

To our customers,

---

## Old Company Name in Catalogs and Other Documents

---

On April 1<sup>st</sup>, 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 1<sup>st</sup>, 2010  
Renesas Electronics Corporation

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

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

# RENEASAS TECHNICAL UPD

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

Product Category	Development Environment		Document No.	TN-OS*-076A/EA	Rev.	1.0
Title	Notes on Use of the Variable-Size Memory Pool and Shared Stack Functions in the HI2000/3		Information Category	Usage Limitation		
Applicable Product	HS0200ITZE1SRE, HS0200ITZE1SRE-E, HS0200ITZE1SRB, HS0200ITZE1SRB-E, HS0200ITZE1SRS, and HS0200ITZE1SRS-E		Lot No.	Reference Document	HI2000/3 Hitachi Industrial Realtime Operating System for H8S Series User's Manual ADE-702-208A Revision 2.0	
			V1.00 r2 V1.10 r1			

Thank you for using our product.

Please observe the notes listed below on use of the variable-size pool and shared stack functions.

These problems will be fixed in the following revision (V1.1.02).

## 1. Variable-Size Memory Pool Function

### 1.1 Release of a Wait Queue

[Problem]

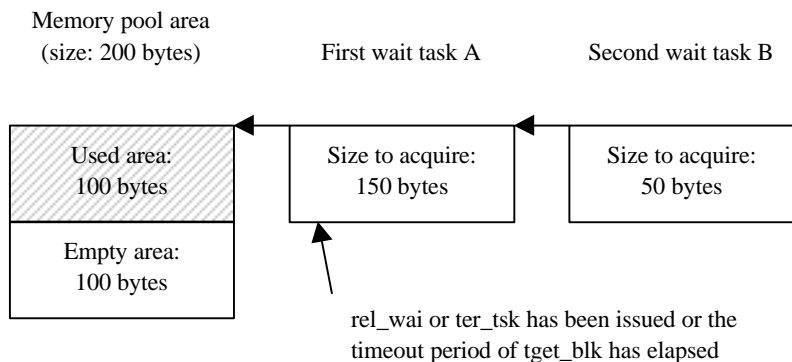
When there are multiple wait queues for a variable-size memory pool, the second and later wait queues may not be released. In this case, tasks in the WAIT state will be released the next time the `rel_blk` system call is issued.

[Conditions]

This problem may arise when both of the following conditions are satisfied.

- There are multiple tasks waiting to acquire a memory block.
- The WAIT state of the first task to acquire a memory block has been released by timeout of `tget_blk` or system call `rel_wai` or `ter_tsk` (excluding the `rel_blk` system call).

Example:



In the above figure, task A and task B are waiting to acquire 150 bytes and 50 bytes, respectively, of the 200-byte memory pool. When timeout of task A occurs after the specified timeout period of `tget_blk` has elapsed or the WAIT state of task A has been released by system call `rel_wai` or `ter_tsk` issued to task A, task B still remains in the WAIT state and acquires no memory block even though it has now become the first wait task.

[Solution]

Keep sufficient variable-size memory pool area so that no task will enter the WAIT state.

## 1.2 Size of a Memory Pool Area

### [Problem]

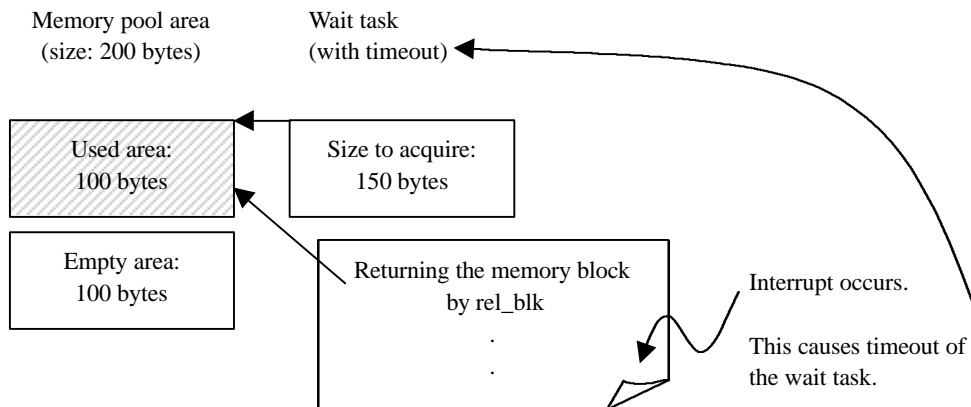
The available size of a variable-size memory pool may be reduced.

### [Conditions]

This problem may arise when both of the following conditions are satisfied.

- (a) There is a task waiting to acquire a memory block.
- (b) The WAIT state of the first wait task is released while the memory block is being returned.

Example:



In the above figure, there is a wait task with timeout for the memory pool. When a timer interrupt occurs and causes timeout as the timeout period remained for the wait task is zero while the memory block is being returned by `rel_blk`, 150 bytes of the memory pool area will be used and thus the empty area will be 50 bytes. The wait task has already had timeout and acquired no memory block. However, the memory pool assumes that the memory block is acquired.

### [Solution]

Keep sufficient variable-size memory pool area so that no task will enter the WAIT state.

## 2. Shared Stack Function

### [Problem]

Tasks waiting to use a shared stack may not be executed.

### [Condition]

The addresses for the lower 16 bits of the task stack area (section name: `h2sstack`) and the OS work area (section name: `hi8_2s_ram`) are the same.

### [Solution]

Allocate different addresses for the lower 16 bits of the task stack area and the OS work area.