

μ ITRON4.0仕様
Ver.4.02.00からVer.4.03.00への改訂箇所

作成者：トロン協会 ITRON仕様検討グループ ITRON仕様書メンテナンスWG
最終更新：2006年11月14日

今回のμ ITRON4.0仕様Ver.4.02.00からVer.4.03.00への改訂では、「ベーシックプロファイル」に関する記述と「実装毎に規定すべき事項(実装定義)一覧表」を追加するとともに、わかりにくい表現・曖昧な表現・不正確な表現の修正や、仕様の本体以外の修正を行った。

このメモは、そのうち、仕様書本体(第2章~第5章)およびリファレンス(第7章)の追記箇所および改訂箇所とその理由をまとめたものである。

このメモ中のページ番号は、Ver.4.03.00でのページ番号を示す。

追記

- ベーシックプロファイルに関する記述
 - ページiii 記述形式「【B】はベーシックプロファイルを示す。」
 - 各対象サービスコールのプロファイル表示部分に【B】を追記
 - P15 図1-2 の後にベーシックプロファイルに関する説明を追記
 - P15 図1-2 のスタンダードプロファイルと自動車制御用プロファイルの間に、「ベーシックプロファイル」を追記
 - 5.2 ベーシックプロファイルに関する説明
- 7.10 実装毎に規定すべき事項(実装定義)一覧表
- P360 静的API索引

仕様書改訂

【改訂箇所】

- P23 : (E) データ型
ITRON仕様でビット数が規定されていないデータ型に対して、実装で定義されるデータ型のビット数よりも少ない有効ビット数ないしはデータ型で表現できる範囲よりも狭い有効範囲を、実装定義に定めることができる。

ITRON仕様でビット数が規定されていないデータ型に対して、C言語の型のビット数よりも少ない有効ビット数ないしは型で表現できる範囲よりも狭い有効範囲を、実装定義に定めることができる。

【改訂理由】

理解しやすさを改善するため。

【改訂箇所】

- P27 : 【μ ITRON3.0仕様との相違】3~4行目
サブエラーコードは、エラーが発生した原因をより細かく報告することを目的としており、主にデバッグ時に使うことを想定している。

サブエラーコードは、エラーが発生した原因をより細かく報告することを目的としている。

【改訂理由】

サブエラーコードは、ソフトウェア部品における実装とデバッグを想定していると判断した。カーネルでの実装と誤解しないように削除した。

【改訂箇所】

- P50 (a)実行状態
- P51 (d)休止状態
- P62 19行目
- P65 【補足説明】2行目
「別に規定されている場合を除いて」の記述を削除。

【改訂理由】

「別に規定」は仕様書で明確に規定されていないため、実装独自を示すものと判断し、仕様としての混乱を避けるために削除した。

【改訂箇所】

- P61 : 【補足説明】の3行目の終わり

CPU例外ハンドラ内で自タスクを広義の待ち状態にする可能性のあるサービスコールを呼び出し可能とした場合であっても、その振る舞いはこの仕様上は未定義である。エラーを報告する場合には、E_CTX エラーを返す。

CPU例外ハンドラ内で自タスクを広義の待ち状態にする可能性のあるサービスコールを呼び出した場合の振る舞いは未定義である。ただし、エラーを報告する場合はE_CTXを返すものとする。

【改訂理由】

理解しやすさを改善するため。

【改訂箇所】

- P63 の下から13~11行目

ただし、割込みハンドラ内で CPUロック解除状態にするための方法を、実装定義で用意することが必要である。

ただし、CPUロック状態または過渡的な状態で、割込みハンドラを実行開始する場合、割込みハンドラ内で CPUロック解除状態にするための方法を、実装定義で用意することが必要である。

【改訂理由】

ここでの実装定義は、「CPUロック状態または過渡的な状態で、割込みハンドラを実行開始する場合」という実装依存仕様を前提とすることを明記するため。

【改訂箇所】

-P65 : 【補足説明】1行目

ディスパッチ禁止状態から自タスクを広義の待ち状態にする可能性のあるサービスコールが呼び出された場合の振舞いを未定義とする制限は、別に規定されている場合を除いて、サービスコール単位に適用される。具体的には、ポーリングを行うサービスコール（例えば pol_sem）は、自タスクを待ち状態にする可能性がないため、ディスパッチ禁止状態でも呼び出すことができる。それに対して、待ち状態に入る可能性のあるサービスコール（例えば twai_sem）は、タイムアウト時間に TMO_POL（ポーリング）を指定して呼び出した場合でも、その振舞いは未定義である。

ディスパッチ禁止状態から自タスクを広義の待ち状態にする可能性のあるサービスコールが呼び出された場合に実装定義でE_CTXエラーとする仕様は、サービスコール単位に適用される。具体的には、タイムアウト付きのサービスコール（例えば twai_sem）において、タイムアウト時間に TMO_POL（ポーリング）を指定して呼び出した場合の振る舞いを、E_CTXエラーとするか、ポーリングを行うサービスコール（例えば pol_sem）と同じ動作とするかは実装定義とする。

【改訂理由】

Ver.4.00.00からVer.4.01.00への改訂箇所(ア-13)(1)に関連する修正漏れ。理解しやすさを改善するため。

【改訂箇所】

- P71 3.7 システム初期化手順 の1~3行目(記述場所の移動と表現の変更)

以上はあくまでも概念モデルを示すものであり、実際のシステム初期化手順は、この概念モデルと同じ振舞いをする限りは、実装依存に最適化することができる。

これはあくまでも概念モデルを示すものであり、実際のシステム初期化手順は、この概念モデルと同じ振舞いをする限りは、実装依存に最適化することができる。

【改訂理由】

システム初期化手順そのものが、実装依存であり、「3.7 システム初期化手順」内に記述される「実装定義」仕様は、実装依存仕様を前提とすることを明記するため。

【改訂箇所】

- P90 : ext_tsk【機能】の最後の文

サービスコール内でエラーを検出した場合の振舞いは実装定義とする。

サービスコール内のエラーを検出するか否かと、もし検出した場合の振る舞いは実装定義とする。

【改訂理由】
エラーの検出についても、実装定義と明記することが妥当であると考えられるため。

【改訂箇所】
- P91 : exd_tsk【機能】の最後の文
サービスコール内でエラーを検出した場合の振る舞いは実装定義とする。

サービスコール内のエラーを検出するか否かと、もし検出した場合の振る舞いは実装定義とする。

【改訂理由】
エラーの検出についても、実装定義と明記することが妥当であると考えられるため。

【改訂箇所】
-P102 :【補足説明】
TMAX_WUPCNTは、タスクの起床待ちをサポートしない場合には未定義、タスクの起床要求キューイングをサポートしない場合には 0とする。また、タスクの強制待ち状態をサポートしない場合には、TMAX_SUSCNTは未定義とする。したがって、TMAX_SUSCNTが 0に定義されることはない。

TMAX_WUPCNTは、タスクの起床待ちをサポートしない場合には定義しない。タスクの起床要求キューイングをサポートしない場合には 0とする。また、タスクの強制待ち状態をサポートしない場合には、TMAX_SUSCNTは定義しない。したがって、TMAX_SUSCNTが 0に定義されることはない。

【改訂理由】
P74に「カーネル構成定数/マクロは、それに関連する機能をサポートしていない場合には定義しない。」と定義されているため。

【改訂箇所】
-P109 sus_tsk の【補足説明】
このサービスコールは、パラメータによっては自タスクを強制待ち状態にする可能性があるにもかかわらず、ディスパッチ禁止状態でも呼び出すことができる。したがって、「ディスパッチ禁止状態から自タスクを広義の待ち状態にする可能性のあるサービスコールが呼び出された場合の振る舞いは未定義とする制限は、サービスコール単位に適用される」という原則の例外となっている。

【補足説明】
このサービスコールは、パラメータによっては自タスクを強制待ち状態にする可能性があるにもかかわらず、ディスパッチ禁止状態でも呼び出すことができる。したがって、「ディスパッチ禁止状態から自タスクを広義の待ち状態にする可能性のあるサービスコールが呼び出された場合に実装定義でE_CTXエラーとする仕様は、サービスコール単位に適用される」という原則の例外となっている。

【改訂理由】
引用元 (P.65の【補足説明】) を変更したため。

【改訂箇所】
- P114 : rsm_tsk/ frsm_tsk【補足説明】の2文目以降
実装独自に、非タスクコンテキストから実行中のタスクを強制待ち状態に移行させるサービスコールを追加した場合や、ディスパッチ禁止状態で自タスクを強制待ち状態に移行させるサービスコールを呼出し可能とした場合には、自タスクの強制待ち要求ネスト数が 1 以上になっている場合がある。このような場合の振る舞いは、実装定義とする。

削除。

【改訂理由】
「対象タスクに自タスクを指定することはできない。」記述と矛盾するため。また、実装独自仕様を、仕様として記述する必要性はないと判断した。

【改訂箇所】

- P230 19～21行目

また、可変長メモリプールから取得可能な最大のメモリブロックサイズよりも大きい値が blkkszに指定された場合、実装依存で E_PARエラーを返すことができる。

また、生成時に指定する可変長メモリプールのメモリ領域のサイズよりも大きい値が blkkszに指定された場合、実装依存で E_PARエラーを返すことができる。

【改訂理由】

「取得可能な最大のメモリブロックサイズ」は、生成時の最大サイズを示すのか、他の制限によるサイズを示すのかが、不明確であるため。

【改訂箇所】

- P291 4.10 の2文目

拡張サービスコールを呼び出すための機能は、標準のサービスコールを呼び出すために用いることもできる。

削除。

【改訂理由】

P295 【機能】の9行目「このサービスコールで、拡張サービスコール以外の標準のサービスコールを呼び出せるかどうかは実装定義である。」の記述と矛盾するため。

誤記修正

- P38

(誤) rdv rendezvous

(正) rdv rendezvous

- P229

(誤) パラメータエラー (p_blk, tmoutが不正)

(正) パラメータエラー (blkksz, p_blk, tmoutが不正)

会社名変更

- P314

(誤) 日立エンジニアリング株式会社

(正) 株式会社日立情報制御ソリューションズ

- P314

(旧) 株式会社エルミックシステム

(新) エルミック・ウェスコム株式会社

メーカーコード新規割り付け

- P314

0x0119 株式会社日新システムズ

以上