

Revised by: ITRON Specification Maintenance Working Group, ITRON Specification Study Group, TRON Association

Last update: January 30, 2008

◆ Treatment of This Document

The μITRON4.0 Specification Ver. 4.03.02 is a revised version that the change described in ‘Details on Revision’ of this document has been added in the μITRON4.0 Specification Ver. 4.03.00.

◆ Overview of Revision

Description on restricted tasks is clarified.

◆ Details on Revision

p. 314: Description in chapter 5.3.1, Restricted Tasks, is changed as follows.

[Before change]

- A restricted task cannot enter the WAITING state.

[After change]

- A restricted task cannot enter the blocked state.

The following contents will be added before [Supplemental Information].

Although rot_rdq or irot_rdq is not included in the automotive control profile, the behavior is implementation-dependent when the kernel supports rot_rdq and irot_rdq and restricted tasks are included in the runnable tasks with the specified priority.

An application must not specify the priority including the restricted tasks as the specified priority of rot_rdq or irot_rdq.

Furthermore, the following contents will be added in [Supplemental Information].

Restrictions on restricted tasks are specified so that the precedence between restricted tasks are decided only by activation of these tasks. This enables restricted tasks to share the same stack area with other restricted tasks.

The automotive control profile does not contain rot_rdq or irot_rdq for the same reason.

The behavior is implementation-dependent when the kernel supports rot_rdq and irot_rdq and restricted tasks are included in the runnable tasks with the specified priority because this applies to a case when an application uses the restricted task wrongly and it has been determined that the behavior needs not be specified in the μITRON4.0 Specification.

For example, when the runnable restricted tasks are included in the specified priority of rot_rdq or irot_rdq, the following will be assumed as the behavior of the kernel.

(1) An error E_NOSPT is always returned when there are runnable restricted tasks with the specified priority.

(2) An error E_NOSPT is returned only when the restricted task is a task with the highest precedence of all the runnable tasks with the specified priority.

(3) E_OK is always returned without rotating the precedence of the tasks when there are runnable restricted tasks with the specified priority.

(4) E_OK is returned without rotating the precedence of the tasks only when the restricted task is a task with the highest precedence of all the runnable tasks with the specified priority.