To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Date: Jun.17.2004

RENESAS TECHNICAL UPDATE

Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan
RenesasTechnology Corp.

Product Category	User Development Environment		Document No.	TN-OS*-081A/EA	Rev.	1.0
Title	HI7000/4, HI7700/4, HI7750/4 Correction of Manual regarding stack size		Information Category	Correction or Supplement of Document		
Applicable Product	(1) HI7000/4 (2) HI7700/4 (3) HI7750/4 For details, refer to below	Lot No. All version	Reference Document	HI7000/4 series User's Manual REJ10B0060_0300H Rev.3.00		

The following manual has errors about stack size. For this reason, stack size may be insufficient or it may become large vainly.

Manual : HI7000/4 series User's Manual Rev.3.0 (REJ10B0060_0300H)

Please improve setting of stack size with reference to the following document.

Document : Correction of HI7000/4 series User's Manual regarding stack size (HI7XXX-4_STACK_040608(E))

Object product:

Product Name	Type Number	Version
HI7000/4	R0R40700TRW011, R0R40700TRW015, R0R40700TRW01A, R0R40700TRW01K, R0R40700TRW01U, R0R40700TRW01Z, R0R40700TXW011, R0R40700TXW015, R0R40700TXW01A, R0R40700TXW01K, R0R40700TXW01U, R0R40700TXW01Z, HS0700ITI41SRE, HS0700ITI41SRB, HS0700ITI41SRS, HS0700ITI41SRE-E, HS0700ITI41SRB-E, HS0700ITI41SRS-E	All version (V1.00r1, V1.01r1, V1.0Ar1, V1.0Br1, V1.0Cr1, V1.0.04, V1.0.05)
HI7700/4	R0R40770TRW011, R0R40770TRW015, R0R40770TRW01A, R0R40770TRW01K, R0R40770TRW01U, R0R40770TRW01Z, R0R40770TXW011, R0R40770TXW015, R0R40770TXW01A, R0R40770TXW01K, R0R40770TXW01U, R0R40770TXW01Z, HS0770ITI41SRE, HS0770ITI41SRB, HS0770ITI41SRS, HS0770ITI41SRE-E, HS0770ITI41SRB-E, HS0770ITI41SRS-E	All version (V1.00r1, V1.01r1, V1.0Ar1, V1.0Br1, V1.0Cr1, V1.1.00, V1.2.00, V1.3.00)
HI7750/4	R0R40775TRW011, R0R40775TRW015, R0R40775TRW01A, R0R40775TRW01K, R0R40775TRW01U, R0R40775TRW01Z, R0R40775TXW011, R0R40775TXW015, R0R40775TXW01A, R0R40775TXW01K, R0R40775TXW01U, R0R40775TXW01Z, HS0775ITI41SRE, HS0775ITI41SRB, HS0775ITI41SRS, HS0775ITI41SRE-E, HS0775ITI41SRB-E, HS0775ITI41SRS-E	All version (V1.00r1, V1.01r1, V1.0Ar1, V1.0Br1, V1.0Cr1, V1.0.04, V1.0.05, V1.1.00)

Correction of HI7000/4 series User's Manual regarding stack size

1. Summary

The following manual has errors about stack size. For this reason, stack size may be insufficient or it may become large vainly.

Manual : HI7000/4 series User's Manual Rev.3.0 (REJ10B0060_0300H)

2. HI7000/4

2.1 The stack for tasks may be insufficient

Phenomenon

The stack for tasks may be insufficient. In the worst case, the size running short is 36-bytes.

Conditions

[A] Static conditions

CFG_LOWINTNST ≥ 7

[B] Dynamic conditions

All interruption in the system, except NMI, occurs continuously almost simultaneous during task execution with SR.IMASK=0, the stack may be insufficient.

Correction of Manual

Table C.5 in the manual is corrected as given in 4.1 Correction of Table C.5(Task Stack) (It has influence only on HI7000/4).

2.2 The stack for direct interrupt handlers may be insufficient

Phenomenon

The stack for direct interrupt handlers may be insufficient. In the worst case, the size running short is 76-bytes.

Conditions

[A] Static conditions

The interrupt nest counts, which level is lower than or equal to CFG_KNLMSKLVL and higher than the interrupt level, is not 0.

[B] Dynamic conditions

All interruption which level is higher than the interrupt level in the system, except NMI, occurs continuously almost simultaneous during the interrupt handler execution, the stack may be insufficient.

Correction of Manual

Table C.6 in the manual is corrected as given in 4.2 Correction of Table C.6(Interrupt Handler Stack).

2.3 The stack for normal interrupt handlers (CFG_IRQSTKSZ) may become large vainly

Phenomenon

The stack for normal interrupt handler may become large vainly.

Conditions

None

Correction of Manual

Table C.6 in the manual is corrected as given in 4.2 Correction of Table C.6(Interrupt Handler Stack).

And the formula described in p.351, "HI7000/4" is corrected as given in 4.3 Correction of formula of CFG_IRQSTKSZ for "HI7000/4" in p.351.

2.4 The stack for time event handlers (CFG_TMRSTKSZ) may be insufficient

Phenomenon

The stack for time event handlers may be insufficient.

Conditions

[A] Static conditions

Conditions = (1) or (2) or (3)

(1) The interrupt nest counts, which level is lower than or equal to CFG_KNLMSKLVL and higher than CFG_TIMINTLVL, is not 0, and both time event handlers and timer interrupt routine (_kernel_tmrint()) does not use service call.

(2) All time event handler does not call service call, and timer interrupt routine calls service call or uses 32-bytes or more of stack.

(3) The interrupt nest counts, which level is lower than or equal to CFG_KNLMSKLVL and higher than CFG_TIMINTLVL, is not 0.

[B] Dynamic conditions

All interruption which level is higher than CFG_TIMINTLVL in the system, except NMI, occurs continuously almost simultaneous during the kernel timer interrupt handler execution, the stack may be insufficient.

Correction of Manual

"Appendix C.8 Stack Size Used by a Timer Handler" is corrected as given in 4.6 Correction of "Appendix C.8 Stack Size Used by a Time Event Handler and Timer Interrupt Routine".

2.5 The stack for initialization may become large vainly

Phenomenon

The stack for initialization routines may become large vainly.

Conditions

[A] Static conditions

One or more initialization routines are registered on configurator.

[B] Dynamic conditions

None

Correction of Manual

Table C.8 in the manual is corrected as given in 4.7 Correction of Table C.8(Initialization Routine Stack).

2.6 The stack for kernel may be insufficient

Phenomenon

The stack for kernel may be insufficient.

Conditions

[A] Static conditions

Conditions = (1) and ((2) or (3))

(1) CFG_TIMUSE is checked.

(2) Timer initialization routine(_kernel_tmrini( )) uses 252-bytes or more of stack.

(3) Timer initialization routine calls service call when the routine uses 112-bytes or more of stacks.

[B] Dynamic conditions

None

Correction of Manual

The description shown in 4.8 Add "Appendix C.10 Timer Initialization Routine Stack" is added to the manual.

3. HI7700/4, HI7750/4

3.1 The stack for interrupt handlers(CFG_IRQSTKSZ) may be insufficient

Phenomenon

The stack for interrupt handlers may be insufficient.

Conditions

[A] Static conditions

Conditions = (1) or (2)

(1) All the following conditions are fulfilled about a certain interruption level.

(a) The interrupt level is lower than or equal to CFG_KNLMSKLVL, and is different from CFG_TIMINTLVL.

(b) All interrupt handler of the interrupt level does not call service call.

(2) All time event handler does not call service call, and timer interrupt routine calls service call or uses 32-bytes or more of stack.

[B] Dynamic conditions

All interruption in the system, except NMI, occurs continuously almost simultaneous during the interrupt handler with the lowest interrupt level execution, the stack may be insufficient.

Correction of Manual

Table C.6 in the manual is corrected as given in 4.2 Correction of Table C.6(Interrupt Handler Stack).

And the formula described in p.351, "HI7700/4" and "HI7750/4" are corrected as given in 4.4 Correction of formula of CFG_IRQSTKSZ for "HI7700/4" in p.351 and 4.5 Correction of formula of CFG_IRQSTKSZ for "HI7750/4" in p.351.

3.2 The stack for initialization routines may become large vainly

Phenomenon

The stack for initialization routines may become large vainly.

Conditions

[A] Static conditions

One or more initialization routines are registered on configurator.

[B] Dynamic conditions

None

Correction of Manual

Table C.8 in the manual is corrected as given in 4.7 Correction of Table C.8(Initialization Routine Stack).

3.3 The stack for kernel may be insufficient

Phenomenon

The stack for kernel may be insufficient.

Conditions

[A] Static conditions

Conditions = (1) and ((2) or (3))

(1) CFG_TIMUSE is checked. And optimized timer driver is not chosen in HI7700/4.

(2)Timer initialization routine(_kernel_tmrini()) uses following size or more of stacks.

HI7700/4 : 208-bytes

HI7750/4 : 204-bytes

(3)Timer initialization routine calls service call when the routine uses following size or more of stack.

HI7700/4 : 68-bytes

HI7750/4 : 60-bytes

[B] Dynamic conditions

None

Correction of Manual

The description shown in 4.8 Add "Appendix C.10 Timer Initialization Routine Stack" is added to the manual.

4. Correction and Addition of Manual

Hatching means addition / change portion.

4.1 Correction of Table C.5(Task Stack) (It has influence only on HI7000/4)

Table C.5 ~~Interrupt Handler Stack Size~~ Task Stack Size

		Stack Size (Byte)		
Item		**HI7000/4**	**HI7700/4**	**HI7750/4**
Size obtained in C.4 and C.5				
Mandatory		~~184~~ 140	184 *4	196
Task	TA_COP0 attribute included *3	56	56	—
	TA_COP1 attribute included	—	—	64
	TA_COP2 attribute included	—	—	64
	Static stacks	8	8	8
Checks CFG_TRACE		24	24	24
Addition considering nested interrupts		*1	—	—
The added value when the NMI is used		*2	—	—
Total				

Note:

1. ~~12 x CFG_UPPINTNST + 20 x CFG_LOWINTNST~~
   12 × CFG_UPPINTNST + 24 × CFG_LOWINTNST + 20
   However, when CFG_LOWINTNST is 0, calculate an underline part as 0.
2. (stack size used by the NMI interrupt handler calculated as shown in appendixes C.4 and C.5 + 8) x NMI nest count
   When there is not nesting, the NMI nest count is 1. When there is not possibility of NMI nesting, the nest count is 1.
3. In the HI7700/4, TA_COP0 attribute can be set or cleared by vchg_cop service call. This additional size is necessary when TA_COP0 attribute is set by vchg_cop.
4 With the optimization timer driver or the DSP standby control function, it becomes 208.

4.2 Correction of Table C.6(Interrupt Handler Stack)

Table C.6 Interrupt Handler Stack Size

	Stack Size (Byte)			
	HI7000/4		**HI7700/4**	**HI7750/4**
Item	**Direct Interrupt Handler**	**Normal Interrupt Handler**		
Size obtained in C.4 and C.5				
Calls service call from the interrupt handlers	**~~188~~** **---(must not call)**	**~~184~~ 140**	**~~184~~ 140***3	**~~192~~ 144**
Checks CFG_TRACE	24	24	24	24
Addition considering nested interrupts	*1	**~~*1~~**	—	—
Added value when the NMI is used *3	*2	—	—	—
Total				

Note: 1. **~~12 x (the nest count of direct interrupts that are higher than CFG_KNLMSKLVL and the kernel interrupt level) +20 x (the nest count of direct interrupts that are lower than CFG_KNLMSKLVL and higher than the kernel interrupt level)~~**

12 x (the nest count of direct interrupts that are higher than CFG_KNLMSKLVL and the kernel interrupt level) +24 x (the nest count of direct interrupts that are lower than CFG_KNLMSKLVL and higher than the kernel interrupt level) + 20

However, when (the nest count of direct interrupts that are lower than CFG_KNLMSKLVL and higher than the kernel interrupt level) is 0, calculate an underline part as 0.

2. (stack size used by the NMI interrupt handler calculated as shown in appendixes C.4 and C.5 + 8) x NMI nest count

When there is not nesting, the NMI nest count is 1. When there is not possibility of NMI nesting, the nest count is 1.

3 With the optimization timer driver or the DSP standby control function, it becomes **~~208~~ 164**.

4.3 Correction of formula of CFG_IRQSTKSZ for "HI7000/4" in p.351

[Before]

CFG_IRQSTKSZ = ∑ (The stack area of the handler that uses the largest stack area) + 28 + (stack size used by the NMI interrupt handler calculated as shown in appendixes C.4 and C.5 + 8) × NMI nest count

[After]

CFG_IRQSTKSZ = ∑ (The stack area of the handler that uses the largest stack area) + 4

+12 × CFG_UPPINTNST + 24 × (CFG_LOWINTNST-1) + 20

+ (stack size used by the NMI interrupt handler calculated as shown in appendixes C.4 and C.5 + 8)

× NMI nest count

However, when CFG_LOWINTNST 1, calculate an underline part as 0.

4.4 Correction of formula of CFG_IRQSTKSZ for "HI7700/4" in p.351

[Before]

CFG_IRQSTKSZ = ∑ (The stack area of the handler that uses the largest stack area) + 28 + (stack size used by the NMI interrupt handler calculated as shown in appendixes C.4 and C.5 + 44) × NMI nest count

[After]

CFG_IRQSTKSZ = ∑ (The stack area of the handler that uses the largest stack area) + 4
+ 44 × ((Number of interrupt levels in the system, except NMI) - 1)
+ (stack size used by the NMI interrupt handler calculated as shown in appendixes C.4 and C.5
+ 44) × NMI nest count

4.5 Correction of formula of CFG_IRQSTKSZ for "HI7750/4" in p.351

[Before]

CFG_IRQSTKSZ = ∑ (The stack area of the handler that uses the largest stack area) + 28 + (stack size used by the NMI interrupt handler calculated as shown in appendixes C.4 and C.5 + 48) × NMI nest count

[After]

CFG_IRQSTKSZ = ∑ (The stack area of the handler that uses the largest stack area) + 4
+ 48 × ((Number of interrupt levels in the system, except NMI) - 1)
+ (stack size used by the NMI interrupt handler calculated as shown in appendixes C.4 and C.5
+ 48) × NMI nest count

4.6 Correction of "Appendix C.8 Stack Size Used by a Time Event Handler and Timer Interrupt Routine"

The size of each time event handler stack **and timer interrupt routine (_kernel_tmrint()) stack** can be determined from appendixes C.4 and C.5.

The size determined by substituting the maximum size of all time event handlers **and timer interrupt routine** into Table C.7 must be assigned to CFG_TMRSTKSZ.

Note, when CFG_ACTION is checked, calculate on condition that the following.

- **Size obtained in C.4 and C.5 : 32**
- **Calls serviceYesall :**

When no time event handler is used, and CFG_ACTION is not checked, calculate on condition that the following.

- **Size obtained in C.4 and C.5 : 0**
- **Calls service call : No**

Table C.7 Time Event Handler and Timer Interrupt Routine Stack Size

	Stack Size (Byte)		
Item	**HI7000/4**	**HI7700/4**	**HI7750/4**
Size obtained in C.4 and C.5			
Mandatory	~~188~~ 144	~~184~~ 140 *3	~~192~~ 144
Calls service call from the time event handlers **or timer interrupt routine (_kernel_tmrint())**	140	140	144
Checks CFG_TRACE	24	—	—
Addition considering nested interrupts	*1	—	—
Addition when the NMI is used	*2	—	—
Total			

Notes: 1 ~~12 x CFG_UPPINTNST + 20 x (the nest count of interrupts that are higher than CFG_TIMINTLVL and lower than CFG_KNLMSKLVL)~~
12 x CFG_UPPINTNST + 24 x (the nest count of interrupts that are higher than CFG_TIMINTLVL and lower than CFG_KNLMSKLVL) + 20
However, when (the nest count of interrupts that are higher than CFG_TIMINTLVL and lower than CFG_KNLMSKLVL) is 0, calculate an underline part as 0.

2. (stack size used by the NMI interrupt handler calculated as shown in appendixes C.4 and C.5 + 8) x NMI nest count
When there is not nesting, the NMI nest count is 1. When there is not possibility of NMI nesting, the nest count is 1.

3 With the optimization timer driver or the DSP standby control function, it becomes ~~208~~ 164.

4.7 Correction of Table C.8(Initialization Routine Stack)

Table C.8 Initialization Routine Stack Size

	Stack Size (Byte)		
Item	**HI7000/4**	**HI7700/4**	**HI7750/4**
Size obtained in C.4 and C.5			
~~Mandatory~~ Calls service call	~~184~~ 140	~~184~~ 140 *2	~~192~~ 144
Checks CFG_TRACE	24	24	24
Addition when the NMI is used	*1	—	—
Total			

Note 1 (stack size used by the NMI interrupt handler calculated as shown in appendixes C.4 and C.5 + 8) x NMI nest count
When there is not nesting, the NMI nest count is 1. When there is not possibility of NMI nesting, the nest count is 1.

2 With the optimization timer driver or the DSP standby control function, it becomes ~~208~~ 164.

4.8 Add "Appendix C.10 Timer Initialization Routine Stack"

The maximum stack size to be used by timer initialization routine (_kernel_tmrini()) is decided as follows.

- **HI7000/4 : 252 bytes**
- **HI7700/4 : 208 bytes**
- **HI7750/4 : 204 bytes**

When the size calculated by Table C.9 exceeds the above size, occupy the stack with the calculated size, and switch stack pointer to the stack. area.

Table C.9 Timer Initialization Routine Stack Size

	Stack Size (Byte)		
Item	**HI7000/4**	**HI7700/4**	**HI7750/4**
Size obtained in C.4 and C.5			
alls service call	**140**	**140 *2**	**144**
Checks CFG_TRACE	**24**	**24**	**24**
Addition when the NMI is used	***1**	**—**	**—**
Total			

Note 1 (stack size used by the NMI interrupt handler calculated as shown in appendixes C.4 and C.5 + 8) x NMI nest count
When there is not nesting, the NMI nest count is 1. When there is not possibility of NMI nesting, the nest count is 1.

2 With the optimization timer driver or the DSP standby control function, it becomes 164.