

## Notes on Using the Real-Time OSes M3T-MR308K/4 and -MR308S/4

Please take note of the following problems in using the real-time OSes M3T-MR308K/4 and -MR308S/4, which are used for the M16C/80 and M32C/80 series of MCUs:

- On issuing `isus_tsk` service calls
- On using the Task Pause function
- On issuing `get_tim` and `iget_tim` service calls

---

### 1. Versions Concerned

Real-time OSes M3T-MR308K/4 and -MR308KS/4 V.4.00 Release 00 and V.4.00 Release 01 (for the M16C/80 and M32C/80 MCU series)

### 2. Problem on Issuing `isus_tsk` Service Calls

#### 2.1 Description

Consider the case where a service call\* has been issued which brings the task having issued this service call to the WAITING state.

If an interrupt has been requested while the above-mentioned service call is being serviced, and the interrupt handler has issued an `isus_tsk` service call, this service call returns `E_OK` as its return value, but the target task of the `isus_tsk` service call will not go to the WAITING-SUSPENDED state.

\* Any one of the service calls described in Condition 2.2-(1) below.

#### 2.2 Conditions

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

- (1) A task issues any one of the following service calls:  
`wai_sem`, `twai_sem`, `wai_flg`, `twai_flg`, `snd_dtq`, `tsnd_dtq`, `rcv_mbx`, `trcv_mbx`, `get_mpf`, `tget_mpf`, `vsnd_dtq`, and `vtsnd_dtq`
- (2) The target object of the service call in (1) has a property of `TA_TPRI`. Note, however, that if this object is an event flag\* that has a property of

TA\_WMUL as well as TA\_TPRI and has not of TA\_CLR, it is not involved in this problem.

- (3) Several tasks other than the task in (1) already join the queue that the task in (1) is to join.
- (4) While the service call in (1) is serviced, an interrupt is requested, and the invoked interrupt handler issues an `isus_tsk` service call.
- (5) The target task of the `isus_tsk` service call issued in (4) is the one that has issued a service call in (1).

\* Any event flag can have properties of TA\_CLR; TA\_WSGL or TA\_WMUL; and TA\_TPRI or TA\_TFIFO.

### 2.3 Workaround

Perform the following procedures to make the target task go to the WAITING-SUSPENDED state:

- (1) Issue an `ista_tsk` service call to invoke the task to go to the WAITING-SUSPENDED state.
- (2) Then issue a `sus_tsk` service call from the above task.

Example:

Presumption: The `DoSuspend` task has the highest priority among all the tasks.

```
-----  
void handler(void)  
{  
  
.....  
  
/* Comment out isus_tsk and issue ista_tsk to invoke task to  
   go to WAITING-SUSPENDED state; pass target task ID to ista_tsk  
   as an invoking code */  
  
/* isus_tsk( ID_task1 ); */  
   ista_tsk( ID_DoSuspend,(VP_INT)ID_task1 );  
  
.....  
  
}  
  
void DoSuspend(VP_INT tskid)  
{
```

```
    sus_tsk(tskid);  
}  
-----
```

## 2.4 Schedule of Fixing the Problem

We plan to fix this problem in the next release of the M3T-MR308K/4 and -MR308S/4.

## 3. Problem on Using the Task Pause Function

### 3.1 Description

Consider the case where the debugging of a program containing the Task Pause function is performed.

If the Task Pause button is pressed while a service call is being serviced, the target task of the service call does not enter the pause state but goes to the object-waiting state though it is displayed in the debugger screen that the task has entered the pause state.

As a result, Pause or Remove Pause operations cannot be performed thereafter.

### 3.2 Conditions

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

- (1) A task issues any one of the following service calls:  
wai\_sem, twai\_sem, wai\_flg, twai\_flg, snd\_dtq, tsnd\_dtq, rcv\_mbx, trcv\_mbx, get\_mpf, tget\_mpf, vsnd\_dtq, and vtsnd\_dtq
- (2) The target object of the service call in (1) has a property of TA\_TPRI. Note, however, that if this object is an event flag\* that has a property of TA\_WMUL as well as TA\_TPRI and has not of TA\_CLR, the object is not involved in this problem.
- (3) Several tasks other than the task in (1) already join the queue that the task in (1) is to join.
- (4) Either of the following emulator debuggers is used:
  - the M32C PC7501 emulator debugger V.1.00 Release 00
  - the M32C PC4701 emulator debugger V.1.00

\* Any event flag can have properties of TA\_CLR; TA\_WSGL or TA\_WMUL; and TA\_TPRI or TA\_TFIFO.

### 3.3 Workaround

If this problem occurs, reset your debugger to restart debugging.

### 3.4 Schedule of Fixing the Problem

We plan to fix this problem in the next release of the M32C emulator debugger.

## 4. Problem on Issuing `get_tim` and `iget_tim` Service Calls

### 4.1 Description

If you acquire system time by issuing a `get_tim` or `iget_tim` service call, the upper 16 bits of `ltime`, a member of a structure giving system time, will have an incorrect value.

### 4.2 Condition

This problem may occur if the interrupt period for acquiring system time is set to a value other than 1 ms in the configuration file.

Example:

```
-----  
system{  
    tic_num = 2;  
    tic_deno = 1;  
    . . . . .  
};  
-----
```

### 4.3 Workaround

Set the interrupt period for acquiring system time to 1 ms in the configuration file.

Example:

```
-----  
system{  
    tic_num = 1;  
    tic_deno = 1;  
    . . . . .  
  
};  
-----
```

#### 4.4 Schedule of Fixing the Problem

We plan to fix this problem in the next release of the M3T-MR308K/4 and -MR308S/4.

---

#### **[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.

© 2010-2016 Renesas Electronics Corporation. All rights reserved.