

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。

# RX600 シリーズ用リアルタイム OS RI600/4

## アプリケーション移行ガイド(HI1000/4⇒RI600/4 編)

本ドキュメントでは、HI1000/4 の C 言語アプリケーションを RI600/4 に移行する際に有用な情報として、HI1000/4 から RI600/4 で変更になった仕様を中心に説明します。

### 目次

1. 概要.....	2
2. パラメータのデータ型とサイズ.....	3
3. RI600/4 で廃止された機能・縮退した機能.....	4
3.1 共有スタック機能の廃止.....	4
3.2 サービスコールトレース機能、および関連サービスコールの廃止.....	4
3.3 CPU例外ハンドラの廃止.....	4
3.4 初期化ルーチン機能の廃止.....	4
3.5 カーネルアイドル処理の非公開化.....	4
4. オブジェクトIDの制約.....	4
5. 可変長メモリプール.....	5
5.1 プールサイズ.....	5
5.2 獲得可能な最大ブロックサイズ.....	5
6. 割込み優先レベル、割込みマスク.....	5
7. 割込みマスク値によるE_CTXエラー.....	5
8. 多重割込み.....	6
9. タイマドライバ.....	6
10. システムダウン.....	6
11. コンパイラの拡張言語仕様.....	6
12. kernel_id.h.....	7
13. カーネル構成マクロおよびref_verで取得される情報.....	7
14. ビルド関連.....	8
14.1 コンフィギュレータの相違.....	8
14.2 ビルド手順の相違(mkritblユーティリティの追加).....	8
14.3 HI1000/4 からRI600/4 へのプロジェクトの移行時の留意事項.....	9
15. 仕様比較一覧.....	11

## 1. 概要

図 1に、HI1000/4 からRI600/4 へのアプリケーション資産の移行の概要を示します。

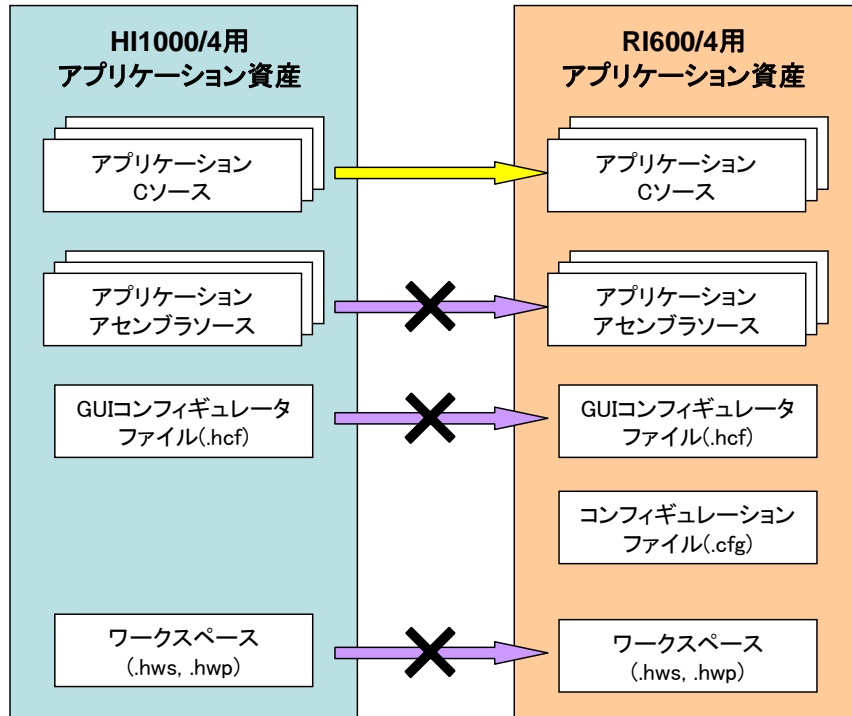


図1 アプリケーション資産の移行概要

### (1) アプリケーション C ソース

コンパイラ仕様差による部分は、アプリケーション C ソースの修正が必要な場合があります。  
これに加え、以降の節に記載の OS 仕様の差異について、必要な箇所の修正をしてください。

### (2) アプリケーションアセンブラソース

アセンブリ言語の互換性は無いため、新たに RX 用に書き直す必要があります。

### (3) GUI コンフィギュレータファイル(.hcf)

RI600/4 でも GUI コンフィギュレータをサポートしていますが、.hcf ファイルの互換性はありません。  
「14.1 コンフィギュレータの相違」を参照してください。

### (4) ワークスペース(.hws, .hwp)

High-performance Embedded Workshop の仕様により、H8SX, H8S ファミリー用に作成したワークスペースは、RX ファミリーでは使用できません。新たに RX 用のワークスペースを作成する必要があります。

## 2. パラメータのデータ型とサイズ

HI1000/4 と RI600/4 のパラメータのデータ型とサイズについて相違を表 1 に示します。  
 網掛け部分をアプリケーションで使用している場合は、使用箇所の見直しを行ってください。  
 特に、FLGPTN 型(イベントフラグのビットパターン)が、16 ビットから 32 ビットに変更されている点に注意してください。

表1 基本データ型の相違

	型	HI1000/4	備考	RI600/4	HI1000/4 との相違
itron.h	B	signed char B;		signed char B;	
	H	signed short H;		signed short H;	
	W	signed long W;		signed long W;	
	D	(未定義)	コンパイル非サポート	signed long long D;	追加
	UB	unsigned char UB;		unsigned char UB;	
	UH	unsigned short UH;		unsigned short UH;	
	UW	unsigned long UW;		unsigned long UW;	
	UD	(未定義)	コンパイル非サポート	unsigned long long UD;	追加
	VB	signed char VB;		signed char VB;	
	VH	signed short VH;		signed short VH;	
	VW	signed long VW;		signed long VW;	
	VD	(未定義)	コンパイル非サポート	signed long long VD;	追加
	VP	void *VP;		void *VP;	
	FP	void (*FP)(void);		void (*FP)(void);	
	INT	H INT;		W INT;	16bit⇒32bit
	UINT	UH UINT;		UW UINT;	16bit⇒32bit
	BOOL	H BOOL;		W BOOL;	16bit⇒32bit
	FN	H FN;	機能無し	(未定義)	削除
	ER	H ER;		W ER;	16bit⇒32bit
	ID	H ID;		H ID;	
	ATR	UH ATR;		UH ATR;	
	STAT	UH STAT;		UH STAT;	
	MODE	UH MODE;		UH MODE;	
	PRI	H PRI;		H PRI;	
	SIZE	UW SIZE;		UW SIZE;	
	TMO	W TMO;		W TMO;	
RELTIM	UW RELTIM;		UW RELTIM;		
SYSTIM	typedef struct { UH utime; UW ltime; } SYSTIM;		typedef struct { UH utime; UW ltime; } SYSTIM;		
VP_INT	W VP_INT;		W VP_INT;		
ER_BOOL	H ER_BOOL;	機能無し	(未定義)	削除	
ER_ID	H ER_ID;	機能無し	(未定義)	削除	
ER_UINT	H ER_UINT;		W ER_UINT;	16bit⇒32bit	
kernel.h	FLGPTN	UH FLGPTN;		UW FLGPTN	16bit⇒32bit
	INHNO	UH INHNO;	機能無し	(未定義)	削除
	EXCNO	UH EXCNO;	機能無し	(未定義)	削除
	IMASK	UH IMASK;		UH IMASK;	

### 3. RI600/4 で廃止された機能・縮退した機能

#### 3.1 共有スタック機能の廃止

RI600/4 では、共有スタック機能をサポートしていません。  
 代替機能もありません。

#### 3.2 サービスコールトレース機能、および関連サービスコールの廃止

RI600/4 では、サービスコールトレース機能をサポートしていません。  
 また、割込みハンドラ開始のトレース取得するサービスコール“`ivbgn_int`”、および割込みハンドラ終了のトレース取得するサービスコール“`ivend_int`”をサポートしていません。  
 これらのサービスコールを使用している場合、その記述を削除する必要があります。

#### 3.3 CPU 例外ハンドラの廃止

RI600/4 では、CPU 例外ハンドラをサポートしていません。  
 CPU 例外はカーネル管理外の割込みとして扱います。

#### 3.4 初期化ルーチン機能の廃止

HI1000/4 では、カーネル起動処理が終わって最初のタスクが実行する前にコンフィギュレータで登録した初期化ルーチンを実行することができますが、RI600/4 では本機能はサポートしていません。  
 初期化ルーチンと同等のやりたい場合は、以下のよう to してください。

- (1) 必要な初期化処理を行うタスクをひとつ設ける。コンフィギュレーション時には、そのタスクだけを「生成後、起動」と設定し、他のすべてのタスクには「生成」だけを設定する。
- (2) (1)のタスクの初期化処理が完了後、“`act_tsk`”または“`sta_tsk`”サービスコールを用いて他のタスクを起動する。

#### 3.5 システムアイドルルーチンの非公開化

HI1000/4 では、カーネルがアイドルに移行した時に `KERNEL_H.SYSTEM_IDLE()` という関数(HI1000/4 に付属のサンプルプログラム: `xxxx_idle.c`)が呼び出される仕様ですが、RI600/4 では本機能をサポートしていません。  
 無負荷時に行いたい処理がある場合は、最も優先度の低いタスクとして実装してください。

## 4. オブジェクト ID の制約

HI1000/4 では、各種オブジェクトの 1～`CFG.MAXxxxID`(各種オブジェクトの最大 ID)のうち、未使用(未生成)の ID 番号が存在することを許しており、未生成の ID 番号を指定してサービスコールを呼び出した場合は、`E.NOEXS` エラーが返る仕様となっています。

RI600/4 では、コンフィギュレーションで「各種オブジェクトの最大 ID」という項目はなく、各種オブジェクトは 1 から順に抜けなく ID 番号を使用する仕様に変更されました。このため、`E.NOEXS` エラーが発生することはなく、ID 抜けがある場合は `cfg600` でエラーを報告します。

## 5. 可変長メモリプール

### 5.1 プールサイズ

RI600/4 では、可変長メモリプールのメモリ管理方式が HI1000/4 から全面変更されています。

具体的には、RI600/4 では割り当てるメモリブロックサイズのバリエーションを限定することで、空き領域の断片化が HI1000/4 よりも発生しにくくしています。

しかし、状況によっては HI1000/4 よりもメモリの利用効率が悪化する可能性もあります。この場合は、メモリプールのサイズを増やす必要があります。

### 5.2 獲得可能な最大ブロックサイズ

HI1000/4 では、獲得可能な最大ブロックサイズはプールサイズ-16(バイト)となっています。

RI600/4 では、コンフィギュレーション時に獲得可能な最大ブロックサイズを定義する仕様に変更されました。なお、前節で述べたメモリブロックサイズのバリエーションは、この情報を元に決定されます。

詳細は、RI600/4 マニュアルの「8.4.12 可変長メモリプール定義(variable\_memorypool[ ])」を参照してください。

## 6. 割込み優先レベル、割込みマスク

HI1000/4 と RI600/4 では、対象 CPU コアが異なるため、以下で扱う割込み優先レベル・マスク値は意味が異なります。

ただし、両者ともに 0 が割込みマスク解除状態を示すという点では同じです。

- ・コンフィギュレーション時に指定するカーネル割込みマスクレベルとタイマ割込み優先レベル
- ・chg\_ims, ichg\_ims, get\_ims, iget\_ims で扱う割込みマスク

## 7. 割込みマスク値による E\_CTX エラー

HI1000/4 では、割込みマスクがカーネル割込みマスクレベルよりも高い状態でサービスコールを呼び出してはならない仕様となっています。

RI600/4 では、このような場合にリターン値として E\_CTX エラーを返す仕様となります。特に、以下のサービスコールでは、リターン値のデータ型が BOOL(=signed long)であるにも関わらず、E\_CTX エラーが返ることに注意してください。

sns\_ctx, sns\_loc, sns\_dsp, sns\_dpn

## 8. 多重割込み

HI1000/4 では、CPU の仕様からすべての割込みハンドラが多重割込みを許可する仕様となっています。

一方、RX600 の仕様では、多重割込みを禁止することもできるようになっています。

RI600/4 では、デフォルトでは割込みハンドラは多重割込みを禁止する設定になります。多重割込みを許可するには、コンフィギュレーション時にその指定を行う必要があります。詳細は、RI600/4 マニュアルの「8.4.15 可変ベクタ割込み定義 (interrupt\_vector[ ])」を参照してください。

## 9. タイマドライバ

HI1000/4 では、タイマドライバをユーザが作成する仕様で、サンプルとして代表的なマイコン用のタイマドライバを提供しています。

RI600/4 では、コンフィギュレータ(cfg600)がマイコン内蔵タイマ(CMT)用のタイマドライバを出力するため、通常はユーザがタイマドライバを作成する必要はありません。もちろん、ユーザが外部タイマ用のドライバを作成することも可能です。

ただし、RI600/4 では、タイマ割込みハンドラはカーネルライブラリ内に実装されているため、HI1000/4 のようにユーザ処理をタイマ割込みハンドラに追加することはできません。これが必要な場合は、周期ハンドラへの置き換えを行ってください。

また、タイマの初期化については、HI1000/4 では vsta\_knl 内で自動的に初期化される仕様となっていますが、RI600/4 ではカーネル起動(vsta\_knl)前にアプリケーション側でタイマを初期化する必要があります。具体的には、cfg600 が出力する“ri\_cmt.h”をインクルードし、\_RI\_init\_cmt()を呼び出してください。

## 10. システムダウン

### (1) 引数

システムダウンに渡される引数の情報量は、RI600/4 では「未定義割込み発生時のタスク ID」が渡らないことを除き、HI1000/4 とほぼ同様ですが、引数仕様が異なります。詳細は、HI1000/4 マニュアル「10. システムダウン時の情報」および RI600/4 マニュアル「6.6 システムダウンルーチン」にて確認してください。

ただし、RI600/4 で未定義割込み時のベクタ番号を引数として受け取るには、cfg600 で-U オプションの指定が必要です。詳細は、RI600/4 マニュアルの「8.5.4 コマンドオプション」を参照してください。

### (2)関数名とファイル

HI1000/4 では、サンプルプログラム xxxx\_sysdwn.c の vsys\_dwn( )がシステムダウンルーチンです。

RI600/4 では、High-performance Embedded Workshop の RI600/4 プロジェクト生成によって生成される resetprg.c にある \_RI\_sys\_dwn()がシステムダウンルーチンになります。

## 11. コンパイラの拡張言語仕様

H8SX/H8S で、#pragma などのコンパイラ専用の拡張言語仕様を使用していた場合は、コンパイラ仕様の相違により、その部分の見直しが必要です。RX 用コンパイラ(ccrx)では、これをチェックする“-check=ch38”オプションを準備しています。



## 12. kernel\_id.h

HI1000/4 では GUI コンフィギュレータが生成していますが、RI600/4 では cfg600 が生成するヘッダファイルになります。

- (1) HI1000/4では、アプリケーションでのkernel\_id.hのインクルードは任意ですが、RI600/4ではタスクやハンドラのエントリ関数でインクルードが必須となりました。
- (2) RI600/4では、kernel\_id.hにタスクやハンドラのエントリ関数のプロトタイプ宣言が出力されるようになっているため、アプリケーション側ではこれを記述しないようにしてください。
- (3) 割込みハンドラは、HI1000/4ではユーザが#pragma interrupt宣言を記述する必要がありますが、RI600/4ではkernel\_id.hに必要な#pragma宣言が出力されるようになっているため、アプリケーション側ではこれを記述しないようにしてください。

## 13. カーネル構成マクロおよび ref\_ver で取得される情報

HI1000/4 と RI600/4 では、表 2 に示すカーネル構成マクロの情報が異なります。

HI1000/4 の GUI コンフィギュレータは、表 2 のカーネル構成マクロを kernel.h からインクルードされる kernel\_macro.h に出力します。RI600/4 では cfg600 が同じマクロを kernel.h からはインクルードされていない kernel\_id.h に出力するように変更されています。

表 2 異なるカーネル構成マクロ

#	マクロ	説明
1	TIC_NUME	タイムティック分子
2	TIC_DENO	タイムティック分母
3	TMAX_TPRI	最大タスク優先度
4	TMAX_MPRI	最大メッセージ優先度
5	VTKNL_LVL	カーネル割込みマスクレベル
6	VTIM_LVL	タイマ割込みレベル

## 14. ビルド関連

### 14.1 コンフィギュレータの相違

HI1000/4 では、GUI コンフィギュレータが提供されています。

RI600/4 では、HI1000/4 と同様の Look & Feel の GUI コンフィギュレータに加え、コマンドラインコンフィギュレータ `cfg600` が追加されています。`cfg600` は、テキスト形式の `cfg` ファイルを入力とします。後者だけを使えば、コンフィギュレーション情報をテキストファイル形式の `cfg` ファイルに記述できるので、保守や変更管理が容易になります。

このことから RI600/4 では、GUI コンフィギュレータは `cfg` ファイルを生成するユーティリティツールの位置づけになります。

なお、HI1000/4 の GUI コンフィギュレータファイル(.hcf)を、RI600/4 で再利用することはできません。

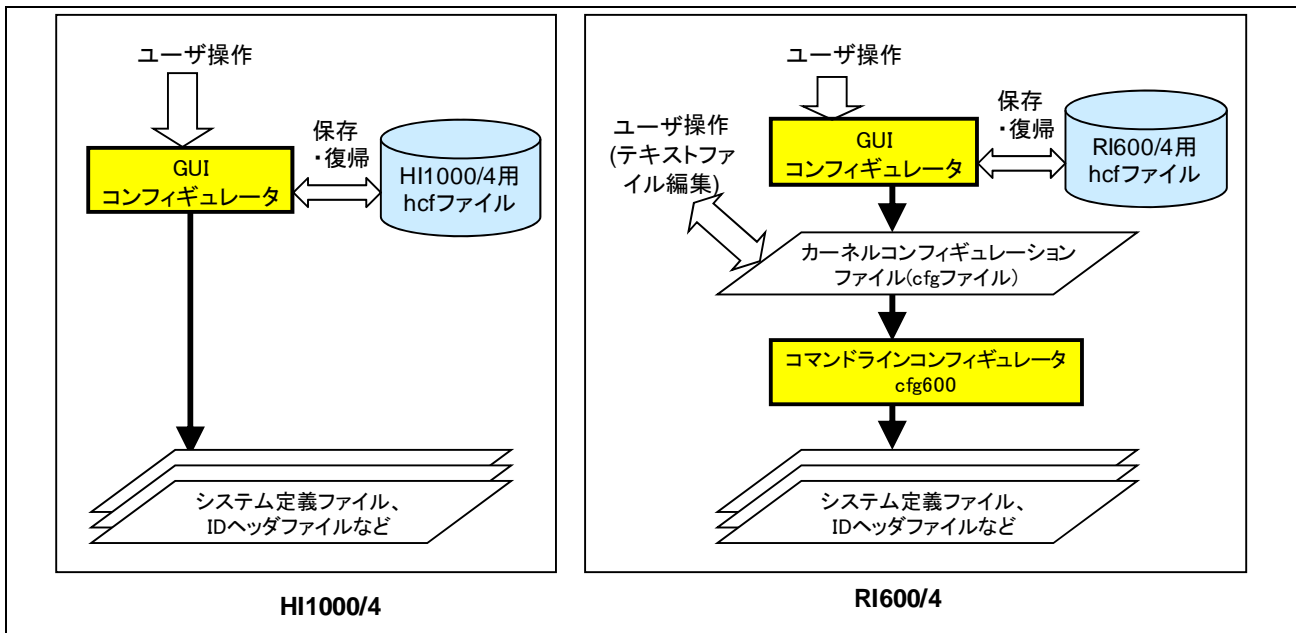


図1 コンフィギュレータの位置付けの変更

### 14.2 ビルド手順の相違(mkritbl ユーティリティの追加)

HI1000/4 のサービスコールは関数ライブラリとして実装されているため、カーネルライブラリとリンクすることで呼び出すことができます。

RI600/4 のサービスコールは、内部的には INT 命令で呼び出すようになっており、シンボルリンクではサービスコールモジュールがリンクできないため、RI600/4 では `mkritbl` と呼ぶユーティリティを用意しています。

アプリケーションをコンパイルすると、`mrc` ファイルが生成されます。`mrc` ファイルには、アプリケーションソース中で使用しているサービスコール情報が出力されます。そして、すべての `mrc` ファイルを `mkritbl` に入力することで、システムで使用しているサービスコールモジュールがリンクされるように定義されたファイル(`ritable.src`)が生成されます。詳細は、RI600/4 マニュアルの「9. テーブル生成ユーティリティ」を参照してください。

### 14.3 HI1000/4 から RI600/4 へのプロジェクトの移行時の留意事項

High-performance Embedded Workshop の仕様により、HI1000/4(H8S, H8SX)用のプロジェクトを RI600/4(RX600)用に流用することができないため、新たに RI600/4 プロジェクトを作成する必要があります。

RX ファミリー用 C/C++コンパイラパッケージ付属の High-performance Embedded Workshop では、RI600/4 用の雛型プロジェクトを生成することができます。この機能を用いてプロジェクトを生成し、必要なアプリケーションファイルを追加し、オプション設定の変更などを行ってください。

以下、特に留意すべき事項について解説します。

#### 14.3.1 リセット関係

##### (1) リセット関数名とソースファイル

HI1000/4 が提供するサンプルプログラム `xxxx_cpuasm.src` にある `_KERNEL_H_CPUINI` がリセット関数です。  
`_KERNEL_H_CPUINI` は、`xxxx_cpu.c` にある `h_cpuini()` を呼び出しています。

RI600/4 の場合は、High-performance Embedded Workshop の「RI600/4 用プロジェクトの生成」機能によって生成される `resetprg.c` にある `PowerON_Reset_PC()` がリセット関数です。

##### (2) リセットベクタの定義

リセットベクタは、HI1000/4 では GUI コンフィギュレータのベクタ#0 に定義する必要がありますが、RI600/4 では GUI コンフィギュレータまたは `cfg` ファイルで、固定ベクタ#31 に定義します。ただ、何も指定しない場合のデフォルトが `PowerON_Reset_PC()` という扱いのため、関数名を変更しない限りはユーザが明示的に固定ベクタ#31 を設定する必要はありません。

##### (3) RI600/4 での `resetprg.c` の位置付け

前述の `resetprg.c` は、リセット関数以外に、以下も含んでいます。

- ・システムダウンルーチン(「10. システムダウン」参照)
- ・`cfg600` が出力するシステム定義ファイル(`kernel_rom.h`, `kernel_ram.h`)のインクルード

なお、`kernel_rom.h`, `kernel_ram.h` は、位置付けとしては HI1000/4 の `kernel_setup.src` に相当します。

また、`resetprg.c` には、コンパイルオプション `"-nostuff"` の指定が必須です。

詳細は、RI600/4 マニュアルの「7.2 スタートアップファイル(`resetprg.c`)の作成」を参照してください。

#### 14.3.2 コンパイラオプション

RI600/4 では、`kernel.h` をインクルードする全ファイルについて、`"-ri600_preinit_mrc"` オプションの指定が必須です。

また、前述の `resetprg.c` には、`"-nostuff"` オプションの指定が必須です。

### 14.3.3 各種ファイルの対応関係

以下に、HI1000/4 のサンプルファイルとの対応関係を示しますので、参考にしてください。

#	概要	HI1000/4 のサンプル	RI600/4
1	共通型定義	(なし)	typedefine.h *
2	I/O 定義	(なし)	iodefine.h *
3	リセット	xxxx_cpuasm.src, xxxx_cpu.h, xxxx_cpu.c	resetprg.c *, hwsetup.c *
4	システムダウン	xxxx_sysdwn.c	resetprg.c *
5	未定義割込みハンドラ	xxxx_ilint.c	cfg600 が ri600.inc 内に生成
6	タイマドライバ	xxxx_tmdrv.c, xxxxtmdrv.c	cfg600 が生成する ri_cmt.h
7	アイドルリング	xxxx_idle.c	(該当機能なし)
8	セクション初期化	(なし)	dbstc.c *
9	標準入出力低水準関数	(なし)	lowsrc.h *, lowsrc.c *, lowlvl.src *
10	標準ライブラリ用メモリ管理低水準関数	(なし)	sbrk.h *, sbrk.c *
11	GUI コンフィギュレータファイル	xxxx.hcf	なし
12	cfg ファイル	(なし)	<プロジェクト名>.cfg
13	初期化ルーチン	GUI コンフィギュレータが生成する kernel_sysini.c(初期化ルーチン呼び出す関数)	(該当機能なし)
14	ベクタテーブル	GUI コンフィギュレータが生成する kernel_vector.src	cfg600 が vector.tpl を生成。これを mkritbl に入力することで、ritable.src 内にベクタテーブルが生成される。
15	ヘッダファイル	hihead%itron.h	inc600%itron.h
16	ヘッダファイル	hihead%kernel.h	inc600%kernel.h
17	ID 名称ヘッダファイル	GUI コンフィギュレータが生成する kernel_id.h	cfg600 が生成する kernel_id.h
18	アプリケーション用カーネルコンフィギュレーションマクロ	GUI コンフィギュレータが生成する kernel_macro.h(kernel.h からインクルードされる)	cfg600 が生成する kernel_id.h(kernel.h からインクルードされない)
19	システム定義ファイル	GUI コンフィギュレータが生成する kernel_setup.src	(1)cfg600 が生成する kernel_rom.h および kernel_ram.h(resetprg.c からインクルードされる) (2)cfg600 が生成する ri600.inc(mkritbl が出力する ritable.src からインクルードされる)
20	カーネルライブラリ	hilib ディレクトリに、動作モード等ごとに用意	lib600%ri600big.lib(ビッグエンディアン用) lib600%ri600lit.lib(リトルエンディアン用)

【注】\*は、High-performance Embedded Workshop の RI600/4 プロジェクト生成機能によって生成されるファイルです。

## 15. 仕様比較一覧

項目	HI1000/4	RI600/4	HI1000/4 との相違補足	備考	
コンフィギュレーション情報の入力	GUI コンフィギュレータに入力	以下のいずれか (1) GUI コンフィギュレータに入力 (2)cfg ファイルに記述	cfg ファイルに記述する方法を追加	「14.1 コンフィギュレータの相違」参照	
データ型	「3. パラメータのデータ型」参照				
オブジェクトの空き ID	可能	不可	不可に変更	「4. オブジェクトIDの制約」参照	
タスク管理	タスク ID	1~255	1~255		
	タスク優先度	1~31	1~255		
	起動回数	255	255		
	拡張情報	16bit	32bit	16bit⇒32bit	「3. パラメータのデータ型」参照参照
	属性	TA_HLNG TA_ASM	TA_HLNG TA_ASM		
	共有スタック	有り	なし	廃止	「3.1. 共有スタック機能」参照
	act_tsk	TEDU	TEDU		
	iact_tsk	NEDU	NEDU		
	can_act	TEDU	TEDU		
	ican_act	(未サポート)	NEDU	追加	
	sta_tsk	TEDU	TEDU		
	ista_tsk	NEDU	NEDU		
	ext_tsk	TEDUL	TEDUL		
	ter_tsk	TEDU	TEDU		
	chg_pri	TEDU	TEDU		
	ichg_pri	(未サポート)	NEDU	追加	
	get_pri	TEDU	TEDU		
	iget_pri	(未サポート)	NEDU	追加	
	ref_tsk	TEDU	TEDU		
	iref_tsk	NEDU	NEDU		
ref_tst	TEDU	TEDU			
iref_tst	NEDU	NEDU			
タスク付属同期	起床回数	255	255		
	サスペンド回数	1	1		
	slp_tsk	TEU	TEU		
	tslp_tsk	TEU	TEU		
	wup_tsk	TEDU	TEDU		
	iwup_tsk	NEDU	NEDU		
	can_wup	TEDU	TEDU		
	ican_wup	(未サポート)	NEDU	追加	
	rel_wai	TEDU	TEDU		
	irel_wai	NEDU	NEDU		
	sus_tsk	TEDU	TEDU		
	isus_tsk	(未サポート)	NEDU	追加	
	rsm_tsk	TEDU	TEDU		
	irms_tsk	(未サポート)	NEDU	追加	
	frsm_tsk	TEDU	TEDU		
	ifrm_tsk	(未サポート)	NEDU	追加	
	dly_tsk	TEU	TEU		
セマフォ	ID 番号	1~255	1~255		
	カウンタ最大値	65535	65535		
	属性	TA_TFIFO	TA_TFIFO TA_TPRI	TA_TPRI(待ち行列をタスク優先度順に管理)を追加	
	sig_sem	TEDU	TEDU		
	isig_sem	NEDU	NEDU		
	wai_sem	TEU	TEU		
	pol_sem	TEDU	TEDU		

項目		HI1000/4	RI600/4	HI1000/4 との相違補足	備考
	ipol_sem	NEDU	NEDU		
	twai_sem	TEU	TEU		
	ref_sem	TEDU	TEDU		
	iref_sem	NEDU	NEDU		
イベントフラグ	ID 番号	1~255	1~255		
	フラグ長	16bit	32bit	16bit⇒32bit	「3. パラメータのデータ型」参照参照
	属性	TA_TFIFO TA_WSGL TA_WMUL TA_CLR	TA_TFIFO TA_TPRI TA_WSGL TA_WMUL TA_CLR	TA_TPRI(待ち行列をタスク優先度順に管理)を追加	
	set_flg	TEDU	TEDU		
	iset_flg	NEDU	NEDU		
	clr_clg	TEDU	TEDU		
	iclr_flg	NEDU	NEDU		
	wai_flg	TEU	TEU		
	pol_flg	TEDU	TEDU		
	ipol_flg	NEDU	NEDU		
	twai_flg	TEU	TEU		
	ref_flg	TEDU	TEDU		
	iref_flg	NEDU	NEDU		
データキュー	ID 番号	1~255	1~255		
	データ長	32bit	32bit		
	属性	TA_TFIFO	TA_TFIFO TA_TPRI	TA_TPRI(待ち行列をタスク優先度順に管理)を追加	
	snd_dtq	TEU	TEU		
	psnd_dtq	TEDU	TEDU		
	ipsnd_dtq	NEDU	NEDU		
	tsnd_dtq	TEU	TEU		
	fsnd_dtq	TEDU	TEDU		
	ifsnd_dtq	NEDU	NEDU		
	rcv_dtq	TEU	TEU		
	prcv_dtq	TEDU	TEDU		
	iprcv_dtq	(未サポート)	NEDU	追加	
	trcv_dtq	TEU	TEU		
ref_dtq	TEDU	TEDU			
iref_dtq	NEDU	NEDU			
メールボックス	ID 番号	1~255	1~255		
	MSG 優先度	1~255	1~255		
	属性	TA_TFIFO TA_TPRI TA_MFIFO TA_MPRI	TA_TFIFO TA_TPRI TA_MFIFO TA_MPRI		
	snd_mbx	TEDU	TEDU		
	isnd_mbx	NEDU	NEDU		
	rcv_mbx	TEU	TEU		
	prcv_mbx	TEDU	TEDU		
	iprcv_mbx	NEDU	NEDU		
	trcv_mbx	TEU	TEU		
	ref_mbx	TEDU	TEDU		
iref_mbx	NEDU	NEDU			
ミューテックス	ID 番号	1~255	1~255		
	属性	TA_CEILING	TA_CEILING		
	loc_mtx	TEU	TEU		
	ploc_mtx	TEDU	TEDU		
	tloc_mtx	TEU	TEU		
	unl_mtrx	TEDU	TEDU		
ref_mtx	TEDU	TEDU			

項目		HI1000/4	RI600/4	HI1000/4 との相違補足	備考
メッセージ バッファ	ID 番号	(未サポート)	1~255	メッセージバッファ機能を追加	
	属性		TA_TFIFO		
	snd_mbf		TEU		
	psnd_mbf		TEDU		
	ipsnd_mbf		NEDU		
	tsnd_mbf		TEU		
	rcv_mbf		TEU		
	prcv_mbf		TEDU		
	trcv_mbf		TEU		
	ref_mbf		TEDU		
	iref_mbf		NEDU		
固定長 メモリ プール	ID 番号	1~255	1~255		
	ブロック数上限	65535	65535		
	ブロックサイズ 上限	65530	65535		
	属性	TA_TFIFO	TA_TFIFO TA_TPRI	TA_TPRI(待ち行列をタスク優先度順 に管理)を追加	
	get_mpf	TEU	TEU		
	pget_mpf	TEDU	TEDU		
	ipget_mpf	NEDU	NEDU		
	tget_mpf	TEU	TEU		
	rel_mpf	TEDU	TEDU		
	irel_mpf	(未サポート)	NEDU	追加	
	iref_mpf	NEDU	NEDU		
可変長 メモリ プール	ID 番号	1~255	1~255		
	管理方式	「5.1 プールサイズ」参照			
	プールサイズ上限	65534	256MB	・64kB⇒256MB	
	獲得可能な最大 ブロックサイズ	プールサイズ-16	コンフィギュレーション定義	獲得可能な最大ブロックサイズをコンフィ ギュレーション定義する仕様に変更	「5.2 獲得可能な最大」参照
	属性	TA_TFIFO	TA_TFIFO		
	get_mpl	TEU	TEU		
	pget_mpl	TEDU	TEDU		
	ipget_mpl	NEDU	NEDU		
	tget_mpl	TEU	TEU		
	rel_mpl	TEDU	TEDU		
	iref_mpl	NEDU	NEDU		
時間管理	システム時刻	符号なし 48bit	符号なし 48bit		
	単位時間	1ms	1ms		
	システム時刻 の更新周期	TIC_NUME /TIC_DENO[ms]	TIC_NUME /TIC_DENO[ms]		
	set_tim	TEDU	TEDU		
	iset_tim	NEDU	NEDU		
	get_tim	TEDU	TEDU		
	iget_tim	NEDU	NEDU		
周期 ハンドラ	ID 番号	1~254	1~255		
	拡張情報	32bit	32bit		
	属性	TA_HLNG TA_ASM TA_STA TA_PHS	TA_HLNG TA_ASM TA_STA TA_PHS		
	sta_cyc	TEDU	TEDU		
	ista_cyc	NEDU	NEDU		
	stp_cyc	TEDU	TEDU		
	istp_cyc	NEDU	NEDU		
	iref_cyc	NEDU	NEDU		
アラーム	ID 番号	(未サポート)	1~255	アラームハンドラ機能を追加	

項目		HI1000/4	RI600/4	HI1000/4 との相違補足	備考
ハンドラ	拡張情報		32bit		
	属性		TA_HLNG, TA_ASM		
	sta_alm		TEDU		
	ista_alm		NEDU		
	stp_alm		TEDU		
	istp_alm		NEDU		
	ref_alm		TEDU		
	iref_alm		NEDU		
システム状態 管理	rot_rdq	TEDU	TEDU		
	irotd_rdq	NEDU	NEDU		
	get_tid	TEDU	TEDU		
	iget_tid	NEDUC	NEDU		
	loc_cpu	TEDUL	TEDUL		
	iloc_cpu	NEDUL	NEDUL		
	unl_cpu	TEDUL	TEDUL		
	iunl_cpu	NEDUL	NEDUL		
	dis_dsp	TEDU	TEDU		
	ena_dsp	TEDU	TEDU		
	sns_ctx	TNEDULC	TNEDUL		
	sns_loc	TNEDULC	TNEDUL		
	sns_dsp	TNEDULC	TNEDUL		
	sns_dpn	TNEDULC	TNEDUL		
	vsta_knl	サポート	サポート		
	ivsta_knl	サポート	サポート		
	vsys_dwn	TEDUL	TEDUL		
ivsys_dwn	NEDULC	NEDUL			
ivbgn_int	NEDU	(未サポート)	廃止	「3.2 サービスコールトレース機能、および関連サービスコールの廃止」参照	
ivend_ont	NEDU				
割り込み管理	chg_ims	TEU	TEDU		「6. 割り込み優先レベル、割り込みマスク」参照
	ichg_ims	NEU	NEDU		
	get_ims	TEDU	TEDU		
	iget_ims	NEDU	NEDU		
	ret_int	N	N		
システム構成管理	CPU 例外ハンドラ	サポート	カーネル管理外割り込みの扱い	「3.3. CPU例外ハンドラ」参照	
	ref_ver	TEDU	TEDU		
	iref_ver	NEDU	NEDU		
オブジェクトリセット機能	vrst_dtq	(未サポート)	TEDU	オブジェクトを初期状態に戻す機能を追加	
	vrst_mbx		TEDU		
	vrst_mpf		TEDU		
	vrst_mpl		TEDU		
	vrst_mbf		TEDU		

- 【注】 N:非タスクコンテキストから呼び出し可能  
 T:タスクコンテキストから呼び出し可能  
 E:ディスパッチ許可状態から呼び出し可能  
 D:ディスパッチ禁止状態から呼び出し可能  
 U:CPU ロック解除状態から呼び出し可能  
 L:CPU ロック状態から呼び出し可能  
 C:CPU 例外ハンドラから呼び出し可能(HI1000/4 のみ)



## ホームページとサポート窓口

ルネサス テクノロジホームページ

<http://japan.renesas.com/>

お問い合わせ先

<http://japan.renesas.com/inquiry>

[csc@renesas.com](mailto:csc@renesas.com)

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
0.01	2008.12.22	—	初版発行[限定配布]
0.70	2009.07.08	—	全面改訂[限定配布]
1.00	2009.10.07	—	正式リリース

すべての商標および登録商標は、それぞれの所有者に帰属します。

### 本資料ご利用に際しての留意事項

1. 本資料は、お客様に用途に応じた適切な弊社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弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証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など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の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、あるいはその他軍事用途の目的で使用しないでください。また、輸出に際しては、「外国為替および外国貿易法」その他輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
4. 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などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりましては、事前に弊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、弊社ホームページ (<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。
5. 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のですが、万一本資料の記述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、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独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。弊社は、適用可否に対する責任を負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会ください。なお、上記用途に使用されたことにより発生した損害等について弊社はその責任を負いかねますのでご了承願います。
8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないでください。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
  - 1) 生命維持装置。
  - 2) 人体に埋め込み使用するもの。
  - 3) 治療行為（患部切り出し、薬剤投与等）を行うもの。
  - 4) その他、直接人命に影響を与えるもの。
9. 本資料に記載された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、実装条件およびその他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて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、弊社はその責任を負いません。
10. 弊社は製品の品質および信頼性の向上に努めておりますが、特に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。弊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、誤動作防止設計などの安全設計（含むハードウェアおよびソフトウェア）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。
11. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなきよう、お客様の責任において十分な安全設計をお願いします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
12. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断りいたします。
13. 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弊社営業窓口までご照会ください。

D039444