RENESAS Tool News

RENESAS TOOL NEWS on July 16, 2005: RSO-M3T-MR308-050716D

A Note on Using the Real-Time OSes M3T-MR308 and M3T-MR30

Please take note of the following problem in using the real-time OSes M3T-MR308 and M3T-MR30 (they are used for the M16C family of MCUs):

- On issuing the isus_tsk system call to a task that is entering the DORMANT state

---

1. **Products and Versions Concerned**
   - M3T-MR308 V.1.00 Release 1 through M3T-MR308 V.1.20 Release 1B
     (for the M32C/90, M32C/80, and M16C/80 series of MCUs)
   - M3T-MR30 V.1.00 Release 1 through M3T-MR30 V.3.30 Release 1
     (for the M16C/60, M16C/30, M16C/Tiny, M16C/20, and M16C/10 series of MCUs)

2. **Description**

   If the isus_tsk system call is issued to a task that is entering the DORMANT state by the ext_tsk system call, either of the following symptoms may arise, prohibiting the system from operating properly:

   (1) The isus_tsk system call does not return E_OBJ but E_OK.

   (2) Even when a task is ready to be carried out with the same priority as the task to which the ext_tsk system call has been issued, the former task is taken off from the ready queue while remaining ready to be performed, which incurs contradiction in the management of the kernel.

   2.1 Condition

   This problem may occur if the following conditions are all satisfied:

   (1) The isus_tsk system call is issued from the interrupt handler, alarm handler, or cyclic handler.

   (2) The task to which isus_tsk is issued is the one that can

enter the DORMANT state.

(3) The handler that issues isus_tsk has been invoked by an interrupt requested while the ext_tsk system call is performed.

(4) The ext_tsk system call in (3) is issued from the task to which isus_tsk has been issued.

3. Workaround

Handle the task to which isus_tsk is issued in an infinite loop so that it cannot be entered to the DORMANT state by ext_tsk as shown in the example below.

Example

```
-----------------------------------------------------
void task1(INT stacd )
{
  . . . . . . . . . . . . . . . . . . . . . . . .
    while(1){
    /* Task carried out */
  . . . . . . . . . . . . . . . . . . . . . . . .
    }
}

void inth(void)
{
  . . . . . . . . . . . . . . . . . . . . . . . .
    isus_tsk(ID_task1);
  . . . . . . . . . . . . . . . . . . . . . . . .
}
-----------------------------------------------------
```

4. Schedule of Fixing the Problem

we plan to fix this problem in its next release.
