RENESAS Tool News

A Note on Using Real-Time OS M3T-MR32R
--On Entering a Task for the Call or Acceptance of a Rendezvous or the Acquisition of a Variable-Size Memory Pool into a SUSPEND State--

Please take note of the following problem in using the M3T-MR32R real-time OS:

- On entering a task for the call or acceptance of a rendezvous or the acquisition of a variable-size memory pool into a SUSPEND state

1. **Versions Concerned**
   M3T-MR32R V.3.00 Release 1 through V.3.50 Release 2

2. **Description**
   When such a task as the call or acceptance of a rendezvous or the acquisition of a variable-size memory pool is entered into a SUSPEND state by an isus_tsk system call, the task will not go to a WAIT-SUSPEND state.

3. **Conditions**
   This problem occurs if the following conditions are all satisfied:

   (1) A task issues any of the following system calls; then goes to a WAIT state:
   acp_por, tacp_por, cal_por, tcal_por, get_blk, and tget_blk

   (2) An isus_tsk system call is issued inside the interrupt handler invoked during the execution of the system call issued in (1).

   (3) The task that is the object of isus_tsk in (2) is the one that issued a system call in (1).

4. **Workaround**
   This problem can be circumvented either of the following ways:

   (1) Before and after issuing the system call stated in Condition (1) above, disable and enable an interrupt respectively.

Example:

```
void task(INT stacd)
{
    . . . . . . . . . . . . . . .

        /* Disable an interrupt */

          asm(" mvfc     R0,PSW¥n"
            " and3     R0,R0,#0xFFBF¥n"
            " mvtc     R0,PSW¥n");

      ercd = cal_por((VP)msg,&amp;size,ID_por2,0x1,7);

        /* Enable the interrupt */
          asm(" mvfc     R0,PSW¥n"
            " or3      R0,R0,#0x0040¥n"
            " mvtc     R0,PSW¥n");

    . . . . . . . . . . . . . . .

}
```
----

(2) Perform the following steps to enter a task into a SUSPEND state:
   1. Issue an ista_tsk system call to invoke the task to enter into SUSPEND state.
   2. Issue a sus_tsk system call from the invoked task mentioned above.

Example:

```
void handler(void)
{
    . . . . . . . . . . . . . .

      /* Comment out isus_tsk, issue ista_tsk to invoke ta
         to enter into SUSPEND state, and pass ID of
         objective task to ista_tsk as startup code */

/*   isus_tsk( ID_task1 );  */
      ista_tsk( ID_DoSuspend,(INT)ID_task1 );

    . . . . . . . . . . . . . .
```

```
        }

        void DoSuspend(INT tskid)
        {
          sus_tsk(tskid);
        }
        -------------------------------------------------------
```

5. **Schedule of Fixing the Problem**

   We plan to fix this problem in our next release of the product.