RENESAS ツールニュース

RENESAS TOOL NEWS 2004年05月01日：RSO-M3T-MR32R-040501D

M3T-MR32R
リファレンスマニュアル訂正のお知らせ

M32Rファミリ用リアルタイムOS M3T-MR32Rリファレンスマニュアルの追加と訂正を連絡します。

1. **該当資料**
   - M3T-MR32R V.3.50 Release 1、およびV.3.50 Release 2に添付のリファレンスマニュアル
   - M3T-MR32R V.3.00 Release 1～V.3.40 Release 1に添付のリファレンスマニュアル（以下、2. (5)と(9)のみの訂正）

2. **内容**

(1) **2.5.1. cre_mbx(Create Mailbox)の「機能説明」**

訂正前
- bufcnt
  対象メイルボックスのメッセージを格納するバッファサイズをここで指定します。 この単位は、バイト数ではなく、メッセージ数を指定します。

訂正後
- bufcnt
  対象メイルボックスのメッセージを格納するバッファサイズをここで指定します。 この単位は、バイト数ではなく、メッセージ数を指定します。
- mbx
  ユーザが確保したメイルボックス領域の先頭アドレスを指定します。

(2) **2.6.1. cre_mbf(Create MessageBuffer)の「機能説明」**

訂正前
- maxmsz
  生成するメッセージバッファで扱うことのできるメッセージの最大長を指定してください。 なお、maxmsz は、MR32R は参照しませんので、この指定は必要ありません。 他のリアルタイムOS との互換性を保つ必要がある場合は、この値を設定してください。

訂正後
- maxmsz
  生成するメッセージバッファで扱うことのできるメッセージの最大長を指定してください。 なお、maxmsz は、MR32R は参照しませんので、この指定は必要あ

りません。 他のリアルタイムOS との互換性を保つ必要がある場合は、この値を設定してください。

- mbf
  ユーザが確保したメッセージバッファ領域の先頭アドレスを指定します。

(3) **2.9.1. cre_mpf(Create Fixed-size Memory Pool)の「機能説明」**

訂正前
- blfsz
  生成するメモリプールの1 ブロックのブロックサイズを指定してください。

訂正後
- blfsz
  生成するメモリプールの1 ブロックのブロックサイズを指定してください。
- mpf
  ユーザが確保した固定長メモリプール領域の先頭アドレスを指定します。

(4) **2.9.9. cre_mpl(Create Variable-size Memory Pool)の「機能説明」**

訂正前
- maxblksz
  MR32R の可変長メモリプールは、4 つの固定長メモリブロックのサイズに分けて、 ユーザが指定したサイズに最も最適なメモリブロックを4つのサイズの中から選択し、 メモリの割り当てを行います。各固定長メモリブロックのサイズは、maxblksz をユーザが指定することで決定します。

訂正後
- maxblksz
  MR32R の可変長メモリプールは、4 つの固定長メモリブロックのサイズに分けて、 ユーザが指定したサイズに最適なメモリブロックを4つのサイズの中から選択し、 メモリの割り当てを行います。各固定長メモリブロックのサイズは、maxblksz をユーザが指定することで決定します。
- mpl
  ユーザが確保した可変長メモリプール領域の先頭アドレスを指定します。

(5) **2.11.2. ref_sys(Refer System Status)の「機能説明」の表中 上から4行目**

訂正前 TSS_LOC　2　タスク部　禁止　禁止訂正後 TSS_LOC　3　タスク部　禁止　禁止

(6) **2.11.3. def_exc(Define Exception Handler)の「アセンブリ言語による呼び出し方法」の「レジスタ設定」**

訂正前 オフセット サイズ　+0　4　excatr　例外ハンドラ属性　+4　4　exchdr　例外ハンドラアドレス　+8　2　tskid　タスクID番号　+12　4　excstksz スタック

サイズ　　訂正後 オフセット サイズ　　+0　4　excatr　例外ハンドラ属性　　+4　4　exchdr　例外ハンドラアドレス　　+8　2　tskid　タスクID番号　　+12　4　excstksz　スタックサイズ　　+16　4　exc　ユーザーが確保した強制例外ハンドラスタック領域の先頭アドレス

(7) **2.11.3. def_exc(Define Exception Handler)の「C 言語による呼び出し方法」の「引数」**

訂正前 typedef struct t_dexc {　　ATR excatr;　/* 例外ハンドラ属性 */　　FP exchdr;　/* 例外ハンドラアドレス*/　　ID tskid;　/* タスクID 番号 */　　W excstksz　/* スタックサイズ */　　}T_DEXC;　　訂正後 typedef struct t_dexc { ATR excatr;　/* 例外ハンドラ属性 */　　FP exchdr;　/* 例外ハンドラアドレス*/ ID tskid;　/* タスクID 番号 */　　W excstksz;　/* スタックサイズ */　　VP exc　　/* ユーザーが確保した強制例外ハンドラスタック　　領域の先頭アドレス*/　　}T_DEXC;

(8) **2.11.3. def_exc(Define Exception Handler)の「機能説明」**

訂正前
- excstksz
  定義する例外ハンドラのスタックサイズを指定してください。本システムコールで、 例外ハンドラが使用するスタック領域のメモリを確保します。
  exchdr=NADR を指定すれば、スタック領域のメモリは解放されます。 指定したサイズ分のメモリが足りない場合は、エラーE_NOMEM を返します。

訂正後
- excstksz
  定義する例外ハンドラのスタックサイズを指定してください。 本システムコールで、例外ハンドラが使用するスタック領域のメモリを確保します。
  exchdr=NADR を指定すれば、スタック領域のメモリは解放されます。 指定したサイズ分のメモリが足りない場合は、エラーE_NOMEM を返します。
- exc
  ユーザが確保した強制例外ハンドラ用スタック領域の先頭アドレスを指定します。

(9) **3.6 共通定数と構造体のパケット形式 の「システム管理関係」**

訂正前 typedef struct t_regs {　　VW r0;　　VW r1;　　VW r2;　　VW r3;　　VW r4;　　VW r5;　　VW r6;　　VW r7;　　VW r8;　　VW r9;　　VW r10;　　VW r11;　　VW r12;　　VW r13;　　VW r14;　　VW sp;　　VW accl;　　VW acch;　　};　　typedef struct t_reit {　　PSW psw;　　FP pc;　　};　　typedef struct t_exc {　　W exckind;　　UW exccd;　　ID tskid;　　UW exeenv;　　};訂正後 /* 例外発生時のレジスタ情報パケット */

```
typedef struct t_regs {
        VW r0;  /* 例外発生時のR0レジスタの内容 */
        VW r1;  /* 例外発生時のR1レジスタの内容 */
        VW r2;  /* 例外発生時のR2レジスタの内容 */
        VW r3;  /* 例外発生時のR3レジスタの内容 */
        VW r4;  /* 例外発生時のR4レジスタの内容 */
        VW r5;  /* 例外発生時のR5レジスタの内容 */
        VW r6;  /* 例外発生時のR6レジスタの内容 */
        VW r7;  /* 例外発生時のR7レジスタの内容 */
        VW r8;  /* 例外発生時のR8レジスタの内容 */
        VW r9;  /* 例外発生時のR9レジスタの内容 */
        VW r10;  /* 例外発生時のR10レジスタの内容 */
        VW r11;  /* 例外発生時のR11レジスタの内容 */
        VW r12;  /* 例外発生時のR12レジスタの内容 */
        VW r13;  /* 例外発生時のR13レジスタの内容 */
        VW r14;  /* 例外発生時のR14レジスタの内容 */
        VW sp;  /* 例外発生時のSPレジスタの内容 */
        VW accl;  /* 例外発生時のACCLレジスタの内容 */
        VW acch;  /* 例外発生時のACCHレジスタの内容 */
        VW acc1l; /* 例外発生時のACC1Lレジスタの内容
(FPU対応カーネル時は、FPSRの値。
M32Rx/D対応カーネル、FPU対応カーネル以外は不定) */
        VW acc1h; /* 例外発生時のACC1Hレジスタの内容
(M32Rx/D以外の場合は不定)      */
        } T_REGS;
        /* EIT 情報パケット */
typedef struct t_reit {
        PSW psw;  /* 例外発生時のPSWレジスタの内容 */
        FP pc;  /* 例外発生時のPCの内容 */
        } T_EIT;
        /* 例外情報パケット */
typedef struct t_exc {
        W exckind; /* 例外の種類 常にEXK_TER=2(強制例外)がセットされる */
        UW exccd;  /* 例外コード */
        ID tskid;  /* タスクID */
        UW exeenv; /* 例外が発生したシステム状態 常にTTE_TSK=0(タスク部
実行中)がセットされる */
        } T_EXC;
```
