

Microcontroller Technical Information

V850 Real-Time OS RX850V4	Document No.	ZMT-F35-10-0006	1/2
	Date issued	March 30, 2011	
Usage Restrictions	Issued by	MCU Tool Product Marketing Department MCU Software Division MCU Business Unit Renesas Electronics Corporation	
Related documents None	Notification classification	√	Usage restriction
			Upgrade
			Document modification
			Other notification

1. Affected products

RX850V4 Ver. 4.42 package (kernel Ver. 4.30) and earlier versions

2. New restrictions

The following restrictions related to the RX850V4 kernel have been added:

- No. 3 Processing cannot return to a restricted task
- No. 4 AND and OR specified together as the wait conditions for the same event flag during processing of the same task
- No. 5 A cyclic handler might start up earlier than the specified cycle time

3. Workarounds

The workarounds below are available for the above restrictions. See attachment 1 for details.

- No. 3 Do not use restricted tasks.
- No. 4 Do not specify AND and OR together as the wait conditions for the same event flag during processing of the same task.
- No. 5 Do one of the following:
 - Do not allow an interrupt handler to issue the service call `sta_cyc` for a cyclic handler for which no PHS attribute is specified.
 - Design the system so that only two cyclic handlers can be waiting to be started up at one time.

4. Modification schedule

- No. 3 This will not be corrected, so regard it as a specification.
- No. 4 This will not be corrected, so regard it as a specification.
- No. 5 This will not be corrected, so regard it as a specification.

5. List of restrictions

A list of restrictions in the RX850V4, including the revision history and detailed information, is shown on attachment 1.

6. Document revision history

V850 Real-Time OS RX850V4 - Usage Restrictions

Document Number	Issued on	Description
ZBG-CD-09-0024	May 11, 2009	Addition of CubeSuite-related restrictions (No. 1 and No. 2)
ZBG-CD-09-0054	October 20, 2009	Addition of restrictions related to RX850V4 kernel (No. 1 and No. 2)
ZMT-F35-10-0006	March 30, 2011	Addition of restrictions related to RX850V4 kernel (No. 3 to No. 5)

Usage Restrictions Related to RX850V4 Kernel

1. Product History

No.	Usage Restrictions	Version	
		V4.22	V4.30
1	Restriction whereby task scheduling is performed incorrectly after executing the service call <code>rot_rdq</code> for which <code>TPRI_SELF</code> is specified	√	
2	Restriction on the service call <code>rot_rdq</code> issued with <code>TPRI_SELF</code> specified when dispatching is disabled	√	
3	Processing cannot return to a restricted task	√	√
4	AND and OR specified together as the wait conditions for the same event flag during processing of the same task	√	√
5	A cyclic handler might start up earlier than the specified cycle time	√	√

√: Applicable

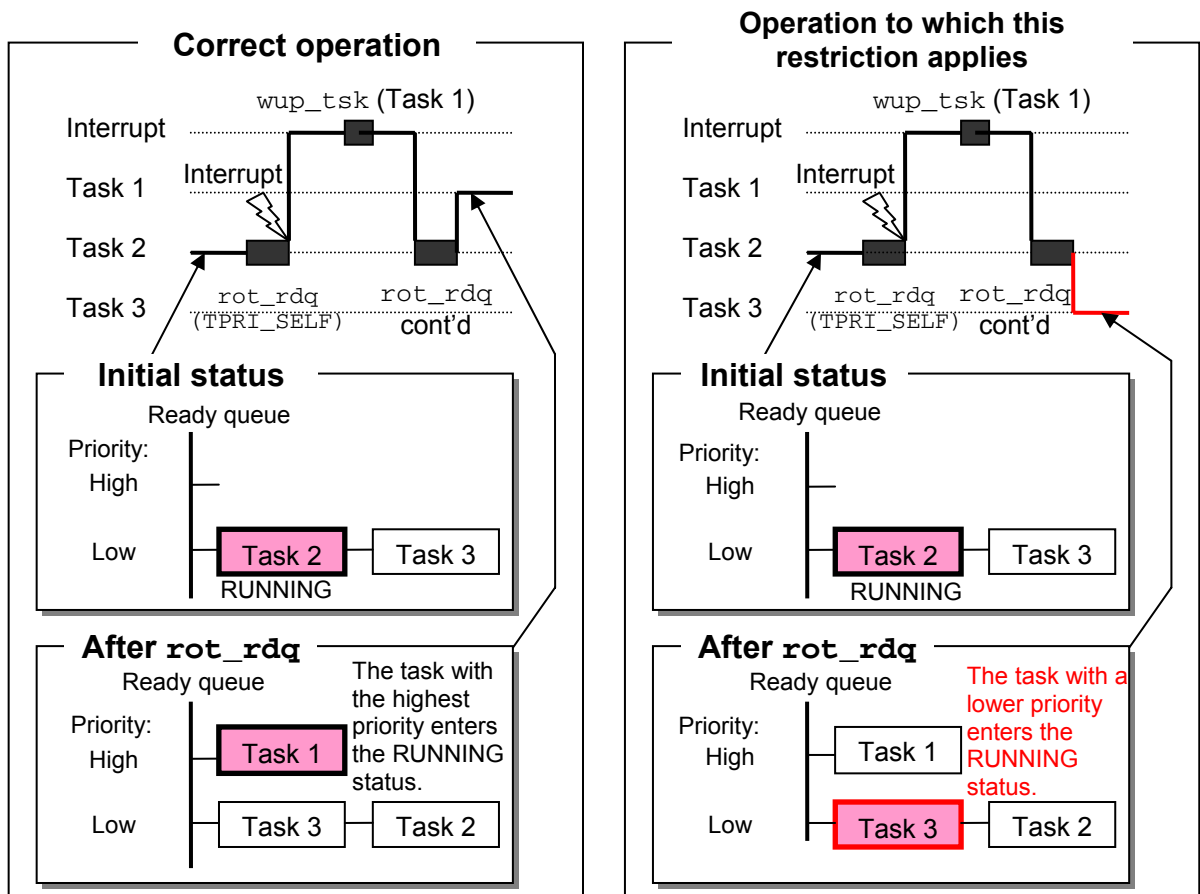
2. Restriction Details

No. 1 Restriction whereby task scheduling is performed incorrectly after executing the service call `rot_rdq` for which `TPRI_SELF` is specified

Description:

If `TPRI_SELF` is specified as the first parameter of the service call `rot_rdq` (to rotate the tasks on the ready queue) in a task and then the task is executed, an interrupt is acknowledged during the processing of the service call `rot_rdq` and, during the interrupt service routine, if a task that has a higher priority than that of the task that executed the service call `rot_rdq` enters the `READY` status, the task that has a lower priority enters the `RUNNING` status after the service call `rot_rdq` finishes, even though the task that has a higher priority should.

This restriction also applies to the service call `ivot_rdq`.



Workaround:

Do one of the following:

- Specify the task priority order before issuing the service call `rot_rdq`.
- Disable interrupts before issuing the service call `rot_rdq`.

Correction:

This issue has been corrected in the RX850V4 Ver. 4.30 kernel.

No. 2 Restriction on the service call `rot_rdq` issued with `TPRI_SELF` specified when dispatching is disabled

Description:

If the service call `rot_rdq` is issued with `TPRI_SELF` specified while dispatching is disabled (by issuing the service call `dis_dsp`), the subsequent operations differ from those prescribed by the specifications.

Originally, the ready queue is rotated and task processing continues. If this restriction applies, however, the ready queue is not rotated, an `E_CTX` error is returned, and task processing continues.

This restriction also applies to the service call `irotd_rdq`.

Workaround:

Do one of the following:

- Specify the task priority order before issuing the service call `rot_rdq`.
- Enable dispatching before issuing the service call `rot_rdq`.

Correction:

This issue has been corrected in the RX850V4 Ver. 4.30 kernel.

No. 3 Processing cannot return to a restricted task

Description:

If an interrupt managed by the OS occurs during OS processing in which restricted task A is switched to another task, B, the context of restricted task A that has been saved in the system stack becomes invalid.

As a result, the processing might not return to restricted task A or, even if the processing returns to restricted task A, the register information might differ from that before the tasks were switched.

Note that interrupts cannot be prevented from occurring during the OS processing period in question.

Workaround:

Do not use restricted tasks.

Correction:

This will not be corrected, so regard it as a specification.

No. 4 AND and OR specified together as the wait conditions for the same event flag during processing of the same task

Description:

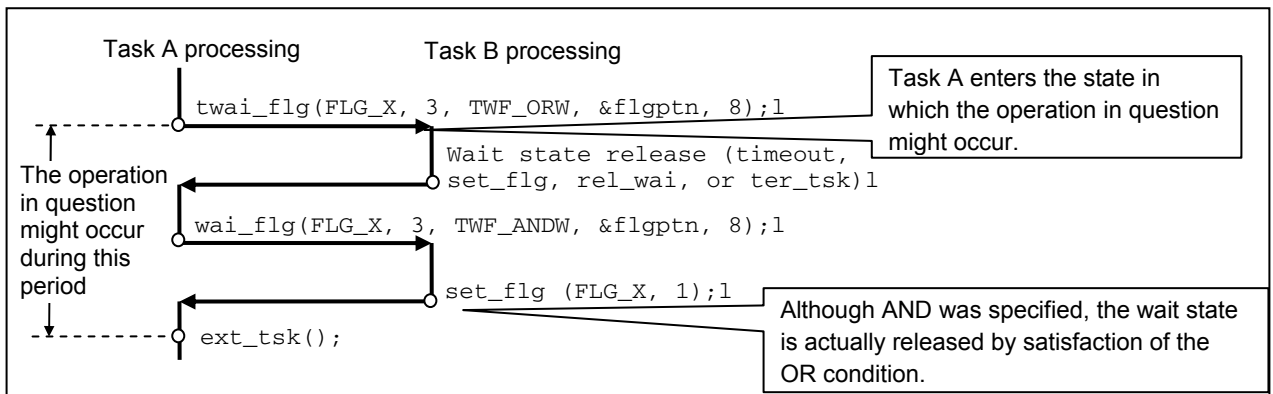
When the service call `wai_flg` or `twai_flg` is issued multiple times with both OR and AND specified at least once as the bit pattern checking condition for the same event flag during processing of the same task, if a subsequent service call is issued with AND specified, the condition might be handled as OR.

The operation in question might occur under the following conditions:

1. During task A processing, `wai_flg` or `twai_flg` is issued for event flag X, with OR specified as the bit pattern checking condition.
Task A then enters the state in which the operation in question might occur.
2. During task A processing, `wai_flg` or `twai_flg` is issued for event flag X, with AND specified as the bit pattern checking condition.
Task A then enters the event flag wait state.
3. During another process, `set_flg` is issued for event flag X, with OR specified to release the wait state.

Although AND is specified as the condition for releasing the wait state in 2, the wait state is actually released by satisfaction of the OR condition.

Example:



To recover from the state in which the operation in question might occur, issue `ext_tsk` or `ter_tsk` for the relevant task, or initialize the operating system.

Workaround:

Do not specify AND and OR together as the wait conditions for the same event flag during processing of the same task.

Correction:

This will not be corrected, so regard it as a specification.

No. 5 A cyclic handler might start up earlier than the specified cycle time

Description:

When an interrupt handler issues the service call `sta_cyc` for a cyclic handler for which no PHS attribute is specified, the cyclic handler might start up earlier than the specified cycle time.

This operation occurs if all the following conditions are satisfied:

- Cyclic handler A starts after its cycle phase elapses following the issuance of the timer interrupt before timer interrupt T (timer interrupt T-1).
- At least two cyclic handlers, excluding cyclic handler A that was started by timer interrupt (T-1), are waiting to start up when timer interrupt T occurs.
- When timer interrupt T occurs, there is a cyclic handler whose cycle time exceeds the time at which cyclic handler A will start up next time.
- When timer interrupt T occurs, another interrupt occurs during the processing to compare the cycle time of cyclic handler A and the other cyclic handlers waiting to start up to determine the handler startup order, and `sta_cyc` is issued for the relevant cyclic handler in that interrupt handler.

When this operation occurs, among the cyclic handlers waiting to start up when timer interrupt T occurs, all the handlers whose cycle time is longer than that of cyclic handler A start up earlier than the specified time. The time until cyclic handler A starts up next time is then brought forward.

Workaround:

Do one of the following:

- Do not allow an interrupt handler to issue the service call `sta_cyc` for a cyclic handler for which no PHS attribute is specified.
- Design the system so that only two cyclic handlers can be waiting to be started up at one time.

Correction:

This will not be corrected, so regard it as a specification.

CubeSuite-Related Usage Restrictions

1. Product History

No.	Usage Restrictions	Version					
		V4.22	V4.30	V4.30 Rev. 2	V4.40	V4.41	V4.42
1	Restriction whereby the Help menu cannot be opened from a message by using the F1 key	√	√				
2	Restriction whereby the task information on the Realtime OS Resource Information panel is not displayed correctly	√	√				

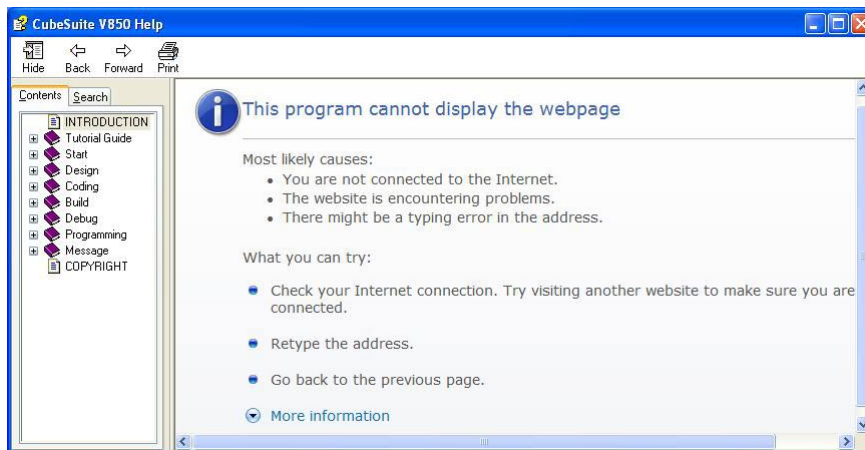
√: Applicable

2. Restriction Details

No. 1 Restriction whereby the **Help** menu cannot be opened from a message by using the **F1** key

Description:

If the **F1** key is pressed while the cursor is pointing to a message output by the build tool in an RX850V4 project, the **Help** menu does not open with an appropriate explanation. Instead, the following **Help** message is displayed:



Workaround:

On the menu bar, click **Help** and then **Help for CubeSuite** to open the **CubeSuite Help** dialog box. Search for the appropriate **Help** message in this dialog box.

Correction:

This issue has been corrected in RX850V4 Ver. 4.30 Rev. 2.

No. 2 Restriction whereby the task information on the **Realtime OS Resource Information** panel is not displayed correctly

Description:

If no task exception handling routine has been defined, some of the task information on the **Realtime OS Resource Information** panel will not be displayed correctly.

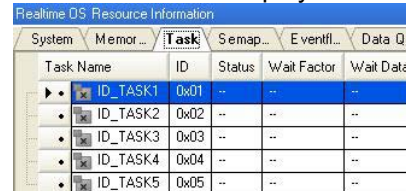
If the statuses, wait factors, and wait data should be displayed, for example, these items are displayed as hyphens if this restriction applies.

Correct display:



Task Name	ID	Status	Wait Factor	Wait Data
ID_TASK1	0x01	Dormant	--	--
ID_TASK2	0x02	Waiting	FLG(0x01) DRW FIFO	0x1
ID_TASK3	0x03	Running	--	--
ID_TASK4	0x04	Ready	--	--
ID_TASK5	0x05	Waiting	SEM(0x01) FIFO	--

Incorrect display:



Task Name	ID	Status	Wait Factor	Wait Data
ID_TASK1	0x01	--	--	--
ID_TASK2	0x02	--	--	--
ID_TASK3	0x03	--	--	--
ID_TASK4	0x04	--	--	--
ID_TASK5	0x05	--	--	--

Workaround:

Define at least one dummy task exception handling routine.

Correction:

This issue has been corrected in RX850V4 Ver. 4.30 Rev. 2.