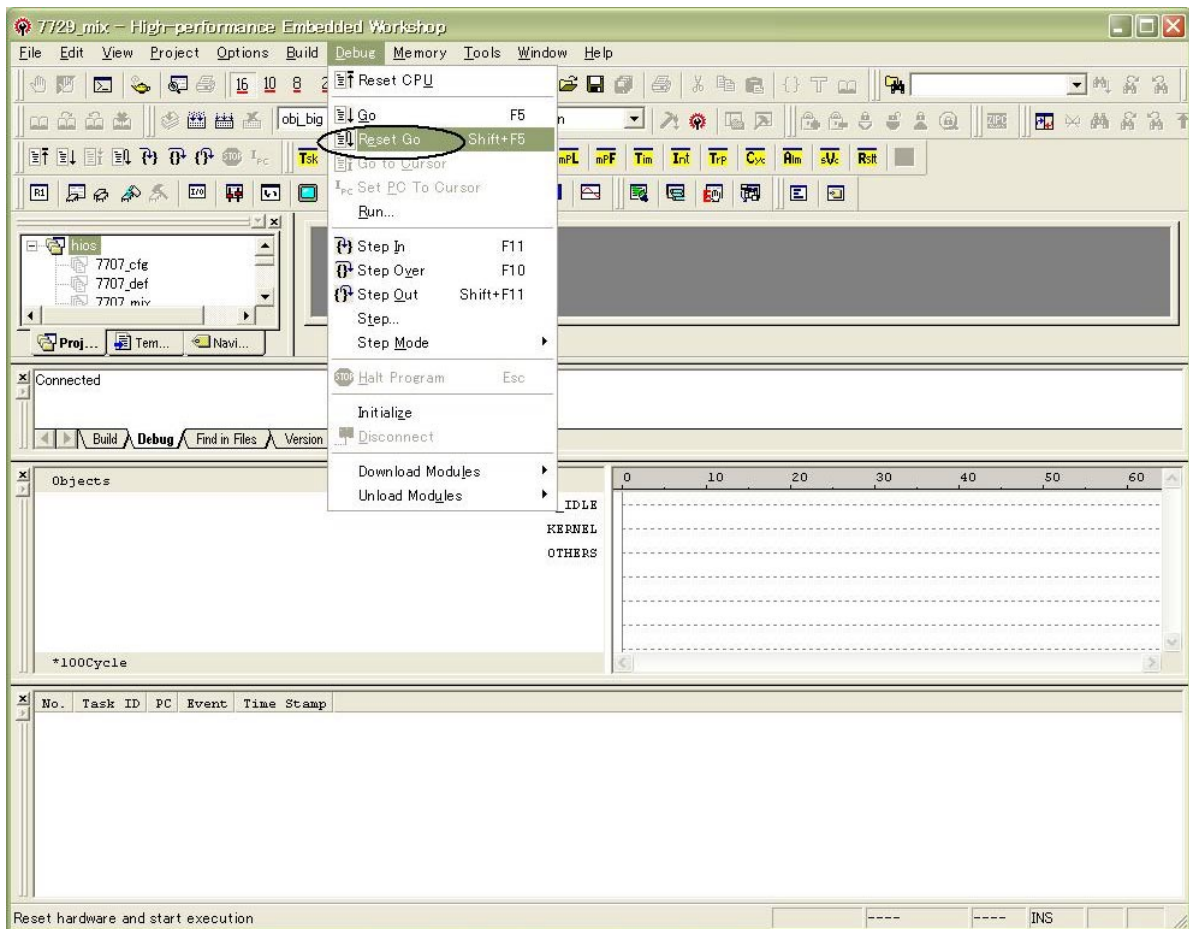


図2.8 プログラムの実行

2. チュートリアル

- (9) プログラムを停止します。[Debug->Halt Program]を選択してください。

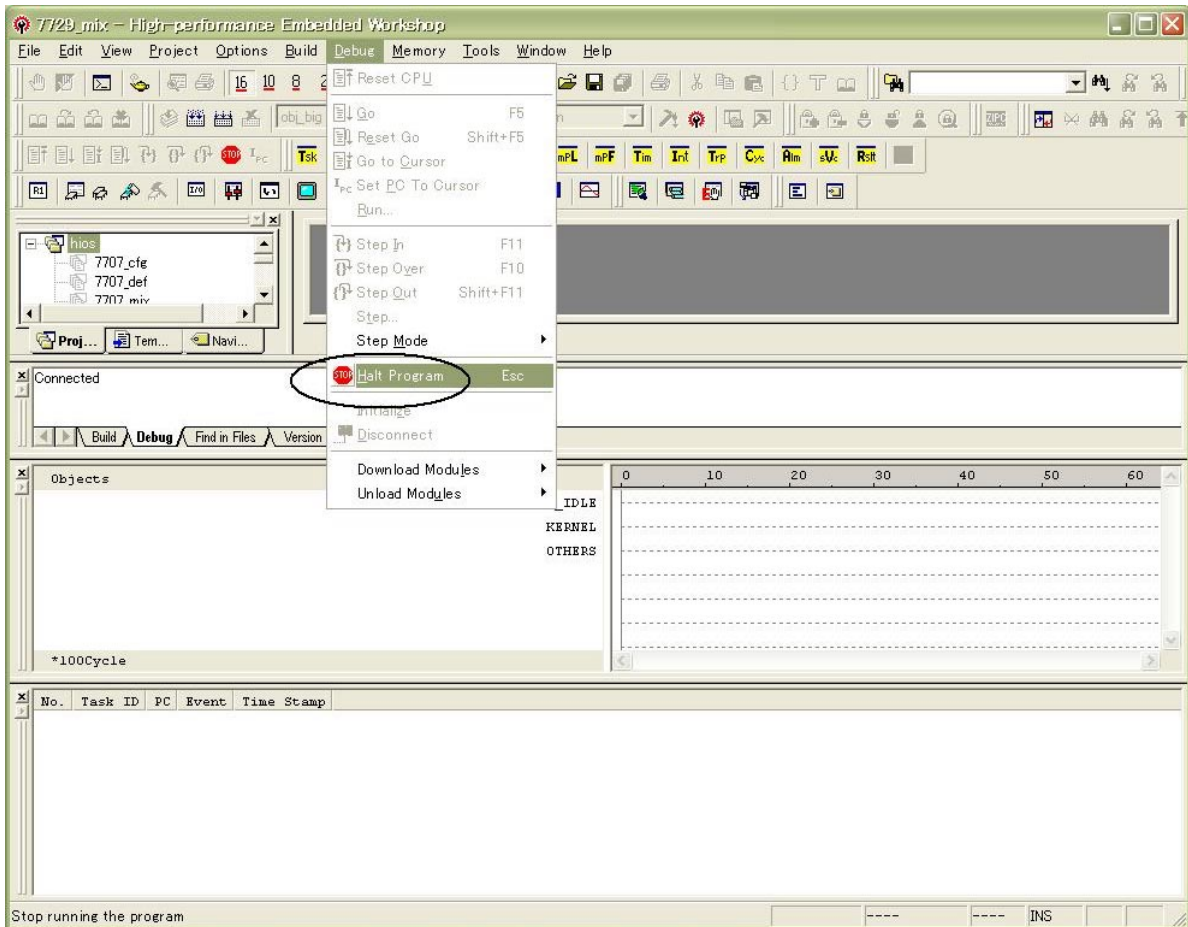


図2.9 プログラムの停止

- (10) 図 2.10のように、Task7 中の while(1);文で停止します。

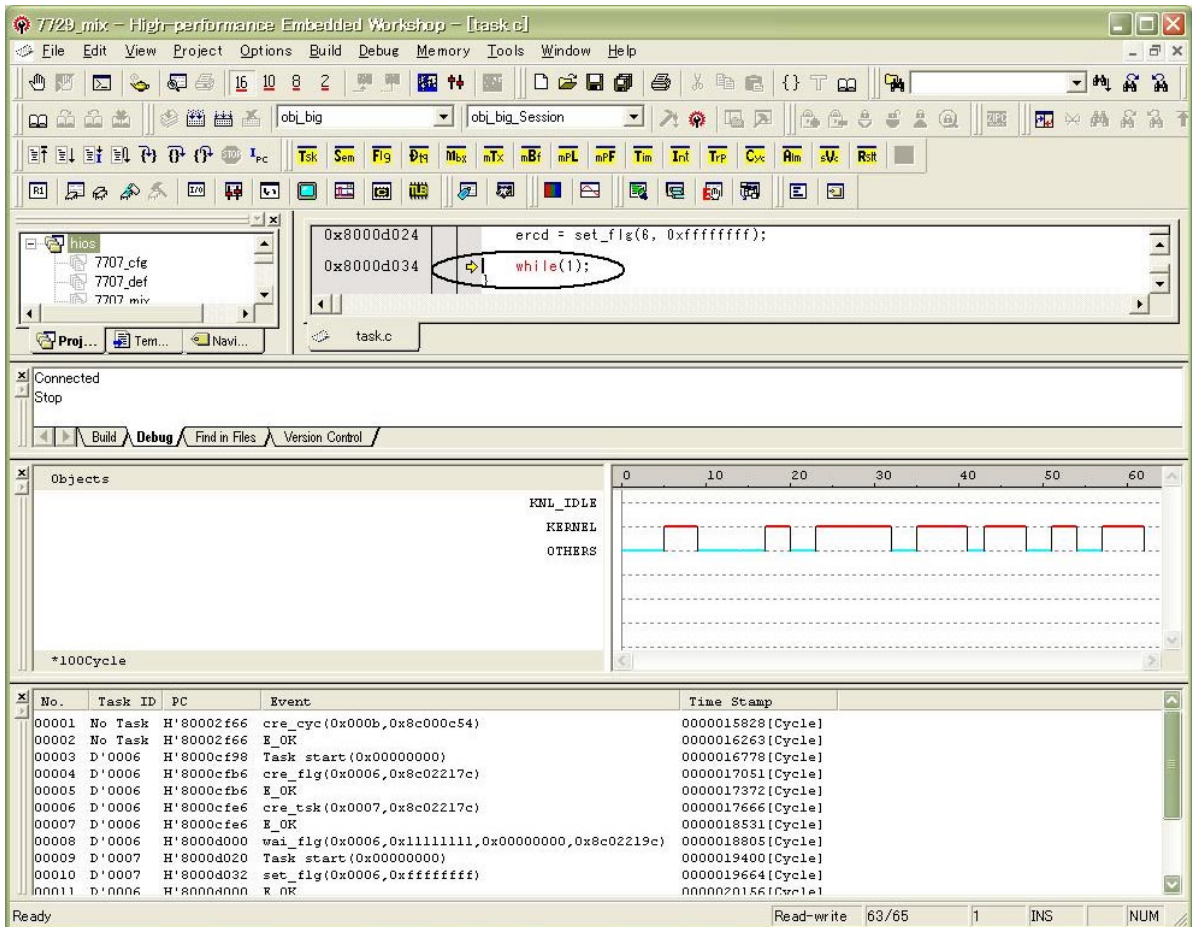


図2.10 プログラム停止時のソースウィンドウ

- (11) どのようにプログラムが実行されたのかを、サービスコールトレースで確認します。ダイアグラム表示を見やすくするために、MainTask(タスク ID 6)と Task7(タスク ID 7)が表示されるようにします。[RTOS Trace]ウィンドウで右クリックし、ポップアップメニューから[Select Diagram Object...]を選択してください。タスク ID 6 と 7 を追加してダイアログボックスを閉じてください。

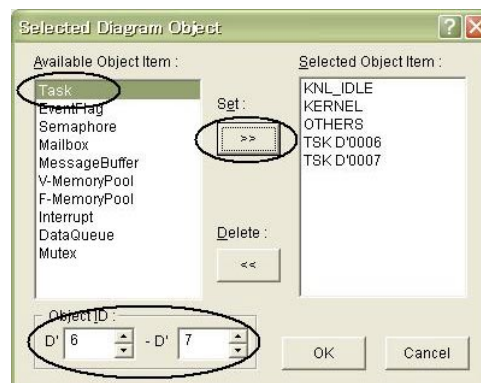


図2.11 [Select Diagram Object]ダイアログボックス

2. チュートリアル

- (12) タスク ID 6 と 7 が[RTOS Trace Diagram]ウィンドウに表示されるようになります。MainTask が wai_flg を発行した後に Task7 にタスクスイッチしたことが分かります。[RTOS Trace Text]ウィンドウのサービスコールが表示されている行をダブルクリックすることで対応するソースウィンドウを開くことができます。

The screenshot shows the HPEW interface with the following components:

- Source Code:** `task.c` showing the function `wai_flg` call: `ercd = wai_flg(6, waiptr, TWF_ANDW, &flgptr);`
- RTOS Trace Diagram:** A timeline showing task execution. A red bar represents `TASK D'0006` and a blue bar represents `TASK D'0007`. A vertical line indicates a task switch from task 6 to task 7.
- RTOS Trace Text:** A log of events. The entry `wai_flg(0x0006, 0x11111111, 0x00000000, 0x8c02219c)` is circled, corresponding to the source code line.

図2.12 [RTOS Trace]ウィンドウ

- (13) [Task]ウィンドウで MainTask と Task7 がどのような状態になっているかを確認してみます。 [View->RTOS->Task]を選択すると、[Task]ウィンドウが開きます。MainTask は DORMANT 状態、Task7 は RUN 状態であることが分かります。

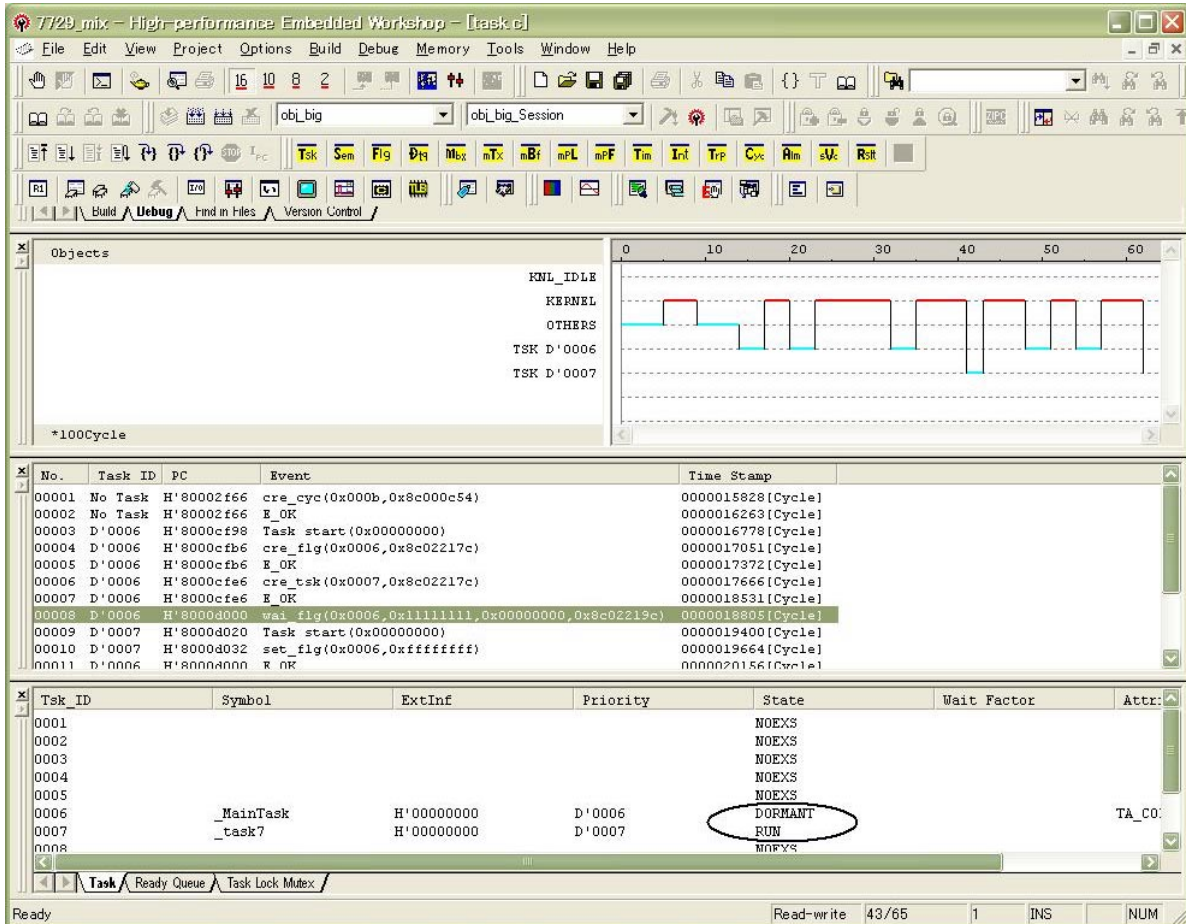


図2.13 [Task]ウィンドウ

2. チュートリアル

- (14) 今度は、MainTask の優先度を Task7 よりも低くして実行してみることにしましょう。そのために、まずは MainTask, Task7 を削除して初期状態に戻すことにします。MainTask を削除するには、[Task]ウィンドウで MainTask を選択(ハイライト)し、右ボタンクリックによるポップアップメニューから[Action->Delete Task...]選択してください。

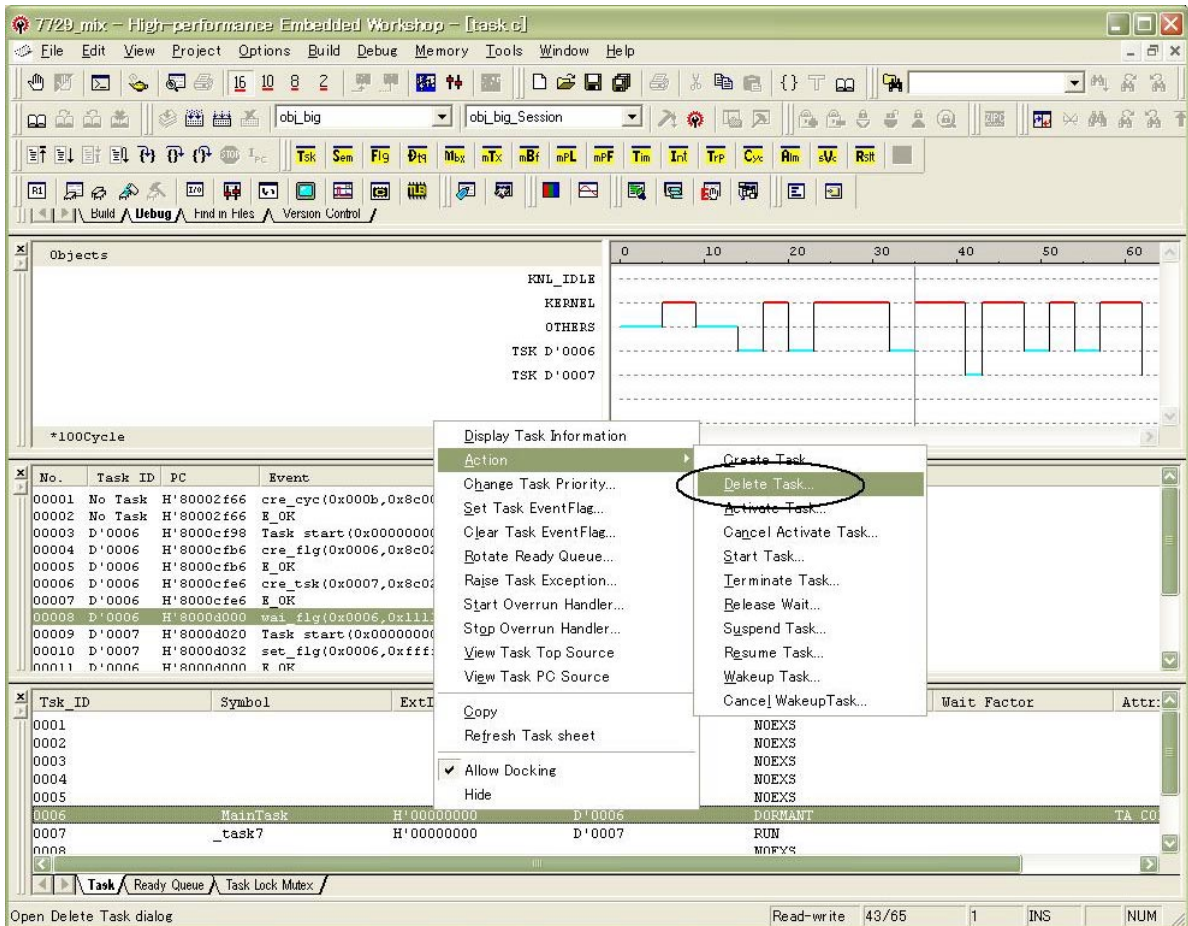


図2.14 [Task]ウィンドウのポップアップメニュー

- (15) 確認ダイアログボックスで[OK]ボタンを押してください。

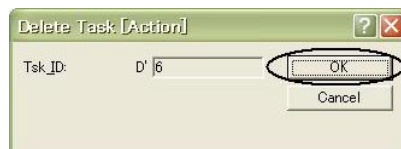


図2.15 [Delete Task]ダイアログボックス

- (16) 一方、Task7 はまだ RUN 状態なので、強制終了させてから削除する必要があります。Task7 を強制終了させるには、[Task]ウィンドウで Task7 を選択(ハイライト)させ、右ボタンクリックによるポップアップメニューから [Action->Terminate Task...]を選択すると、[Terminate Task]ダイアログボックスが開きますので、[OK]を押してください。その後、MainTask と同様にして Task7 を削除してください。



図2.16 [Terminate Task]ダイアログボックス

- (17) タスクの強制終了や削除といったカーネルオブジェクトの操作の結果は、[View->RTOS->Action Result]で確認できます。しかし、今[Action Result]ウィンドウを開いても、何も表示されません。カーネルオブジェクトの操作は DX からカーネルヘサースコールの実行を依頼することによって実現されています。このため、カーネルが実行した後、実際にタスクが削除されます。結果を得るために、[Debug->Go]によりターゲットの実行を再開させてください。自動的に結果が表示されます。その後、[Task]ウィンドウのポップアップメニューより[Refresh task sheet]を選択すると、MainTask(タスク ID 6)と Task7(タスク ID 7)が NOEXS 状態になっていることがわかります。

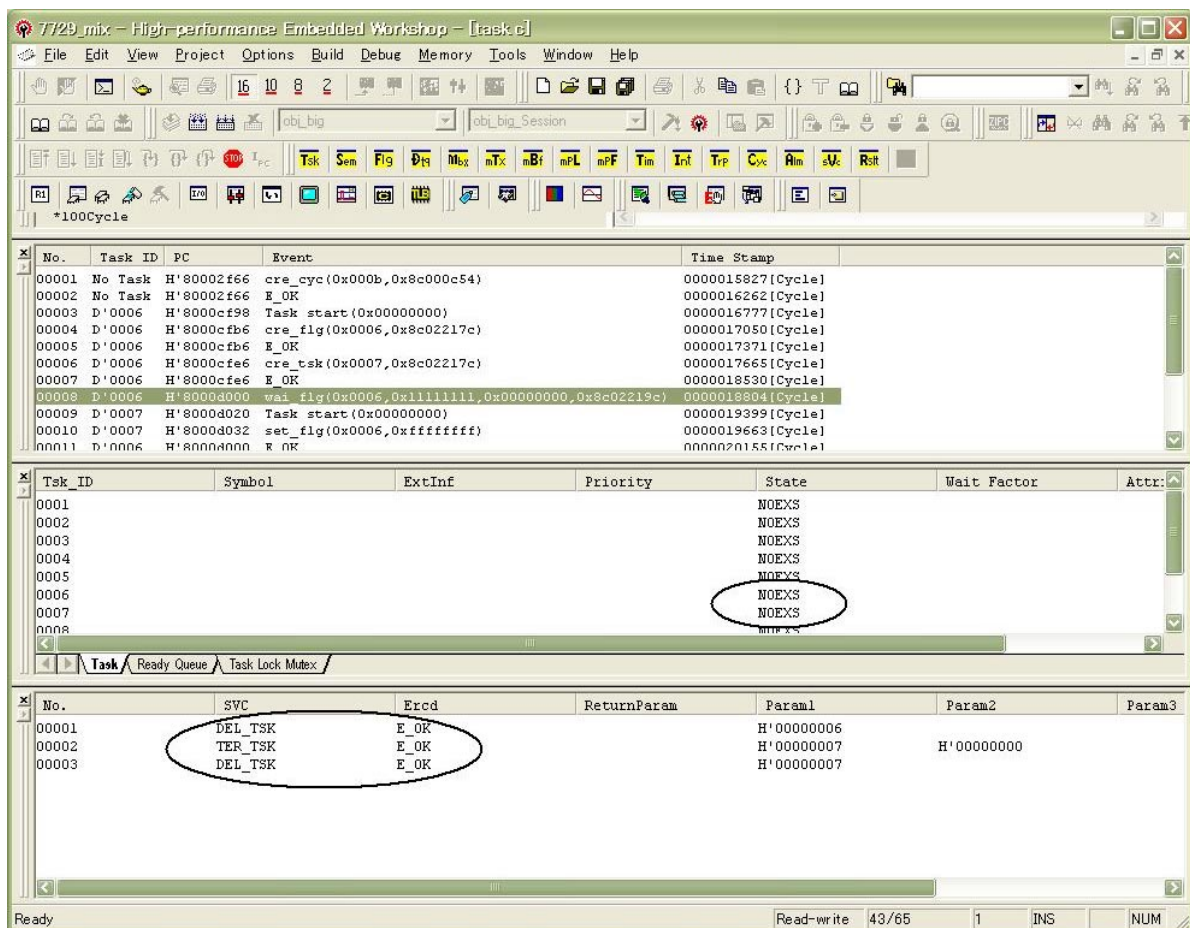


図2.17 オブジェクト操作結果

2. チュートリアル

- (18) [Debug->Halt Program]によりプログラムを停止してください。MainTask をTask7 よりも低い優先度(8)で生成・起動します。[Task]ウィンドウでタスク ID 6 を選択(ハイライト)させ、ポップアップメニューから[Action->Create Task...]を選択すると、[Create Task]ダイアログボックスが開きます。ここでは、図 2.18のように入力してください。

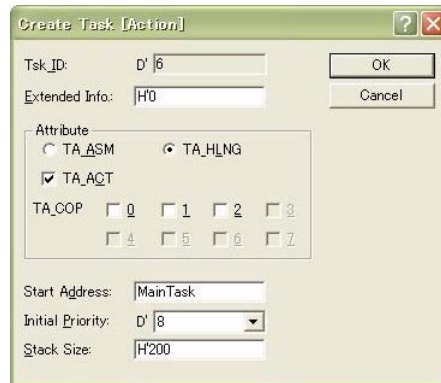


図2.18 [Create Task]ダイアログボックス

- (19) [Task]ウィンドウおよび[Action Result]ウィンドウのポップアップメニューから[Hide]を選択して、ウィンドウを閉じてください。再実行する前に Task7 の while(1);文まで実行したら停止するようにブレークポイントを設定しておきます。図 2.19の丸で囲んだ部分をダブルクリックしてブレークポイントを設定してください。

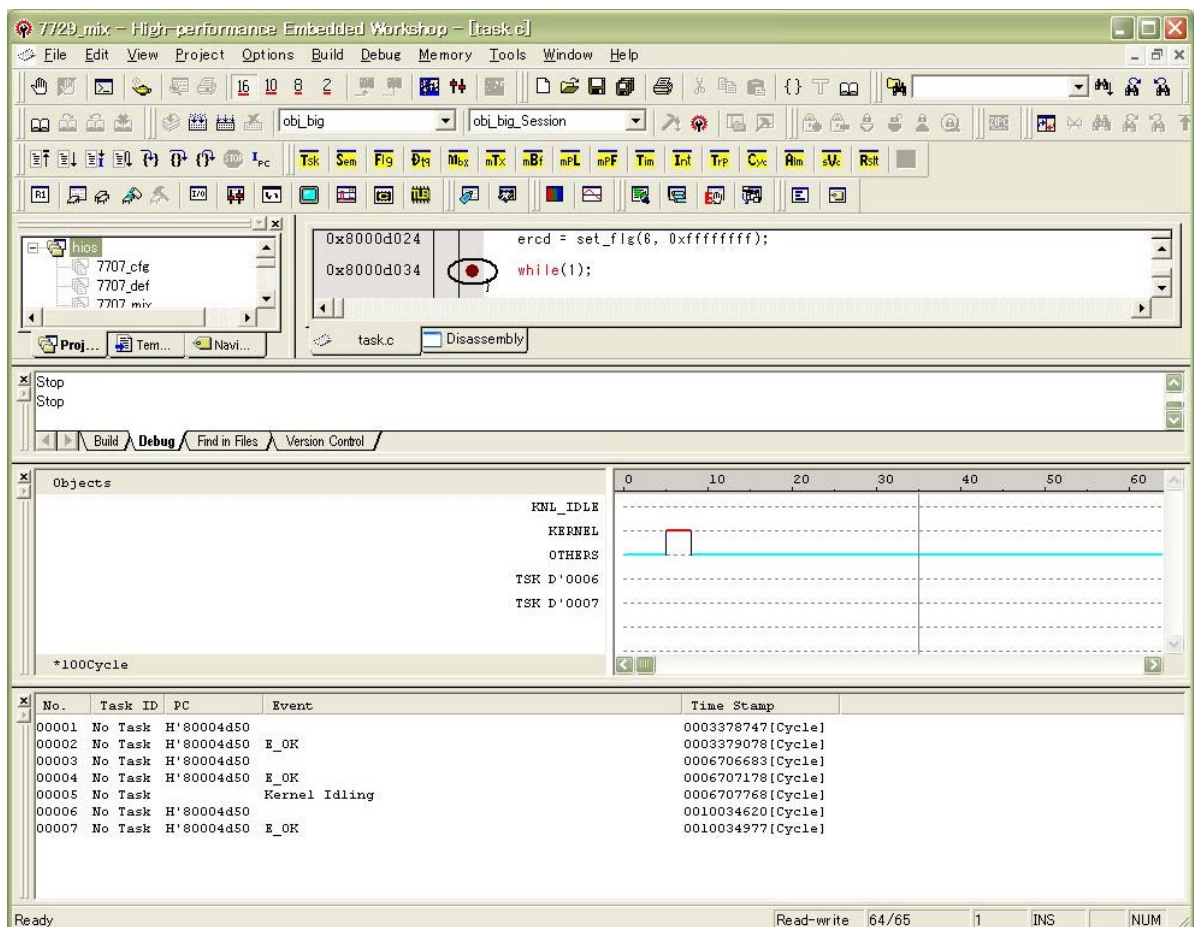


図2.19 ブレークポイントの設定

- (20) ターゲットを再実行させると、MainTask が起動され、先ほど設定した Task7 の while(1);文のブレークポイントで停止します。どのように実行されたのかをサービスコールトレースの表示情報を更新して確認してみてください。MainTask の cre_tsk 後に Task7 にスイッチしたので、MainTask はまだ wai_flg を発行していないことが分かります。

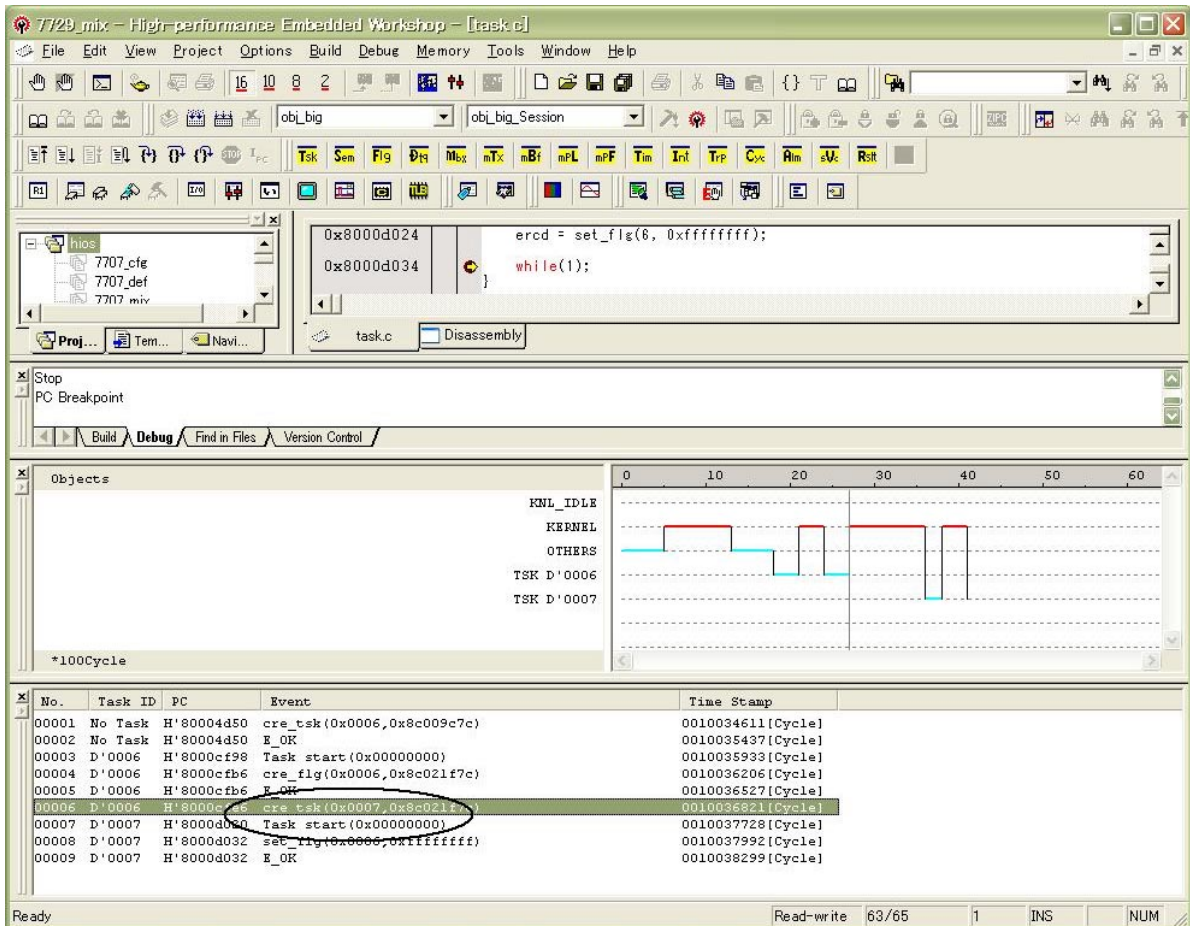


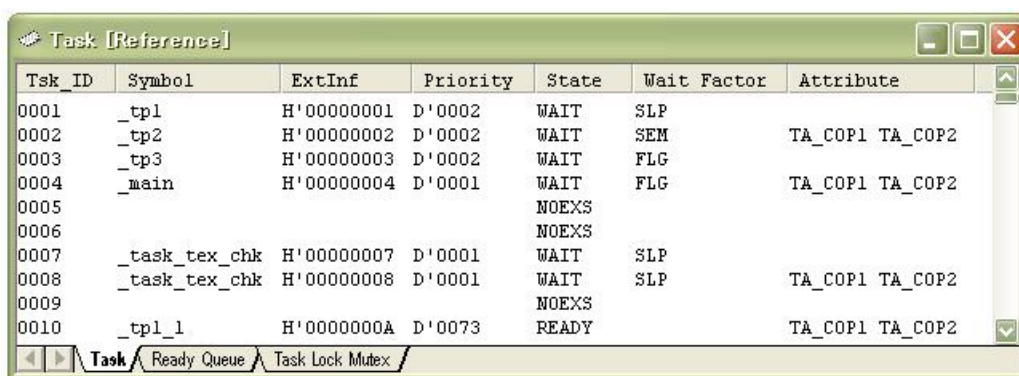
図2.20 [RTOS Trace Diagram]ウィンドウと[RTOS Trace Text]ウィンドウ

3. 機能

本章では、DXの機能概要について簡単に説明します。機能の詳細は、オンラインヘルプを参照してください。

3.1 オブジェクト状態の参照

[View->RTOS]メニューからタスク等の各オブジェクトを選択すると、図3.1のようにオブジェクトの状態を表示するウィンドウが開きます。



Tsk_ID	Symbol	ExtInf	Priority	State	Wait Factor	Attribute
0001	_tp1	H'00000001	D'0002	WAIT	SLP	
0002	_tp2	H'00000002	D'0002	WAIT	SEM	TA_COP1 TA_COP2
0003	_tp3	H'00000003	D'0002	WAIT	FLG	
0004	_main	H'00000004	D'0001	WAIT	FLG	TA_COP1 TA_COP2
0005				NOEXS		
0006				NOEXS		
0007	_task_tex_chk	H'00000007	D'0001	WAIT	SLP	
0008	_task_tex_chk	H'00000008	D'0001	WAIT	SLP	TA_COP1 TA_COP2
0009				NOEXS		
0010	_tp1_1	H'0000000A	D'0073	READY		TA_COP1 TA_COP2

図3.1 タスク状態の表示例

3.2 オブジェクトの操作

各オブジェクトのウィンドウからは、オブジェクトの操作をカーネルに要求することができます。この機能を使用するには、使用するRTOSにおいてDXのオブジェクト操作を行うための構築をする必要があります。

オブジェクト操作の要求は、図3.2に示すように各オブジェクトウィンドウのポップアップメニューから開いたダイアログボックスで行えます。

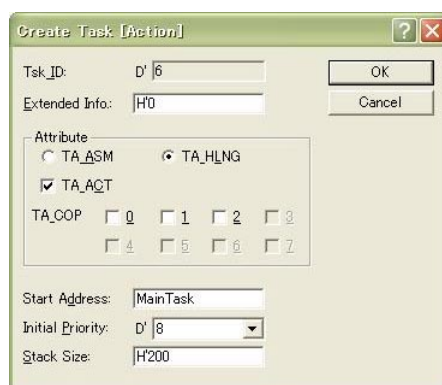
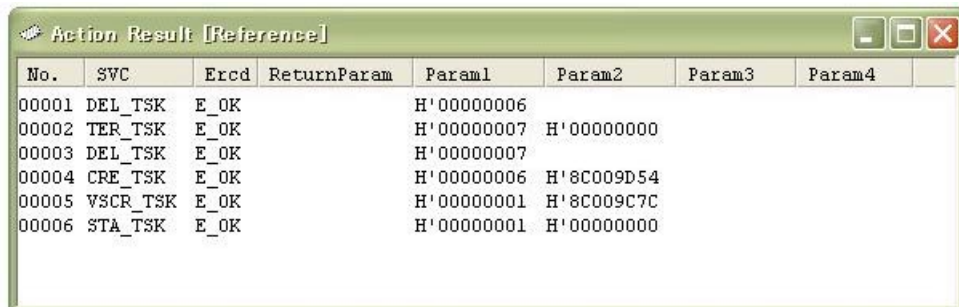


Figure 3.2 shows the 'Create Task [Action]' dialog box. It contains the following fields and options:

- Tsk_ID: D'6
- Extended Info: H0
- Attribute: TA_ASM (radio button), TA_HLNG (radio button), TA_AQT (checkbox checked)
- TA_COP: 0, 1, 2, 3, 4, 5, 6, 7 (checkboxes)
- Start Address: MainTask
- Initial Priority: D'8
- Stack Size: H'200

図3.2 オブジェクト操作の要求例

オブジェクト操作要求はターゲット側に伝えられ、対応するサービスコールが発行されます。DX は、その結果を図 3.3 に示す [Action Result] ウィンドウに表示します。



No.	SVC	ErCd	ReturnParam	Param1	Param2	Param3	Param4
00001	DEL_TSK	E_OK		H'00000006			
00002	TER_TSK	E_OK		H'00000007	H'00000000		
00003	DEL_TSK	E_OK		H'00000007			
00004	CRE_TSK	E_OK		H'00000006	H'8C009D54		
00005	VSCR_TSK	E_OK		H'00000001	H'8C009C7C		
00006	STA_TSK	E_OK		H'00000001	H'00000000		

図3.3 オブジェクト操作の結果

オブジェクト操作要求は、ターゲット実行時に初めて処理されます。ターゲット停止中は、オブジェクト操作の要求は可能ですが、その実行は次にターゲットを実行するまで待たされます。

オブジェクト操作要求は、ターゲットメモリ上にキューイングされ、[Action Result] ウィンドウに結果が表示されることで、キューイングから外れます。キューイングの最大数は 4 です。

ターゲット停止中のオブジェクト操作要求を行っても、次にターゲットを実行するまで決して [Action Result] ウィンドウに結果が表示されることは無いので、ターゲット停止中は同時に要求できるオブジェクト操作要求は 4 回になります。

なお、[Action Result] ウィンドウは、以下のタイミングで表示が更新されます。

- (1) ターゲット実行中 (周期的)
- (2) [Action Result] ウィンドウの右ボタンクリックによるポップアップメニューから [Refresh] を選択したとき
- (3) ターゲットシステムがブレーク (停止) したとき

3.3 サービスコールトレースの表示

[View->RTOS->Trace]を選択すると、図 3.4に示すようにサービスコールの履歴をダイアグラムおよびテキストで表示します。この機能を使用するには、使用する RTOS において DX のサービスコールトレースを表示するための構築を行う必要があります。

サービスコールトレースの履歴情報の元となるデータは、ツール（シミュレータ・デバッガ、エミュレータ）またはターゲットメモリどちらか選択できます。コンフィギュレータによっては「ツール」ではなく、「エミュレータ」と表示されている場合がありますが、シミュレータ・デバッガでも使用可能です。

また、取得された情報をファイルに保存し、そのファイルを読み出して表示することもできます。

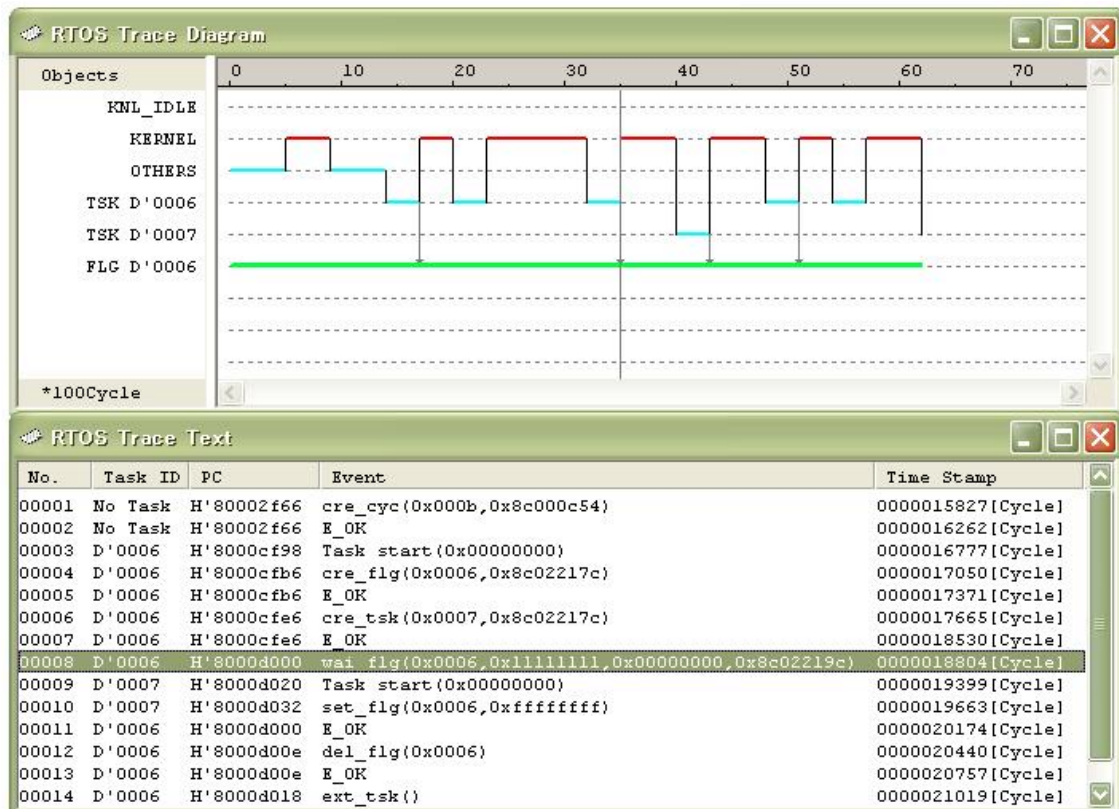


図3.4 [RTOS Trace Diagram]ウィンドウと[RTOS Trace Text]ウィンドウ

3.4 機能一覧

3.4.1 メニュー

DX をインストールすると、HEW の [View]メニューに表 3.1の項目が追加されます。これらから DX の機能にアクセスできます。

表3.1 HEW の [View]メニューに追加される内容

プルダウンメニュー	サブメニュー	機能
RTOS	Task	[Task]ウィンドウのオープン
	Semaphore	[Semaphore]ウィンドウのオープン
	EventFlag	[EventFlag]ウィンドウのオープン
	DataQueue	[DataQueue]ウィンドウのオープン
	Mailbox	[Mailbox]ウィンドウのオープン
	Mutex	[Mutex]ウィンドウのオープン
	MessageBuffer	[MessageBuffer]ウィンドウのオープン
	V-MemoryPool	[V-MemoryPool]ウィンドウのオープン
	F-MemoryPool	[F-MemoryPool]ウィンドウのオープン
	Timer	[Timer]ウィンドウのオープン
	Interrupt	[Interrupt]ウィンドウのオープン
	Trap	[Trap]ウィンドウのオープン
	Cyclic	[Cyclic]ウィンドウのオープン
	Alarm	[Alarm]ウィンドウのオープン
	Extended SVC	[Extended SVC]ウィンドウのオープン
	Action Result	[Action Result]ウィンドウのオープン
	Trece	[Trece]ウィンドウのオープン

また、[View->CPU->Status]を選択することにより、HEW 共通の[Status]ウィンドウに DX の状態を表示します。

なお、使用する RTOS によってはサポートされない機能があります。

3.4.2 ウィンドウとダイアログボックス

表 3.2に、DX のウィンドウ、ダイアログボックスの一覧を示します。ウィンドウ、ダイアログボックスの詳細についてはオンラインヘルプを参照してください。オンラインヘルプは、[Help->RTOS Help]を選択することによりオープンできます。

表3.2 ウィンドウとダイアログボックス一覧

分類	ウィンドウ名称	機能
タスク	[Task]ウィンドウ	・全タスクの状態表示 ・全タスク優先度の待ちタスク行列の表示 ・全タスクのロックされているミューテックスの表示
	[Task Detail Information]ウィンドウ	タスクの詳細情報表示
	[Create Task]ダイアログボックス ^{*1}	タスクの生成
	[Delete Task]ダイアログボックス ^{*1}	タスクの削除
	[Activate Task]ダイアログボックス ^{*1}	タスクの起動
	[Cancel Activate Task]ダイアログボックス ^{*1}	起動要求の無効化
	[Start Task]ダイアログボックス ^{*1}	タスクの起動（起動コード指定）
	[Terminate Task]ダイアログボックス ^{*1}	タスクの強制終了
	[Release Wait]ダイアログボックス ^{*1}	待ち状態の強制解除
	[Suspend Task]ダイアログボックス ^{*1}	強制待ちへの移行
	[Resume Task]ダイアログボックス ^{*1}	強制待ち状態タスクの再開
	[Wakeup Task]ダイアログボックス ^{*1}	タスクの起床
	[Cancel Wakeup Task]ダイアログボックス ^{*1}	起床要求の無効化
	[Change Task Priority]ダイアログボックス ^{*1}	タスク優先度の変更
	[Set Task EventFlag]ダイアログボックス ^{*1}	タスク付属イベントフラグのセット
	[Clear Task EventFlag]ダイアログボックス ^{*1}	タスク付属イベントフラグのクリア
	[Rotate Ready Queue]ダイアログボックス ^{*1}	レディキューの回転
	[Raise Task Exception Routine]ダイアログボックス ^{*1}	タスク例外処理の要求
[Start Overrun Handler]ダイアログボックス ^{*1}	オーバーランハンドラの動作開始	
[Stop Overrun Handler]ダイアログボックス ^{*1}	オーバーランハンドラの動作停止	

【注】 *1 使用する RTOS においてオブジェクト操作のための構築をしなかった場合は使用できません。

表 3.2 ウィンドウとダイアログボックス一覧 (続き)

分類	ウィンドウ名称	機能
セマフォ	[Semaphore]ウィンドウ	・全セマフォの状態表示 ・全セマフォの待ちタスク行列の表示
	[Increment Semaphore Count]ダイアログボックス ^{*1}	セマフォカウントの増加
	[Decrement Semaphore Count]ダイアログボックス ^{*1}	セマフォカウントの減少
イベントフラグ	[EventFlag]ウィンドウ	・全イベントフラグの状態表示 ・全イベントフラグの待ちタスク行列の表示
	[Set EventFlag]ダイアログボックス ^{*1}	イベントフラグのセット
	[Clear EventFlag]ダイアログボックス ^{*1}	イベントフラグのクリア
データキュー	[DataQueue]ウィンドウ	・全データキューの状態表示 ・全データキューの受信データの表示
	[Send Data to DataQueue]ダイアログボックス ^{*1}	データキューへのデータ送信
	[DataQueue Wait Task Queue]ウィンドウ	全データキューの待ちタスク行列の表示
メールボックス	[Mailbox]ウィンドウ	・全メールボックスの状態表示 ・全メールボックスの待ちタスク行列の表示 ・全メールボックスの受信メッセージ行列の表示
	[Send Message to Mailbox]ダイアログボックス ^{*1}	メールボックスへのメッセージ送信
	[Receive Message from Mailbox]ダイアログボックス ^{*1}	メールボックスからのメッセージ受信
ミューテックス	[Mutex]ウィンドウ	・全ミューテックスの状態表示 ・全ミューテックスの待ちタスク行列の表示
メッセージバッファ	[MessageBuffer]ウィンドウ	・全メッセージバッファの状態表示 ・全メッセージバッファの待ちタスク行列の表示 ・全メッセージバッファの受信メッセージ行列
	[Send Message to MessageBuffer]ダイアログボックス ^{*1}	メッセージバッファへのメッセージ送信
可変長メモリプール	[Variable-size MemoryPool]ウィンドウ	・全可変長メモリプールの状態表示 ・全可変長メモリプールの待ちタスク行列の表示
	[Get Variable-size MemoryPool]ダイアログボックス ^{*1}	可変長メモリブロックの獲得
	[Release Variable-size MemoryPool]ダイアログボックス ^{*1}	可変長メモリブロックの返却

【注】 *1 使用する RTOS においてオブジェクト操作のための構築をしなかった場合は使用できません。

表 3.2 ウィンドウとダイアログボックス一覧（続き）

分類	ウィンドウ名称	機能
固定長 メモリプール	[Fixed-size MemoryPool]ウィンドウ	・全固定長メモリプールの状態表示 ・全固定長メモリプールの待ちタスク行列の表示
	[Get Fixed-size MemoryPool]ダイアログボックス ^{*1}	固定長メモリブロックの獲得
	[Release Fixed-size MemoryPool]ダイアログボックス ^{*1}	固定長メモリブロックの返却
タイマ	[Timer]ウィンドウ	・システム時刻の表示 ・タイマへの待ち行列の表示
	[Set Time]ダイアログボックス ^{*1}	システム時刻の変更
割り込みハンドラ, 例外処理ルーチン	[Interrupt Handler]ウィンドウ	全割り込みハンドラ, 例外処理ルーチンの状態表示
トラップ例外処理ルーチン	[Trap Routine]ウィンドウ	全トラップ例外処理ルーチンの状態表示
周期ハンドラ	[Cyclic Handler]ウィンドウ	全周期ハンドラの状態表示
	[Start Cyclic Handler]ダイアログボックス ^{*1}	周期ハンドラの動作開始
	[Stop Cyclic Handler]ダイアログボックス ^{*1}	周期ハンドラの動作停止
アラーム ハンドラ	[Alarm Handler]ウィンドウ	全アラームハンドラの状態表示
	[Start Alarm Handler]ダイアログボックス ^{*1}	アラームハンドラの動作開始
	[Stop Alarm Handler]ダイアログボックス ^{*1}	アラームハンドラの動作停止
拡張 SVC ルーチン	[Extended SVC Routine]ウィンドウ	全拡張 SVC ルーチンの状態表示
結果	[Action Result]ウィンドウ ^{*1}	オブジェクト操作結果の表示
トレース	[RTOS Trace Diagram]ウィンドウ	・オブジェクトアイテムリストの表示 ・トレース情報の表示（図）
	[RTOS Trace Text]ウィンドウ	・オブジェクトアイテムリストの表示 ・トレース情報の表示（テキスト）
	[RTOS Trace Options]ダイアログボックス	トレースの各種設定
	[Select Diagram Object]ダイアログボックス	トレースダイアグラム表示の対象オブジェクトの登録
	[Select Text Object]ダイアログボックス	カーネルリソースのトレース情報の登録
	[Load RTOS Trace Information File]ダイアログボックス	トレース情報ファイルの入力
	[RTOS Trace Statistics]ダイアログボックス	プログラム実行内容の集計表示
	[RTOS Trace Find]ウィンドウ	イベント情報の検索
	[Save As]ダイアログボックス	トレース情報をファイルへ出力

【注】 *1 使用する RTOS においてオブジェクト操作のための構築をしなかった場合は使用できません。

4. 留意・制限事項

4.1 ターゲットシステムのリアルタイム性と周期ハンドラ

DX は、ターゲットシステムのメモリ内容を参照あるいは更新することで、その機能を実現しています。ターゲットシステムが実行中に、以下の機能を使用するとメモリアクセスが発生して、リアルタイム性がなくなります。

- ウィンドウをオープン、更新したとき
- ダイアログボックスで[OK]ボタンを押し、オブジェクト操作を行ったとき

また、使用する RTOS において DX のオブジェクト操作を行うための構築をした場合、自動的に一定周期の DX 用の周期ハンドラがひとつシステムに組み込まれるため、ターゲットのスループットがわずかに低下します。

4.2 表示内容の整合性

DX は、ターゲットシステムのメモリ内容を直接読み出すことでオブジェクトの状態参照機能を実現しています。このため、メモリ内容の読み出しが RTOS カーネル実行中に行われた場合、DX は正しい情報を表示しない場合があります。また、オブジェクトの状態参照をカーネル初期化完了前に行った場合は、DX は正しい情報を表示できません。

付録A トラブルシューティング

A.1 [Action Result]ウィンドウの結果表示

ダイアログボックスからオブジェクトの操作を要求し、ターゲットシステムを実行した後、結果が表示されない場合、DX用の周期ハンドラが正しく動作していないことが考えられます。その原因と対策を以下に示します。

- (1) タイマ割込みが入らない
 - タイマ割込みが発生していない。
タイマドライバを見直し、タイマ割込みが発生するようにしてください。
 - タイマ割込みは発生しているが受け付けられない。
次の場合には、タイマ割込みが受け付けられません。
 - タイマ割込みより高いレベルの割込みの発生頻度が高すぎる。
 - タイマ割込みレベルでマスクされた状態が長く続いている。
タイマ割込みレベル以上の割込みマスク期間はなるべく短くしてください。
- (2) システムダウンが発生した
システムがダウンすると、タイマ割込みが受け付けられない場合があります。
システムダウンが発生しないように対策してください。

デバッグングエクステンション ユーザーズマニュアル

発行年月 2003年9月8日 Rev.1.00

発行 株式会社ルネサス テクノロジ 営業企画統括部
〒100-0004 東京都千代田区大手町 2-6-2

編集 株式会社ルネサス小平セミコン 技術ドキュメント部

デバッグエクステンション
ユーザズマニュアル
HS6400IWIN1SJ



ルネサスエレクトロニクス株式会社
神奈川県川崎市中原区下沼部1753 〒211-8668

RJJ10B0032-0100H