RENESAS Tool News

RENESAS TOOL NEWS on February 1, 2012: 120201/tn1

Notes on Using Real-Time OSes for R8C and M16C MCU Families

When using real-time OSes for the R8C and M16C families of MCUs, take note of the following problems:

- With task switching being delayed
- With dispatch disabled state

---

1. Problem with Task Switching Being Delayed

1.1 Products and Versions Concerned

- M3T-MR30 V.3.00 Release 1 through V.3.30 Release 2
- M3T-MR30/4 V.4.00 Release 00 and later
- M3T-MR308 V.1.00 Release 1 through V.1.20 Release 1B
- M3T-MR308/4 V.4.00 Release 00 and later
- M3T-MR100/4 V.1.00 Release 00 and later

1.2 Description

If an operation or event that necessitates task switching is performed by an interrupt generated while a service call is executed, task switching may not be done just at the end of the service call, but after a certain delay time. That is, task switching may be done after a call is made to any service call that changes a task's state, or after a kernel interrupt is completed.

1.3 Condition

The condition under which this problem arises depends on real-time OSes as follows:

(1) M3T-MR30 V.3.00 Release 1 through V.3.30 Release 2 and
M3T-MR308 V.1.00 Release 1 through V.1.20 Release 1B

The problem may arise when the vrst_blk service call (system call) is used.

(2) M3T-MR30/4 V.4.00 Release 00 and later; and
M3T-MR308/4 V.4.00 Release 00 and later

The problem may arise when either of the following conditions is met:
- The vrst_mpl service call is used.
- The ref_tsk service call is used to reference the task waiting for timeout.

(3) M3T-MR100/4 V.1.00 Release 00 and later

The problem may arise when any of the following conditions is met:
- The chg_pri service call is used with the mutex function.
- The loc_mtx or tloc_mtx service call is used with error E_ILUSE returned as the return value.
- The vrst_mpl service call is used.
- The rel_mpl service call is used with error E_PAR returned as the return value.

1.4 Workaround

To avoid this problem, make a call to the ena_dsp service call immediately after calling the service call involved.
Example:

```
----------------------------------------------------------------------------
--
ER ercd;
. . . . . .
ercd = vrst_mpl(ID_mpl1);   Service call in Condition (2) or (3).
ena_dsp();                  Call made to ena_dsp.
. . . . . .
----------------------------------------------------------------------------
--
```

2. Problem with Dispatch Disabled State

2.1 Products and Versions Concerned

- M3T-MR30/4 V.4.00 Release 00 and later
- MR8C/4 V.1.00 Release 00 and later
- M3T-MR308/4 V.4.00 Release 00 and later
- M3T-MR100/4 V.1.00 Release 00 and later

2.2 Description

Because confliction is generated in a ready queue, tasks in the ready state may not be executed. As a result, your system may not operate properly.

2.3 Conditions

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

(1) Any of the following is done in the dispatch disabled state:

(1.1) Selecting the task in the RUNNING state by the isus_tsk service call.

(1.2) Lowering the current priority of the task in the RUNNING state by the chg_pri or ichg_pri service call.

(1.3) Changing the current priority of the task in the RUNNING state by the rot_rdq or irot_rdq service call while another task is joining the ready queue whose current priority is of the task in the RUNNING state.

(2) The ext_tsk service call is called from the task in the RUNNING state in (1).
In M3T-MR100/4 V.1.01 Release 00, however, any of the following service calls is called: ploc_mtx, unl_mtx, and ext_tsk

2.4 Workaround

To avoid this problem, make a call to the service call in Condition (2) in the dispatch enabled state.

Example:

```
------------------------------------------------------------------------------
--
. . . . . .
    ena_dsp();          Dispatching enabled.
    ext_tsk();          Call made to service call in Condition (2).
------------------------------------------------------------------------------
--
```