

スタック領域の変更方法

RL78、78K0R用 Cコンパイラ CA78K0R

株式会社ルネサス ソリューションズ
ツールビジネス本部 ツール技術部

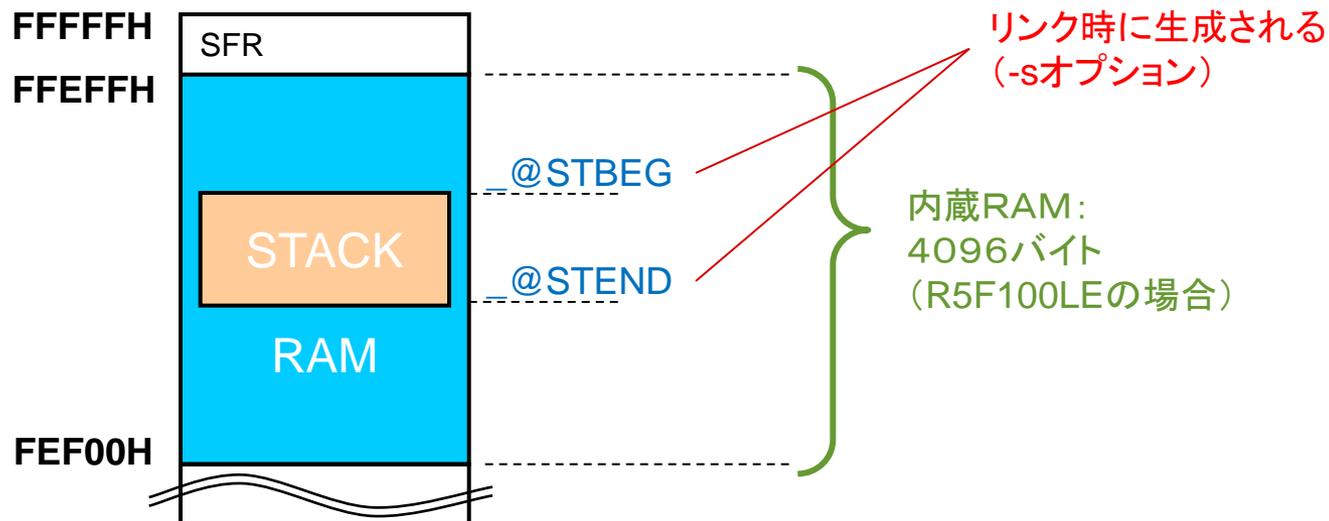
2014/6/20 Rev. 1.00

R20UT3042JJ0100

- デフォルトのスタック領域
- スタック領域を変更するには
- スタック領域用のメモリ領域を作成する方法
- スタックポインタ(SP)にアドレスを設定する方法
 - ソースファイルを登録する方法
 - オブジェクトファイルを登録する方法

デフォルトのスタック領域

- メモリ領域名RAMの中の最大空き領域(gap)にスタック領域が設定される



- デフォルトのスタートアップルーチンで、スタックポインタ(SP)に、上記領域の最上位アドレスを設定している

```
MOVW    SP,#LOWW __@STBEG ;SP <- stack begin address
```

スタック領域を変更するには

- **スタック領域用のメモリ領域を作成する方法(推奨)**
 - 他の変数が配置される可能性低
- **スタックポインタ(SP)にアドレスを設定する方法**
 - 他の変数が配置された領域と重なる可能性あり
 - スタートアップルーチンのソースファイル変更後、オブジェクトを作成する必要あり

スタック領域用のメモリ領域を作成する方法

スタック領域用のメモリ領域作成(1/6)

■ リンクディレクティブファイルで領域確保

- デフォルトのRAM領域を縮小
- 新たにスタック領域用のメモリ領域を作成

■ スタック領域の指定

- リンカのオプションで、新たに作成したメモリ領域をスタック領域として指定

■ リンクディレクティブファイルをCubeSuite+のプロジェクトに登録

- リンクディレクティブファイルをプロジェクトに登録

スタック領域用のメモリ領域作成(2/6)

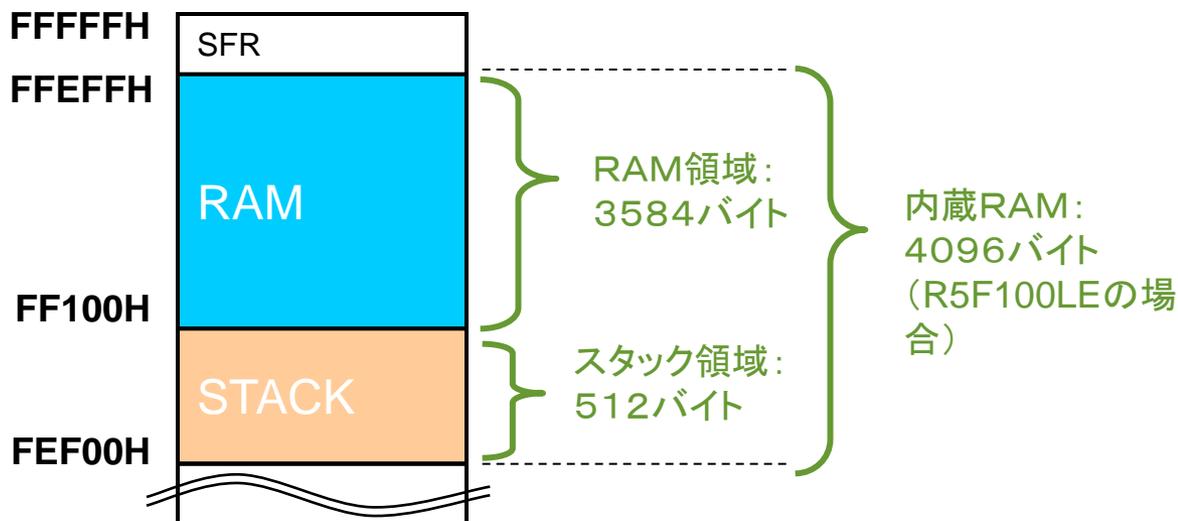
■ リンクディレクティブファイルで領域確保

(例)

```
; File Name is "rl78.dr"  
MEMORY   RAM       : ( OFF100H, OE00H)  
MEMORY   STACK     : ( OFEFO0H, 0200H)
```

RAMのサイズを削減する
(スタック領域を分削除したサイズ)

スタック領域用の定義



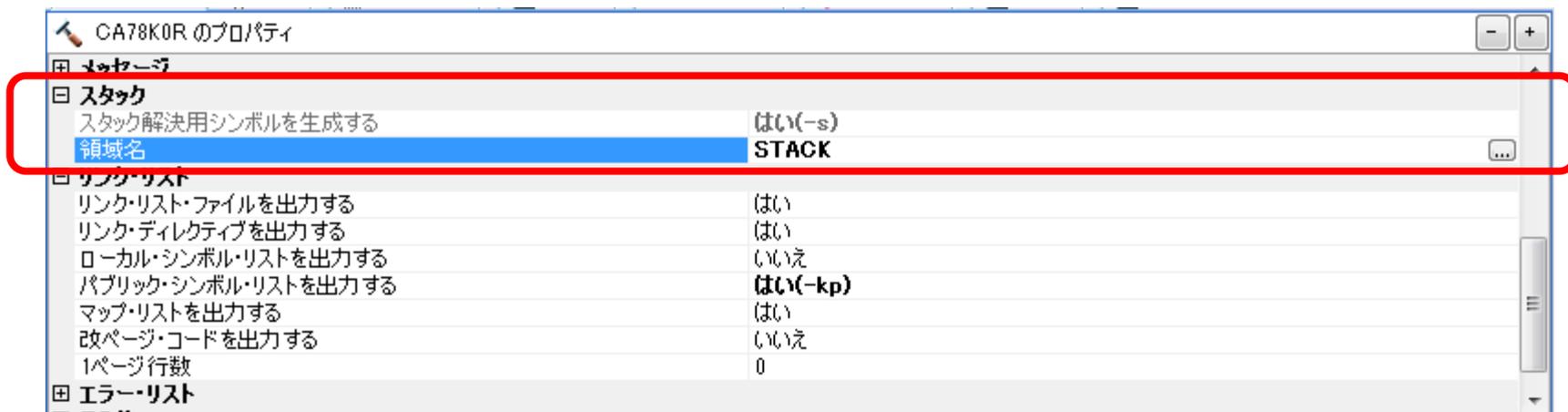
補足: セクションがメモリ領域STACKに配置されてしまう場合には、MERGE指定でメモリ領域RAMへの配置を指定をしてください。

スタック領域用のメモリ領域作成(3/6)

■ スタック領域の指定

● [リンクオプション]タブの「スタック」

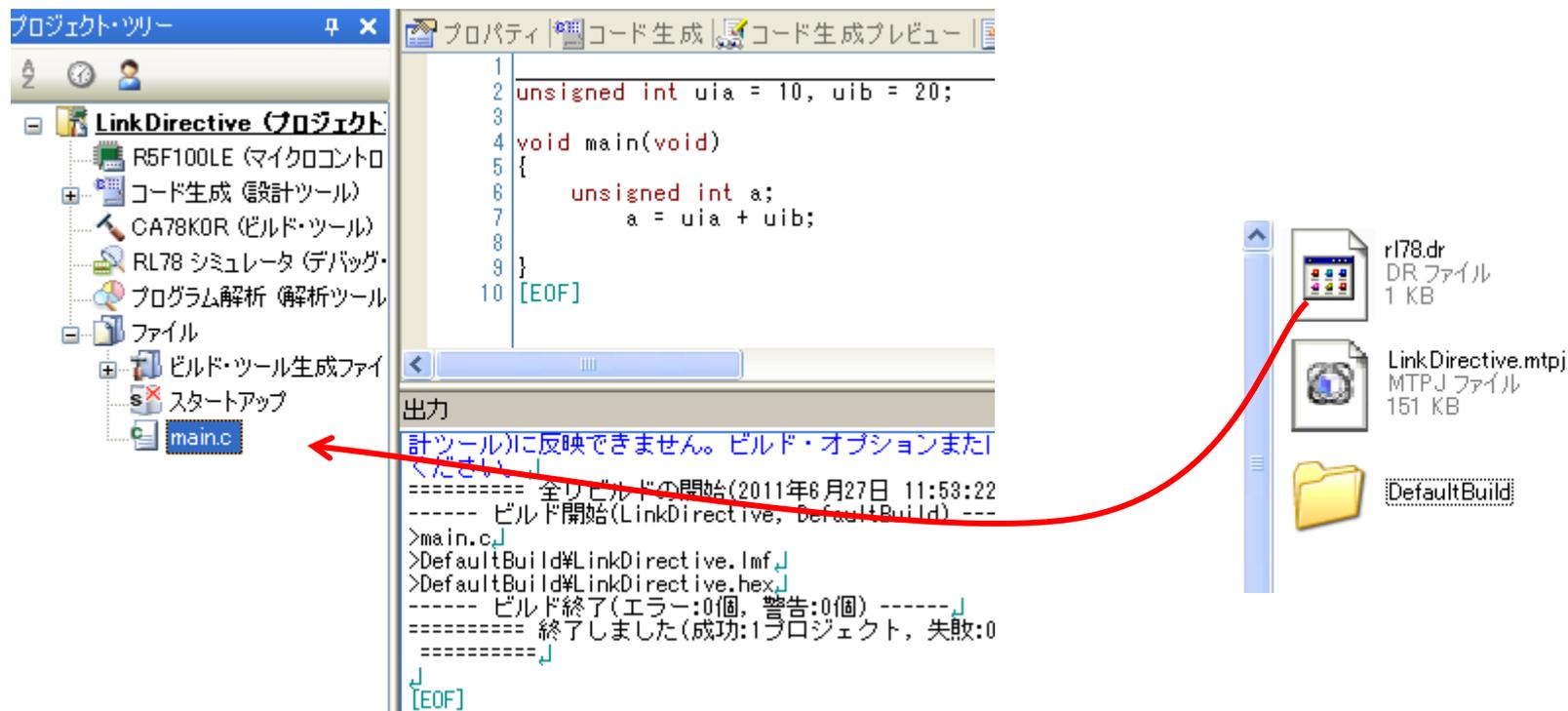
- 「スタック解決用シンボルを生成する」で「はい」を選択(デフォルト)
- 「領域名」を入力(リンクディレクティブファイルで新たに定義した領域名)



スタック領域用のメモリ領域作成(4/6)

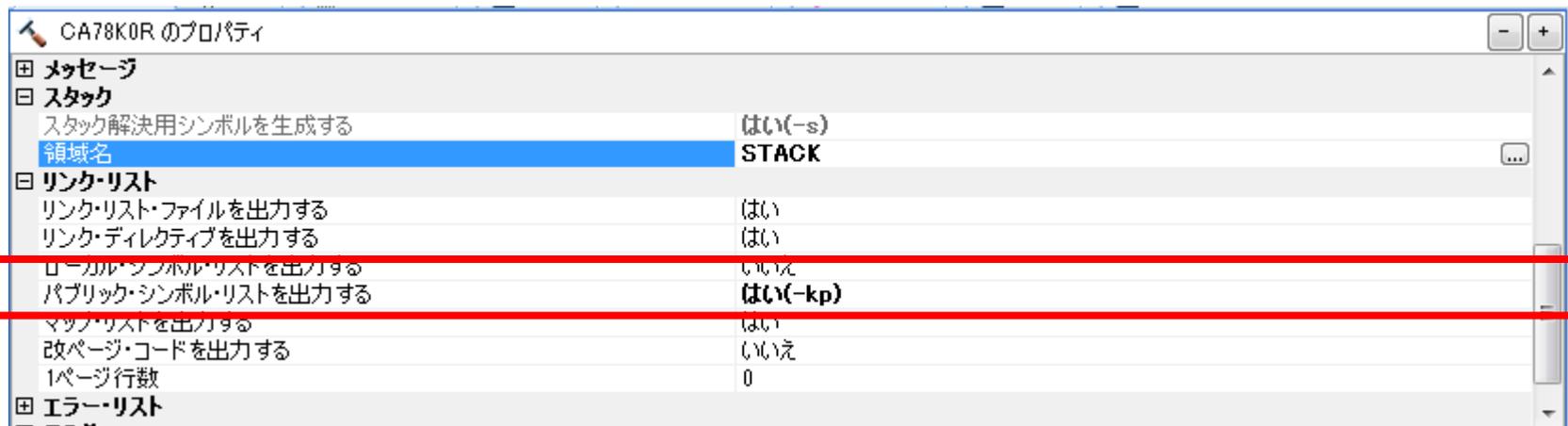
■ リンクディレクティブファイルをCubeSuite+ のプロジェクトに登録

- CubeSuite+ のプロジェクトツリーパネルに、リンクディレクティブファイルを、ドラッグ&ドロップすれば登録できます



スタック領域用のメモリ領域作成(5/6)

- リンク時に生成されるmapファイル上で、スタック用のシンボルを参照するため、パブリックシンボルの情報を出力
 - [リンクオプション]タブの「リンク・リスト」
 - 「パブリック・シンボル・リストを出力する」で「はい」を選択



スタック領域用のメモリ領域作成(6/6)

- ビルド後、プロジェクトツリーパネルの「ビルドツール生成ファイル」カテゴリのmapファイルを確認

The image shows a linker map file (LinkDirective.map) with two windows. The left window displays memory layout details, and the right window displays a public symbol list. Red boxes highlight key information, and yellow callouts provide context.

Memory Layout (Left Window):

- Line 7: MEMORY=STACK
- Line 8: BASE ADDRESS=FEF00H SIZE=00200H
- Line 10: MEMORY=RAM
- Line 11: BASE ADDRESS=FF100H SIZE=00E00H

Public Symbol List (Right Window):

ADDR	VALUE	NAME
000CEH		_@cstart
0017AH		_@cend
FEF98H		_@BRKADR
FEF9AH		_@MEMTOP
FEFBAH		_@MEMBTM
FEF00H		_@FNCTBL
0017AH		_main
FF100H		_@STBEG
00183H		_hdwinit
FF100H		_?INITL
FF100H		_?INIS
FEFBEH		_@STEND

Annotations:

- Yellow callout (top right): **スタック領域 最上位アドレス(_@STBEG)** (Stack area, highest address (_@STBEG))
- Yellow callout (bottom left): **スタック領域 最下位アドレス(_@STEND)** (Stack area, lowest address (_@STEND))

スタックポインタ(SP)にアドレスを設定する方法

スタックポインタ(SP)へのアドレス設定

- スタックポインタ(SP)を設定する処理は、スタートアップルーチンにあります
- そのため、アドレスを設定するように変更するには、スタートアップルーチンの変更が必要です
- スタートアップルーチンの変更方法は次の2通りがあります
 - プロジェクトツリーパネルに、スタートアップルーチンの**ソースファイル**を登録
 - スタックポインタの設定以外に、セクションの初期化に必要なシンボルを編集しないのであれば、この方法を使用可能
 - プロジェクトツリーパネルに、スタートアップルーチンの**オブジェクトファイル**を登録
 - 事前に**batファイル**でオブジェクトファイルの構築が必要
 - スタックポインタの設定以外に、セクションの初期化に必要なシンボルを編集する場合には、この方法でスタートアップルーチンの変更が必要

スタックポインタ(SP)にアドレスを設定する方法 (ソース指定)

スタックポインタ(SP)へのアドレス設定(ソース指定)(1/4)

- システムフォルダからの編集するスタートアップのソースファイル、および、ヘッダファイルをコピー
 - CubeSuite+ のインストールフォルダの「¥CubeSuite+¥CA78K0R¥V1.50¥Src¥cc78k0r¥src」から任意のフォルダにコピーする
 - スタートアップルーチンのソースファイル
 - 対象となるスタートアップルーチンのソースファイル名は次ページ参照
 - スタートアップルーチンでインクルードされているヘッダファイル(上記フォルダに存在しています)
 - def.inc
 - macro.inc

スタックポインタ(SP)へのアドレス設定(ソース指定)(2/4)

■ スタートアップルーチンのソースファイルの種類

ソースファイル名	説明
cstart.asm	通常プロジェクト(標準ライブラリ使用時)
cstartn.asm	通常プロジェクト(標準ライブラリ未使用時)
cstartb.asm	ブート用プロジェクト(標準ライブラリ使用時)
cstartbn.asm	ブート用プロジェクト(標準ライブラリ未使用時)
cstarte.asm	フラッシュ用プロジェクト(標準ライブラリ使用時)
cstarten.asm	フラッシュ用プロジェクト(標準ライブラリ未使用時)

スタックポインタ(SP)へのアドレス設定(ソース指定)(3/4)

■ スタートアップルーチンのソースファイルのプロジェクトへの登録

プロジェクトツリーパネルの「スタートアップ」を選択し、右クリックして「既存のファイルを追加」を選択しファイルを指定する

登録完了するとファイルが表示される

登録完了すると、[コンパイルオプション]タブの「スタートアップ」の「標準のスタートアップを使用する」は、「いいえ」になる

スタートアップ
標準のスタートアップを使用する はい いいえ

スタックポインタ(SP)へのアドレス設定(ソース指定)(4/4)

■ スタートアップルーチンのソースファイルの編集

- コピー先のスタートアップルーチンのソースファイル `cstart***.asm`をエディタで編集

```
MOVW    SP,#OF100H           ;SP <- stack begin address (0FF100H)
```

スタックポインタ(SP)にアドレスを設定する方法 (オブジェクト指定)

スタックポインタ(SP)へのアドレス設定(オブジェクト指定)(1/6)

- システムフォルダからのbatファイルに関するファイルのコピー
 - CubeSuite+ のインストールフォルダの「¥CubeSuite+¥CA78K0R¥V1.50¥Src¥cc78k0r」を任意のフォルダにコピーする
- スタートアップルーチンのファイルの編集
 - コピー先の「cc78k0r¥src¥cstart***.asm」をエディタで編集
 - 編集するスタートアップルーチンのソースファイル名は次ページ参照

```
MOVW    SP,#OF100H           ;SP <- stack begin address(0FF100H)
```

スタックポインタ(SP)へのアドレス設定(オブジェクト指定)(2/6)

■ スタートアップルーチンのソースファイルの種類

ソースファイル名	説明
cstart.asm	通常プロジェクト(標準ライブラリ使用時)
cstartn.asm	通常プロジェクト(標準ライブラリ未使用時)
cstartb.asm	ブート用プロジェクト(標準ライブラリ使用時)
cstartbn.asm	ブート用プロジェクト(標準ライブラリ未使用時)
cstarte.asm	フラッシュ用プロジェクト(標準ライブラリ使用時)
cstarten.asm	フラッシュ用プロジェクト(標準ライブラリ未使用時)

スタックポインタ(SP)へのアドレス設定(オブジェクト指定)(3/6)

- スタートアップルーチンの再構築を行う
- Windowsのコマンドプロンプトを起動する
- コマンドプロンプト上で次の操作をする
 - コピー先のフォルダに移動する

(例) `cd C:¥sample¥cc78k0r¥bat`

コピー先のフォルダ

- (例) ● PATHを追加する(batファイル中のra78k0r.exeを実行するため)

`set path=%PATH%;C:¥Program Files¥Renesas Electronics¥CubeSuite+¥CA78K0R¥V1.60¥bin`

CubeSuite+の
インストールフォルダ

CA78K0Rのバージョン

スタックポインタ(SP)へのアドレス設定(オブジェクト指定) (4/6)

- batファイルの「mkstup」を実行する
 - 次の形式で実行してください
 - mkstup 品種指定名 DIFファイルパス指定オプション

(例)

```
mkstup f100le -y"C:¥Program Files¥Renesas Electronics¥CubeSuite+¥Device¥RL78¥Devicefile"
```

品種指定名

CubeSuite+の
インストールフォルダ

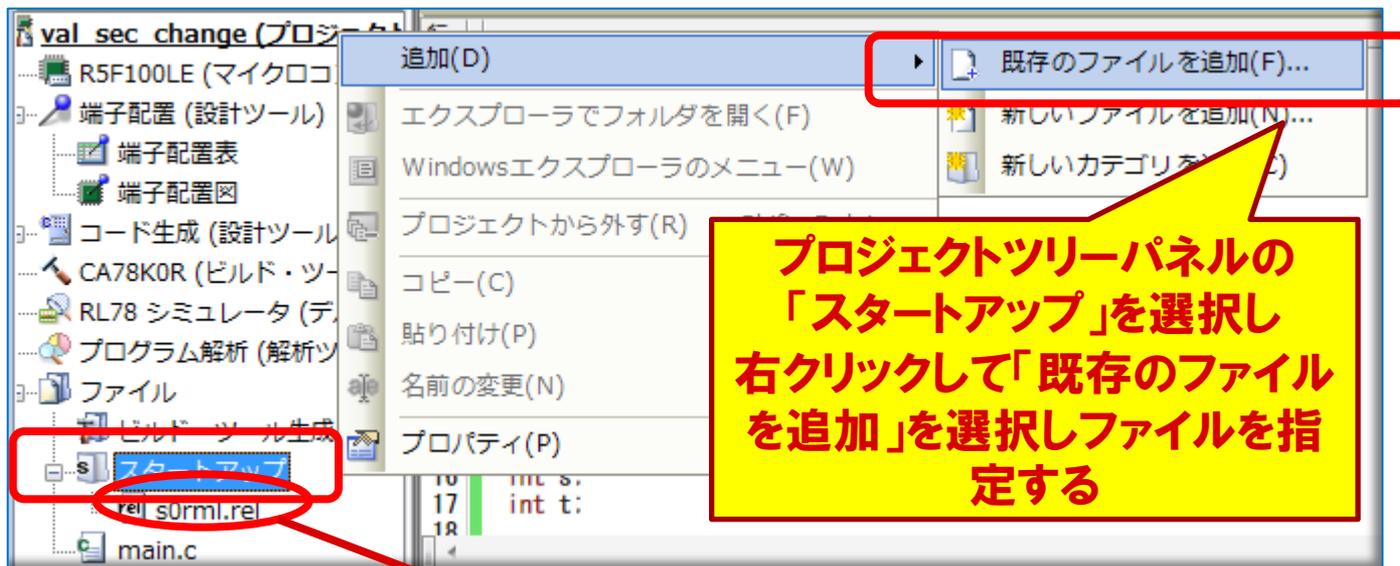
RL78
(78K0Rの場合は78K0R)

- 「¥cc78k0r¥lib」にスタートアップルーチンのファイルが生成される

スタックポインタ(SP)へのアドレス設定(オブジェクト指定) (5/6)

■ スタートアップのプロジェクトへの登録

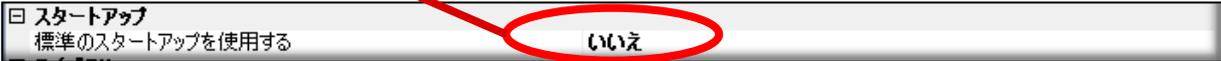
- 登録するスタートアップルーチンのオブジェクトファイル名は次ページ参照



プロジェクトツリーパネルの「スタートアップ」を選択し右クリックして「既存のファイルを追加」を選択しファイルを指定する

登録完了するとファイルが表示される

登録完了すると、[コンパイルオプション]タブの「スタートアップ」の「標準のスタートアップを使用する」は、「いいえ」になる



スタックポインタ(SP)へのアドレス設定(オブジェクト指定)(6/6)

■ スタートアップ・ファイルの命名規則

- lib78k0r¥s0r<model> <lib> <flash>.rel

– <model>

m : ミディアム・モデル(スモール・モデル兼用)

l : ラージ・モデル

– <lib>

なし : 標準ライブラリ固定領域を使用しない場合

l : 標準ライブラリ固定領域を使用する場合

– <flash>

なし : 通常用

b : ブート領域用

e : フラッシュ領域用

RENEASAS

ルネサス ソリューションズ株式会社

© 2014 Renesas Solutions Corp.