

コンパイラパッケージ

Call Walker を使ったスタック見積もり方法（CS+編）

要旨

CS+のコンパイラパッケージには、スタック算出ユーティリティの Call Walker が付属しています。Call Walker を使用すると、ビルド時に集計されたスタック情報を解析して、各関数ツリーで消費されるスタックサイズを確認することができます。本ドキュメントでは Call Walker の使い方について説明します。

対象ソフトウェア： Call Walker V.2.05 以降（CS+同梱）

概要

Call Walker は、最適化リンケージエディタが出力したスタック使用量情報ファイル (*.sni)、または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) を読み込んで関数の呼び出し関係をつリー形式で表示し、関数シンボル単位のスタック情報（シンボル名、属性、アドレス、サイズ、静的なスタックサイズ、ファイル名）をリスト形式で表示します。

また、動的なスタックサイズを見積もるためにユーザによるスタック使用量の編集が可能で、編集した情報は呼び出し情報ファイル (*.cal) として保存して、読み込むことができます。

目次

1. Call Walker の概要	3
1.1 ファイル構成.....	3
1.2 スタックについて.....	3
1.3 Call Walker によるスタック使用量の表示.....	4
1.4 Call Walker 使用時の注意点	4
2. スタック情報ファイルの作成方法.....	5
2.1 スタック使用量情報ファイル (*.sni)	5
2.2 プロファイル情報ファイル (*.pro)	5
3. Call Walker の起動	6
4. メインウィンドウ.....	8
4.1 メインウィンドウの構成.....	8
4.2 呼び出し情報ビュー	9
4.3 シンボル詳細ビュー	12
4.4 Standard Library Version (標準ライブラリバージョン選択)	12
5. メニューバーのコマンド説明.....	13
5.1 File メニュー	13
5.2 Edit メニュー.....	16
5.3 View メニュー	17
5.4 Tools メニュー	19
5.5 Help メニュー.....	20
6. アセンブラプログラムのスタック使用量表示.....	21
6.1 「.STACK」アセンブラ拡張機能制御命令.....	21
6.2 アセンブラプログラム記述例.....	21
7. スタック情報の編集	22
7.1 Add コマンドによるシンボルの追加	22
7.2 Modify コマンドによるシンボル情報の変更.....	26
8. スタック情報のマージ.....	29
8.1 マージの手順 (例)	29
8.2 マージタイプの選択.....	30
9. ユーザライブラリファイルによるスタックサイズ表示変更	32
付録 1. インフォメーションメッセージ.....	35
付録 2. エラーメッセージ.....	36
改訂記録.....	39

1. Call Walker の概要

1.1 ファイル構成

Call Walker は、最適化リンケージエディタが出力したスタック使用量情報ファイル (*.sni) を読み込んで、静的なスタック使用量をシンボル別に表示します。

以下に、本ツールと使用される各ファイルの関係を示します。

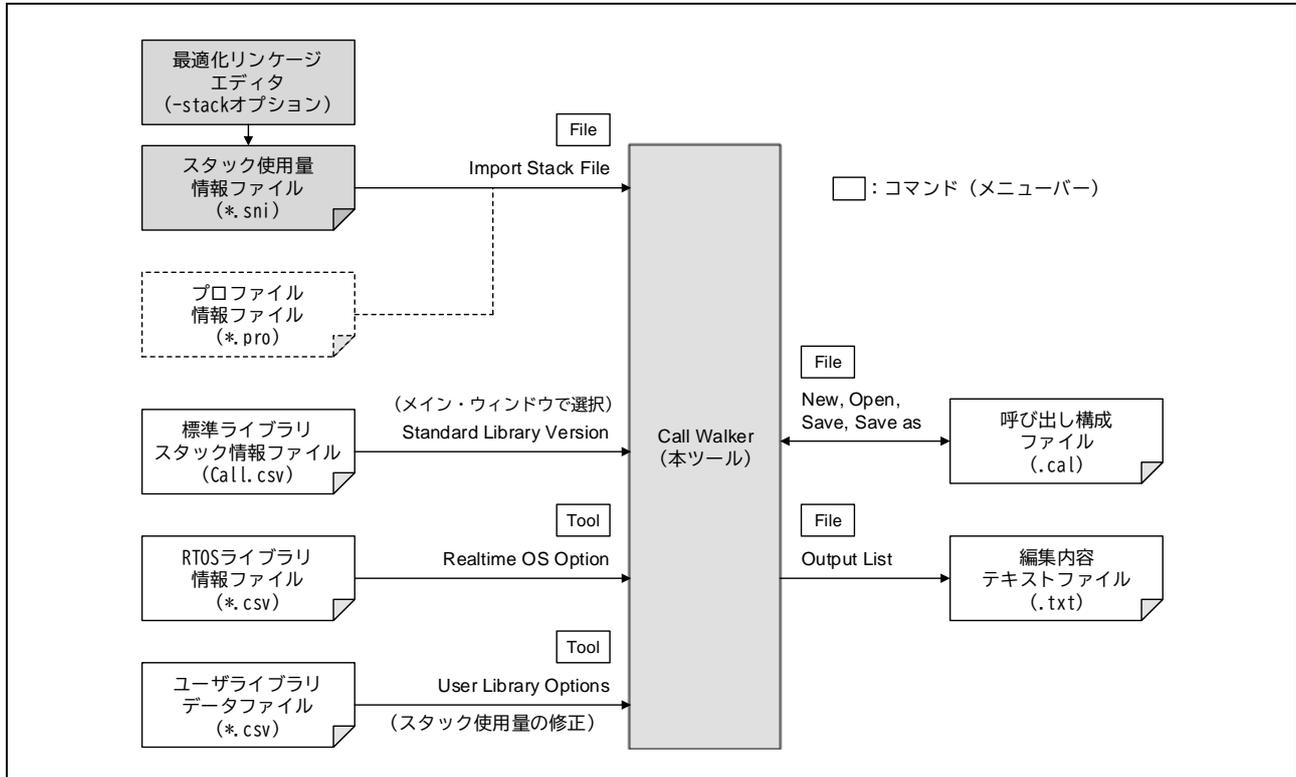


図 1-1. Call Walker と関連ファイル

1.2 スタックについて

一般的に、メモリ上のスタックは、下記の作業領域として使用されます。

- 関数のリターン PC (戻り先アドレス)
- 関数の引数
- 関数の戻り値
- ローカル変数
- テンポラリ変数 (コンパイラが処理のため一時的に使用する変数)
- 例外処理コンテキスト

ただし、実際のスタックの扱いは、CPU のアーキテクチャとコンパイラの仕様に依存します (たとえば、関数呼び出し時の引数や戻り値のデータ格納場所、最適化の設定とスタック使用の関係など)。

スタックの扱いについては、使用するコンパイラ、リアルタイム OS のユーザーズマニュアルを参照ください。

1.3 Call Walker によるスタック使用量の表示

最適化リンケージエディタの出力を基に算出された静的なスタック使用量の最大値は、Call Walker の呼び出し情報ビュー (左側ウィンドウ) の最上部の (Max:) に表示された値となります。ただし、制約事項がありますので、「1.4 Call Walker 使用時の注意点」を必ずお読みください。

スタック使用量の最大値は、関数ツリー (第 1 階層のシンボル) のスタック使用量の最大値を示します。

第 2 階層以下のスタック使用量は、View メニューの「Show Required Stack / Show Used Stack」の設定により表示が異なります (「図 5-3. Show Required Stack / Show Used Stack の表示」を参照してください)。

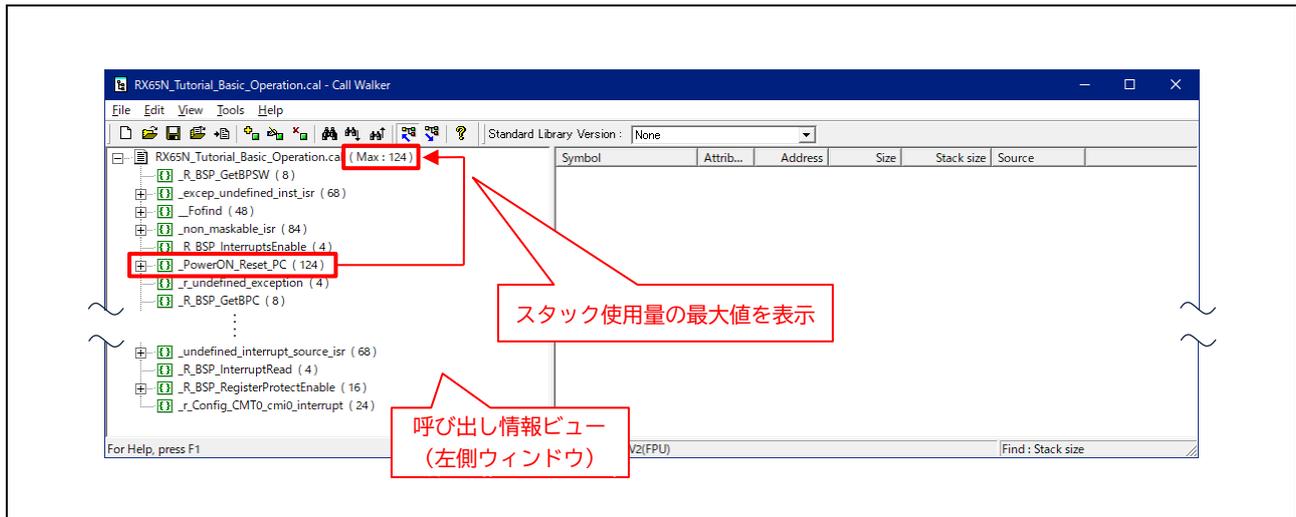


図 1-2. Call Walker によるスタック使用量の最大値表示

1.4 Call Walker 使用時の注意点

本ツールの使用にあたっては、次の点にご注意ください。

- 同じプログラムでも、ビルド設定 (最適化オプション等) でスタック使用量は変化します。
- 再帰関数や多重割り込みなど、動的なスタック使用量は表示できません。
- 関数ポインタ経由で呼び出された場合のスタック使用量は計算できません。
- 最適化リンケージエディタが出力するスタック使用量情報ファイル (*.sni) で、インラインアセンブラの記述内でスタックは消費されないものとして扱われます。

Call Walker でスタック使用量を計測できない場合は、手計算によりスタック使用量を求めるか、スタック使用量が最大となる経路になるよう関数シンボルを編集して、最大スタック使用量を計算してください。

2. スタック情報ファイルの作成方法

2.1 スタック使用量情報ファイル (*.sni)

CS+のプロジェクト・ツリー・パネルから「ビルドツール」の「プロパティ」を開きます。[リンク・オプション] タブを選択し、[その他] カテゴリの [スタック使用量情報ファイルを出力する] プロパティで「はい」を選択します。

この操作により、最適化リンケージエディタの「-stack」オプションが使用されるようになります。

この設定を行ってからプログラムをビルドすると、スタック使用量情報ファイル (*.sni) が生成されます (デフォルト設定では、プロジェクトフォルダ直下の「DefaultBuild」フォルダ内)。

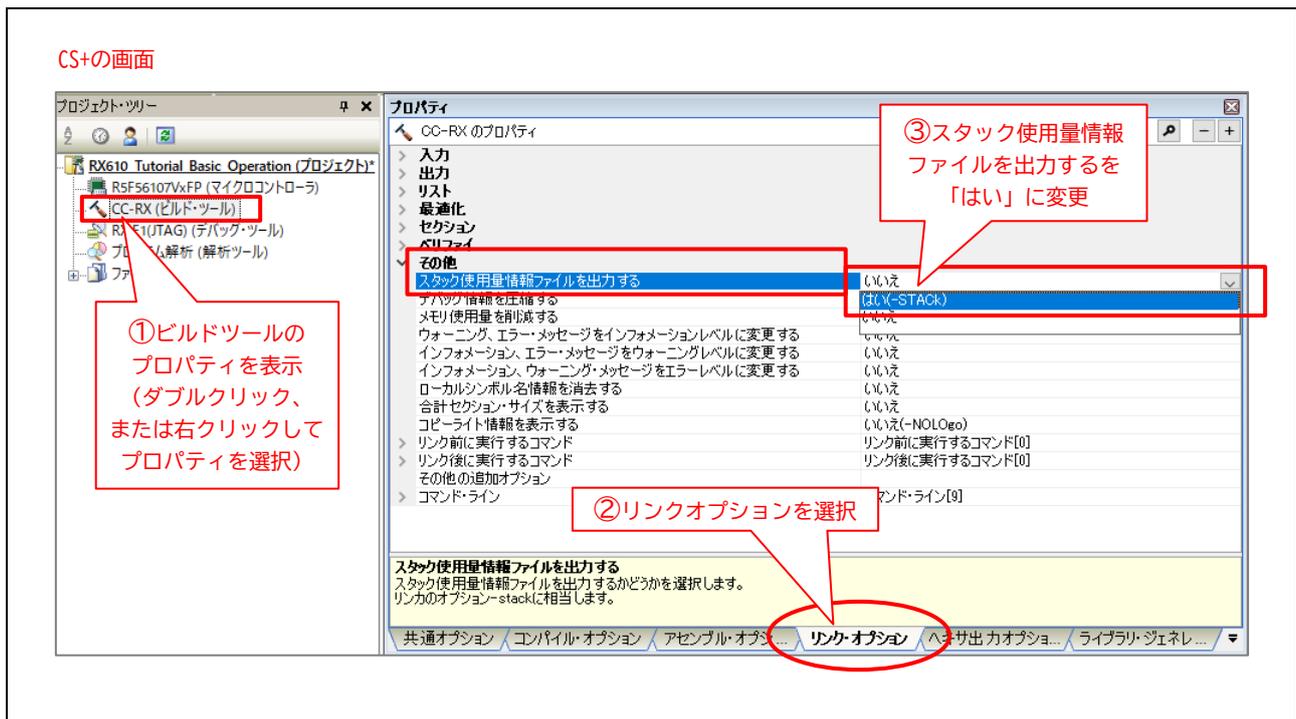


図 2-1. CC-RX のリンクオプションの設定

2.2 プロファイル情報ファイル (*.pro)

Call Walker には、HEW (High-performance Embedded Workshop) がサポートしていたプロファイル情報ファイル (*.pro) をインポートする機能があります。

3. Call Walker の起動

Call Walker は CS+に組み込まれており、ツールバーからの起動が可能です。また、Windows のスタートメニューから Call Walker を単独で起動することもできます（その場合、最適化リンケージエディタ出力情報は手動でインポートします）。

(1) CS+のツールバーからの起動

前章「2. スタック情報ファイルの作成方法」に従ってプログラムをビルドした後、以下の操作で Call Walker を起動します。

CS+のツールバー > [ツール] > [スタック見積もりツールの起動^注] をクリック

注. ツールバーのメニューに無い場合は、CS+の [ツール] > [プラグインの管理] で「追加機能」タブの「スタック見積もりツール」にチェックが入っていることを確認してください。

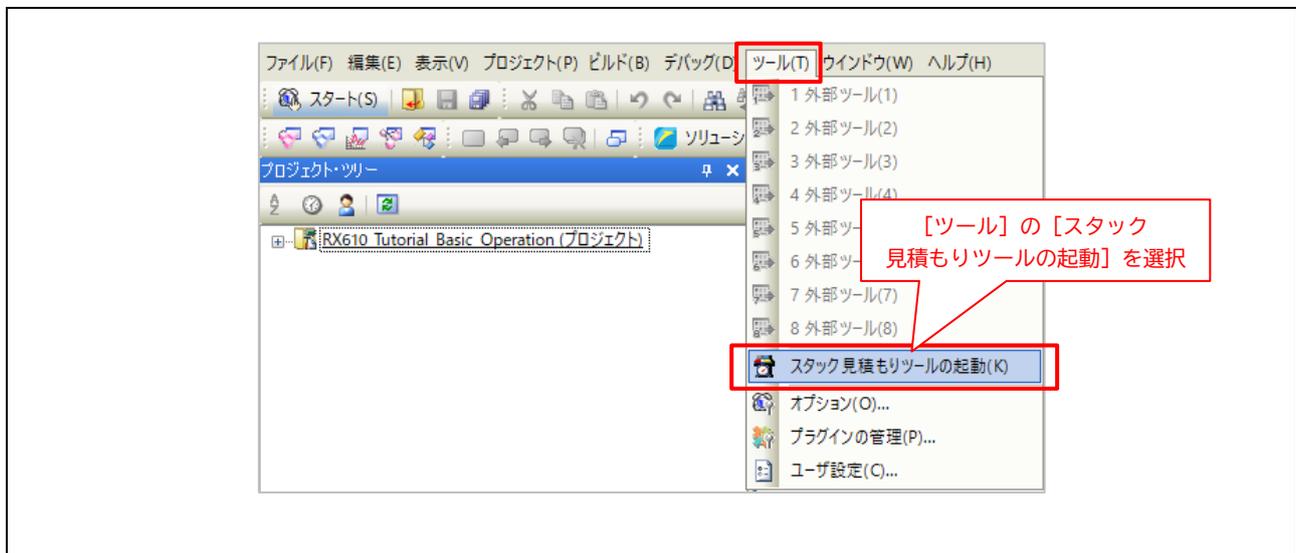


図 3-1. CS+ のツールバーからの Call Walker 起動

(2) Windows のスタートメニューからの起動

Call Walker は、CS+のコンテンツの1つとして単独で起動することもできます。

スタートメニュー > アプリ一覧 > Renesas Electronics CS+ > Call Walker を選択します。

Call Walker が起動したら、スタック使用量情報ファイル (.sni) またはプロファイル情報ファイル (.pro) をインポートしてください。

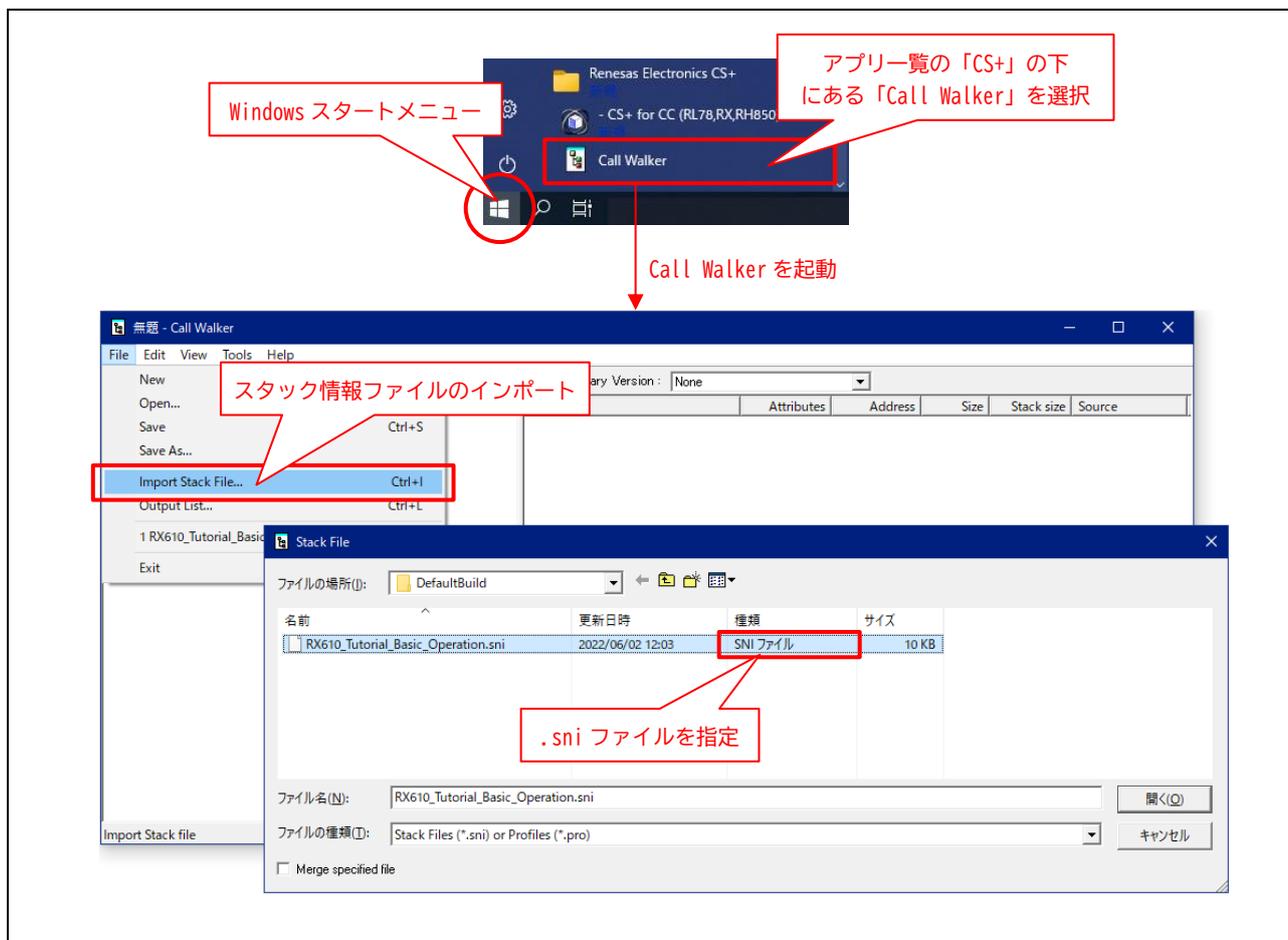


図 3-2. スタートメニューからの Call Walker 起動とスタック情報のインポート

4. メインウィンドウ

4.1 メインウィンドウの構成

Call Walker が起動するとメインウィンドウが開きます。Call Walker は、一度に1つのスタック情報ファイルのみを開くことができます。

メインウィンドウは、以下のような構成になっています。

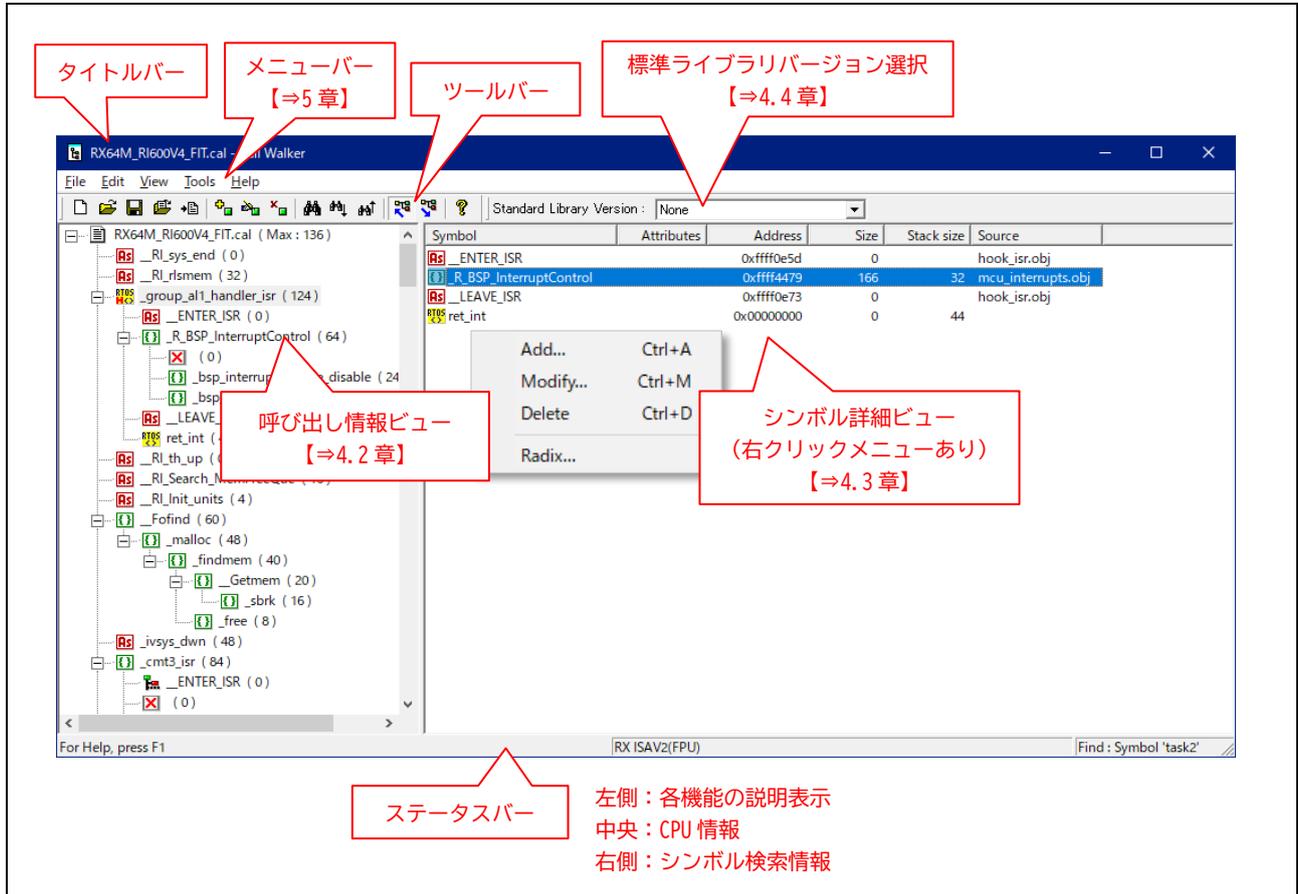


図 4-1. Call Walker のメインウィンドウと各部の名称

表 4-1. シンボル区分記号 (アイコン)

アイコン	説明
	編集中のファイル
	アセンブラのラベル
	C/C++の関数
	再帰呼び出し (リカーシブ) 関数または循環関数
	RTOS 関数
	参照元シンボル不明関数
	アドレス参照未解決関数
	省略表示シンボル
	RTOS ハンドラ関数
	RTOS タスク関数
	可変個引数を持つ関数

4.2 呼び出し情報ビュー

呼び出し情報ビュー（左側ウィンドウ）には、関数シンボル間のリンク階層構造が表示されます。各シンボル名の右側カッコ内に、使用しているスタック使用量が表示されます。

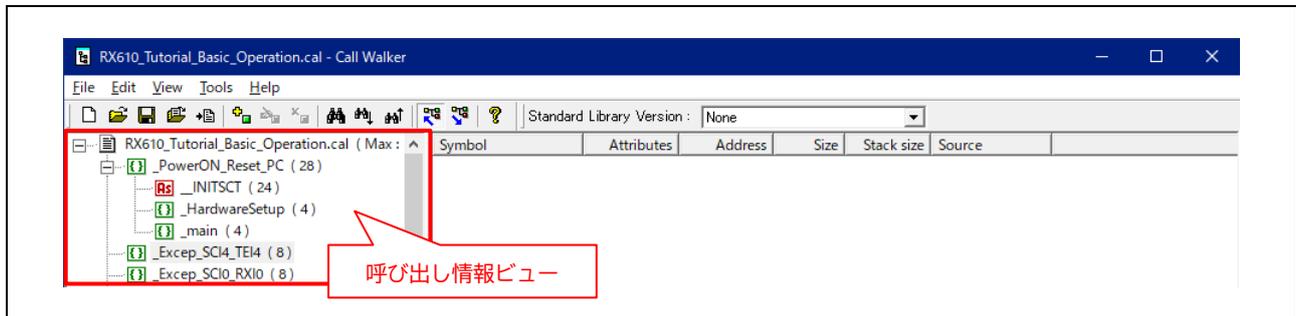


図 4-2. 呼び出し情報ビュー

(1) 各シンボルの説明

シンボルの区分記号（アイコン）とその説明を以下に示します。

	編集中のファイル
	アセンブラのシンボル
	C/C++の関数
	再起呼び出し（リカーシブコール）関数または循環関数
<p>(a) 再起呼び出し（リカーシブコール）関数 関数内で自関数を呼び出す場合に表示されます。</p> <p>【例】</p> <pre>void func(int x) { x++; if(x != OFF) func(x); if(x == MAX) return; } </pre>	
<p>(b) 循環関数 間接的に自関数を呼び出す場合に表示されます。</p> <p>【例】</p> <pre>void func1(int a) { func2(10); } void func2(int b) { func1(9); } </pre>	

	RTOS 関数 (ITRON などのシンボル)
---	-------------------------

	参照元シンボル不明関数
---	-------------

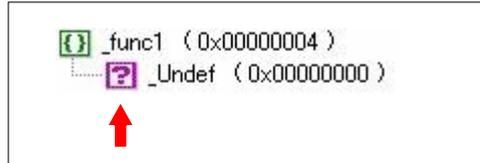
下記例の場合、関数 func1() で関数 Undef() を呼び出していますが、関数 Undef() の実体がない場合、関数 Undef() に本アイコンが表示されます。

実際には実体のない関数呼び出しは、リンク時にエラーになりますが、リンクオプション change_message を使用することにより、エラーをウォーニングに変更できます。

ウォーニングにすると、ロードモジュールが作成できるので、スタック情報ファイルも作成されます。

【例】

```
void func1(void)
{
    Undef();
}
```



	アドレス参照未解決関数
---	-------------

関数をポインタで間接的に呼び出した場合に表示されます。

【例】

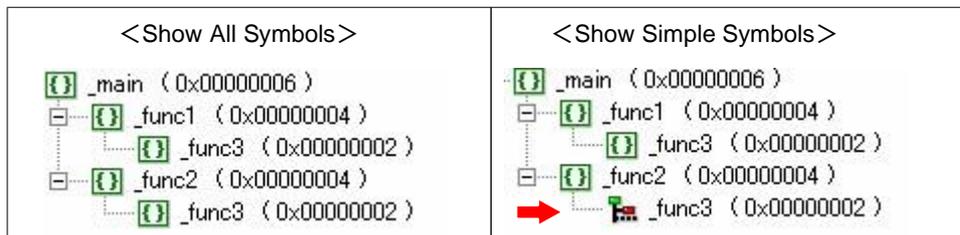
```
void main(void)
{
    void (*ptr)() = subfunc;
    ptr();
    while(1);
}
```



	省略表示シンボル
---	----------

本ツールでは、リンク階層を全て表示するため、サイズが大きい場合、表示量も膨大なものになります。そこで、各シンボルの階層表示について最初の1つだけを表示するようにし、他のシンボルは階層表示を省略化することで表示量を少なくすることができます。

【例】



表示形式は、[View] > [Show All Symbols/Show Simple Symbols] で切り替えることができます。

	RTOS ハンドラ関数 (#pragma almhandler、#pragma cychandler、#pragma interrupt)
	RTOS タスク関数 (#pragma task、#pragma taskexception)
	可変個引数を持つ関数

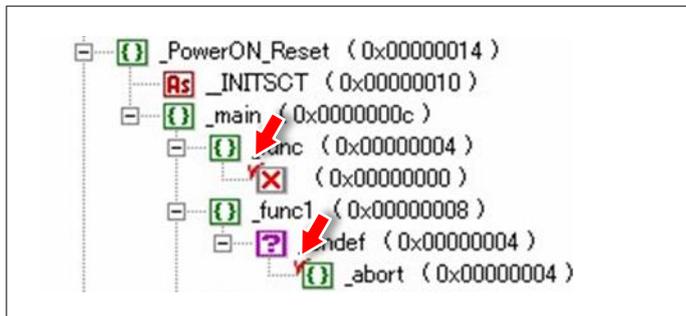
(2) スタックサイズの表示

呼び出し情報ビューに表示されたシンボル名の右側カッコ内の数値はスタック使用量を示します。表示形式は、View メニューの「Show Required Stack / Show Used Stack」で切り替えられます（「5.3 View メニュー」を参照ください）。

折りたたまれて省略表示されているシンボルには、省略された子関数を呼び出した時のスタック使用量の最大値を表示します。

(3) シンボルの位置の移動

呼び出し情報ビューでシンボルをドラッグ&ドロップすると、シンボルの位置を移動できます。シンボルの移動や編集を行うと、左側の呼び出し情報ビューの該当シンボルに変更を示すチェックが付きます（下図の矢印部分）。



4.3 シンボル詳細ビュー

シンボル詳細ビュー（右側ウィンドウ）には、「呼び出し情報ビュー」で選択されているシンボルが参照するシンボル情報の一覧を表示します。



図 4-3. シンボル詳細ビュー

シンボル詳細ビューの見出し部分の各項目をクリックすることで、昇順⇄降順の切り替えができます。各表示項目の内容は以下の通りです。

表 4-2. シンボル詳細ビュー各項目の内容

項目	内容
Symbol	記号内容は、呼び出し情報ビューと同じです。
Attributes	シンボルの属性 R (Runtime Library) : 実行時ルーチン O (Created by Optimization) : 最適化作成関数 I (Interrupt) : 割り込み関数 S (Static) : 静的関数 V (Virtual) : 仮想関数 L (Use Local Stack) : ローカルスタック使用関数
Address	シンボルの配置アドレス
Size	シンボルのサイズ
Stack size	シンボルが使用しているスタックのサイズ (スタックサイズの値が 0xffffffff の場合は、空白表示します。)
Source	シンボルのソースが収められているファイル名称

4.4 Standard Library Version (標準ライブラリバージョン選択)

現在オープンしているスタック情報ファイルを作成した際の、標準ライブラリバージョンを選択します。これにより、標準ライブラリ内アセンブラ関数のスタック使用量を表示します。

- 注 1. RX ファミリー用 CC-RX コンパイラの場合は標準ライブラリバージョンを選択する必要はありません (None のままご使用ください)。
2. CC-RL V1.03.00 以下を使用の場合、Standard_library_RL78_V1 を選択してください。
CC-RL V1.04.00 以上を使用の場合、Standard_library_RL78_V2 を選択してください。

5. メニューバーのコマンド説明

5.1 File メニュー

File	Edit	View	Tools	Help
New				Ctrl+N
Open...				Ctrl+O
Save				Ctrl+S
Save As...				
Import Stack File...				Ctrl+I
Output List...				Ctrl+L
1 RX610_Tutorial_Basic_Operation.cal				
Exit				

表 5-1. File メニューの内容

コマンド	内容	アイコンとショートカット
New	編集情報をクリアし、新規作成します。編集中の内容を保存していない場合は、情報クリア前に確認のダイアログボックスを表示します。新規作成ファイル（無題）の場合は、ファイル名指定保存（Save As コマンド参照）になります。	 Ctrl+N
Open	一覧よりオープンするファイルを選択、あるいはファイル名を入力してファイルをオープンします。 「Merge specified file」チェックボックスにチェックを入れると、現在編集中的の内容と指定したファイルの内容を、Tools メニューの Merge Option コマンドで指定した方法に従ってマージします。 (図 5-1 を参照)	 Ctrl+O
Save	現在編集中的のファイル (*.cal) を上書きで保存します。新規作成したファイル（無題）の時は、ファイル名指定保存となります	 Ctrl+S
Save As	ファイル名 (*.cal) を指定して保存します。	-
Import Stack File	最適化リンケージエディタが出力するスタック使用量情報ファイル (*.sni) を読み込み、編集データを作成します。 「Merge specified file」チェックボックスにチェックを入れると、現在編集中的の内容と指定したファイルの内容を、Tools メニューの Merge Option コマンドで指定した方法に従ってマージします。 (図 5-2 を参照)	 Ctrl+I
Output List	現在編集中的の内容を、テキスト形式 (*.txt) で出力します。	 Ctrl+L
Recent File	File メニューの中に直近で使用した編集ファイル（4 つまで）が一覧で表示されます。次回起動時は、この一覧から選択することによりオープン操作が容易になります。	-
Exit	本ツールを終了します。現在編集中的の内容を保存していない場合、終了前に保存確認のダイアログボックスを表示します。	-

(1) Open コマンドのダイアログパネル

Open を選択すると、以下のダイアログボックスを表示します。

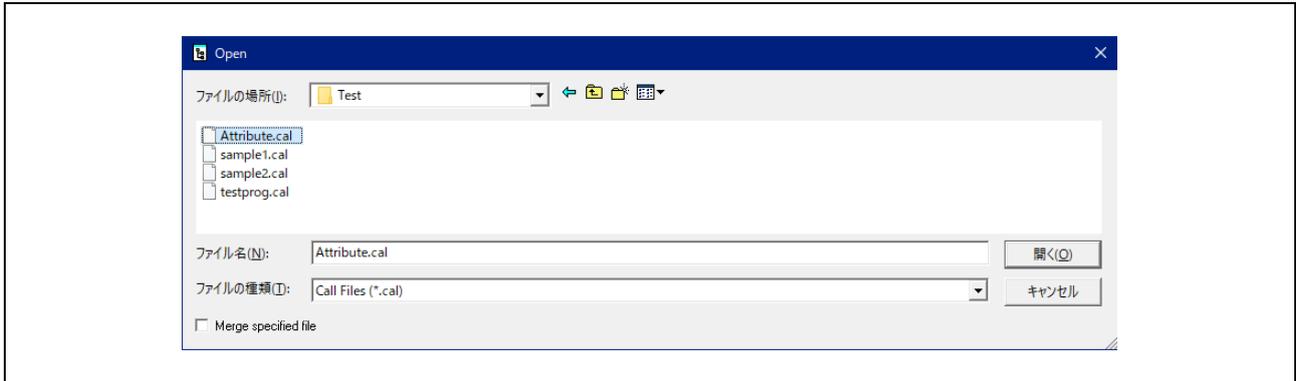


図 5-1. Open コマンドのダイアログパネル

一覧よりオープンするファイルを選択、あるいはファイル名を入力し、「開く」ボタンをクリックするとファイルをオープンします。

その際、「Merge specified file」チェックボックスにチェックを入れると、現在編集中的の内容と指定したファイルの内容を、Merge Option コマンドで指定した方法に従ってマージします（マージ編集方法の指定については、Tools メニューの Merge Option コマンドを参照してください）。チェックしない場合は、現在編集中的の情報を破棄しファイルをオープンします。

ファイルオープン時、現在編集中的の内容を保存していない場合は、ファイルオープン前に保存確認のダイアログボックスを表示します。「はい」ボタンをクリックすると、現在編集中的の内容をファイルに保存します。「いいえ」ボタンをクリックすると現在編集中的の内容を破棄します。「キャンセル」ボタンをクリックすると、ファイルオープンを中止します。

(2) Import Stack File のダイアログパネル

Import Stack File を選択すると、以下のダイアログボックスを表示します。

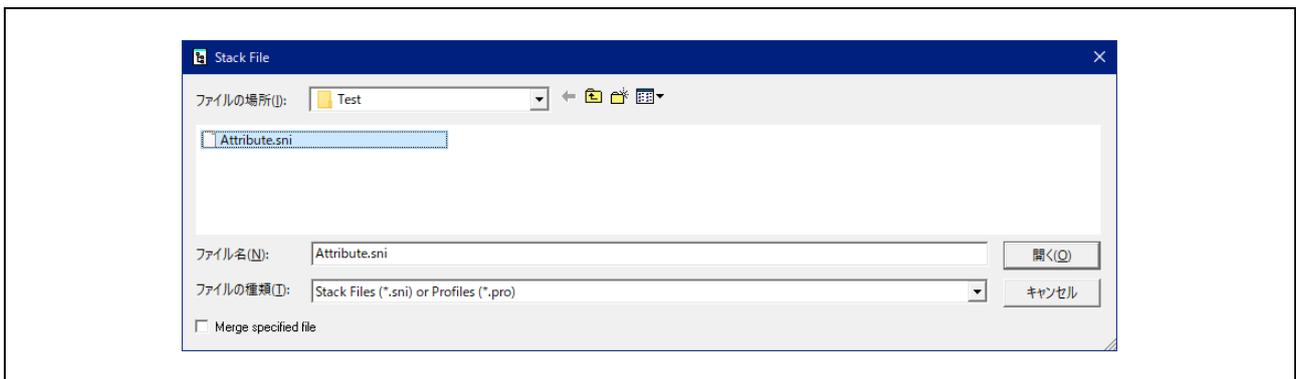


図 5-2. Import Stack File のダイアログパネル

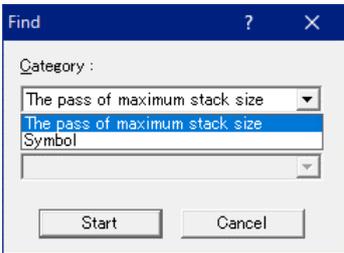
ここで、ファイル一覧よりファイル名を選択、または直接ファイル名を入力し、「開く」ボタンを押すとファイルをオープンし、データを編集し画面表示します。

その際、「Merge specified file」チェックボックスにチェックを入れると、現在編集中の内容と指定したファイルの内容を、Merge Option コマンドで指定した方法に従ってマージします（マージの編集方法指定については、Tools メニューの Merge Option コマンドを参照してください）。チェックしない場合は、現在編集中の情報を破棄しファイルをオープンします。

5.2 Edit メニュー

Edit	View	Tools	Help
Add...			Ctrl+A
Modify...			Ctrl+M
Delete			Ctrl+D
Find...			Ctrl+F
Find Next			F3
Find Previous			Shift+F3

表 5-2. Edit メニューの内容

コマンド	内容	アイコンとショートカット
Add	現在選択中のシンボル直下の階層に、ダイアログパネルから新しくシンボル情報を追加します。 (「7. スタック情報の編集」を参照してください)	 Ctrl + A
Modify	現在選択中のシンボル情報をダイアログパネルで変更します。ただし、RTOS シンボル選択時は Modify コマンドを実行することはできません。 (「7. スタック情報の編集」を参照してください)	 Ctrl + M
Delete	現在選択中のシンボル情報およびその下位にあるシンボルを削除します。 (消去前の確認メッセージはありません。)	 Ctrl + D
Find	呼び出し情報からシンボル情報を後方検索します。 <ul style="list-style-type: none"> The pass of maximum stack size : スタック使用量が最大のパスを検索 Symbol : シンボル名で検索 	 Ctrl + F
Find Next	先に Find コマンドで検索した検索情報 (ステータスバー「Find:」に表示している内容) で後方検索します。	 F3
Find Previous	先に Find コマンドで検索した検索情報 (ステータスバー「Find:」に表示している内容) で前方検索します。	 Shift + F3

5.3 View メニュー

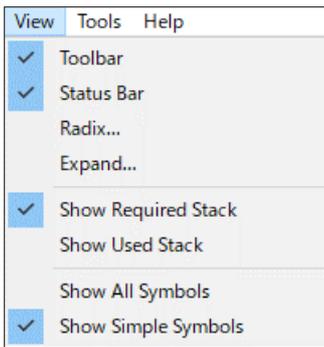
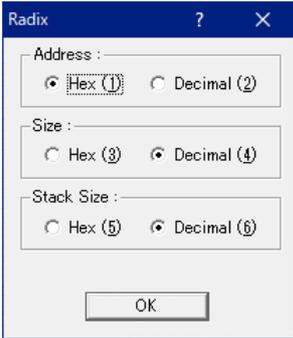


表 5-3. View メニューの内容

コマンド	内容	アイコンとショートカット
Toolbar	ツールバーの表示、非表示を切り替えます。ツールバーを表示しているときは、このコマンド名の横にチェックマークを表示します。	-
Status Bar	ステータスバーの表示、非表示を切り替えます。ステータスバーを表示しているときは、このコマンド名の横にチェックマークを表示します。	-
Radix	シンボル情報内の数値部分 (Address、Size、Stack size) の基数表記をダイアログボックスで切り替えます。「呼び出し情報ビュー」のカッコ内数値 (使用スタックサイズ合計) の表示は Stack size の選択と連動します。 	-
Expand	選択したシンボルの呼び出しシンボルのリストを展開します。ダイアログボックスで展開レベルを指定、または Expand All: ですべて展開します。 	-
Show Required Stack / Show Used Stack	呼び出し情報ビューのスタックサイズ表示形式を設定します。 <ul style="list-style-type: none"> Show Required Stack (必要スタック表示形式) : 下から上へスタック使用量が積み重なっていきます。 Show Used Stack (使用済みスタック表示形式) : 上から下へスタック使用量が積み重なっていきます。 (図 5-3 を参照)	 Show Required Stack  Show Used Stack
Show All Symbols / Show Simple Symbols	呼び出し情報ビューのシンボルの省略表示形式を設定します。 <ul style="list-style-type: none"> Show All Symbols (全表示) : 全てのシンボルを省略せずに表示します。 Show Simple Symbols (省略表示) : 各シンボルについて、呼び出し情報ビューの最初に表示されているもののみ全呼び出し構造を表示し、他のシンボルは省略形式で表示します。 	-

(1) Show Required Stack / Show Used Stack の表示

呼び出し情報ビューのスタックサイズ表示形式の設定と表示を以下に示します。

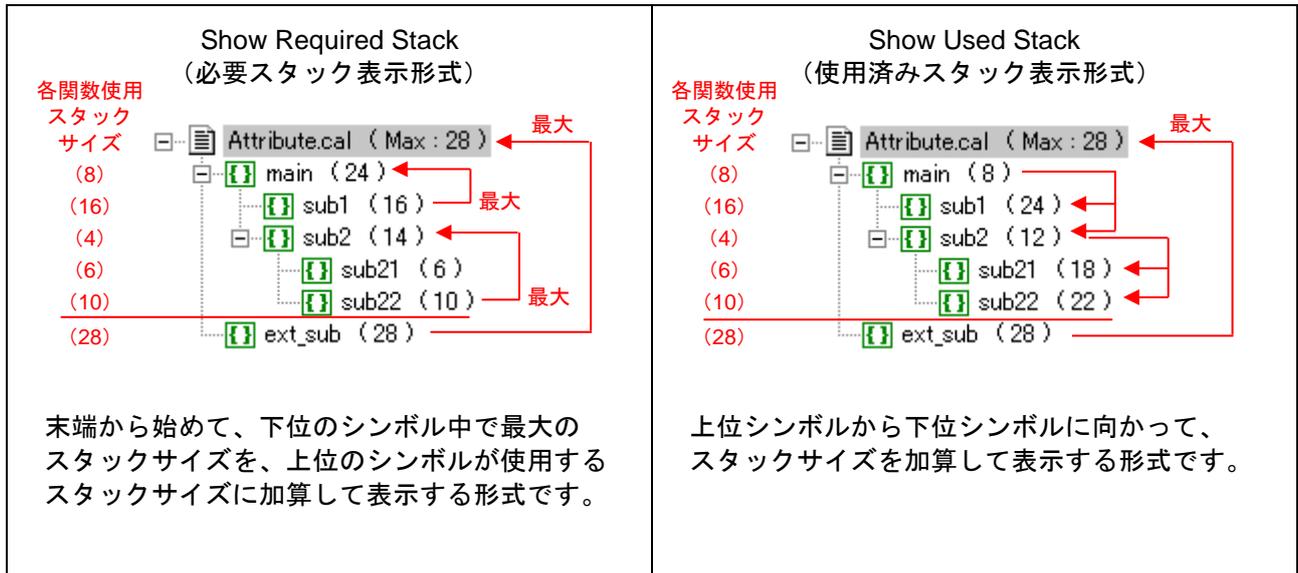


図 5-3. Show Required Stack / Show Used Stack の表示

5.4 Tools メニュー

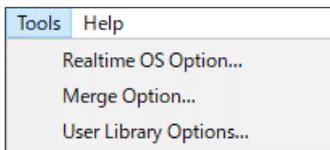
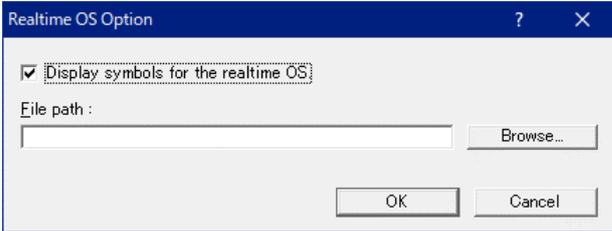


表 5-4. Tools メニューの内容

コマンド	内容	アイコンとショートカット
Realtime OS Option	<p>RTOS 標準関数の表示設定を行います。ダイアログボックスで「Display symbols for the realtime OS」にチェックを入れると、RTOS ライブラリデータファイルを指定できます。</p>  <p>RTOS ライブラリデータファイルは、拡張子「.csv」で作成されています。（ただし本ツールと同じパスにある「Call.csv」（標準ライブラリスタック情報ファイル）は、ここで使用するファイルではありませんので指定しないでください。）</p> <p>RX ファミリー用リアルタイム OS RI600V4 の場合は、カーネルソース（アセンブラ）で STACK 制御命令を使ってスタックサイズを記述しているため、スタック使用量は Call Walker に反映済みです。</p>	-
Merge Option	<p>ダイアログボックスから、編集情報のマージを行う際の編集方法の設定を行います。</p> <p>詳しくは「8. スタック情報のマージ」を参照ください。</p>	-
User Library Options	<p>すでに Call Walker で表示しているシンボルのスタックサイズを、読み込んだユーザライブラリデータファイルに記述されたスタックサイズに置き換え、表示内容を更新します。</p> <p>作成したユーザライブラリデータファイル (*.csv) を、このコマンドのダイアログボックスで指定し、Add ボタンを押すことでスタック情報が更新されます。</p> <p>詳しくは「9. ユーザライブラリファイル」を参照ください。</p>	-

5.5 Help メニュー



表 5-5. Help メニューの内容

コマンド	内容	アイコンとショートカット
Help Topics	本ツールに関するヘルプを参照します。各コマンドの詳細は、このヘルプを参照ください。	-
About Call Walker	本ツールのバージョンや著作権等を参照します。	

6. アセンブラプログラムのスタック使用量表示

C/C++プログラムと異なり、アセンブラプログラムは、アセンブルをしてもスタック使用量を自動で算出することができません。このため、Call Walker にスタック使用量を表示するためのアセンブラ拡張機能制御命令「.STACK」が用意されています。

備考. 「.STACK」制御命令は、Call Walker にスタックサイズを表示させる機能です。プログラムの動作に影響を与えるものではありません。

6.1 「.STACK」アセンブラ拡張機能制御命令

「.STACK」は、シンボルに対して、Call Walker で参照するスタック使用量を定義します。

「.STACK」制御命令の記述方法は次の通りです。

```
.STACK△<シンボル>=<数値>           △:1個以上の半角スペース
```

- 1つのシンボルに対して定義できるスタック値は、1度のみ有効とします。2度以上指定した場合は、その定義を無効とします。
- CC-RX、CC-RHの場合は、指定できるスタック値は、0x0~0xFFFFFFFFCの範囲の4の倍数のみとし、それ以外を指定した場合はその定義を無効とします。
- CC-RLの場合は、指定できるスタック値は、0x0~0xFFFEの範囲の2の倍数のみとし、それ以外を指定した場合はその定義を無効とします。
- <数値>は定数値とし、かつ前方参照シンボル、外部参照シンボル、相対アドレスシンボルを使わずに指定してください。

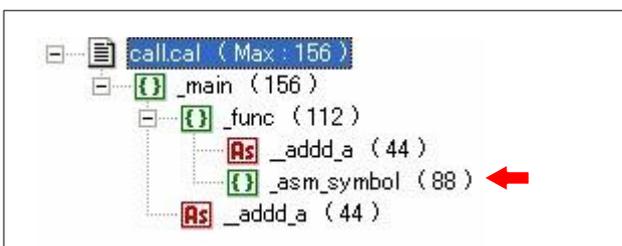
6.2 アセンブラプログラム記述例

_asm_symbol関数のスタックサイズを「88」に設定します。

```
.GLB _asm_symbol
.SECTION P, CODE
_asm_symbol:
.STACK _asm_symbol=88
;
RTS
.END
```

[Call Walker の表示例]

Call Walker 上で、_asm_symbol関数のスタック使用量が「88」と表示されます。



7. スタック情報の編集

本ツールは静的なスタック最大使用量を算出できますが、多重割り込みなどによる、動的な最大使用量を計算するためには、ユーザ側で情報ファイルを編集する必要があります。

ファイルをオープン中、右側のシンボル詳細ビューでシンボル名を選択し、Edit メニューの Add、Modify または Delete コマンドで、シンボルの追加、変更、または削除ができます (シンボル詳細ビューで右クリックしても、同様の操作が行えます)。

また、左側の呼び出し情報ビューでシンボルをドラッグ&ドロップすると、シンボルの位置を変更できます。

Call Walker は静的なスタック最大使用量を算出できますが、多重割り込みによる動的な最大使用量などを計算に含めるには、ユーザが情報ファイルを編集する必要があります。

シンボルを選択し、Edit メニューの各コマンド (Add/Modify/Delete) を使用することで、情報の編集 (シンボルの追加、変更、または削除) ができます。また、右側の「シンボル詳細ビュー」で右クリックメニューから同様の操作を選択できます (メニューにある Radix については「5.3 View メニュー」を参照ください)。

左側の「呼び出し情報ビュー」でシンボルをドラッグ&ドロップすると、シンボルの位置を変更できます。

なお、移動や Add/Modify コマンドで編集したシンボルには、変更を示すチェックマークが付きます (4.2 章 (3) を参照してください)。

7.1 Add コマンドによるシンボルの追加

シンボル情報を追加する方法として新規にシンボルを登録する方法と、既存のシンボルから選択する方法があり、どちらか一方を選択して追加します。

新規シンボル入力時は、「New Symbol」チェックボタンにチェックを入れ、シンボル名称および他の項目を入力、チェックまたは、選択します。シンボル情報は Address をキーとして管理しています。

既存シンボルからの選択時は、「New Symbol」チェックボタンにチェックを入れなくて、「Symbol list」から該当のシンボルを選択します。他の項目は、既存の情報を反映させるため、変更できません。「Symbol list」は、リスト内の各項目について、ヘッダ (項目表示部分) をクリックすることで、ソートが可能です (ソート動作については、メインウィンドウのシンボル詳細ビューと同様に動作します)。

再帰呼び出し関数、循環関数、および省略表示シンボルの下位にシンボルを追加することはできません。

RTOS 関数については、編集情報内にある既存のシンボルを選択して追加することはできませんが、新規シンボルを RTOS として追加することはできません。ただし、新規に追加したシンボルのシンボル名が、RTOS 変換対象シンボルである場合、追加したシンボルは RTOS シンボルとして表示されます。

シンボル情報を入力または選択し OK ボタンをクリックすると、選択シンボル直下に入力したシンボル情報を追加します。

追加シンボルと同名のシンボルが他に設定されており、そのシンボルが下位シンボルを持つとき、下位シンボルも追加します。また、別層に選択シンボルと同名のシンボルが設定されている場合、その直下にも同シンボル構成を追加します。

なお、Address、Size および Stack Size の初期表示時の基数表記は、Radix コマンドの設定に依存しません。手入力時は、Radix コマンドの設定には依存せず、先頭に 0x を付けることで 16 進扱いとし、数字のみの場合は 10 進扱いとします。

(1) 新規シンボルの追加

Add コマンドのダイアログパネル左側の「New symbol」チェックボックスにチェックを入れると、選択中のシンボル直下の階層に新規シンボルを作成できます。ダイアログパネル各項目で、新規シンボルのシンボル名、シンボル区分、属性、アドレス、スタック使用量、オブジェクトファイルなどを設定できます。

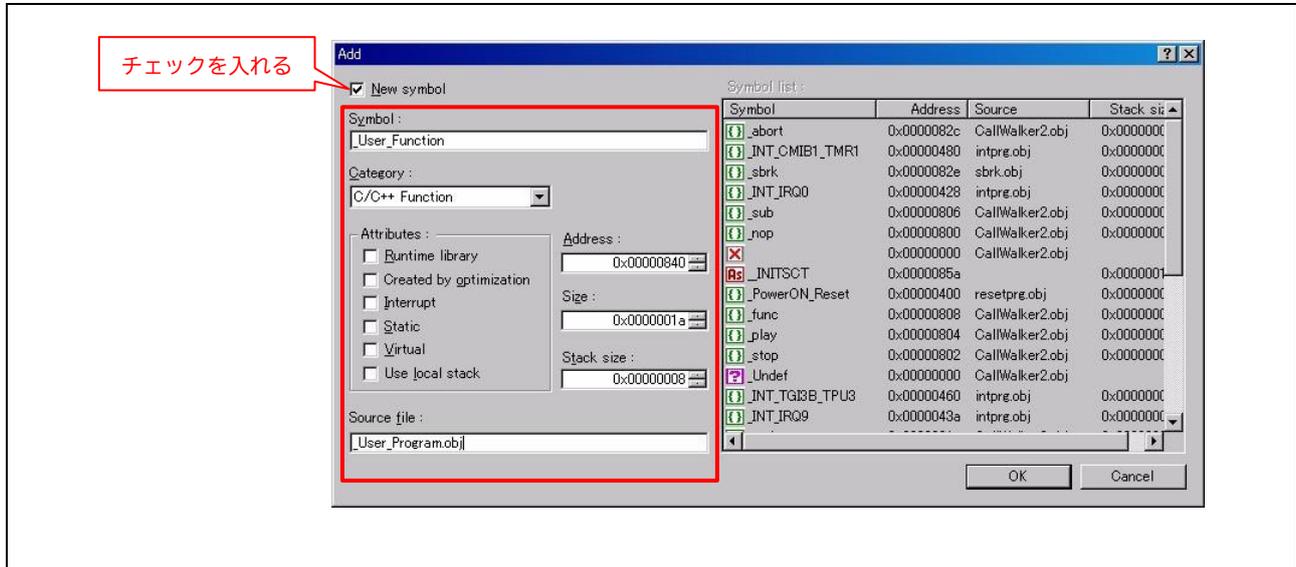


図 7-1. Add コマンドのダイアログパネル（新規シンボルの追加）

(2) 既存シンボルの追加

シンボルを選択し、Edit メニューの Add コマンドをクリックすると、以下のダイアログパネルが表示されます。右側の一覧は現在のファイル内のシンボルです。選択中のシンボル直下の階層に、既存のシンボルを追加する場合は、このリストからシンボルを選択し OK ボタンをクリックします。

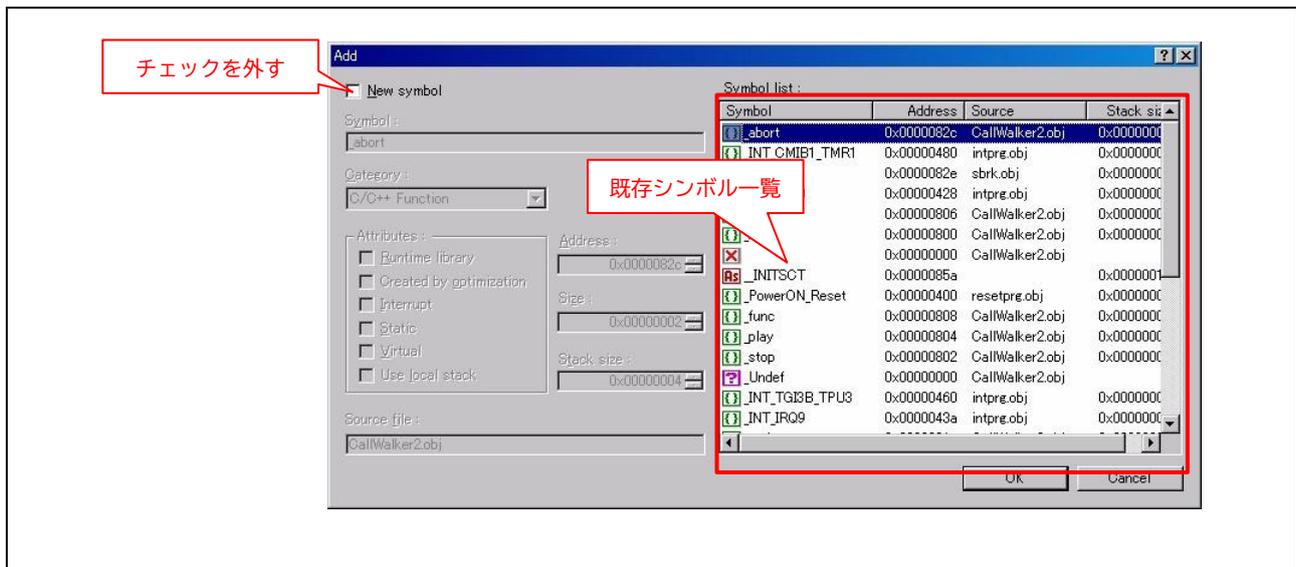


図 7-2. Add コマンドのダイアログパネル（既存シンボルの追加）

(3) シンボル追加の例

図 7-3 において、

- ① : sub1 の呼び出しシンボルに sub3 を追加します。
- ② : sub3 の呼び出しシンボルとして sub4 が設定されているので、sub4 も追加されます。
- ③ : sub2 の呼び出しシンボルとして sub1 が設定されているので、同様に sub1 の下に sub3 と sub4 が追加されます。

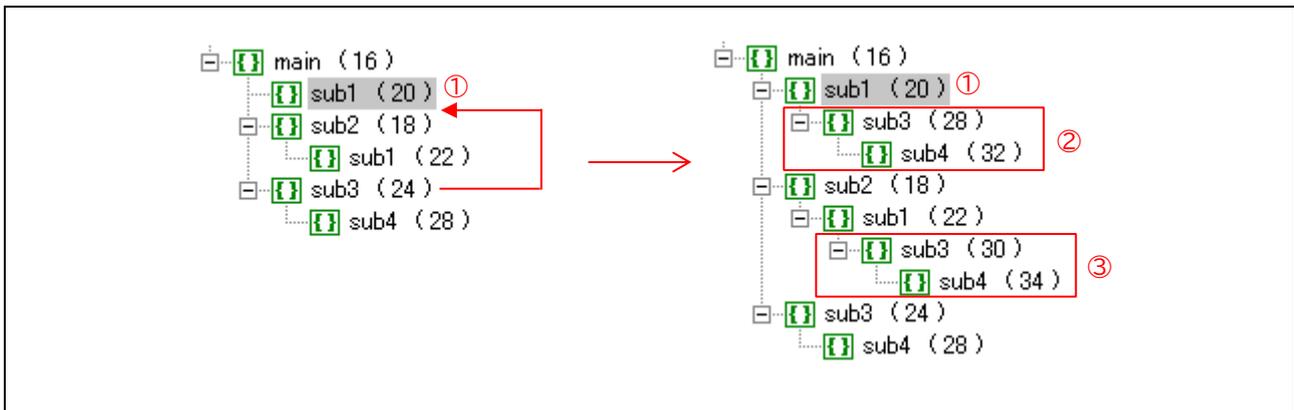


図 7-3. シンボル追加例

図 7-3 の例のように、指定位置以外にも追加した場合、複数追加したことを示すインフォメーションメッセージを表示します。

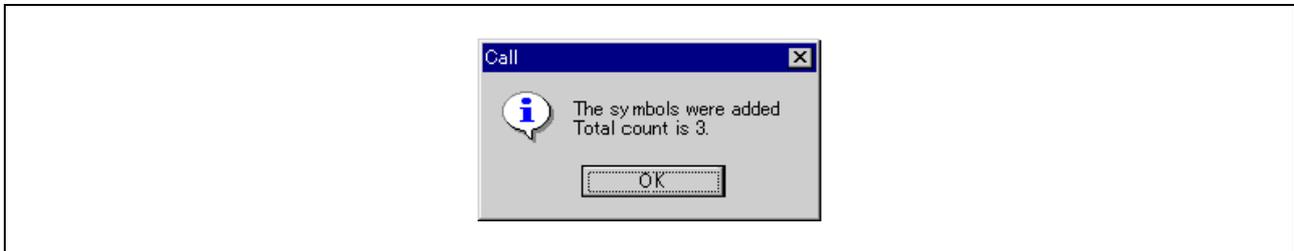


図 7-4. シンボル複数箇所追加のメッセージ

カウントの対象は追加指定したシンボルのみで、下位シンボルはカウント対象外です。呼び出し情報ビューを省略形式で表示している場合は、省略されたシンボルはカウントされません。

また、既存のシンボルとアドレスが同じ場合も新規シンボルを追加できます。新規シンボルが既存のシンボルとアドレス、名称、および種類は同じで、スタックサイズが異なる場合、これらを可変個引数を持つ関数シンボルに変更するか否かを確認するメッセージを表示します。OK を選択するとこれらは可変個引数を持つ関数シンボルになり、新規シンボルは既存のシンボルのスタックサイズ以外のすべての情報を継承します。

図 7-5 において、

- ① : main の呼び出しシンボルとして sub1 を追加します。
- ② : 新規シンボルは既存の sub1 とアドレス・名称・種類が同じで、スタックサイズが異なります。これらを可変個引数を持つ関数シンボルに変更するか否かを確認するメッセージを表示します。
- ③ : sub2 (既存のシンボル) は sub1 の呼び出しシンボルとして設定されているため、新規シンボルの呼び出しシンボルとして sub2 も追加します。

他のケースではエラーになります。

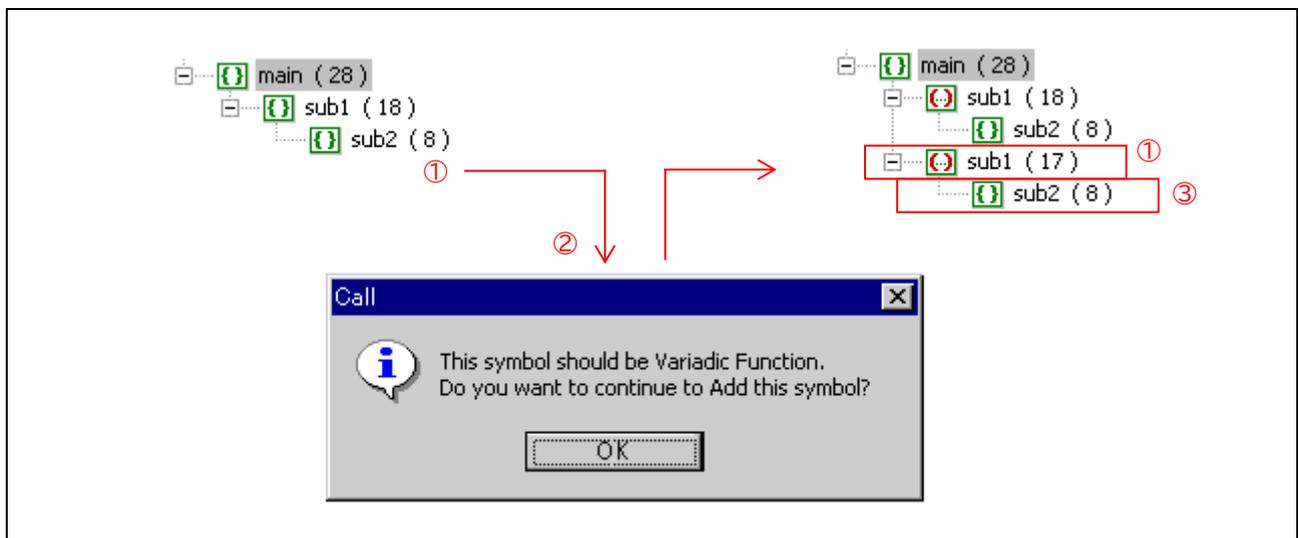


図 7-5. 同じアドレスを持つシンボルの追加例

7.2 Modify コマンドによるシンボル情報の変更

(1) 選択中のシンボル情報変更

情報を変更したいシンボルを選択し、Edit メニューの Modify コマンドをクリックすると、以下のダイアログボックスが表示されます。ここで各種情報を変更することができます。

ただし、RTOS シンボル選択時は Modify コマンドを実行することはできません。

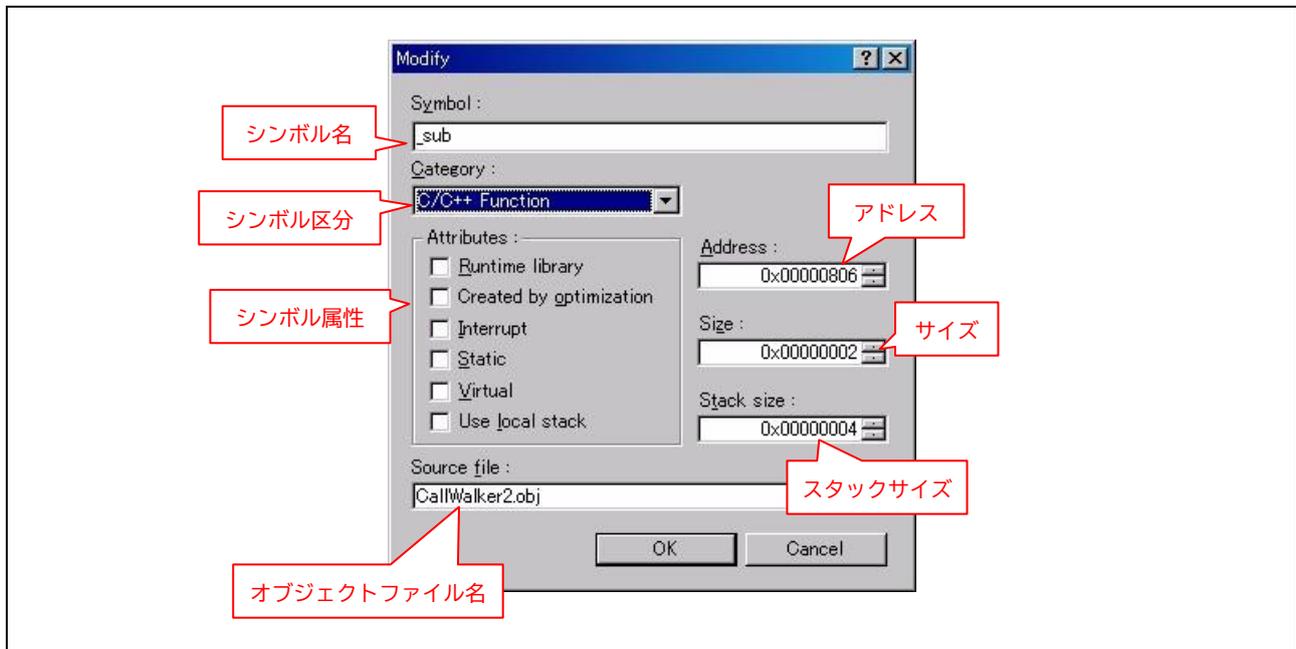


図 7-6. Modify コマンドのダイアログパネル

ダイアログボックス内の情報を入力、チェックまたは選択し、OK ボタンをクリックすると入力したシンボル情報に変更します。

シンボル情報は Address をキーとして管理しています。

なお、Address、Size および Stack Size の初期表示時の基数表記は、Radix コマンドの設定に依存します。手入力時は、Radix コマンドの設定には依存せず、先頭に 0x を付けることで 16 進扱いとし、数字のみ場合は 10 進扱いとします。

本コマンドを使用してシンボル種別を RTOS 関数には変更できません。ただし、シンボル名を変更した際に、変更後のシンボル名が RTOS シンボルと一致した場合は、RTOS シンボルとして表示されます。

変更したシンボルが指定位置以外にも存在する場合、それらすべてに対して変更します。その際、複数変更したことを示すインフォメーションメッセージを表示します。

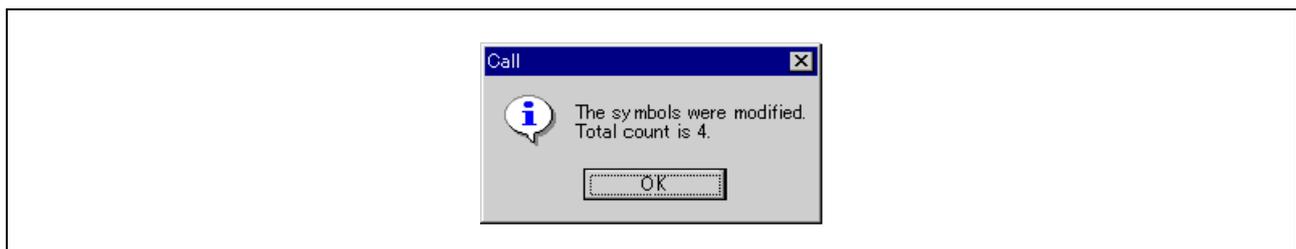


図 7-7. シンボル複数箇所追加のメッセージ

カウントの対象は変更指定したシンボルのみで、下位シンボルはカウント対象外です。呼び出し情報ビューを省略形式で表示している場合は、省略されたシンボルはカウントされません。

(2) シンボル変更のルール

シンボルの変更に以下のルールがあります。

(a) 可変個引数を持つ関数シンボルを選択した場合

シンボルの種類とアドレスの両方を変更する場合、可変個引数を持たない関数シンボルの情報を変更するときと同様にこのシンボルのみ変更されます。

シンボルの種類以外の情報を変更した場合、次のようになります。

- 変更した新しいアドレスが他のシンボルと同じ場合、エラーが発生します。
- アドレス、名称、ソースファイル、サイズ、または属性を変更した場合、可変個引数を持つ関数シンボルのグループ（同じアドレス・名称を持つ可変個引数を持つ関数シンボルを含む）のすべてのシンボルが変更されます。
- 変更した新しいスタックサイズが、可変個引数を持つ関数シンボルのグループの他のシンボルのスタックサイズと同じ場合、エラーが発生します。

可変個引数を持つ関数のシンボル情報の変更例を以下に示します。

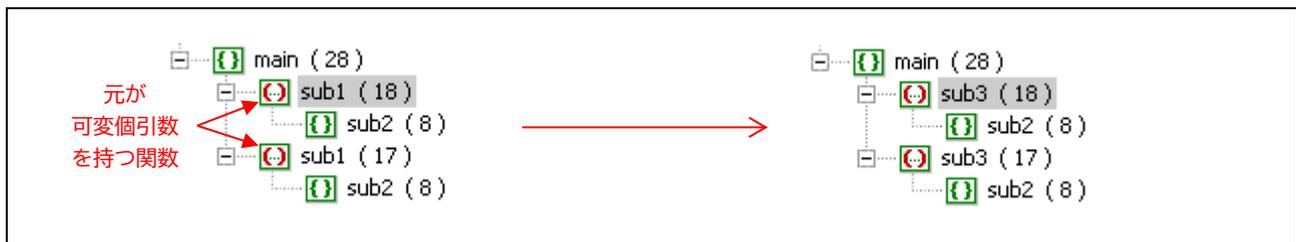


図 7-8. 可変個引数を持つ関数のシンボル情報の変更例

シンボル名 sub1（最初のシンボル）を sub3 へ変更した場合、これら 2 つのシンボルは 1 つの可変個引数を持つ関数シンボルのグループに属するので、2 番目のシンボルも sub3 へ変更されます。

(b) 可変個引数を持たない関数シンボルを選択した場合

変更した新しいアドレスが他のシンボルと異なる場合、指定したシンボルの情報は通常通り変更されません。

変更した新しいアドレスが他のシンボルと同じ場合、以下のようになります。

- Modify ダイアログボックスの入力情報と同じアドレス、名称を持つ選択したシンボルまたは既存のシンボルに呼び出しシンボルがある場合、エラーが発生します。
- 既存のシンボルと Modify ダイアログボックスで指定した Symbol、Address、Category が同じで Stack size が異なる場合、これらを可変個引数を持つ関数シンボルに変更するか否かを確認するメッセージを表示します。OK を選択するとこれらは可変個引数を持つ関数シンボルになり、Modify ダイアログボックスの Stack size 以外の入力情報を継承します。

可変個引数を持たない関数のシンボル情報の変更例を以下に示します。

図 7-9 において、

- ① : sub1 を sub2 に変更します。Modify ダイアログボックスでの指定は、既存のシンボル (sub2) の情報と Symbol、Address、Category は同じで Stack size が異なります。sub1 と sub2 に呼び出しシンボルはないものとします。
- ② : 可変個引数を持つ関数シンボルに変更するか否かを確認するメッセージを表示します。
- ③ : OK を選択すると、sub1 から変更した sub2 は、Modify ダイアログボックスで指定した情報に変更されます。ただし、Stack size の値は元のシンボルの値を保持します。

他のケースではエラーになります。

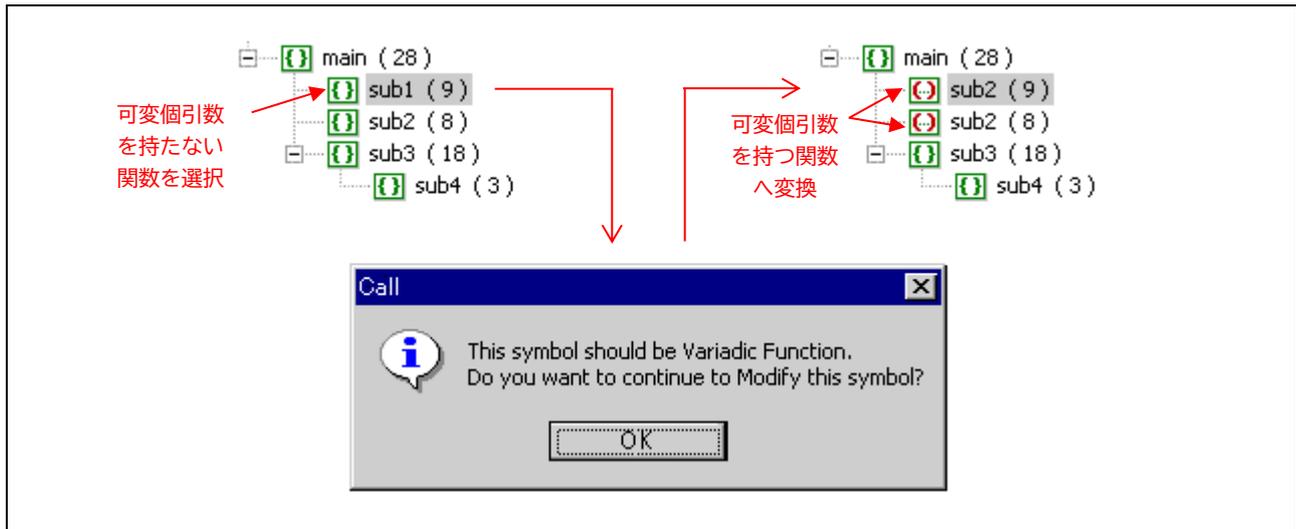


図 7-9. 可変個引数を持たない関数のシンボル情報の変更例

8. スタック情報のマージ

Call Walker 編集情報の保存後、編集中のスタック情報ファイルと、他のスタック情報ファイルをマージすることができます。

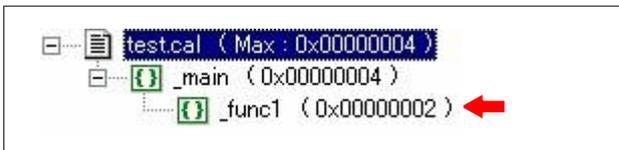
この機能を使用することで、編集したスタック情報が、再ビルド後のスタック情報に上書きされるのを防ぐことができます。

8.1 マージの手順 (例)

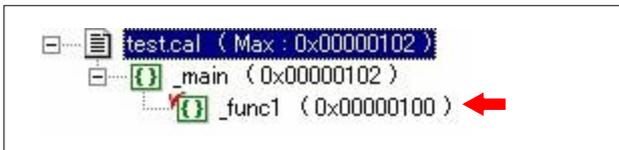
① test.c の内容

```
void main(void)
{
    func1();
}
```

② test.c をビルドし、Call Walker でスタック情報を開きます。



③ 内容を変更します。(func1 のスタック使用量を 0x00000100 に変更)

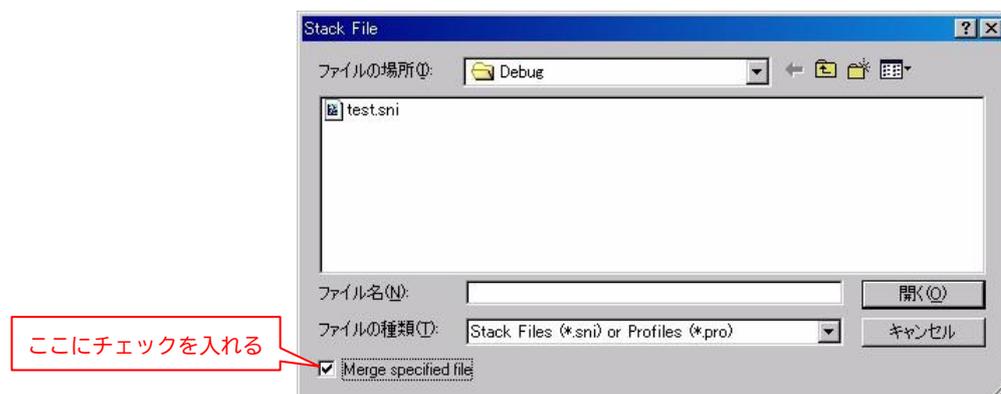


④ test.c の内容を変更します。(func2 の呼び出しを追加)

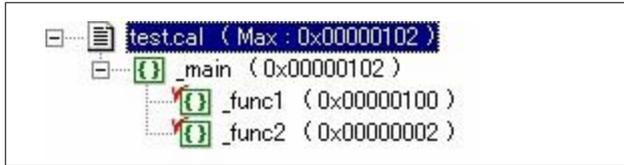
```
void main(void)
{
    func1();
    func2();
}
```

⑤ Call Walker は開いたまま、test.c を再ビルドします。

⑥ 開いたままの Call Walker で test.sni をインポートします。このとき「Merge specified file」にチェックを入れ、「開く」ボタンをクリックします。



- ⑦ ⑥の操作でインポートを行うと、③で変更した func1 のスタック使用量はそのまま、func2 の情報が追加になります。これがスタック情報のマージです。



もし、⑥で「Maerge specified file」をチェックしなかった場合は、以下のように③で変更した func1 のスタック使用量は、変更前の値に戻ります。



8.2 マージタイプの選択

マージする方法（マージタイプ）は5種類あり、マージオプションのダイアログパネルで変更することができます。マージオプションは、[Tools] > [Merge Option] で設定します。

ダイアログパネルの Merge type でマージタイプを選択すると、Sample view に選択中の設定によるマージ編集例が、Description 欄に編集方法の内容と編集例についての説明が表示されます。

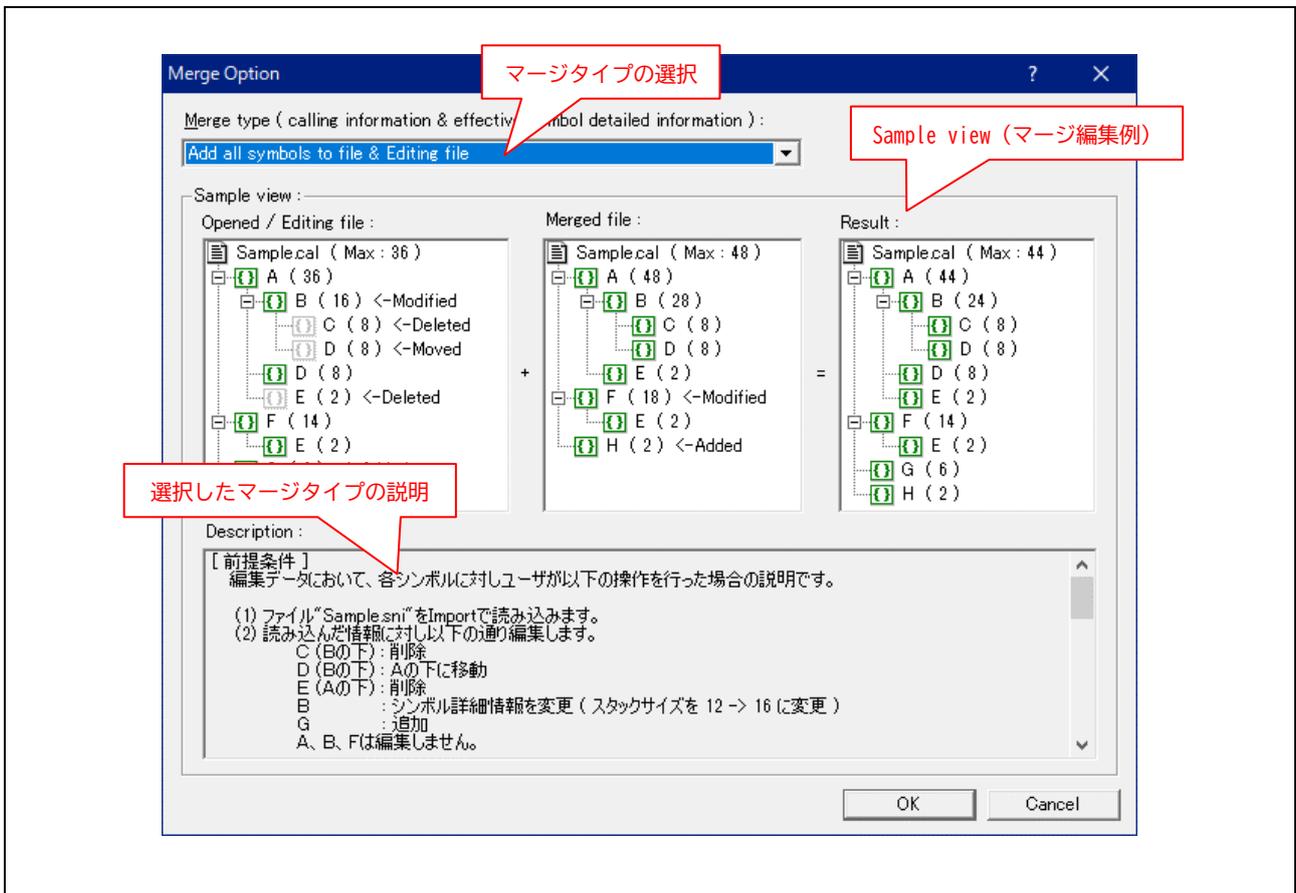


図 8-1. マージオプションのダイアログパネル

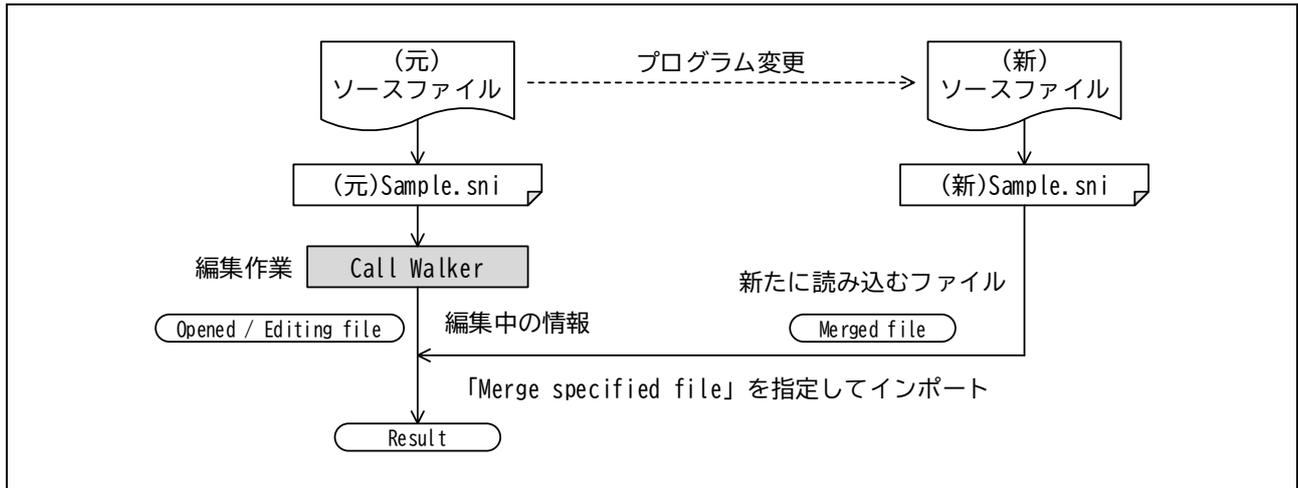


図 8-2. 補足図：サンプルビューの処理説明 (前提条件)

マージタイプ：

- (1) Not merge calling information & Merged file
- (2) Add all symbols to file & Editing file
- (3) Add all symbols to file & Merge file
- (4) Add only new symbols to file & Editing file
- (5) Add only new symbols to file & Merged file

表 8-1. マージタイプとマージ処理の概要

マージタイプ	呼び出し情報	シンボル詳細情報
(1)	Not merge calling information : ・ 編集中の情報を維持します (新たに読み込んだファイル内の呼び出し情報はマージしません)。	Merge file : ・ 新たに読み込んだファイルのシンボル詳細情報を有効とします (編集中の情報で変更を行ったシンボルの詳細情報は失われます)。
(2)	Add all symbols to file : ・ 新たに読み込んだファイル内にあり、編集中の情報にないシンボルを追加します。 ・ 削除されたシンボルは再び追加します。 ・ 他の位置に移動されたシンボルは、移動前の場所に再び追加します。このとき移動後の位置にあるシンボルはそのまま残ります。	Editing file : ・ 編集中の情報を有効とし、新たに読み込んだファイルのシンボル詳細は無視します。 ・ ただし、新たに読み込んだファイルから新規追加するシンボルについては、新たに読み込んだファイルの内容を反映します。
(3)	Add all symbols to file : (2)と同じ	Merge file : (1)と同じ
(4)	Add only new symbols to file : ・ 新たに読み込んだファイル内にあり、編集中の情報にないシンボルを追加します。 ・ 削除されたシンボルは追加しません。 ・ 他の位置に移動されたシンボルは追加しません。	Editing file : (2)と同じ
(5)	Add only new symbols to file : (4)と同じ	Merge file : (1)と同じ

9. ユーザライブラリファイルによるスタックサイズ表示変更

Tool メニューの User Library Options コマンドは、すでに Call Walker で表示しているシンボルのスタックサイズを、読み込んだユーザライブラリデータファイルに記述されたスタックサイズに置き換え、表示内容を更新します。

ユーザライブラリデータファイルのフォーマットは次の通りです。

```
[Display Module]
既表示シンボル名, スタックサイズ
[Non Display Module]
非表示にするシンボル名
```

記述例を以下に示します (テキストエディタで作成し、ファイル名*.csv で保存)。

```
[Display Module]
_sub1, 500
[Non Display Module]
_sub2
```

作成したユーザライブラリデータファイル (*.csv) を、User Library Options コマンドのダイアログボックスで指定し、スタック情報を更新します。

Tool メニューの User Library Options コマンドを選択して、ダイアログボックスを開きます。

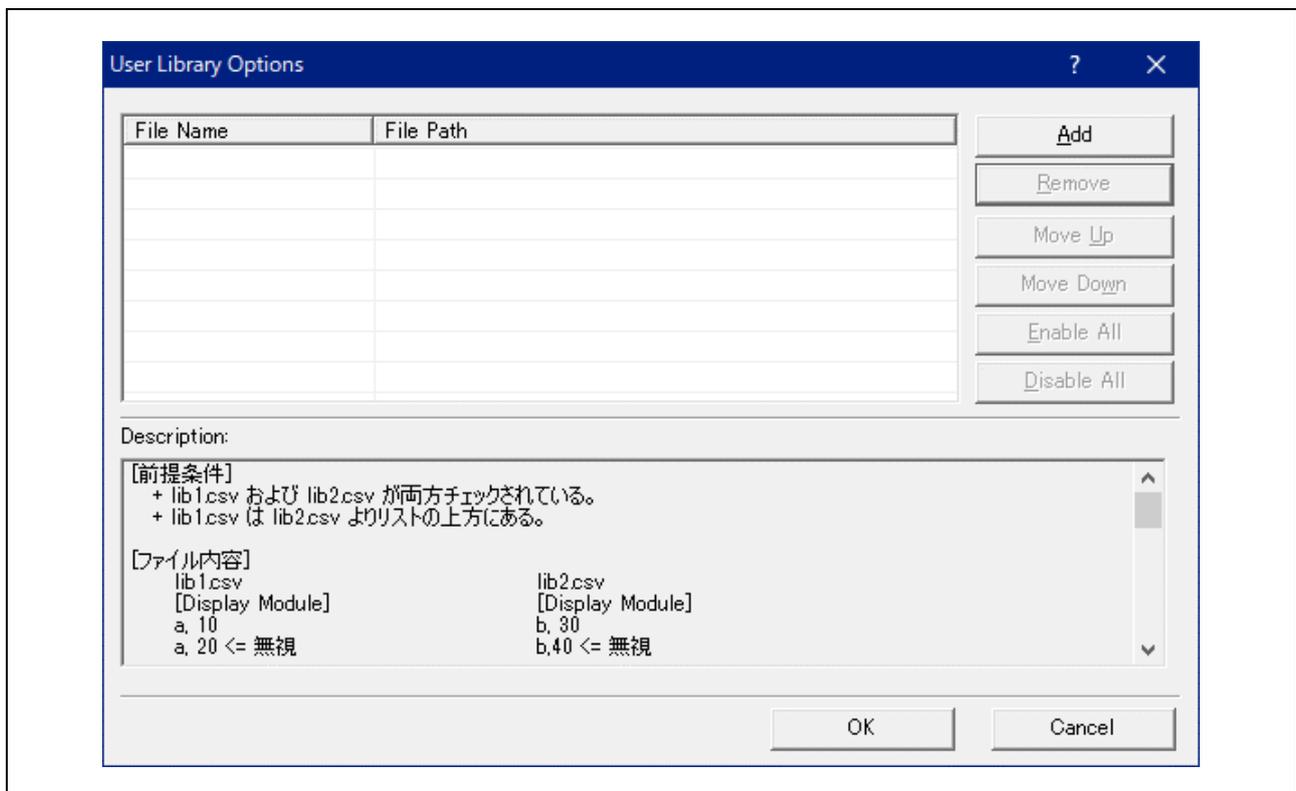


図 9-1. User Library Options コマンドのダイアログボックス

Add ボタンをクリックして、ユーザライブラリデータファイルを指定します。

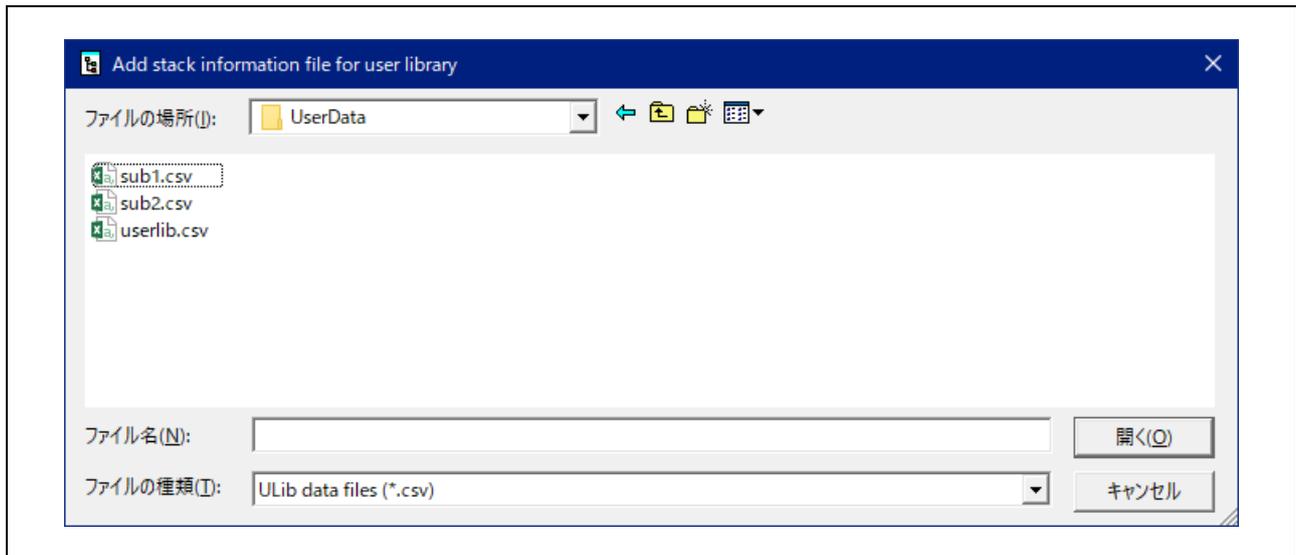


図 9-2. ユーザライブラリデータファイルの追加

各ユーザライブラリデータファイルを有効化するには、対応するチェックボックスをチェックします。複数のユーザライブラリデータファイルを選択し、同時に有効化できます。

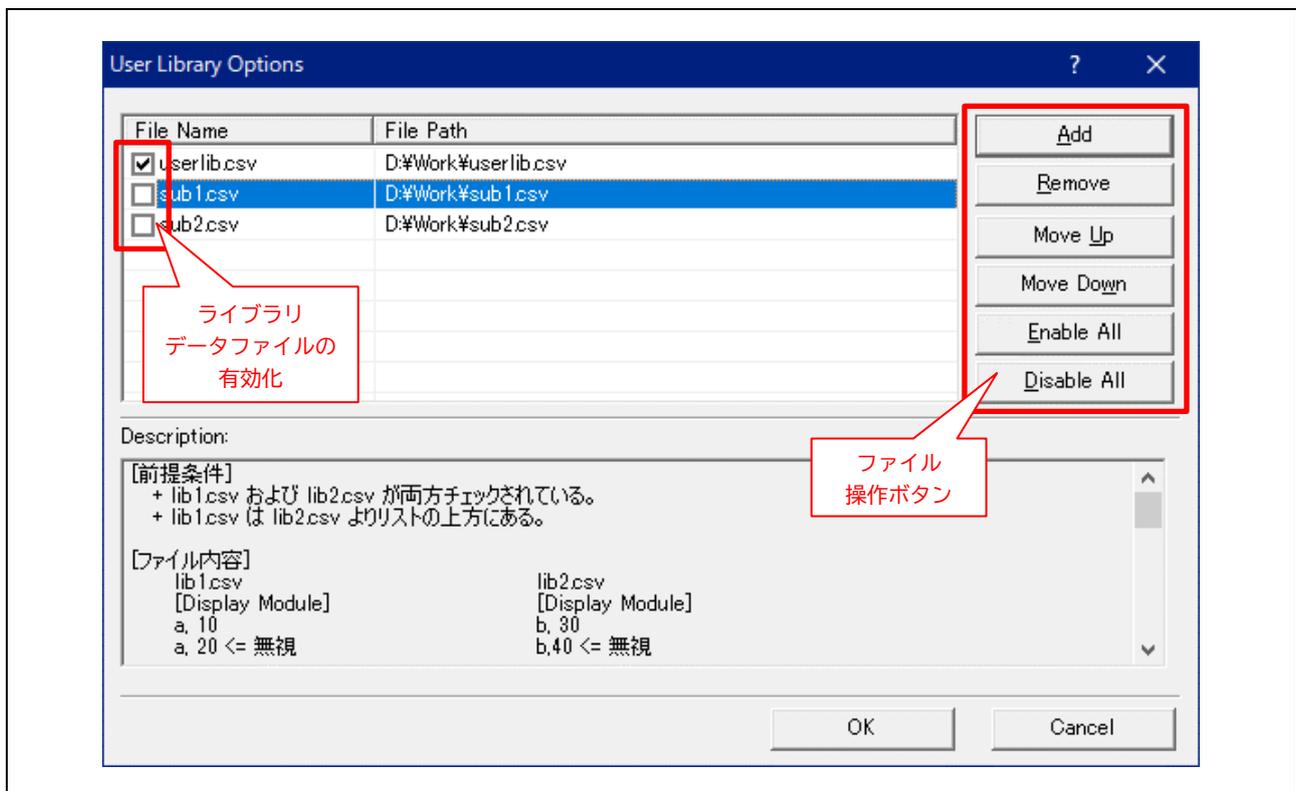


図 9-3. User Library Options コマンド ダイアログボックスの操作

表 9-1. User Library Options コマンドのダイアログボックスの操作ボタン

ボタン	操作
Remove	選択したユーザライブラリファイルをリストから削除する
Move Up	選択したファイルを上へ移動する
Move Down	選択したファイルを下へ移動する
Enable All	ユーザライブラリファイルをすべて有効化する
Disable All	ユーザライブラリファイルをすべて無効化する

OK ボタンをクリックすると、シンボルの現在のスタックサイズをファイルから読み込んだスタックサイズに置き換え、表示内容を更新します。

スタックサイズの置き換えの例を以下に示します。

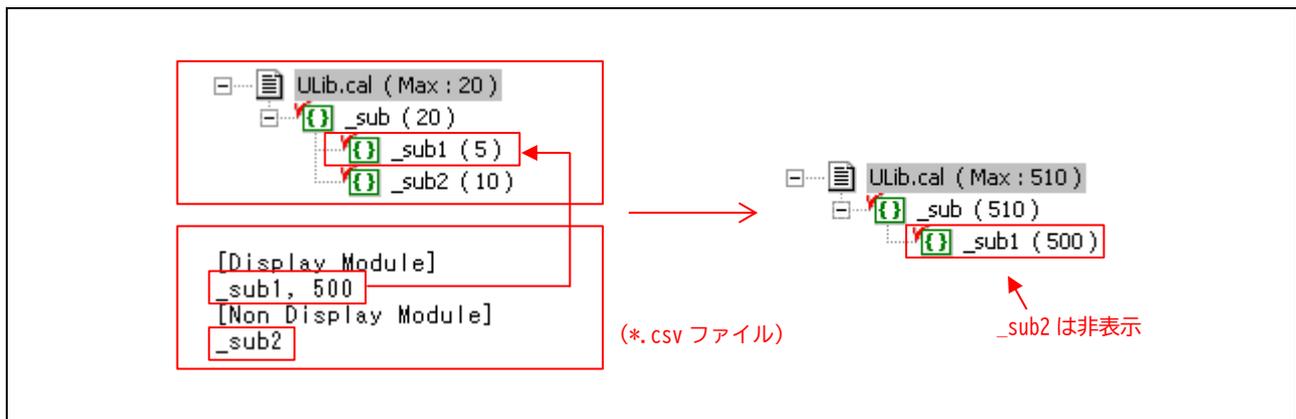


図 9-4. スタックサイズの置き換えの例

付録 1. インフォメーションメッセージ

1	メッセージ	The symbols were added. Total count is <加算シンボル数>.
	意味	Add コマンド実行時、指定した位置以外にもシンボルが追加されました。
	説明	Add コマンドでシンボルを追加した時に、追加指定した位置のシンボルと同じシンボルが他の位置に存在した場合、その位置にも同様に追加します。追加した位置が指定した位置以外にも存在する場合にこのメッセージを表示します。
	ツールの処理	処理を続行します。
	ユーザの対処	特になし。
2	メッセージ	The symbol were modified. Total count is <変更シンボル数>.
	意味	Modify コマンド実行時、指定した位置以外のシンボルも変更されました。
	説明	Modify コマンドで変更したシンボルが指定した位置以外にも存在する場合、そのシンボルに対しても同様に変更します。変更したシンボルが指定した位置以外にも存在する場合にこのメッセージを表示します。
	ツールの処理	処理を続行します。
	ユーザの対処	特になし。
3	メッセージ	The symbols were removed. Total count is <削除シンボル数>.
	意味	Delete コマンド実行時、指定した位置以外のシンボルも削除されました。
	説明	Delete コマンドでシンボルを削除した時に、削除指定したシンボルの参照元シンボルと同じものが指定以外の位置に存在し、かつそのシンボルが削除指定したシンボルを参照している場合、そのシンボルについても同様に削除します。削除したシンボルが指定した位置以外にも存在する場合にこのメッセージを表示します。
	ツールの処理	処理を続行します。
	ユーザの対処	特になし。
4	メッセージ	Call Walker has finished searching a symbol
	意味	検索処理が完了しました。(Find Option : 最大スタックサイズ)
	説明	最小スタックサイズ (Find Next) または最大スタックサイズ (Find Previous) を持つシンボルを見つけると、このメッセージを表示します。
	ツールの処理	処理を中断します。
	ユーザの対処	特になし。
5	メッセージ	Cannot find the symbol <Symbol name>
	意味	シンボルが見つかりませんでした。
	ツールの処理	処理を中断します。
	ユーザの対処	探したいシンボル名を確認し、修正してください。
6	メッセージ	The number of items is too large. Cannot display all.
	意味	項目数が多いため表示できませんでした。
	ツールの処理	処理を中断します。
	ユーザの対処	表示オプションの確認: 「Show All Symbols」を選択している場合、「Show Simple Symbol」に変更してください。 「Show All Symbols」を選択していない場合、シンボル情報の全体を参照するには「Output List」を使用してください。
7	メッセージ	This symbol should be Variadic Function. Do you want to continue to [Add/Modify] this symbol?
	意味	シンボルを可変個引数を持つ関数に変更可能な場合に、このメッセージを表示します。
	ツールの処理	操作結果に依存します。
	ユーザの対処	OK を選択すると、シンボルを可変個引数を持つ関数に変更します。 キャンセルを選択すると、ウォーニングメッセージを表示して処理を中断します。

付録 2. エラーメッセージ

(1/3)

1	メッセージ	Cannot open a file. '<Error file>'
	意味	<Error file>で示すファイルがオープンできません。
	ツールの処理	処理を中断します。
	ユーザの対処	正しいファイル名を指定してください。 使用中のストレージに問題がないか確認してください。
2	メッセージ	Cannot read a file. '<Error file>'
	意味	<Error file>で示すファイルを読み込み中にエラーが発生しました。
	ツールの処理	処理を中断します。
	ユーザの対処	ファイルが破損していないか確認してください。 使用中のストレージに問題がないか確認してください。
3	メッセージ	Cannot write a file. '<Error file>'
	意味	<Error file>で示すファイルを書き込み中エラーが発生しました。
	ツールの処理	処理を中断します。
	ユーザの対処	書き込み場所、書き込み権限を確認してください。 書き込みプロテクトがかかっていないか確認してください。
4	メッセージ	Cannot close a file. '<Error file>'
	意味	<Error file>で示すファイルをクローズする際にエラーが発生しました。
	ツールの処理	処理を中断します。
	ユーザの対処	メモリ、ストレージに問題がないか確認してください。
5	メッセージ	The version of file which is not supported was specified. '<Error file>'
	意味	<Error file>で示すファイルのバージョンはサポートされていません。
	ツールの処理	処理を中断します。
	ユーザの対処	正しいファイル名を指定してください。
6	メッセージ	The contents of the specified file are wrong. '<Error file>'
	意味	<Error file>で示すファイルの内容が不正です。
	ツールの処理	処理を中断します。
	ユーザの対処	正しいファイル名を指定してください。
7	メッセージ	Not enough memory.
	意味	メモリに空き容量がありません。
	ツールの処理	処理を中断します。
	ユーザの対処	他のアプリケーションを終了して、メモリの空き容量を増やしてください。
8	メッセージ	Cannot move here ... <Detail Message>. <Detail Message> <ul style="list-style-type: none"> • Same symbol exists on destination symbol: … 移動先に同名シンボルがあります。 • Move to omitted symbol: … 移動先が再帰呼び出し関数または循環関数です。 • Move to recursive symbol: … 移動先が省略表示シンボルです。 • Move to lower symbol: … 移動先が自シンボルの下位です。 • Move to own symbol: … 移動先が自シンボルを直接参照しているシンボル (位置不変) です。 • Move to symbol without name: … 移動先が名称のないアドレス参照未解決シンボルです。
	意味	シンボル情報のドラッグアンドドロップ操作で、移動できない場所に移動しようとしてしました。
	ツールの処理	処理を中断します。
	ユーザの対処	上記<Detail Message>に示す場所へ移動できません。移動可能な位置を確認し、指定してください。

9	メッセージ	Failed to input ... <Detail Message> <Detail Message> <ul style="list-style-type: none"> Illegal value input on Address/Size/Stack Size/Expansion Level: <ul style="list-style-type: none"> … Address, Size, Stack size, または Expansion Level に不正値が入力されています。 Symbol name was not selected: … シンボル名称が選択されていません。 Symbol name was not input: … シンボル名称が入力されていません。 A symbol of same address already exists: … 入力されたアドレスは既に登録されています。 Specified symbol already exists: … 指定されたシンボルはすでに同階層に存在しています。 A filename cannot contain any of the following characters (¥/:;*?"<>): <ul style="list-style-type: none"> … Source File にファイル名に使用できない文字が使用されています。 Symbol name is not input or selected: … シンボル名が入力または選択されていません。 A symbol name cannot contain character (): <ul style="list-style-type: none"> … シンボル名に使用できない文字が使用されています。
	意味	Add/Modify/Expansion の各ダイアログボックスで、不正値が入力されたため各コマンドを実行できません。
	ツールの処理	処理を中断し、再びダイアログボックスの入力に戻ります。
	ユーザの対処	入力内容を確認し、正しい値を入力してください。 Address/Size/Stack size は、数字のみは 10 進、先頭に 0x を付けると 16 進と認識します。 Expansion Level の入力値の範囲は、10 進のみで 1~32 になります。 それ以外の文字を入力できません。また、入力値の範囲は、0 (0x00000000) ~ 4294967295 (0xffffffff) になります。 Source File でファイル名に使用できない文字 (¥/:;*?"<>) を使うとエラーになります。
10	メッセージ	Failed to input ... Symbol name is not input or selected
	意味	Find ダイアログボックスでシンボル名称が選択または入力されていません。
	ツールの処理	処理を中断します。
	ユーザの対処	シンボル名称の検索時は、Symbol コンボボックスから選択するか、シンボル名称またはその一部を入力してください。
11	メッセージ	Shell version is too old.
	意味	シェルのバージョンが古いため本ツールを実行できません。
	ツールの処理	処理を中断します。
	ユーザの対処	本ツールの動作環境を確認してください。
12	メッセージ	Illegal file format '<Error File>'
	意味	オープン指定したファイルのフォーマットは、本ツールで使用する呼び出し情報ファイルのフォーマットではありません。
	ツールの処理	処理を中断します。
	ユーザの対処	指定したファイルに誤りがないか確認してください。 また、Ver.1.0 および Ver.1.1 Release 1 で作成した呼び出し情報ファイルを指定した場合も、本エラーメッセージを出力します。
13	メッセージ	Illegal input file name specified. Illegal output file name specified.
	意味	入力 (出力) ファイルが指定されていません。
	ツールの処理	処理を中断します。
	ユーザの対処	コマンドライン起動時、-list オプションのみが指定されていないか、あるいは -import オプションのファイル名に空白文字列が指定されていないか確認してください。
14	メッセージ	Internal error has occurred. '<Detail Message>'
	意味	本ツールの内部エラーが発生しました。
	ツールの処理	処理を中断します。
	ユーザの対処	再起動してください。再起動しても同じ現象が発生する場合は、本ツールをインストールし直してください。

15	メッセージ	Syntax error.
	意味	コマンドラインのシンタックスエラーです。
	ツールの処理	処理を中断します。
	ユーザの対処	コマンドライン起動時、複数の入力ファイル名または出力ファイル名を指定していないか確認してください。 オプションとして-import および-list オプション以外を指定していないか確認してください。
16	メッセージ	File already exists '<Path of file>'
	意味	指定したファイルはファイルリストに存在するため追加できません。
	ツールの処理	処理を中断します。
	ユーザの対処	追加したいファイル名を確認し、ファイルパスが間違っている場合は正しいファイルを選択してください。
17	メッセージ	File does not exist or File has illegal format in User Library Options...
	意味	指定したファイルは、User Library Options ダイアログボックスに存在しないか、または間違っ たフォーマットです。
	ツールの処理	このファイルを選択しません。
	ユーザの対処	特にありません。このファイルはツールが自動的に選択しません。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2022.7.16	—	初版発行 (CS+編)

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れしないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 $V_{IL}(\text{Max.})$ から $V_{IH}(\text{Min.})$ までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 $V_{IL}(\text{Max.})$ から $V_{IH}(\text{Min.})$ までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違えば、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
 2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
 3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
 4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
 5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
 6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通管制（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等
当社製品は、データシート等により高信頼性、Harsh environment向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
 7. あらゆる半導体製品は、外部攻撃からの安全性を100%保証されているわけではありません。当社ハードウェア/ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限られません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因してまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア/ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
 8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
 9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
 10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
 11. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
 12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものとしたします。
 13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
 14. 本資料に記載されている内容または当社製品についてご不明点がございましたら、当社の営業担当者までお問合せください。
- 注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。
- 注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。