To our customers,

---

**Old Company Name in Catalogs and Other Documents**

On April 1\(^{st}\), 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](http://www.renesas.com)

---

April 1\(^{st}\), 2010
Renesas Electronics Corporation

---

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

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))

<table>
<thead>
<tr>
<th>Object product:</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Type Number</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>HI7000/4</td>
<td>R0R40700TRW011, R0R40700TRW015, R0R40700TRW01A, R0R40700TRW01K, R0R40700TRW01U, R0R40700TRW01Z, R0R40700TXW011, R0R40700TXW015, R0R40700TXW01A, R0R40700TXW01K, R0R40700TXW01U, R0R40700TXW01Z, HS0700IT41SRE, HS0700IT41SRB, HS0700IT41SRS, HS0700IT41SRE-E, HS0700IT41SRB-E, HS0700IT41SRS-E</td>
<td>All version (V1.00r1, V1.01r1, V1.0Ar1, V1.0Br1, V1.0Cr1, V1.0.04, V1.0.05)</td>
</tr>
<tr>
<td>HI7700/4</td>
<td>R0R40770TRW011, R0R40770TRW015, R0R40770TRW01A, R0R40770TRW01K, R0R40770TRW01U, R0R40770TRW01Z, R0R40770TXW011, R0R40770TXW015, R0R40770TXW01A, R0R40770TXW01K, R0R40770TXW01U, R0R40770TXW01Z, HS0770IT41SRE, HS0770IT41SRB, HS0770IT41SRS, HS0770IT41SRE-E, HS0770IT41SRB-E, HS0770IT41SRS-E</td>
<td>All version (V1.00r1, V1.01r1, V1.0Ar1, V1.0Br1, V1.0Cr1, V1.1.00, V1.2.00, V1.3.00)</td>
</tr>
<tr>
<td>HI7750/4</td>
<td>R0R40775TRW011, R0R40775TRW015, R0R40775TRW01A, R0R40775TRW01K, R0R40775TRW01U, R0R40775TRW01Z, R0R40775TXW011, R0R40775TXW015, R0R40775TXW01A, R0R40775TXW01K, R0R40775TXW01U, R0R40775TXW01Z, HS0775IT41SRE, HS0775IT41SRB, HS0775IT41SRS, HS0775IT41SRE-E, HS0775IT41SRB-E, HS0775IT41SRS-E</td>
<td>All version (V1.00r1, V1.01r1, V1.0Ar1, V1.0Br1, V1.0Cr1, V1.0.04, V1.0.05, V1.1.00)</td>
</tr>
</tbody>
</table>
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_KNLSKLVL 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 even t 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**

<table>
<thead>
<tr>
<th>Item</th>
<th>Stack Size (Byte)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HI7000/4</td>
</tr>
<tr>
<td>Size obtained in C.4 and C.5</td>
<td></td>
</tr>
<tr>
<td>Mandatory</td>
<td>184</td>
</tr>
<tr>
<td>Task</td>
<td></td>
</tr>
<tr>
<td>TA_COP0 attribute included *3</td>
<td>56</td>
</tr>
<tr>
<td>TA_COP1 attribute included</td>
<td>—</td>
</tr>
<tr>
<td>TA_COP2 attribute included</td>
<td>—</td>
</tr>
<tr>
<td>Static stacks</td>
<td>8</td>
</tr>
<tr>
<td>Checks CFG_TRACE</td>
<td>24</td>
</tr>
<tr>
<td>Addition considering nested interrupts</td>
<td>*1</td>
</tr>
<tr>
<td>The added value when the NMI is used</td>
<td>*2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Note:**
1. $12 \times \text{CFG\_UPPINTNST} + 24 \times \text{CFG\_LOWINTNST}$
   
   However, when $\text{CFG\_LOWINTNST}$ is 0, calculate an underline part as 0.
2. $(\text{stack size used by the NMI interrupt handler calculated as shown in appendixes C.4 and C.5 } + 8) \times \text{NMI nest count}$
   
   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

<table>
<thead>
<tr>
<th>Item</th>
<th>HI7000/4</th>
<th>HI7700/4</th>
<th>HI7750/4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Direct Interrupt Handler</td>
<td>Normal Interrupt Handler</td>
<td></td>
</tr>
<tr>
<td>Size obtained in C.4 and C.5</td>
<td>188</td>
<td>184</td>
<td>184</td>
</tr>
<tr>
<td>Calls service call from the interrupt handlers</td>
<td><em>(must not call)</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Checks CFG_TRACE</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>Addition considering nested interrupts</td>
<td>*1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Added value when the NMI is used *3</td>
<td>*2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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) + 20
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]

\[
\text{CFG_IRQSTKSZ} = \sum (\text{The stack area of the handler that uses the largest stack area}) + 28 + (\text{stack size used by the NMI interrupt handler calculated as shown in appendixes C.4 and C.5 + 8}) \times \text{NMI nest count}
\]

[After]

\[
\text{CFG_IRQSTKSZ} = \sum (\text{The stack area of the handler that uses the largest stack area}) + 4 + 12 \times \text{CFG_UPPINTNST} + 24 \times (\text{CFG_LOWINTNST} - 1) + 20 \\
	\times \text{NMI nest count}
\]

However, when \(\text{CFG_LOWINTNST} \leq 1\), calculate an underline part as 0.
4.4 Correction of formula of CFG_IRQSTKSZ for "HI7700/4" in p.351

[Before]

\[\text{CFG_IRQSTKSZ} = \sum (\text{The stack area of the handler that uses the largest stack area}) + 28 + (\text{stack size used by the NMI interrupt handler calculated as shown in appendixes C.4 and C.5} + 44) \times \text{NMI nest count}\]

[After]

\[\text{CFG_IRQSTKSZ} = \sum (\text{The stack area of the handler that uses the largest stack area}) + 4 + 44 \times (\text{Number of interrupt levels in the system, except NMI}) - 1) + (\text{stack size used by the NMI interrupt handler calculated as shown in appendixes C.4 and C.5} + 44) \times \text{NMI nest count}\]

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

[Before]

\[\text{CFG_IRQSTKSZ} = \sum (\text{The stack area of the handler that uses the largest stack area}) + 28 + (\text{stack size used by the NMI interrupt handler calculated as shown in appendixes C.4 and C.5} + 48) \times \text{NMI nest count}\]

[After]

\[\text{CFG_IRQSTKSZ} = \sum (\text{The stack area of the handler that uses the largest stack area}) + 4 + 48 \times (\text{Number of interrupt levels in the system, except NMI}) - 1) + (\text{stack size used by the NMI interrupt handler calculated as shown in appendixes C.4 and C.5} + 48) \times \text{NMI nest count}\]
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 service call : Yes

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Stack Size (Byte)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HI7000/4</td>
</tr>
<tr>
<td>Size obtained in C.4 and C.5</td>
<td></td>
</tr>
<tr>
<td>Mandatory</td>
<td>188 144</td>
</tr>
<tr>
<td>Calls service call from the time event handlers or timer interrupt routine (_kernel_tmrint())</td>
<td>140</td>
</tr>
<tr>
<td>Checks CFG_TRACE</td>
<td>24</td>
</tr>
<tr>
<td>Addition considering nested interrupts</td>
<td>*1</td>
</tr>
<tr>
<td>Addition when the NMI is used</td>
<td>*2</td>
</tr>
<tr>
<td>Total</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Item</th>
<th>HI7000/4</th>
<th>HI7700/4</th>
<th>HI7750/4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size obtained in C.4 and C.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mandatory Calls service call</td>
<td>144 140</td>
<td>144 140</td>
<td>192 144</td>
</tr>
<tr>
<td>Checks CFG_TRACE</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>Addition when the NMI is used</td>
<td>*1</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

Note 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

<table>
<thead>
<tr>
<th>Item</th>
<th>Stack Size (Byte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size obtained in C.4 and C.5</td>
<td></td>
</tr>
<tr>
<td>alls service call</td>
<td>140</td>
</tr>
<tr>
<td>Checks CFG_TRACE</td>
<td>24</td>
</tr>
<tr>
<td>Addition when the NMI is used</td>
<td>*1</td>
</tr>
<tr>
<td>Total</td>
<td></td>
</tr>
</tbody>
</table>

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.

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