

リアルタイムOS MRシリーズご使用上のお願い

リアルタイムOS MRシリーズの使用上の注意事項を連絡します。

- set_flgまたはiset_flgサービスコールとichg_priサービスコールを組み合わせて使用する際の注意事項
- TA_TPRI属性オブジェクトの使用に関する注意事項
- 複数のタスクが待っているイベントフラグをクリアする際の注意事項

1. set_flgまたはiset_flgサービスコールとichg_priサービスコールを組み合わせて使用する際の注意事項

1.1 該当製品

M32Rファミリ M32R/ECUシリーズ M3T-MR32R/4 V.4.00 Release 00
M16Cファミリ M32C/80, M16C/80シリーズ M3T-MR308/4
V.4.00 Release 00 ~ V.4.00 Release 02, および V.4.00 Release 02A
M16Cファミリ M16C/60, M16C/30, M16C/20, M16C/10, および M16C/Tiny
シリーズ M3T-MR30/4 V.4.00 Release 00

1.2 内容

set_flgまたはiset_flgサービスコール処理中に割り込みが発生し、その割り込みハンドラからichg_priを発行した場合、優先度を変更したタスクの待ち解除条件を満たしたとしても待ちが解除されない場合があります。

1.3 発生条件

1.3.1 M3T-MR30/4およびM3T-MR308/4の場合

下記条件をすべて満たす場合に問題が発生する場合があります。

- (1)TA_WMUL属性のイベントフラグを対象にset_flgまたはiset_flgサービスコールを発行した。
- (2)上記(1)のサービスコール発行中に割り込みが発生し、その割り込みハンドラにおいてichg_priサービスコールを発行した。
- (3)上記(2)で発行したichg_priサービスコールの対象タスクが、上記(1)で発行したset_flgまたはiset_flgサービスコールの対象イベントフラグの待ち行列に

つながれている。

1.3.2 M3T-MR32R/4の場合

下記条件をすべて満たす場合に問題が発生する場合があります。

- (1)TA_WMUL属性およびTA_TPRI属性を持つイベントフラグのIDを引数にして set_flgまたはiset_flgサービスコールを発行した。
- (2)上記(1)のサービスコール発行中に割り込みが発生し、その割り込みハンドラにおいてichg_priサービスコールを発行した。
- (3)上記(2)で発行したichg_priサービスコールの対象タスクが、上記(1)で発行したset_flgまたはiset_flgサービスコールの対象イベントフラグの待ち行列につながれている。

1.4 回避策

項目1.3.1または1.3.2の発生条件(1)のset_flgもしくはiset_flgサービスコール呼び出しの前後で割り込みの禁止および許可を行ってください。

1.4.1 M3T-MR30/4およびM3T-MR308/4の記述例

(1)タスクからset_flgサービスコールを発行した場合

割り込み許可フラグのセット、クリアを使用せず、プロセッサ割り込み優先レベルIPLをカーネル割り込みマスクレベル(OS割り込み禁止レベル)に変更して割り込みを禁止し、その後元の値に戻すことによって割り込みを許可します。

```
-----  
void task(VP_INT exinf)  
{  
    :  
    /* 割り込み禁止処理 */  
    #pragma ASM  
    ; 下記例はカーネル割り込みマスクレベル(OS割り込み禁止レベル)が7の  
    ; 場合を示しています。  
    LDIBL #7  
    NOP  
    NOP  
    NOP  
    #pragma ENDASM  
    set_flg(ID_flg,0x000000F0);  
    /* 割り込み許可処理 */  
    #pragma ASM  
    ; タスク実行中のIPLの値は、通常0です。  
    LDIBL #0  
    NOP  
    NOP  
    NOP  
    #pragma ENDASM  
    :  
}
```

}

(2)割り込みハンドラからiset_flgサービスコールを発行した場合

下記(a),(b)のいずれかの対策を実施してください。

(a)タスクと同様にプロセッサ割り込み優先レベルIPLをカーネル割り込みマスクレベル(OS割り込み禁止レベル)に変更して割り込みを禁止し、元の値に戻すことによって割り込みを許可します。

(b)割り込み許可フラグのクリアとセットによって割り込みの禁止と許可を行います。

void inthand(void)

{

:

/* 回避例(a) */

/* 割り込み禁止処理 */

#pragma ASM

; 下記例はカーネル割り込みマスクレベル(OS割り込み禁止レベル)が7の場合を示しています。

LDIPL #7

NOP

NOP

NOP

#pragma ENDASM

iset_flg(ID_flg,0x00F0);

/* 割り込み許可処理 */

#pragma ASM

; 割り込みハンドラからからiset_flgを発行する直前のIPL値に戻します。

LDIPL #3

NOP

NOP

NOP

#pragma ENDASM

:

/* 回避例(b) */

/* 割り込み禁止処理 */

#pragma ASM

FCLR I

#pragma ENDASM

iset_flg(ID_flg,0x00F0);

/* 割り込み許可処理 */

#pragma ASM

FSET I

NOP

```
#pragma ENDASM
```

```
  :  
}
```

注意:

ご使用になるマイコンのLDIPL命令によるIPLが変更されるタイミング、FSET, FCLR命令によるIフラグ変化が反映されるタイミングに合わせてLDIPL, FSET, およびFCLR命令の後に適宜NOP命令挿入または削除などの処理を実施してください。

1.4.2 M3T-MR32R/4の記述例

```
-----  
void task(VP_INT exinf)  
{  
  :  
  /*割り込み禁止処理*/  
  asm(" mvfc    R0,PSW¥n"  
      " and3    R0,R0,#0xFFBF¥n"  
      " mvtc    R0,PSW¥n");  
  set_flg(ID_flg,0x000000F0);  
  /*割り込み許可処理*/  
  asm(" mvfc    R0,PSW¥n"  
      " or3     R0,R0,#0x0040¥n"  
      " mvtc    R0,PSW¥n");  
  :  
}  
-----
```

1.5 恒久対策

次期バージョンで改修する予定です。

2. TA_TPRI属性オブジェクトの使用に関する注意事項

2.1 該当製品

M32Rファミリ M32R/ECUシリーズ M3T-MR32R/4 V.4.00 Release 00
M16Cファミリ M32C/80, M16C/80シリーズ M3T-MR308/4
V.4.00 Release 00 ~ V.4.00 Release 02, および V.4.00 Release 02A
M16Cファミリ M16C/60, M16C/30, M16C/20, M16C/10, および M16C/Tiny
シリーズ M3T-MR30/4 V.4.00 Release 00

2.2 内容

TA_TPRI属性のオブジェクトを対象に待ち状態に移行するサービスコールを呼び出し、そのサービスコール中に発生した割り込みハンドラがその待ち状態を解除するようなサービスコールを複数回呼び出した場合、正しくタスク状態が変化しない場合があります。さらに、正しく待ち行列またはレディキューにつな

がれず、不定番地へ不定値の書き込みが発生する場合があります。

2.3 発生条件

2.3.1 M3T-MR30/4およびM3T-MR308/4の場合

発生条件A, Bのいずれかを満たす場合に発生します。

(1)発生条件A

以下(a)から(c)の条件を満たす場合に必ず発生します。

(a)以下に示すTA_TPRI属性のオブジェクトの待ち行列に複数のタスクがつながれている。

- ・セマフォ待ち行列
- ・データキュー送信待ち行列
- ・longデータキュー送信待ち行列(M3T-MR30/4の場合のみ)
- ・shortデータキュー送信待ち行列(M3T-MR308/4の場合のみ)
- ・固定長メモリプールメモリ獲得待ち行列
- ・メールボックスメッセージ受信待ち行列

(b)上記(a)のオブジェクトを対象に待ち状態に移行するサービスコールを発行した。

(c)上記(b)で発行したサービスコール処理中に割り込みが発生し、その割り込みハンドラにおいて上記(a)で待ち行列につながれているすべてのタスクの待ちを解除した。

発生条件Aの発生例1:

既に2つのタスクがTA_TPRI属性を持つID_sem1のセマフォ待ち行列につながれているものとします。(条件(a)に該当)

```
-----  
void task1(VP_INT exinf)  
{  
    :  
    wai_sem(ID_sem1); /* 条件(b) */  
    :  
}  
/* 上記wai_sem実行中に割り込みが発生し、対応する割り込みハンドラ  
   inthandが実行される */  
void inthand(void)  
{  
    :  
    isig_sem(ID_sem1); /* 条件(c) */  
    :  
    isig_sem(ID_sem1); /* 条件(c) */  
    :  
}  
-----
```

動作説明:

発生条件を満たした場合、本来task1は、セマフォ待ち状態に移行し、セマフォ待ち行列につながれます。

しかし、実際にはセマフォ状態に移行するもののセマフォ待ち行列につながれず不定番地への不定値の書き込みが発生します。

発生条件Aの発生例2:

既に2つのタスクがTA_TPRI属性を持つID_sem1のセマフォ待ち行列につながれているものとします。(条件(a)に該当)

```
-----  
void task1(VP_INT exinf)  
{  
    :  
    wai_sem(ID_sem1); /* 条件(b) */  
    :  
}  
/* 上記wai_sem実行中に割り込みが発生し、対応する割り込みハンドラ  
   inthandが実行される。さらにinthand実行中に多重割り込みが発生し  
   inthand2が実行される。 */  
void inthand(void)  
{  
    :  
    isig_sem(ID_sem1); /* 条件(c) */  
    :  
}  
void inthand2(void)  
{  
    :  
    isig_sem(ID_sem1); /* 条件(c) */  
    :  
    isig_sem(ID_sem1); /* 条件(c) */  
    :  
}  
-----
```

動作説明:

発生条件を満たした場合、本来task1は、割り込みハンドラinthand2で発行された2番目のisig_semサービスコール発行によってセマフォを獲得するため、待ち状態に移行することはありません。

しかし、実際にはセマフォ状態に移行し、さらにセマフォ待ち行列にもつながれず不定番地への不定値の書き込みが発生します。

(2)発生条件B

以下(a)と(b)の条件を満たし、さらに条件(i)もしくは(ii)のいずれかの条件を満たす場合に発生することがあります。

(a)TA_TPRI属性, TA_WMUL属性, およびTA_CLR属性のすべてを持つイベントフラグのイベントフラグ待ち行列に複数のタスクが存在する。

(b)上記(a)のイベントフラグを対象にwai_flgもしくはtwai_flgサービスコールを実行した。この場合、wai_flgもしくはtwai_flgサービスコールの引数waipnは、待ち条件を満たさない。

条件(i)

上記(b)で発行したwai_flgもしくはtwai_flgサービスコール実行中に割り込みが発生し、その割り込みハンドラにおいて上記(a)のイベントフラグを対象にiset_flgサービスコールが発行され、上記(b)で発行したwai_flgもしくはtwai_flgサービスコールの待ち条件を満たすようにビットパターンをセットした。

条件(ii)

上記(b)で発行したwai_flgもしくはtwai_flg実行中に割り込みが発生し、その割り込みハンドラにおいて上記(a)のイベントフラグを対象にiset_flgサービスコールが複数回発行され、そのイベントフラグ待ちキューにつながっているすべてのタスクの待ちが解除された。

発生条件Bの発生例1:

既に2つのタスクがTA_TPRI属性, TA_WMUL属性, およびTA_CLR属性を持つID_flg1のイベントフラグ待ち行列につながれているものとします。

(条件(a)に該当)

```
-----  
void task1(VP_INT exinf)  
{  
    FLGPTN flgptn;  
    :  
    wai_flg(ID_flg1,(FLGPTN)0x000F,TWF_ANDW,&flgptn); /* 条件(b) */  
    :  
}  
/* 上記wai_flg実行中に割り込みが発生し、対応する割り込みハンドラ  
   inthandが実行される。 */  
void inthand(void)  
{  
    :  
    iset_flg(ID_flg1,(FLGPTN)0x000F); /* 条件(i) */  
    :  
}  
-----
```

動作説明:

発生条件を満たした場合、本来task1は、割り込みハンドラinthandで発行されたiset_flgサービスコール発行によって待ち条件が満たされるため、待ち状態に移行することはありません。

しかし、実際にはイベントフラグ待ち状態に移行し、さらにイベントフラグ待ち行列につながれます。

発生条件Bの発生例2:

既に2つのタスク(task2およびtask3)がTA_TPRI属性, TA_WMUL属性, およびTA_CLR属性を持つID_flg1のイベントフラグ待ち行列につながれているものとします。(条件(a)に該当)

```
-----  
void task1(VP_INT exinf)  
{  
    FLGPTN flgptn;  
    :  
    wai_flg(ID_flg1,(FLGPTN)0x000F,TWF_ANDW,&flgptn); /* 条件(b) */  
    :  
}  
void task2(VP_INT exinf)  
{  
    FLGPTN flgptn;  
    :  
    wai_flg(ID_flg1,(FLGPTN)0x00F0,TWF_ANDW,&flgptn); /* 条件(a) */  
    :  
}  
void task3(VP_INT exinf)  
{  
    FLGPTN flgptn;  
    :  
    wai_flg(ID_flg1,(FLGPTN)0x0F00,TWF_ANDW,&flgptn); /* 条件(a) */  
    :  
}  
/* 上記wai_flg実行中に割り込みが発生し、対応する割り込みハンドラ  
   inthandが実行される。 */  
void inthand(void)  
{  
    :  
    iset_flg(ID_flg1,(FLGPTN)0x00F0); /* 条件(ii) */  
    :  
    iset_flg(ID_flg1,(FLGPTN)0x0F00); /* 条件(ii) */  
    :  
}
```

動作説明:

発生条件を満たした場合、本来task1は、割り込みハンドラinthandで発行されたiset_flgサービスコール発行によって待ち条件が満たされることはないため、イベントフラグ待ち状態に移行し、イベントフラグ待ち行列につながれます。しかし、実際にはイベントフラグ待ち状態に移行しますがイベントフラグ待ち行列につながれず、不定番地への不定値の書き込みが

発生することがあります。

2.3.2 M3T-MR32R/4の場合

発生条件A, Bのいずれかを満たす場合に発生することがあります。

(1)発生条件A

以下(a)から(c)の条件をすべて満たした場合に必ず発生します。

(a)以下に示すTA_TPRI属性のオブジェクトの待ち行列に4つ以上のタスクが
つながれている。

- ・セマフォ待ち行列
- ・データキュー送信待ち行列
- ・固定長メモリプールメモリ獲得待ち行列
- ・メールボックスメッセージ受信待ち行列

(b)上記(a)のオブジェクトを対象に待ち状態に移行するサービスコールを
発行した。

(c)上記(b)で発行したサービスコール処理中に割り込みが発生し、その割
り込みハンドラにおいてサービスコールを発行し、上記(a)で待ち行列
につながれているすべてのタスクの待ちを解除し、さらに上記(b)に示
すサービスコール発行タスクの待ち解除条件を満たした。

発生条件Aの発生例:

既に4つのタスクがTA_TPRI属性を持つID_sem1のセマフォ待ち行列につなが
れているものとします。(条件(a)に該当)

```
-----  
void task1(VP_INT exinf)  
{  
    :  
    wai_sem(ID_sem1); /* 条件(b) */  
    :  
}  
/* 上記wai_sem実行中に割り込みが発生し、対応する割り込みハンドラ  
   inthandが実行される */  
void inthand(void)  
{  
    :  
    isig_sem(ID_sem1); /* 条件(c) */  
    :  
}
```

動作説明:

発生条件を満たした場合、本来task1は、割り込みハンドラinhandで発行されたisig_semによってセマフォを獲得し、待ち状態に移行することはありません。しかし、実際にはセマフォ状態に移行し、さらにセマフォ待ち行列につながれてしまいます。また、セマフォカウントは本来0となるべきですが、1となります。

(2)発生条件B

以下(a)から(c)の条件をすべて満たした場合に発生することがあります。

- (a)TA_TPRI属性, TA_WMUL属性両方を持つイベントフラグのイベントフラグ待ち行列に4つ以上のタスクが存在する。
- (b)上記(a)のイベントフラグを対象にwai_flgもしくはtwai_flgサービスコールを実行した。この場合、wai_flgもしくはtwai_flgサービスコールの引数waipnは待ち条件を満たさない。
- (c)上記(b)で発行したwai_flgもしくはtwai_flgサービスコール実行中に割り込みが発生し、その割り込みハンドラにおいて上記(a)のイベントフラグを対象にiset_flgサービスコールが発行され、上記(b)で発行したwai_flgもしくはtwai_flgサービスコールの待ち条件を満たすようにイベントフラグのビットパターンをセットした。

発生条件Bの発生例:

既に4つのタスクがTA_TPRI属性, TA_WMUL属性両方を持つID_flg1のイベントフラグ待ち行列につながれているものとしします。(条件(a)に該当)

```
void task1(VP_INT exinf)
{
    FLGPTN flgptn;
    :
    wai_flg(ID_flg1,(FLGPTN)0x000F,TWF_ANDW,&flgptn); /* 条件(2) */
    :
}
/* 上記wai_flg実行中に割り込みが発生し、対応する割り込みハンドラ
   inhandが実行される。 */
void inhand(void)
{
    :
    iset_flg(ID_flg1,(FLGPTN)0x000F); /* 条件(i) */
    :
}
```

動作説明:

発生条件を満たした場合、本来task1は、割り込みハンドラinhandで発行されたiset_flgサービスコール発行によって待ち条件が満たされるため、待ち状態に移行することはありません。

しかし、実際にはイベントフラグ待ち状態に移行し、さらにイベントフラグ待ち行列につながれます。

2.4 回避策

発生条件(b)のサービスコール呼び出しの前後で割り込みの禁止、許可を行ってください。

M3T-MR30/4,M3T-MR308/4の記述例:

割り込み許可フラグのセット、クリアを使用せず、プロセッサ割り込み優先レベルIPLをカーネル割り込みマスクレベル(OS割り込み禁止レベル)に変更して割り込みを禁止し、元の値に戻すことによって割り込を許可します。

```
-----  
void task(VP_INT exinf)  
{  
    :  
    /* 割り込み禁止処理 */  
    #pragma ASM  
    ; 下記例はカーネル割り込みマスクレベル(OS割り込み禁止レベル)が7の場合  
    ; を示しています。  
    LDIPL #7  
    NOP  
    NOP  
    NOP  
    #pragma ENDASM  
    wai_sem(ID_sem1); /* 条件(b) */  
    /* 割り込み許可処理 */  
    #pragma ASM  
    ; タスク実行中のIPLの値は、通常0です。  
    LDIPL #0  
    NOP  
    NOP  
    NOP  
    #pragma ENDASM  
    :  
}
```

注: ご使用になるマイコンのLDIPL命令によるIPLが変更されるタイミング、FSET, FCLR命令によるIフラグ変化が反映されるタイミングに合わせてLDIPL, FSET, FCLR命令の後に適宜NOP命令挿入,削除などの処理を実施してください。

M3T-MR32R/4の記述例:

```
-----  
void task(VP_INT exind)  
{
```

```

:
/*割り込み禁止処理*/
asm(" mvfc    R0,PSW¥n"
     " and3    R0,R0,#0xFFBF¥n"
     " mvtc    R0,PSW¥n");
wai_sem(ID_sem1); /* 条件(2) */
/*割り込み許可処理*/
asm(" mvfc    R0,PSW¥n"
     " or3     R0,R0,#0x0040¥n"
     " mvtc    R0,PSW¥n");
:
}
-----

```

2.5 恒久対策

次期バージョンで改修する予定です。

3. 複数のタスクが待っているイベントフラグをクリアする際の注意事項

3.1 該当製品

M32Rファミリ M32R/ECUシリーズ M3T-MR32RおよびM3T-MR32R/4の全バージョン
M16Cファミリ M32C/80, M16C/80シリーズ M3T-MR308およびM3T-MR308/4の全バージョン
M16Cファミリ M16C/60, M16C/30, M16C/20, M16C/10, および M16C/Tiny
シリーズ M3T-MR30およびM3T-MR30/4の全バージョン
7700ファミリ 79xxシリーズ M3T-MR79の全バージョン

3.2 内容

複数のタスクが待っているイベントフラグを対象にset_flgもしくはiset_flgサービスコールを発行し、そのサービスコール実行中に発生した割り込みハンドラから同じイベントフラグを対象としてiset_flgサービスコールを発行した場合、前者のset_flg,iset_flgサービスコールによってタスクの待ちが解除される場合であっても待ちが解除されないことがあります。

3.3 発生条件

以下の条件をすべて満たす場合に発生することがあります。

- (1) イベントフラグのイベントフラグ待ち行列に複数のタスクがつかがれている。
- (2) 上記(1)のイベントフラグを対象にそのイベントフラグ待ち行列のタスクの待ち条件を満たすようなビットパターンを引数としてset_flgもしくは、iset_flgサービスコールを発行した。
- (3) 上記(2)のset_flgもしくはiset_flgサービスコール実行中に割り込みが発生し、その割り込みハンドラから上記(1)のイベントフラグを対象にiset_flgサービスコールを発行し、イベントフラグをクリアする処理(注)を行った。

注： イベントフラグをクリアする処理とは以下の場合を示します。

- (i) TA_CLR属性を持つイベントフラグを対象に(3)で示すiset_flgサービスコールの発行によってイベントフラグを待っているタスクの待ち状態が解除されさらにイベントフラグのビットパターンがクリアされる場合。
(M3T-MR30/4, M3T-MR308/4, およびM3T-MR32R/4のみ該当します。)
- (ii) (3)で示すiset_flgサービスコールを発行し、クリア指定(TWF_CLR)付の待ち条件で待っているタスクの待ちを解除し、イベントフラグのビットパターンをクリアする場合。
(M3T-MR30, M3T-MR308, M3T-MR32R, およびM3T-MR79のみ該当します。)
- (iii) (3)で示すiset_flgサービスコールの発行とともにiclr_flgまたは、clr_flgサービスコールを発行し、イベントフラグのビットパターンを変更した場合。

3.4 回避策

発生条件(2)のサービスコール呼び出しの前後で割り込みの禁止、許可を行ってください。

記述例は、1.4を参照ください。

3.5 恒久対策

M3T-MR79以外の製品は、次期バージョンで改修する予定です。

M3T-MR79の改修予定はありません。

[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。