

Notes on Using the Real-Time OS --M3T-MR100/4-- for the R32C/100 MCU Series

Please take note of the following problems in using the real-time OS-- M3T-MR100/4--for the R32C/100 series of MCUs:

- With using the timeout function
 - With placing a fixed-length memory pool in a memory area
-

1. Problem with Using the timeout Function

1.1 Product and Version Concerned

M3T-MR100/4 V.1.00 Release 00 (for the R32C/100 series)

1.2 Description

When 128 or more tasks are waiting for delays or timeouts at a time, the data that the kernel manages may become incorrect. So handling of system-clock interrupts and that of service calls for clearing waiting states are not completed properly, resulting in an incorrect operation (for example, hanging-up) being performed.

1.3 Conditions

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

- (1) In the program exist 128 or more tasks at a time which are waiting for delays or timeouts.
- (2) Generation of a system-clock interrupt or issuance of a service call for clearing the waiting state of a task waiting for a delay or a timeout clears the task's waiting state. Here, the clearance of the task's waiting state includes the forced clearance of the waiting state and a timeout.

1.4 Workaround

No measures to avoid this problem are available. Design your program so that it might not contain 128 or more tasks waiting for delays or timeouts at a time.

1.5 Schedule of Fixing the Problem

This problem has been resolved in M3T-MR100/4 V.1.01 Release 00, which has been published on May 20, 2009.

2. Problem with Placing a Fixed-Sized Memory Pool in a Memory Area

2.1 Product and Version Concerned

M3T-MR100/4 V.1.00 Release 00 (for the R32C/100 series)

2.2 Description

When a fixed-sized memory pool is placed at an address exceeding 0x10000, the kernel may write values in an incorrect area.

2.3 Conditions

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

- (1) A fixed-sized memory pool is placed at an address exceeding 0x10000.
- (2) The pget_mpf, tget_mpf, or get_mpf service call is issued to acquire a block of the fixed-sized memory pool.

2.4 Workaround

No measures to avoid this problem are available. Place any fixed-sized memory pool at an address not exceeding 0x10000.

2.5 Schedule of Fixing the Problem

This problem has been resolved in M3T-MR100/4 V.1.01 Release 00, which has been published on May 20, 2009.

[Disclaimer]

The past news contents have been based on information at the time of publication. Now changed or invalid information may be included. The URLs in the Tool News also may be subject to change or become invalid without prior notice.