RENESAS Tool News

RENESAS TOOL NEWS on November 1, 2011: 111101/tn7

Notes on Using Real-Time OSes for SuperH Family

When using real-time OSes for the SuperH family of MCUs, take note of the following problems:

- With the iset_flg service call
- With the SPIN_WriteLock() function in the spin-lock library
- With a descriptive error about the vsns_tmr service call in the HI7200/MP V.1.00 user's manual (Rev.1.01)

---

1. Problem with the iset_flg Service Call

1.1 Products and Versions Concerned

- HI7000/4 (for SH-2A, SH2A-FPU, SH-2, SH2-DSP, and SH-1)
  V.2.03 Release 00 and earlier
- HI7700/4 (for SH4AL-DSP, SH3-DSP, and SH-3)
  V.2.03 Release 04 and earlier
- HI7750/4 (for SH-4A and SH-4)
  V.2.02 Release 06 and earlier
- HI7300/PX (for SH-4A and SH4AL-DSP)
  V.1.02 Release 02 and earlier
- HI7200/MP (for SH2A-DUAL)
  V.1.00 Release 05 and earlier

1.2 Description

If the iset_flg service call has been called from an interrupt handler or a time-event handler, events set by this iset_flg service call may disappear.

1.3 Conditions

This problem may arise if Conditions (1) and (2) are satisfied, or Conditions (1) and (3) are satisfied.

(1) An event flag has the TA_CLR attribute.

(2) A task has made a call to set_flg, and any handler has made a call to iset_flg.

(3) A task has not made a call to set_flg, and two or more

handlers with different priority levels have made a call to iset_flg.

Note that in (2) and (3), a time-event handler is interpreted as a handler whose priority is the timer interrupt level.

1.4 Workaround

To avoid this problem, make a call to set_flg and iset_flg after setting the kernel interrupt masking level. Several examples are shown below where the above level is 13.

(1) In HI7000/4, HI7700/4, HI7750/4, and HI7200/MP

a. Changes are made to set_flg called from the task in Condition (2).

```
-----------------------------------------------------------
#include <machine.h>       // Added
 . . . . . . . . . . . . . . . . . . . . . . . .
int imask;                 // Added
 . . . . . . . . . . . . . . . . . . . . . . . .
imask = get_imask();       // Added
chg_ims(SR_IMS13);         // Added
iset_flg(...);             // set_flg replaced with iset_flg
ichg_ims(imask);           // Added
 . . . . . . . . . . . . . . . . . . . . . . . .
-----------------------------------------------------------
```

b. Changes are made to iset_flg called from the handlers in Conditions (2) and (3). These changes are unnecessary to the interrupt handler with the highest priority in the system.

```
-----------------------------------------------------------
#include <machine.h>         // Added
 . . . . . . . . . . . . . . . . . . . . . . . .
int imask;                   // Added
 . . . . . . . . . . . . . . . . . . . . . . . .
imask = get_imask();         // Added
set_imask(13);               // Added
iset_flg(...);               // iset_flg concerned
set_imask(imask);            // Added
 . . . . . . . . . . . . . . . . . . . . . . . .
-----------------------------------------------------------
```

(2) In HI7300/PX

a. Changes are made to set_flg called from the task in Condition (2), which belongs to the user domain.

```
-----------------------------------------------------------
```

```
IMASK imask;                // Added
 . . . . . . . . . . . . . . . . . . . . . . . .
get_ims(&imask);            // Added
chg_ims(SR_IMS13);          // Added
set_flg(...);               // set_flg concerned
chg_ims(imask);             // Added
 . . . . . . . . . . . . . . . . . . . . . . . .
-------------------------------------------------------
```

b. Changes are made to set_flg called from the task in Condition (2), which belongs to the kernel domain.

```
-------------------------------------------------------
#include <machine.h>        // Added
 . . . . . . . . . . . . . . . . . . . . . . . .
int imask;                  // Added
 . . . . . . . . . . . . . . . . . . . . . . . .
imask = get_imask();        // Added
chg_ims(SR_IMS13);          // Added
set_flg(...);               // set_flg concerned
chg_ims(imask);             // Added
 . . . . . . . . . . . . . . . . . . . . . . . .
-------------------------------------------------------
```

c. Changes are made to iset_flg called from the handlers in Conditions (2) and (3). These changes are unnecessary to the interrupt handler with the highest priority in the system.

```
-------------------------------------------------------
#include <machine.h>        // Added
 . . . . . . . . . . . . . . . . . . . . . . . .
int imask;                  // Added
 . . . . . . . . . . . . . . . . . . . . . . . .
imask = get_imask();        // Added
set_imask(13);              // Added
iset_flg(...);              // iset_flg concerned
set_imask(imask);           // Added
 . . . . . . . . . . . . . . . . . . . . . . . .
-------------------------------------------------------
```

2. Problem with the SPIN_WriteLock() Function in the Spin-Lock Library

2.1 Product and Versions Concerned

- HI7200/MP (for SH2A-DUAL)
  V.1.00 Release 05 and earlier

2.2 Description

Even if ReadLock has been already performed, the SPIN_WriteLock() function may successfully performed.

2.3 Conditions

This problem may arise if either of the following conditions is satisfied:

(1) At the time when SPIN_WriteLock() is called, ReadLock has been performed and the R0 register cleared to 0.

(2) At the time when SPIN_WriteLock() is called, ReadLock is not performed and the R0 register not cleared to 0; however, during the execution of SPIN_WriteLock(), another CPU or task acquires ReadLock.

2.4 Workaround

To avoid this problem, change the line containing "_SPIN_WriteLock" and later, which starts from about 132th line of the rwlock.src file as follows. The rwlock.src file resides in the path of "spinlock¥spinlock¥source¥," which is under the directory where HI7200/MP has been installed.

```
-----------------------------------------------------------------------
        .export     _SPIN_WriteLock
_SPIN_WriteLock:
        .stack      _SPIN_WriteLock=0       ; frame size=0
WriteLock_WriteRetry:
        tas.b   @r4                     ; At first, write-lock
        bf      WriteLock_WriteRetry        ; for busy loop
;
; Check read-lock
        mov.b   @(ucReadLock,r4),r1     -> Replace r1 with r0.
        cmp/eq  #0,r0
        bf      WriteLock_ReadRetry
;
        rts
        nop
-----------------------------------------------------------------------
```

3. Problem with a Descriptive Error about the vsns_tmr Service Call in the HI7200/MP V.1.00 User's Manual (Rev.1.01)

3.1 Manual Concerned

Title: HI7200/MP V.1.00 User's Manual (Rev.1.01)
Document No.: REJ10J1727-0101

3.2 Description

An error is found in Article 6.22.6, Reference Timer State (vsns_tmr).
Rectify the 5th and 6th lines in this article as follows (interchange TRUE and FALSE with each other):

For:
TRUE is returned when the kernel timer is stopped and FALSE is returned when the kernel timer is operating
Read:
FALSE is returned when the kernel timer is stopped and TRUE is returned when the kernel timer is operating

4. Schedule of Fixing the Problems

Problems 1 and 2 have already been fixed in the following versions of real-time OSes for the SuperH families:

- HI7000/4 V.2.03 Release 01
- HI7700/4 V.2.04 Release 00
- HI7750/4 V.2.03 Release 00
- HI7300/PX V.1.03 Release 00
- HI7200/MP V.1.01 Release 00

For how to update your products to the above, see RENESAS TOOL NEWS Document No. 111101/tn8 at:
http://tool-support.renesas.com/eng/toolnews/111101/tn8.htm
This Web page will be opened on November 21, 2011.
