

RI600/4 V.1.01 Release 00

R20UT0797JJ0100

リリースノート

Rev.1.00

2011.10.14

この度は、RI600/4 をご使用いただきまして、誠にありがとうございます。

本資料では、本製品をお使いいただく上での制限事項および注意事項を記載しております。ご使用前に、必ずお読みくださいますようお願い申し上げます。

目次

1.	製品構成.....	2
2.	ツールニュース.....	2
3.	ユーザ登録のお願い.....	3
4.	対象デバイスについて.....	4
5.	動作環境.....	4
5.1	ハードウェア環境.....	4
5.2	ソフトウェア環境.....	4
5.3	対応ツール.....	4
6.	インストール・アンインストール.....	4
7.	タイマテンプレートファイル.....	5
8.	カーネルソースコードのビルド方法.....	5
9.	スタック使用量について.....	6
9.1	システムクロック割込みハンドラのスタック使用量(ϵ_1 , ϵ_2 , ϵ_3).....	6
9.2	サービスクールスタック使用量(α).....	6
9.3	カーネルライブラリをビルドした場合.....	9
10.	前バージョンからの変更点.....	10
10.1	制限事項の解除.....	10
10.2	isus_tsk.....	10
10.3	リアルタイム OS デバッグ機能対応.....	10
10.4	割込みハンドラでの ACC レジスタ保証.....	10
10.5	固定ベクタ割込み定義 (interrupt_fvector[]).....	11
10.6	タイマテンプレートファイルの追加.....	11
10.7	バージョン情報.....	11
11.	注意事項.....	12
11.1	RX610 グループ使用時の注意事項.....	12
11.2	High-performance Embedded Workshop に関する注意事項.....	12
11.3	0 番地に関する注意事項.....	12
12.	制限事項.....	12
13.	ドキュメント訂正.....	13

1. 製品構成

RI600/4 は型名により、契約形態と提供物が異なります。

型名	契約形態	提供物
R0R5RX00TRW011	評価契約, インストール可能な PC は 1 台	A
R0R5RX00TRW01A	評価契約, インストール可能な PC は無制限	A
R0R5RX00TRW01K	量産契約, 量産数は 3000 台まで	A
R0R5RX00TRW01U	量産契約, 量産数は無制限	A
R0R5RX00TRW01Z	量産契約, 量産数は無制限, ソースコード付き	B

提供物は以下となります。

提供物	ツール名	
B	A	カーネル オブジェクト
		コマンドラインコンフィギュレータ cfg600
		GUI コンフィギュレータ GUI600
		テーブル生成ユーティリティ mkritbl
	カーネル ソースコード	

2. ツールニュース

ツールニュースは、ツール製品を購入いただいたお客様によりよくご使用いただくために、製品の情報や各種案内を提供することを目的として発行しています。

ツールニュースで新製品の情報、バージョンアップの情報、使用上のご注意等、最新の情報を入手いただき、開発にお役立てください。

特に、本バージョンリリース後の情報は本リリースノートには掲載されないため、必ず最新のツールニュースを確認してください。ツールニュースは以下の 2 種類の形で提供しています。

- 電子メール版ツールニュース
製品をご購入いただき、なおかつユーザ登録いただいたお客様に配信しています。(月 2 回発行)
- Web版ツールニュース
すべてのお客様に閲覧いただけます。(月 2 回更新, 5 日と 20 日頃)
電子メール版ツールニュースを発行した約 3 日~1 週間後に閲覧可能となります。
URL : <http://tool-support.renesas.com/jpn/toolnews/index.htm>

3. ユーザ登録のお願い

バージョンアップ情報や技術サポートなどのサービスを受けるためにユーザ登録を行って下さい。ユーザ登録をされていない場合は、これらのサービスを受けることができません。また、ご購入後 30 日以内に登録して下さいようお願い申し上げます。

ユーザ登録方法

インストールすると、インストールディレクトリ下に、"support¥ri600-4¥regist.txt"が生成されます。このファイルにシリアル番号を記入の上、すべてカット& ペーストして以下の電子メールアドレス宛に送付して下さい。

【ユーザ登録窓口】

regist_tool@renesas.com

※ルネサスエレクトロニクス(株)の個人情報保護方針につきましては、以下をご覧ください。

ホームページ : <http://japan.renesas.com/privacy/>

ユーザ登録でご提供頂きました個人情報は、お客様のサポート活動に活用させて頂き、そのために必要な範囲でルネサスエレクトロニクス(株)、およびその関係会社、ならびに特約店に、電子データ、書面により提供させて頂きますので、ご了承の程お願い申し上げます。なお、提供を希望されない場合は、提供を停止させて頂きますので、お問い合わせ時にその旨ご連絡ください。その場合、サポート範囲が制約される場合がございます。

4. 対象デバイスについて

本製品は、以下のデバイスに対応しています。

RX600 シリーズ MCU
RX200 シリーズ MCU

5. 動作環境

本製品を使用するには、次の環境が必要になります。

5.1 ハードウェア環境

メモリ容量：推奨 256MB 以上。最低 128MB 以上
ディスプレイ：800×600 以上の解像度

5.2 ソフトウェア環境

次のソフトウェア環境に対応しています。

Windows XP (32bit 版のみ)
Windows Vista (32bit 版, 64bit 版)
Windows 7 (32bit 版, 64bit 版)

いずれの場合も、最新の Service Pack がインストールされていることを推奨します。

5.3 対応ツール

本製品は次の開発ツールに対応しています。

ツール名	バージョン
RX ファミリー用 C/C++コンパイラパッケージ	V1.00 Release 00以降

V1.01 Release 00以降

ただし、High-performance Embedded Workshop で RI600/4 対応プロジェクトを生成するには、RX ファミリー用 C/C++コンパイラパッケージ V.1.02 Release 00 以降が必要です。これ以前のバージョンで生成した RI600/4 プロジェクトでは、ビルド設定が不適切になります。

6. インストール・アンインストール

インストール、アンインストールには、Windows の管理者権限が必要です。

インストールするには、提供する CD のルートディレクトリにある setup.exe を起動し、画面に表示されるインストールの指示に従ってください。インストール時は、あらかじめ他のアプリケーションを終了させておいてください。

アンインストールするには、コントロールパネルの[プログラムの追加と削除]から、「RI600/4 V.1.01 Release 00」を選択してください。

7. タイマプレートファイル

以下に、RI600/4 が提供するタイマプレートファイルと、対応している MCU を示します。

なお、タイマプレートファイルは、コンフィギュレーションファイルの"clock.template"に指定するファイルです。

プレートファイル	対応 MCU
rx610.tpl	RX600 シリーズ RX610 グループ
rx62t.tpl	RX600 シリーズ RX62T グループ
rx62n.tpl	RX600 シリーズ RX62N グループ
rx630.tpl	RX600 シリーズ RX630 グループ
rx210.tpl	RX200 シリーズ RX210 グループ

8. カーネルソースコードのビルド方法¹

カーネルのソースコードは、"<インストールディレクトリ>%src600"に格納されます。カーネルをビルドするためには、ソースコードインストールディレクトリに移動し、"nmake.exe"²を実行します。

カーネルビルド時には、コンパイラが必要とする環境変数の設定が必要です。

ビルド例：

```
C:\%RI600-4%v101r00%src600> nmake (RET)
```

カーネルライブラリのビルド後、カーネルライブラリは以下のディレクトリに生成されます。

カーネルライブラリ名	内容
product%big%debug%ri600big.lib	デバッグ情報付きビッグエンディアンライブラリ
product%big%release%ri600big.lib	デバッグ情報なしビッグエンディアンライブラリ
product%little%debug%ri600lit.lib	デバッグ情報付きリトルエンディアンライブラリ
product%little%release%ri600lit.lib	デバッグ情報なしリトルエンディアンライブラリ

製品インストールディレクトリに対する書き込み権限がない場合、"src600"ディレクトリを書き込み可能なディレクトリにコピーしてビルドしてください。ビルド後、製品インストールディレクトリに対する書き込み権限のあるユーザにて、生成されたライブラリをインストールディレクトリ下の"lib600"ディレクトリにコピーしてください。

¹ ソースコードが付属するのは、ソース付き量産契約版(R0R5RX00TRW01Z)のみです。

² "nmake.exe"は、米国 Microsoft Corporation により提供されるプロジェクトをビルドするためのツールです。"nmake.exe"は、Microsoft Visual Studio 2008 等に含まれています。

9. スタック使用量について

9.1 システムクロック割込みハンドラのスタック使用量($\epsilon 1$, $\epsilon 2$, $\epsilon 3$)

RI600/4 ユーザーズマニュアルの 12.4 節に記載の $\epsilon 1$, $\epsilon 2$ および $\epsilon 3$ の値は、以下の通りです。

$\epsilon 1 = 104$
 $\epsilon 2 = 104$
 $\epsilon 3 = 192$

9.2 サービスコールのスタック使用量(α)

サービスコールでは、以下のようにスタックを使用します。

(1) タスクコンテキストから呼び出された場合

タスクコンテキスト実行中のスタックはユーザスタックです。サービスコールでは、

- (a)ユーザスタック(呼び出し元スタック)
- (b)システムスタック

を使用します。

(2)非タスクコンテキストから呼び出された場合

非タスクコンテキスト実行中のスタックはシステムスタックです。サービスコールでは、

- (c)システムスタック(呼び出し元スタック)

を使用します。

サービスコールが使用する呼び出し元のスタック((a),(c))の使用量は、Call Walker によって表示されます。

また、(b)および(c)のサイズは、マニュアル 12.4 節に記載のようにシステムスタックの使用量を算出するために必要となります (12.4 節では α と表記しています)。以下に、各サービスコールの(a)~(c)のサイズを示します。

	サービスコール	ユーザスタック 使用サイズ(a)	システムスタック 使用サイズ((b), (c))	備考
タスク管理機能				
1	act_tsk	0	24	
2	iact_tsk	0	24	
3	can_act	0	24	
4	ican_act	0	24	
5	sta_tsk	0	24	
6	ista_tsk	0	24	
7	ext_tsk	0	64	タスク開始関数からのリターン時にも ext_tsk が呼び出されます。
8	ter_tsk	0	128	
9	chg_pri	0	36	
10	ichg_pri	0	52	
11	get_pri	0	24	
12	iget_pri	0	24	
13	ref_tsk	0	28	
14	iref_tsk	0	28	
15	ref_tst	0	24	
16	iref_tst	0	24	
タスク付属同期機能				

	サービスコール	ユーザスタック 使用サイズ(a)	システムスタック 使用サイズ((b), (c))	備考
17	slp_tsk	0	24	
18	tslp_tsk	0	24	
19	wup_tsk	0	40	
20	iwup_tsk	0	52	
21	can_wup	0	24	
22	ican_wup	0	24	
23	rel_wai	0	112	
24	irel_wai	0	128	
25	sus_tsk	0	24	
26	isus_tsk	0	24	
27	rsm_tsk	0	24	
28	irms_tsk	0	24	
29	frsm_tsk	0	24	
30	ifrs_tsk	0	24	
31	dly_tsk	0	24	
セマフォ				
32	sig_sem	0	44	
33	isig_sem	0	60	
34	wai_sem	0	32	
35	pol_sem	0	24	
36	ipol_sem	0	24	
37	twai_sem	0	36	
38	ref_sem	0	24	
39	iref_sem	0	24	
イベントフラグ				
40	set_flg	0	48	
41	iset_flg	0	64	
42	clr_flg	0	24	
43	iclr_flg	0	24	
44	wai_flg	0	44	
45	pol_flg	0	24	
46	ipol_flg	0	24	
47	twai_flg	0	48	
48	ref_flg	0	24	
49	iref_flg	0	24	
データキュー				
50	snd_dtq	0	36	
51	psnd_dtq	0	32	
52	ipsnd_dtq	0	52	
53	tsnd_dtq	0	40	
54	fsnd_dtq	0	32	
55	ifsnd_dtq	0	52	
56	rcv_dtq	0	32	
57	prcv_dtq	0	32	
58	iprcv_dtq	0	52	
59	trcv_dtq	0	32	
60	ref_dtq	0	24	
61	iref_dtq	0	24	
メールボックス				
62	snd_mbx	0	40	
63	isnd_mbx	0	56	
64	rcv_mbx	0	32	
65	prcv_mbx	0	28	
66	iprcv_mbx	0	28	
67	trcv_mbx	0	36	

	サービスコール	ユーザスタック 使用サイズ(a)	システムスタック 使用サイズ((b), (c))	備考
68	ref_mbx	0	24	
69	iref_mbx	0	24	
ミューテックス				
70	loc_mtx	0	40	
71	ploc_mtx	0	24	
72	tloc_mtx	0	44	
73	unl_mtx	0	56	
74	ref_mtx	0	24	
メッセージバッファ				
75	snd_mbf	0	44	
76	psnd_mbf	0	44	
77	ipsnd_mbf	0	60	
78	tsnd_mbf	0	44	
79	rcv_mbf	0	56	
80	prcv_mbf	0	56	
81	trcv_mbf	0	56	
82	ref_mbf	0	24	
83	iref_mbf	0	24	
固定長メモリプール				
84	get_mpf	0	48	
85	pget_mpf	0	36	
86	ipget_mpf	0	36	
87	tget_mpf	0	48	
88	rel_mpf	20	36	
89	irel_mpf	0	56	
90	ref_mpf	0	24	
91	iref_mpf	0	24	
可変長メモリプール				
92	get_mpl	36	92	
93	pget_mpl	0	108	
94	ipget_mpl	0	108	
95	tget_mpl	36	92	
96	rel_mpl	0	104	
97	ref_mpl	0	24	
98	iref_mpl	0	24	
時間管理機能				
99	set_tim	0	24	
100	iset_tim	0	24	
101	get_tim	0	24	
102	iget_tim	0	24	
周期ハンドラ				
103	sta_cyc	0	24	
104	ista_cyc	0	24	
105	stp_cyc	0	24	
106	istp_cyc	0	24	
107	ref_cyc	0	24	
108	iref_cyc	0	24	
アラームハンドラ				
109	sta_alm	0	24	
110	ista_alm	0	24	
111	stp_alm	0	24	
112	istp_alm	0	24	
113	ref_alm	0	24	
114	iref_alm	0	24	
システム状態管理機能				

	サービスコール	ユーザスタック 使用サイズ(a)	システムスタック 使用サイズ((b), (c))	備考
115	rot_rdq	0	24	
116	irotd_rdq	0	24	
117	get_tid	0	24	
118	iget_tid	0	24	
119	loc_cpu	0	24	
120	iloc_cpu	0	24	
121	unl_cpu	0	24	
122	iunl_cpu	0	24	
123	dis_dsp	0	24	
124	ena_dsp	0	24	
125	sns_ctx	0	24	
126	sns_loc	0	24	
127	sns_dsp	0	24	
128	sns_dpn	0	24	
129	vsta_knl	0	64	システムスタックポインタを初期化後に使 用します。
130	ivsta_knl	0	64	
131	vsys_dwn	0	16	
132	ivsys_dwn	0	16	
割込み管理機能				
133	chg_ims	0	28	
134	ichg_ims	0	16	
135	get_ims	4	0	
136	iget_ims	0	4	
137	ret_int	0	32	
システム構成管理機能				
138	ref_ver	0	24	
139	iref_ver	0	24	
オブジェクトリセット機能				
140	vrst_dtq	0	48	
141	vrst_mbx	0	24	
142	vrst_mbf	0	48	
143	vrst_mpf	0	48	
144	vrst_mpl	0	68	

9.3 カーネルライブラリをビルドした場合

コンパイラのバージョンやオプション設定を変更してカーネルライブラリをビルドした場合、カーネルのスタック使用量が変わる場合があるので、注意してください。

10. 前バージョンからの変更点

本章では、前バージョン(V.1.00 Release 02)からの変更点を説明します。

10.1 制限事項の解除

以下の制限事項を解除しました。制限事項の詳細は、該当のツールニュースを参照してください。

- (1) loc_mtx, tloc_mtx, chg_priの問題
ツールニュースURL : <http://tool-support.renesas.com/jpn/toolnews/date2011.htm#d111101>
- (2) irel_mpfの問題
ツールニュースURL : <http://tool-support.renesas.com/jpn/toolnews/date2011.htm#d111101>
- (3) ディスパッチ禁止状態の問題
ツールニュースURL : <http://tool-support.renesas.com/jpn/toolnews/date2011.htm#d111101>
- (4) GUIコンフィギュレータが異常終了する問題
ツールニュースURL : <http://tool-support.renesas.com/jpn/toolnews/date2011.htm#d111101>

10.2 isus_tsk

ディスパッチ禁止状態の時に isus_tsk で実行状態のタスクを指定した場合に、E_OBJ エラーを返す仕様に変更しました。

10.3 リアルタイムOSデバッグ機能対応

High-performance Embedded Workshop のリアルタイム OS デバッグ機能の OS トレースが利用可能になりました。

なお、割り込みハンドラをトレースするには、割り込みハンドラ定義時に”pragma_switch=H”の指定が必要です。詳細は、本資料の「13(13)「8.4.15 (4) PRAGMA拡張機能に渡すスイッチ(pragma_switch)」および「13(14)「8.4.16 固定ベクタ割り込み定義(interrupt_fvector[])」」を参照してください。

リアルタイム OS デバッグ機能については、以下を参照してください。

http://japan.renesas.com/products/tools/os/os_aware_debugging/ecxos/osaware_hew_mid_level_landing.jsp

本機能を使用するには RX ファミリ用 C/C++コンパイラパッケージ V.1.02 Release 00 以降が必要です。

10.4 割り込みハンドラでのACCレジスタ保証

interrupt_vector[],pragma_switch および interrupt_fvector[],pragma_switch に、以下の指定ができるようになりました。

ACC : 割り込みハンドラで ACC レジスタを保証する"acc" スイッチを渡します。

NOACC : 割り込みハンドラで ACC レジスタを保証しない"no_acc" スイッチを渡します。

本機能を使用するには RX ファミリ用 C/C++コンパイラパッケージ V.1.01 Release 00 以降が必要です。

10.5 固定ベクタ割込み定義 (interrupt_fvector[])

(1) entry_address に数値を指定可能

前バージョンでは、entry_address には関数のみを指定可能でしたが、本バージョンでは数値も指定可能となりました。これにより、固定ベクタ領域 (0xFFFFFFFF80~0xFFFFFFFF 番地) に割り当てられたレジスタの設定が容易になりました。

(2) 定義省略時の扱い

ベクタ 0

一部の MCU では、ベクタ 0(0xFFFFFFFF80 番地)にエンディアン選択レジスタが割り当てられています。interrupt_fvector[0]の定義を省略した場合は、コンパイラの endian オプションに連動してエンディアン選択レジスタが自動設定されます。

ベクタ 1~15

これらの定義を省略した場合、前バージョンではベクタの内容はカーネルの未定義割込み処理プログラムの開始アドレスとなっていたましたが、本バージョンでは 0xFFFFFFFF となります。

10.6 タイマテンプレートファイルの追加

rx62n.tpl, rx62t.tpl, rx630.tpl, rx210.tplを追加しました。詳細は「7. タイマテンプレートファイル」を参照してください。

10.7 バージョン情報

項目	変更前	変更後
TKENREL_MAKER, ref_ver および iref_ver で返る T_RVER.maker	0x0115	0x011B
TKERNEL_PRID, ref_ver および iref_ver で返る T_RVER.prid	0x0015	0x0003
TKERNEL_PRVER, ref_ver および iref_ver で返る T_RVER.prver	0x0102	0x0110

11. 注意事項

11.1 RX610 グループ使用時の注意事項

RX610 グループの PSW.IPL は 3 ビット構成のため、以下は必ず 8 未満としてください。

- chg_ims, ichg_ims で指定する割込みマスク値
- cfg ファイルの system.system_IPL 設定値
- cfg ファイルの clock.IPL 設定値

11.2 High-performance Embedded Workshopに関する注意事項

High-performance Embedded Workshop で RI600/4 対応プロジェクトを生成した場合、生成される cfg ファイルの以下の箇所は常に RX610 グループ用の設定となります。RX610 グループ以外の MCU を使用する場合は、これらの設定を適切に変更してください。

- clock.template : "rx610.tpl" が設定されます。
- clock.timer_clock : "25MHz" が設定されます。

11.3 0 番地に関する注意事項

以下を 0 番地に配置しないでください。

- 固定長メモリプール領域のセクション (memorypool[].section)
- 可変長メモリプール領域のセクション (variable_memorypool[].mpl_section)
- メールボックスに送信するメッセージのアドレス

12. 制限事項

RI600/4 V.1.01 Release 00 に制限事項はありません。

13. ドキュメント訂正

以下のとおり、ユーザーズマニュアル(ドキュメント番号：RJJ10J2562-0100)を訂正します。

(1) 「5.6.5 タスクの強制終了(ter_tsk)」 エラーコード説明

【正】

E_OBJ オブジェクト状態エラー(tskid のタスクが休止状態)

【誤】

E_OBJ オブジェクト状態エラー(tskid のタスクが休止状態でない)

(2) 「5.6.8 タスクの状態参照(ref_tsk, iref_tsk)」

「◆tskpri」と「◆tskbpri」の説明を以下に訂正します。

◆ tskpri *

タスクの現在優先度です。タスクが休止状態の場合は、不定値を返します。

◆ tskbpri *

タスクのベース優先度です。タスクが休止状態の場合は、不定値を返します。

(3) 「5.7.5 強制待ち状態への移行(sus_tsk, isus_tsk)」 エラーコード説明

【正】

E_OBJ オブジェクト状態エラー

(1) tskid のタスクが休止状態

(2) ディスパッチ禁止状態の時に、 isus_tsk で実行状態のタスクを指定

【誤】

E_OBJ オブジェクト状態エラー(tskid のタスクが休止状態)

(4) 「5.12.3 ミューテックスの状態参照(ref_mtx)」

「■エラーコード」欄の E_CTX エラーの以下の注記を削除します。

~~注：以下のケースでは、E_CTX エラーは検出されません。~~

~~(1)非タスクコンテキストからの ref_mtx の呼び出し~~

(5) 表 5.19

【正】

表 5.19 メッセージバッファ機能サービスコール一覧

	サービスコール *1	...
1	snd_mbf	...
2	psnd_mbf	...
3	ipsnd_mbf	...
4	tsnd_mbf	...
5	rcv_mbf	...
6	prcv_mbf	...
7	trcv_mbf	...
8	ref_mbf	...
9	iref_mbf	...

【誤】

表 5.19 メッセージバッファ機能サービスコール一覧

No	サービスコール *1	...
1	snd_mbf [R]	...
2	psnd_mbf [R]	...
3	ipsnd_mbf	...
4	tsnd_mbf [R]	...
5	rcv_mbf [R]	...
6	prcv_mbf [R]	...
7	trcv_mbf [R]	...
8	ref_mbf [R]	...
9	iref_mbf	...

(6) 「5.21.1 (1) maker」

以下に訂正します。

maker は、このカーネルを作ったメーカーを表します。本カーネルでは、ルネサスエレクトロニクスを意味する 0x011B です。

(7) 「5.21.1 (2) prid」

以下に訂正します。

prid は、カーネルや VLSI の種類を区別する番号を表します。本カーネルでは、0x0003 が返ります。

(8) 表 5.34

【正】

分類	マクロ	定義内容	定義場所	説明
...
カーネル構成	VTMAX_TSK	task[]定義数	kernel_id.h	最大タスク ID

	TKERNEL_MAKER	0x011B	kernel.h	カーネルのメーカーコード
	TKERNEL_PRID	0x0003	kernel.h	カーネルの識別番号

	TKERNEL_PRVER	0x0110	kernel.h	カーネルのバージョン番号
	VTMAX_AREASIZE	0x10000000	kernel.h	各種領域サイズの最大値
...
エラーコード	E_NOSPT	-9	itron.h	未サポート機能
...

【誤】

分類	マクロ	定義内容	定義場所	説明
...
カーネル構成	VTMAX_TSK	task[]定義数	kernel_id.h	最大タスク ID

	TKERNEL_MAKER	0x0115	kernel.h	カーネルのメーカーコード
	TKERNEL_PRID	0x0015	kernel.h	カーネルの識別番号

	TKERNEL_PRVER	0x0100	kernel.h	カーネルのバージョン番号
	VTMAX_AREASIZE	0x20000000	kernel.h	各種領域サイズの最大値
...
エラーコード	E_NOSPT	-11	itron.h	未サポート機能
...

(9) 「6.3.3 起動時の CPU 状態」

【正】

タスクはユーザモードで実行されるので、特権命令は実行できません。なお、アセンブラでは、特権命令の使用を Warning とする機能(-chkpm オプション)があります。

【誤】

タスクはユーザモードで実行されるので、特権命令は実行できません。例えば、コンパイラが提供する組み込み関数のうち、ヘッダ smachine.h で提供される機能は特権命令を使用するので、注意してください。なお、アセンブラでは、特権命令の使用を Warning とする機能(-chkpm オプション)があります。

(10) 「6.8 (2) ハンドラ」

以下に訂正します。

(2) 割込みハンドラ

アプリケーション内に前述の DSP 機能命令を使用するタスクおよびハンドラがひとつでもある場合は、すべての割込みハンドラが ACC レジスタを保証する必要があります。その方法として、以下の3つがあります。

- (a) コンパイラの `save_acc` オプションを使用する。
- (b) `cfg` ファイルですべての割込みハンドラ定義時に、`pragma_switch` に”ACC”を指定する。
- (c) 割込みハンドラが明示的に ACC を保証する

ユーザーズマニュアルの図 6.7 のようにして、ACC レジスタを保証してください。

(11) 図 6.7

最後から2行目を訂正します。

【正】

```
set_acc(&st_acc); // ACC レジスタを復帰
```

【誤】

```
set_acc(st_acc); // ACC レジスタを復帰
```

(12) 「7.6 CPU の動作モード」

【正】

タスクはユーザモード(P_{SW}.P_M=1)、ハンドラやカーネルはスーパーバイザモード(P_{SW}.P_M=0)で実行します。

ユーザモードで特権命令を実行すると、CPU は特権命令例外を検出します。ユーザモードで実行するプログラムでは、特権命令を使用しないように注意してください。

なお、アセンブラには特権命令の記述をワーニングとする `chkpm` オプションがあるので、必要に応じて利用してください。

【誤】

タスクはユーザモード(P_{SW}.P_M=1)、ハンドラやカーネルはスーパーバイザモード(P_{SW}.P_M=0)で実行します。

ユーザモードで特権命令を実行すると、CPU は特権命令例外を検出します。ユーザモードで実行するプログラムでは、特権命令を使用しないように注意してください。

なお、特権命令は以下の場合に使用されます。

- コンパイラの `smachine.h` によって提供される組み込み関数を使用
- アセンブリ言語プログラムで、特権命令を記述

なお、アセンブラには特権命令の記述をワーニングとする `chkpm` オプションがあるので、必要に応じて利用してください。

(13) 「8.4.15 (4) PRAGMA 拡張機能に渡すスイッチ(pragma_switch)」

「【定義範囲】」に、以下を追記します。

ACC : 割込みハンドラで ACC レジスタを保証する"acc" スイッチを渡します。

NOACC : 割込みハンドラで ACC レジスタを保証しない"no_acc" スイッチを渡します。

H : 割込みハンドラをトレースする(リアルタイム OS デバッグ機能の OS トレース)

また、以下の説明を追加します。

ACC レジスタの保証については、以下を参照してください。

pragma_switch の 設定	コンパイラの"save_acc"オプション	
	なし	あり
"ACC", "NOACC" ともに無し	"acc", "no_acc" いずれのス イッチも渡されません。ACC は保証されません。	"acc", "no_acc" いずれのスイッ チも渡されません。ACC は保証されま す。
"ACC"あり	"acc"スイッチが渡されます。ACC は保証されます。	
"NOACC"あり	"no_acc"スイッチが渡されます。ACC は保証されません。	

(14) 「8.4.16 固定ベクタ割込み定義(interrupt_fvector[])」

以下に訂正します。

interrupt_fvector[]は、固定ベクタに対する割込みハンドラを定義する定義項目です。固定ベクタの要因は、すべてカーネル管理外割込みの扱いとなります。

固定割込みベクタの各要因には、マイコン仕様ではベクタ番号が割り振られていませんが、RI600/4 では各ベクタアドレスに対して表 8.8 に示すようにベクタ番号を割り当てています。

本定義を行わない場合の振る舞いも表 8.8 に示します。

表 8.8 固定割込みベクタ

ベクタアドレス	ベクタ番号	要因 *1	定義省略時の扱い	
0xFFFFFFFF80	0	エンディアン選択レジスタ	コンパイラの endian オプションに応じて、以下が設定されます。 ・ endian=little の場合 : 0xFFFFFFFFF ・ endian=big の場合 : 0xFFFFFFFF8	
0xFFFFFFFF84	1	(予約領域)	0xFFFFFFFFF	
0xFFFFFFFF88	2	オプション機能選択レジスタ 1		
0xFFFFFFFF8C	3	オプション機能選択レジスタ 0		
0xFFFFFFFF90	4	(予約領域)		
0xFFFFFFFF94	5	(予約領域)		
0xFFFFFFFF98	6	(予約領域)		
0xFFFFFFFF9C	7	ROM コードプロテクト(フラッシュメモリ)		
0xFFFFFA0	8	オンチップデバッグ ID コードプロテクト(フラッシュメモリ)		
0xFFFFFA4	9			
0xFFFFFA8	10			
0xFFFFFAC	11			
0xFFFFFB0	12	(予約領域)		
0xFFFFFB4	13	(予約領域)		
0xFFFFFB8	14	(予約領域)		
0xFFFFFBC	15	(予約領域)		
0xFFFFFC0	16	(予約領域)		システムダウン
0xFFFFFC4	17	(予約領域)		
0xFFFFFC8	18	(予約領域)		
0xFFFFFCC	19	(予約領域)		
0xFFFFFD0	20	特権命令例外		
0xFFFFFD4	21	アクセス例外		
0xFFFFFD8	22	(予約領域)		
0xFFFFFDC	23	未定義命令例外		
0xFFFFFE0	24	(予約領域)		
0xFFFFFE4	25	浮動小数点例外		
0xFFFFFE8	26	(予約領域)		
0xFFFFFEC	27	(予約領域)		
0xFFFFFF0	28	(予約領域)		
0xFFFFFF4	29	(予約領域)		
0xFFFFFF8	30	ノンマスカブル割込み		
0xFFFFFFC	31	リセット	PowerON_Reset_PC()	

【注】 MCU 品種によって異なります。

なお、cfg600 はここで指定した割込みに関する割込み制御レジスタ(IPL 等)や、割込み要因等の初期設定のコードは生成しません。初期設定はスタートアップファイル中もしくは、開発されるアプリケーションにあ

わせて作成していただく必要があります。

■注意

マイコン仕様で予約となっているベクタには定義しないでください。

■形式

```
interrupt_fvector[(1)ベクタ番号]{
    entry_address      = (2)ハンドラの開始アドレス;
    pragma_switch     = (3)PRAGMA 拡張機能に渡すスイッチ;
};
```

■内容

(1) ベクタ番号

【説明】表 8.8 を参考にベクタ番号を定義します。

【定義形式】数値

【定義範囲】0～31

【省略時の扱い】省略不可(エラー)

(2) ハンドラの開始アドレス(entry_address)

【説明】割込みハンドラの実行開始アドレス、または固定ベクタへの設定値を定義します。

【定義形式】関数名または数値

【定義範囲】数値の場合は 0~0xFFFFFFFF

【省略時の扱い】省略不可(エラー)

(3) PRAGMA 拡張機能に渡すスイッチ(pragma_switch)

【説明】cfg600 は、entry_address で指定された関数を、割込み関数として kenrel_id.h に #pragma interrupt ディレクティブを出力します。この pragma ディレクティブに渡すスイッチを指定します。

pragma 仕様の詳細は、コンパイラのマニュアルを参照してください。

なお、entry_address に数値を指定した場合、およびベクタ番号 31(リセット)については、#pragma interrupt ディレクティブは出力されません。

【定義形式】シンボル

【定義範囲】以下を指定できます。複数指定する場合は、カンマで区切ってください。なお、"ACC"と "NOACC"を同時に指定することはできません。

S : 割込みハンドラで使用するレジスタ数を制限する"save"スイッチを渡します。

ACC : 割込みハンドラで ACC レジスタを保証するコードを生成する"acc"スイッチを渡します。

NOACC : 割込みハンドラで ACC レジスタを保証するコードを生成しない"no_acc"スイッチを渡します。

H : 割込みハンドラをトレースします(High-performance Embedded Workshop リアルタイム OS デバッグ機能の OS トレース)

ACC レジスタの扱いについては、以下を参照してください。

pragma_switch の設定	コンパイラの"save_acc"オプション	
	なし	あり
"ACC", "NOACC"とも無し	"acc", "no_acc" いずれのスイッチも渡されません。ACC は保証されません。	"acc", "no_acc" いずれのスイッチも渡されません。ACC は保証されます。
"ACC"あり	"acc"スイッチが渡されます。ACC は保証されます。	
"NOACC"あり	"no_acc"スイッチが渡されます。ACC は保証されません。	

【省略時の扱い】何もスイッチを渡しません。

(15) 「11.1 概要」

【正】

表 11.1 サンプルプログラムの関数一覧

関数名	種類	ID 番号	優先度	機能
task1()	タスク	1	1	"task1 running"を出力します。
task2()	タスク	2	2	"task2 running"を出力します。
cyh1()	周期ハンドラ	1	—	task1 を起床します。

【誤】

表 11.1 サンプルプログラムの関数一覧

関数名	種類	ID 番号	優先度	機能
main()	タスク	1	1	task1,task2 を起動させます。
task1()	タスク	2	2	"task1 running"を出力します。
task2()	タスク	3	3	"task2 running"を出力します。
cyh1()	周期ハンドラ	1	—	task1 を起床します。

【正】

以下に, 処理内容を説明します。

task1 は次の順で動作します。

【誤】

以下に, 処理内容を説明します。

main タスクは task1,task2,cyh1 を起動し, 自タスクを終了させます。

task1 は次の順で動作します。

ホームページとサポート窓口

- ルネサス エレクトロニクスホームページ
<http://japan.renesas.com/>
- お問い合わせ先
<http://japan.renesas.com/inquiry>

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。

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て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、誤りがな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。



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<http://japan.renesas.com/inquiry>