お客様各位

カタログ等資料中の旧社名の扱いについて

拝啓　時下ますますご清栄のこととお喜び申し上げます。平素は格別のご高配を賜り厚く御礼申し上げます。

さて、２０１７年５月１日を以ってルネサス セミコンダクタ パッケージ&テスト ソリューションズ株式会社の半導体製造装置をはじめとする各種産業用制御ボードの受託開発・製造および画像認識システム開発・製造・販売事業を日立マクセル株式会社へ譲渡したことにより、当該事業は日立マクセル株式会社の子会社として新設されるマクセルシステムテック株式会社に承継されております。

従いまして、ドキュメント等資料中には、旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い申し上げます。

敬具

２０１７年５月１日

マクセルシステムテック株式会社

【発行】マクセルシステムテック（http://www.systemtech.maxell.co.jp/）

【お問い合わせ先】denki-support@maxell.co.jp

maxell

マクセルシステムテック株式会社

Smalight® OS V3 アプリケーションノート
スタック解析ツール「Call Walker」
Smalight OSライブラリデータの利用方法(R8C/35A)

本ドキュメントでは、ルネサスマイコン向けコンパイラパッケージに同梱されているスタック情報解析ツール「Call Walker」を利用した Smalight OS 導入システムのスタック算出方法について説明します。

目 次

1. 前提条件

1.1 適用範囲

- Smalight OS V3.00 以降(R8C/35A)
- スタック情報解析ツール「Call Walker」 V1.05 以降
- 対応コンパイラ
  M3T-NC30WA Ver5.43 Release 00 以降

Smalight OS **開発環境は** M3T-NC30WA Ver5.42 Release 00 **以降(上位互換)となっております。**
Call Waler **をご利用される場合、コンパイラのアップデートが必要になります。**

1.2 Smalight OS ライブラリデータ

本ドキュメントと一緒に同梱される Smalight OS ライブラリデータファイル一覧です。

表1 Smalight OSライブラリデータファイル一覧

CPU コア	モード	割込制御モード	ライブラリデータファイル名
R8C/35A	-	-	CW_SmalightOS_r8c1_***vrr*** _v***xxx***.csv

vrr : Smalight OS バージョン
xxx : ライブラリデータファイルのバージョン

2. Call Walker 使用方法

Call Walker はリンケージエディタが出力したスタック情報ファイル(*.sni)、シミュレータデバッガが出力したプロファイル情報ファイル(*.pro)を入力に静的なスタックサイズを算出できる便利なツールです。以下に、スタック情報ファイル(*.sni)の作成方法を示します。

プロファイル情報ファイル(*.pro)の作成方法などの詳細な使い方については、Call Walker のヘルプ、または、ルネサステクノロジの Web サイトから提供されるアプリケーションノートをご参照ください。

2.1 スタック情報ファイル作成方法

スタック情報ファイルは、使用するコンパイラの最適化リンカのオプション設定により出力できる様になります。

Smalight OS 提供のワークスペースを開き、 [ビルド]-[Renesas M16C Standard Toolchain …]メニューを選択し、[Renesas M16C Standard Toolchain]ダイアログを表示します。

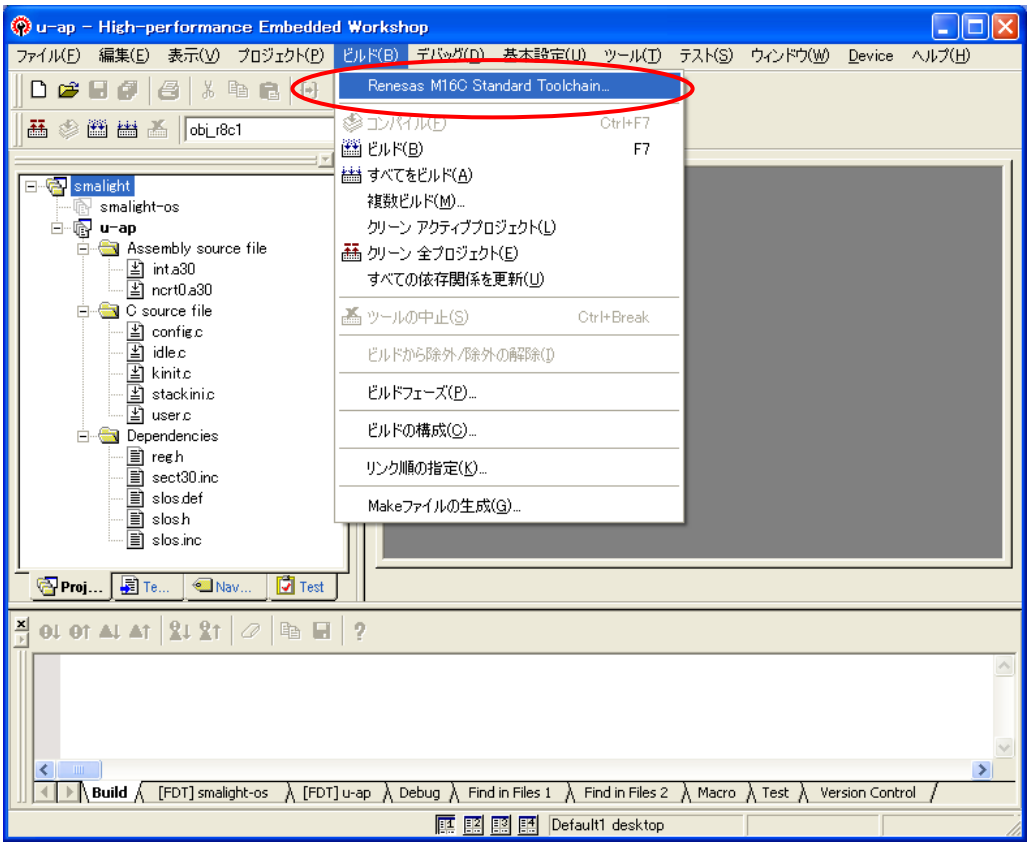

図1 最適化リンカのオプション設定①

[Renesas M16C Standard Toolchain]ダイアログの[コンパイラ]タグを選択し、[オブジェクト]カテゴリを選択します。

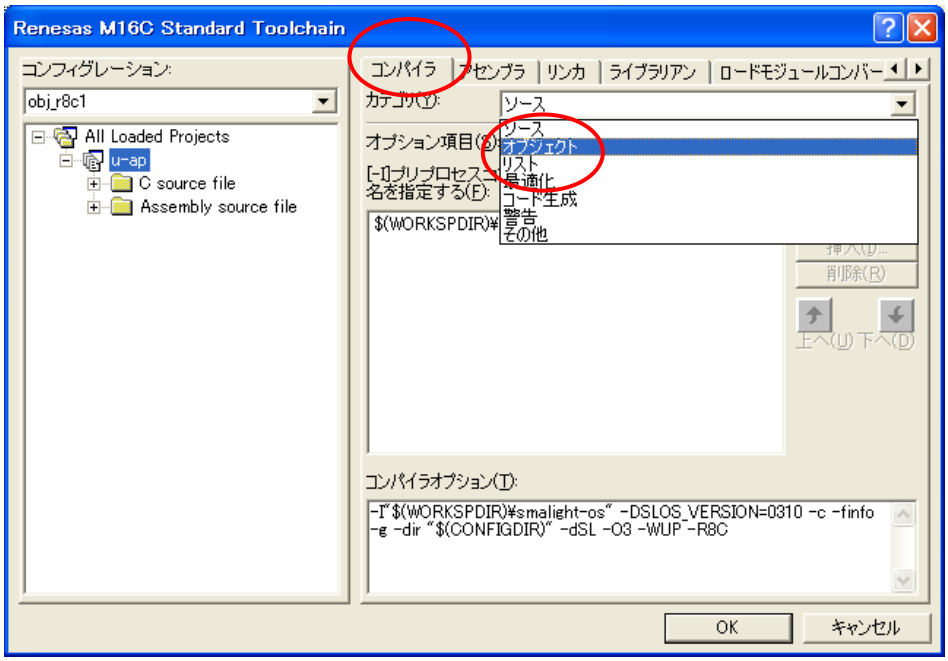

図2 最適化リンカのオプション設定②

[デバックオプション]の[-finfo]チェックボックスをチェックします。本設定を有効にしてから、ビルドするとスタック情報ファイルが出力されます。出力先ディレクトリはロードモジュール(*.x30 等)が出力されるコンフィギュレーションディレクトリです。

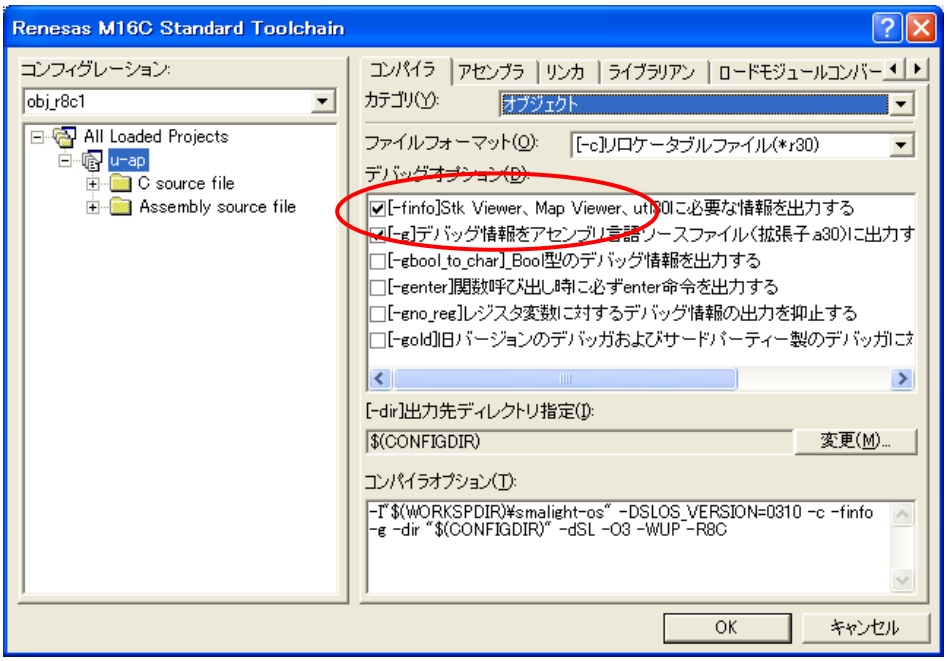

図3 最適化リンカのオプション設定③

2.2 Call Walker の起動

■ スタートメニューからの起動

Windows のスタートメニューから[プログラム]-[Renesas]-[High-performance Embedded Workshop]-[Call Walker]を選択する。

■ High-performance Embedded Workshop のメニューから起動

High-performance Embedded Workshop の[ツール]-[Call Walker]を選択する。High-performance Embedded Workshop のバージョンにより、本メニューが存在しない場合があります。その場合は、スタートメニューから起動してください。

2.3 Call Walker でスタック情報ファイルを読み込む

Call Walker を起動し[File]-[Import Stack File]メニューを選択します。[Stack File]ダイアログが表示されますので、作成したスタック情報ファイルを選択します。

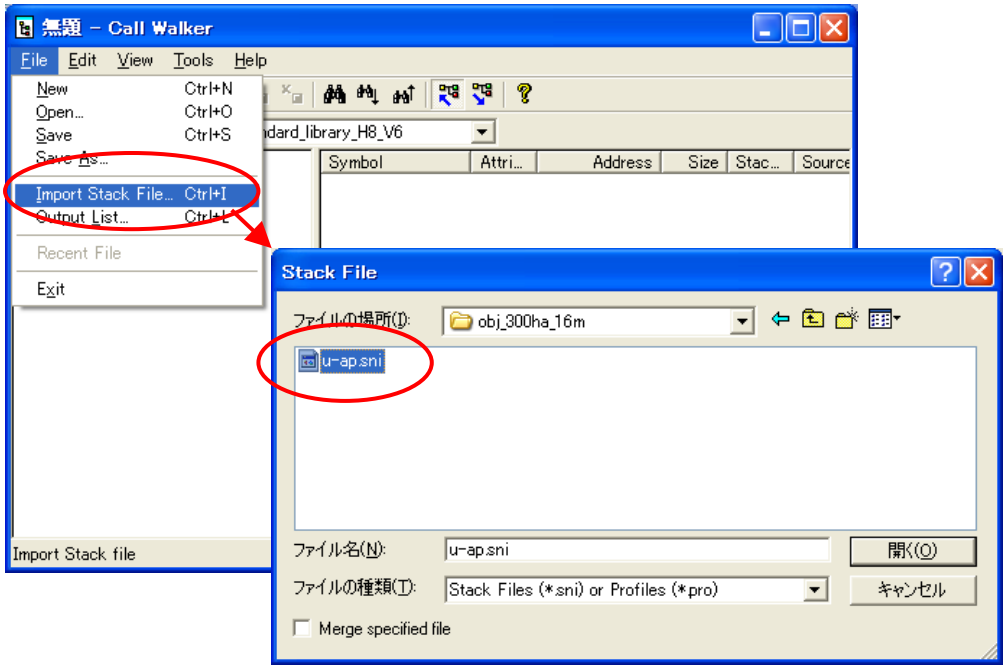

図4 Call Walkerスタック情報ファイルの読み込み

スタック情報ファイルを読み込むと、各関数のスタック情報が表示されます。

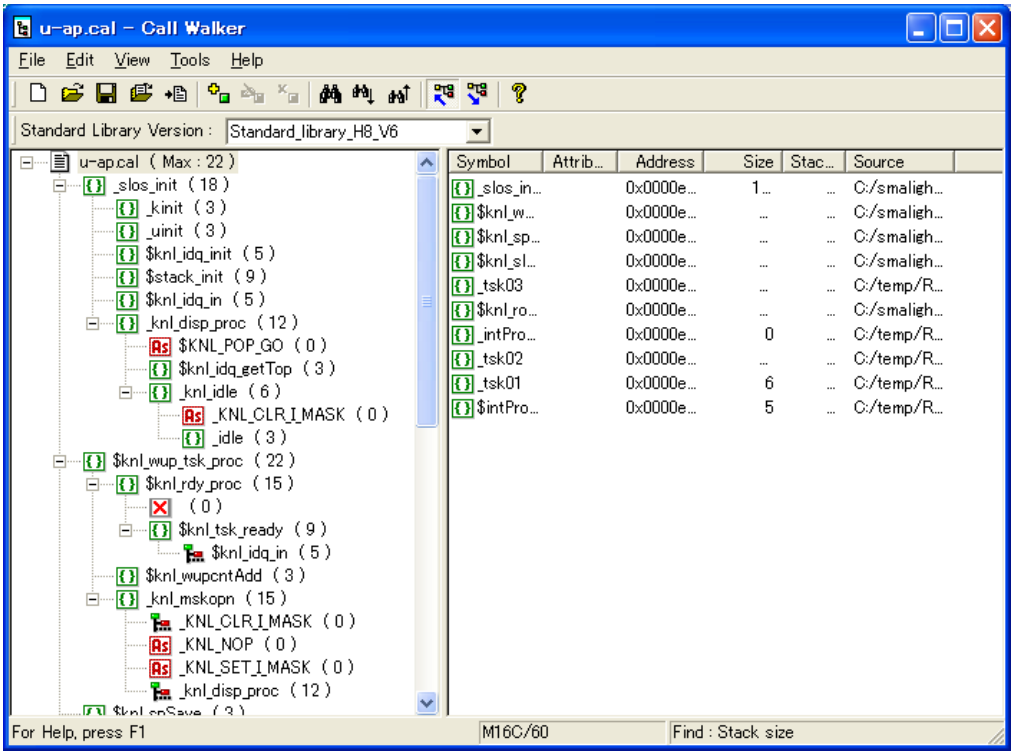

図5 Call Walkerスタック情報表示

2.4 Call Walker のスタック表示について

Call Walker で表示されるスタック情報について説明します。

 C 言語で記述された関数

C 言語で記述された関数は､関数の呼び出しによるスタック消費を考慮した値が表示されます。

 アセンブラで記述された関数

アセンブラで記述された関数のスタックは自動的に算出できないため、スタック情報は 0 と表示されます。

但し、標準ライブラリで準備されたアセンブラ記述の関数については、データベース化されたスタック測定結果が自動的に取り込まれて正しく表示されます。

 省略表示シンボル

Call Walker では、全ての関数の呼び出しを含む情報を表示する為、複数箇所で呼び出される関数を表示すると膨大な表示量となります。既に表示済みの関数については省略シンボルで関数ネスト表示を簡略化されます。

 アドレス参照未解決関数

(*func)() 等、関数アドレスを格納した変数を利用した関数呼び出しがある場合、この表示となります。リンク時点でアドレス解決しない関数としてスタック情報は 0 と表示されます。

その他の表記については、Call Walker のヘルプ、または、ルネサステクノロジの Web サイトから提供されるアプリケーションノートをご参照ください。

2.5 Smalight OS ライブラリデータ

Call Walker でスタック情報ファイルを読み込んだ状態では、Smalight OS のサービスコールは、スタックサイズ 0 で表示されます(アセンブラで記述された関数として認識されるため)。そのため、タスク関数などのスタック情報が正しく表示されていない状態となります。

また、通常、ユーザが意識する必要のない Smalight OS の内部関数なども合わせて表示されるため、Call Walker 表示が見づらくなります。

そこで、Smalight OS ライブラリデータを利用することで、Smalight OS サービスコールのスタックサイズ情報を外部から取り込み、正しくタスク関数のスタックサイズを表示することができます。

2.6 Smalight OS ライブラリデータの取り込み

Call Walker の[Tools]-[Realtime OS Option …]メニューを選択すると、[Realtime OS Option]ダイアログが表示されます。[Realtime OS Option]ダイアログにて、[Display symbols for the realtime OS]チェックボタンを有効にします。[Browse]ボタンをクリックして、ファイル選択ダイアログから、Smalight OS ライブラリデータを選択します。使用する CPU により選択ファイルが異なるなる為、注意ください。

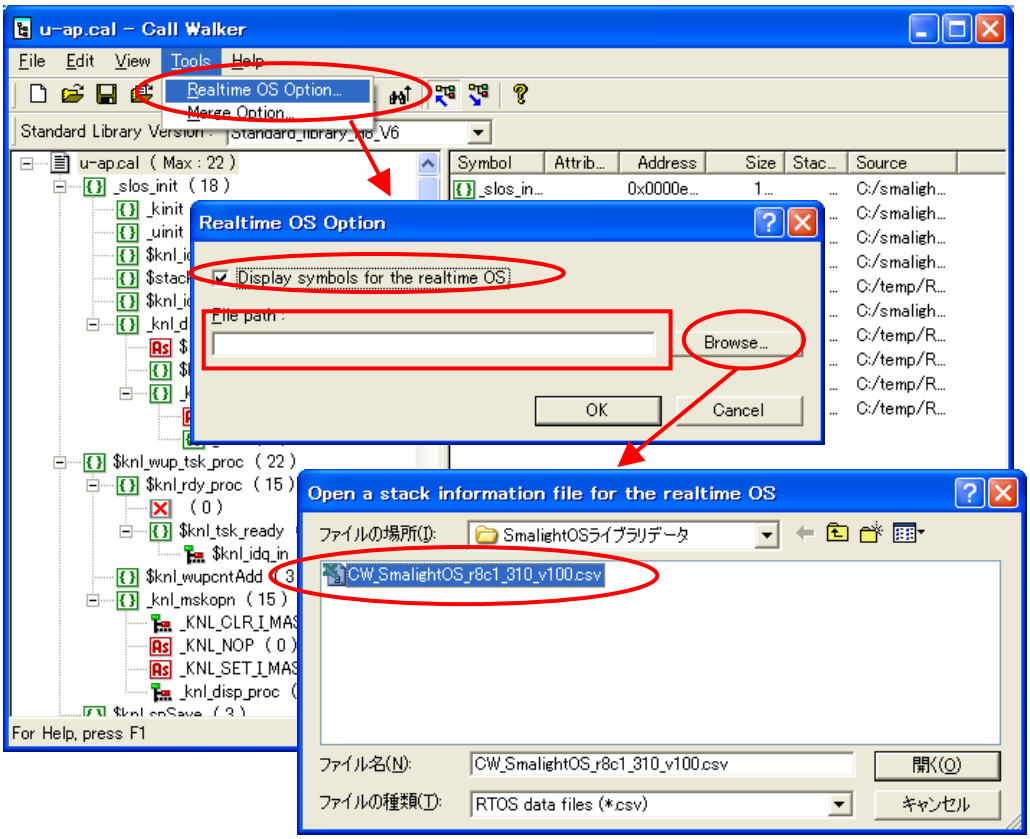

図6 Call Walker RTOSライブラリデータの取り込み

Smalight OS ライブラリデータを取り込んだ Call Walker は以下の様な表示になります。

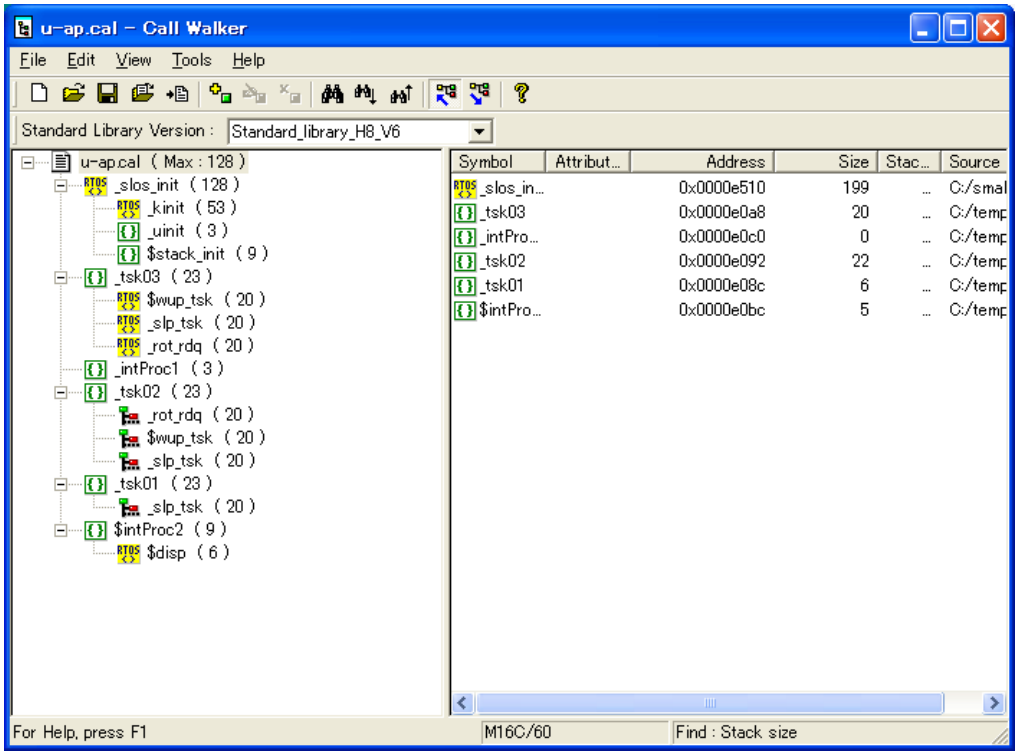

図7 Call Walker RTOSライブラリデータの取り込み後のスタック情報表示

Smalight OS ライブラリデータを取り込むことで表示されます。リアルタイム OS サービスコールのスタックサイズが表示されます。

3. Smalight OS 導入システムのスタック算出

Smalight OS には、3 種類のスタックがあります。

- タスクスタック … タスク毎に1つスタックが必要となります。
- 割込みスタック … 割込みレベル毎に1つのスタックが必要となります。
- OS スタック … OS 動作時に使用されるスタックでシステムに 1 つ必要となります。

3.1 タスクスタック

タスクスタックは、タスク毎に 1 つのスタックが必要になります。各タスクのスタックサイズを確認し、OS コンフィギュレーション情報として設定が必要になります。

2 章の手順により、Call Walker ではタスク関数のスタックサイズが算出されました。

タスクスタックとして考える場合、タスク関数のスタックサイズではスタック不足となります。割込みで(割込み発生元の)タスクスタックが使用されるからです。割込みを一切使用していないシステム、または、タスク実行中に一切、割込みを受け付けない場合を除き、タスクスタックは割込みで消費されるスタックサイズを加算する必要があります。

Smalight OS 環境下での割込み発生元で準備すべきスタックサイズを以下に示します。この値は使用する CPU により異なります。

表2 割込み発生元で使用されるdisp有割込みのスタックサイズ

CPU コア	モード	割込み用のスタックサイズ	備考
R8C/35A	-	20 **バイト**	

例）R8C/35A の場合

① Call Walker で求めたタスク関数 tsk02 のスタックサイズ … 36 バイト
② タスク側で用意する割込み用スタックサイズ …. 20 バイト(R8C/35A)

準備するタスクスタック ＝ ① 36 バイト ＋ ② 20 バイト ＝ 56 バイト

3.2 割込みスタック

Smalight OS を導入したシステムでは以下の 2 種類の割込みが存在します。

- disp 無割込み … カーネルマスクレベルを超える割込レベルで動作する割込み
  OS 管理下でない為、OS サービスコール発行は禁止されます。
- disp 有割込み … カーネルマスクレベル以下の割込レベルで動作する割込み
  OS 管理で動作し、通常、割込みを抜けるとき、割込み発生元へ
  戻らずに OS に制御を移します。

Smalight OS の割込みに関する詳細は、Smalight OS リファレンスマニュアルを参照ください。

割込みスタックは、通常、割込みレベル毎に１つのスタックを準備することで正しく動作できます。１つの割込みレベルに複数の割込みが存在する場合、それらの割込みの中で、一番スタックサイズの大きい割込みのスタックサイズを設定してください。

3.2.1 disp 無割込み

2 章の手順により、Call Walker では割込み関数のスタックサイズが算出されました。

(1) 多重割込みを行っていない場合、または、多重割込みで最上位レベルの割込みの場合
Call Walker の算出値をそのまま設定してください。

(2) 多重割込みで最上位レベルでない場合
Call Walker の算出値に、上位割込みが割込み発生元で消費するスタックサイズを加算した値を設定してください。

上位レベルの割込みで使用されるスタックサイズは、使用する CPU のデータシートに記載される割込み動作を確認してください。

3.2.2 disp 有割込み

2 章の手順により、Call Walker では割込み関数のスタックサイズが算出されました。

まず、disp 割込み関数のスタックサイズに disp 有割込み受付部で消費されるスタックサイズを加算します。disp 有割込み受付部で消費されるスタックサイズを以下に示します。このスタックサイズは使用する CPU により異なります。

表3 disp有割込み受付部で消費されるスタックサイズ

CPU コア	モード	disp 有割込み受付部スタックサイズ	備考
R8C/35A	-	4 **バイト**	

(1) 多重割込みを行っていない場合、または、多重割込みで最上位レベルの割込みの場合
上記の算出値をそのまま設定してください。

例）R8C/35A の場合

① Call Walker で求めたタスク関数 intProc2 のスタックサイズ … 12 バイト
② disp 有割込み受付部スタックサイズ …. 4 バイト(R8C/35A)

準備する disp 有割込みスタック ＝ ① 12 バイト ＋ ② 4 バイト ＝ 16 バイト

(2) 多重割込みで最上位レベルでない場合
上記の算出値に、上位割込みが割込み発生元で消費するスタックサイズを加算した値を設定してください。

例）R8C/35A の場合

① Call Walker で求めたタスク関数 intProc2 のスタックサイズ … 12 バイト
② disp 有割込み受付部スタックサイズ …. 4 バイト(R8C/35A)
③ 上位レベルの割込みスタックサイズ …. 32 バイト(R8C/35A)

準備する disp 有割込みスタック
＝ ① 12 バイト ＋ ② 4 バイト ＋ ③ 32 バイト ＝ 48 バイト

上位レベル割込みが disp 有割込みの場合、割込み発生元で消費するスタックサイズは、「表 2 割込み発生元で使用される disp 有割込みのスタックサイズ」を参照ください。disp 無割込みの場合、使用する CPU のデータシートに記載される割込み動作を確認してください。

3.2.3 周期ハンドラ使用時の注意事項

周期ハンドラ(callback_cychrr：名称はユーザ任意)は、ユーザ任意で実装する周期タイマハンドラから呼び出されます。周期ハンドラ関数のスタックサイズは、周期タイマハンドラ割込み関数のスタックサイズとして考慮する必要があります。

周期タイマハンドラ関数(Ex. intTim)を Call Walker で確認すると以下の様な表示となります。

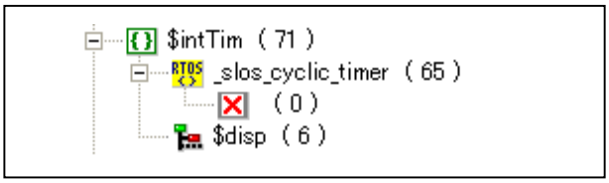

図8 周期ハンドラ使用時の表示

表示の (アドレス参照未解決関数)が、周期ハンドラに該当します。使用する全ての周期ハンドラ関数の中で、最大のスタックサイズをあてはめてください。その上で周期タイマハンドラ(disp 有割込み)のスタックサイズを求めてください。

3.3 OS スタック

OS スタックは、start.src (アセンブラソース)内のシンボル _knl_sp で定義され、通常、問題なく動作できるスタックサイズが設定されております。

下記のコールバックルーチンは、OS スタックを使用して動作します。ユーザ側でこれらのコールバックルーチンを修正することにより、コールバックルーチン単体でのスタックサイズが、デフォルトで定義される OS スタックサイズをオーバーする場合、OS スタックサイズを修正してください。

- kinit() … OS オブジェクトの初期化(Smalight Configurator にて出力されます)
- uinit() … ユーザ初期化処理
- stack_init() … タスクスタックの初期化
- idle() … アイドル状態時の処理

4. 注意事項

・ スタックサイズ不足が原因でプログラムの誤動作が発生した場合、原因の解析が非常に困難となります。本ドキュメントで説明するスタックサイズ算出を参考に、余裕を持ったスタックサイズを設定する様にお願いします。

・本ドキュメントに記載されるスタックサイズ算出方法は、静的なスタックサイズ算出で、実際に消費されるスタックサイズ(動的なスタックサイズ)は、プログラムの動作条件(動作パス)などにより異なる(少なくなる)場合があります。

・Smalight OS ライブラリの作成は、M3T-NC30WA Ver5.42 Release 00 で行われております。Call Walker にて、一部の OS 内部関数が不正に表示される場合がございます。

ホームページとサポート窓口

(株)ルネサス北日本セミコンダクタ ホームページ
http://www.kitasemi.renesas.com/

Smalight ホームページ
http://www.kitasemi.renesas.com/product/smalight/

お問合せ先
soft.support@kitasemi.renesas.com

改定記録

Rev	発行日	改定内容	
		ページ	ポイント
1.00	2009.4.1	―	**初版発行**

安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご留意ください。

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切な弊社製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、弊社は、予告なしに、本資料に記載した製品または仕様を変更することがあります。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したものですが万一本資料の記述誤りに起因する損害がお客様に生じた場合には、弊社はその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。弊社は、適用可否に対する責任は負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、弊社へご照会ください。
7. 本資料の転載、複製については、文書による弊社の事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気付きの点がございましたら弊社までご照会ください。