

## M16Cシリーズ用リアルタイムOS M3T-MR30/4 V.4.00 Release 00 ご使用上のお願い

M16Cシリーズ\*用リアルタイムOS M3T-MR30/4 V.4.00 Release 00 の使用上の 注意事項を連絡します。

\*M16C/60, /30, /20, /10, および /Tinyシリーズの総称です。

- ter\_tskサービスコールの使用に関する注意事項

### 1. 内容

タイムアウト指定付サービスコール\*により待ち状態になっているタスクにter\_tskサービスコールを発行した場合、そのタスクがオブジェクト待ちキューから削除されません。その結果、2項の例に示すような現象が発生します。

\*twai\_sem,twai\_flg,trcv\_mbx,tsnd\_dtq,trcv\_dtq,tget\_mpf,vtsnd\_dtq, および vtrcv\_dtqサービスコール

### 2. 現象

#### 例1

```
-----  
void main( VP_INT stacd )  
{  
    sta_tsk(ID_task1,0);  
    sta_tsk(ID_task2,0);  
  
    ter_tsk(ID_task1);  
    sig_sem(ID_sem1);          /* (1) */  
}  
void task1( VP_INT stacd )  
{  
    twai_sem(ID_sem1,100);
```

```

}
void task2( VP_INT stacd )
{
    wai_sem(ID_sem1);
}

```

-----

本来は、上記(1)のsig\_semサービスコールを発行するとtask2のセマフォ待ち状態が解除され、task2はRUNNING状態に移行します。

しかし、本問題により、task2のセマフォ待ち状態は解除されず、task1がRUNNING状態になります。

補足: タスク優先度は、task1が一番高く、以下 task2, mainの順序。

## 例2

```

-----
void main( VP_INT stacd )
{
    sta_tsk(ID_task1,0);
    sta_tsk(ID_task2,0);

    ter_tsk(ID_task1);
    sta_tsk(ID_task1,0);    /* (1) */
    while(1){}            /* (2) */
}
void task1( VP_INT stacd )
{
    twai_sem(ID_sem1,100);
}

void task2( VP_INT stacd )
{
    wai_sem(ID_sem1);
}
void func(void)
{
    while(1){}            /* (3) */
}

```

-----

本来は、上記(1)で発行したsta\_tskサービスコールにより、task1,task2ともにセマフォ待ち状態に移行しmainの上記(2)を実行します。

しかし、本問題によりセマフォ待ちキューおよびレディキューが破壊され、待ち状態であるにもかかわらずtask2が実行され続けます。

その結果、本来実行されることのないtask2に続く関数funcの上記(3)が実行され、システムが暴走してしまいます。

補足: タスク優先度は、task1が一番高く、以下 task2, mainの順序。

### 3. 発生条件

以下の条件をすべて満たす場合に問題が発生します。

- (1) ter\_tskサービスコールを使用している。
- (2) ter\_tskサービスコールの対象タスクが、以下のいずれかのサービスコール発行により待ち状態となっている。

twai\_sem, twai\_flg, trcv\_mbx, tsnd\_dtq, trcv\_dtq, tget\_mpf, vtsnd\_dtq,  
および vtrcv\_dtqサービスコール

### 4. 回避策

以下の手順で回避してください。

- (1) dis\_dspサービスコール発行によりディスパッチを禁止します。  
これによりrel\_waiサービスコールによるタスク切り替えが発生しなくなります。
- (2) rel\_waiサービスコール発行により、対象タスクの待ち状態を強制解除します。
- (3) 対象タスクに対してter\_tskサービスコールを発行します。
- (4) ena\_dspサービスコール発行によりディスパッチ禁止状態を解除します。

記述例

```
-----  
void main( VP_INT stacd )  
{  
    sta_tsk(ID_task1,0);  
  
    dis_dsp();           /* 回避策の(1) */  
    rel_wai(ID_task1);   /* 回避策の(2) */  
    ter_tsk(ID_task1);   /* 回避策の(3) */  
    ena_dsp();           /* 回避策の(4) */  
  
}  
void task1( VP_INT stacd )  
{  
    twai_sem(ID_sem1,100);  
}  
-----
```

### 5. 恒久対策

次期バージョンで改修する予定です。

---

**【免責事項】**

過去のニュース内容は発行当時の情報をもとにしており、現時点では変更された情報や無効な情報が含まれている場合があります。ニュース本文中のURLを予告なしに変更または中止することがありますので、あらかじめご承知ください。

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