

# 変数セクションの追加方法

RL78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

R20UT3043JJ0100

- **変数のセクションの追加するには**
- **変数セクションの初期化**
- **変数セクションの初期化のイメージ**
- **Cソース上でのセクション指定**
- **Cソース上でのセクション指定の注意**
- **スタートアップルーチン、rom.asmの編集の準備**
- **rom.asmの変更**
- **スタートアップルーチンの変更**
- **スタートアップルーチン、ライブラリの再構築**
- **スタートアップルーチン、ライブラリのプロジェクト登録**

# 変数のセクションの追加するには(1/2)

- Cソース上でのセクション指定を追加
  - #pragma section で変数のセクション名を変更
- rom.asmの変更
  - デフォルトのセクション用の定義しか入っていないので追加が必要
    - セクション最後シンボルの追加
- スタートアップルーチンの変更
  - デフォルトのセクションの処理しか入っていないので追加が必要
    - セクション最後シンボルのEXTRN宣言
    - セクション先頭シンボル定義の追加
    - コピールーチンの追加
- スタートアップルーチンのオブジェクト、ライブラリの再構築
- スタートアップルーチン、ライブラリのプロジェクト登録

## 変数のセクションの追加するには(2/2)

- Cソース上でのセクション指定を追加
  - #pragma section で変数のセクション名を変更
- rom.asmの変更
  - デフォルトのセクション用の定義しか入っていないので追加が必要
    - セクション最後シンボルの追加
- スタートアップルーチンの変更
  - デフォルトのセクションの処理しか入っていないので追加が必要
    - セクション最後シンボルのEXTRN宣言
    - セクション先頭シンボル定義の追加
    - コピールーチンの追加
- スタートアップルーチンのオブジェクト、ライブラリの再構築
- スタートアップルーチン、ライブラリのプロジェクト登録

**次の条件をすべて満たす場合、上記の灰色文字部分は必要ありません。**

- ◆ 初期値なし変数のみ(初期値あり変数セクションではない)
- ◆ 初期値なしの変数領域の0初期化が不要(値が不定で問題ない)

# 変数セクションの初期化

- 変数のセクションを追加した際には、そのセクションの初期化が必要です
  - 初期値なし変数: 0初期化
  - 初期値あり変数: 初期値のコピー
  
- この初期化の処理はスタートアップルーチンで行います
  - 初期化の際には、各セクションの**先頭と最後のアドレス**が必要  
これらのシンボルを使用して、0初期化、または、初期値のコピー
  - 先頭のアドレス
    - スタートアップルーチンで、各セクションの**先頭のシンボル**を定義
  - 最後のアドレス
    - rom.asm(ライブラリで定義)で、各セクションの**最後のシンボル**を定義
  - リンク時に次の順番でリンク(CubeSuite+では自動で設定されます)
    - スタートアップルーチン: 先頭にリンク
    - rom.asm(ライブラリで定義): 最後にリンク

# 変数セクションの初期化のイメージ

スタートアップルーチン

セクション先頭シンボル

ソースn.c

ユーザ変数シンボル  
:

ソースn+1.c

ユーザ変数シンボル  
:  
:  
:

rom.asm(ライブラリ中に存在)

セクション最後シンボル

変数のセクション

セクション先頭シンボル

ユーザ変数シンボル  
:

ユーザ変数シンボル  
:  
:  
:

セクション最後シンボル

スタートアップルーチンで定義することにより先頭にリンク

スタートアップルーチンで  
“先頭”から“最後”までを  
初期化

```
LINIT3:  
MOV ES,#HIGHW _@RLINIT  
MOV A,ES:[HL]  
MOV ES,#HIGHW _@INITL  
MOV ES:[DE],A  
INCW HL  
INCW DE  
LINIT4:  
MOVW AX,HL  
CMPW AX,#LOWW _?RLINIT  
BNZ $LINIT3
```

rom.asm(ライブラリ)で定義することにより  
ユーザセクションより後ろ(最後)にリンク

リンク

# Cソース上でのセクション指定(1/3)

## ■ #pragma section を使用

- デフォルトで出力されるセクションの名前を変更する
- 記述形式

- #pragma section **コンパイラ出力セクション名** **変更セクション名**
- **コンパイラ出力セクション名は次ページ参照**

## ■ 初期値あり変数の場合

- 初期値のコピー元のセクションの名前の変更も必要

(例)

```
#pragma section @@INIT UINIT  
#pragma section @@R_INIT R_UINIT  
#pragma section @@DATA UDATA  
  
int a;  
int b=0xff;  
  
#pragma section @@INIT @@INIT  
#pragma section @@R_INIT @@R_INIT  
#pragma section @@DATA @@DATA
```

ユーザセクション名に変更

デフォルトのセクション名に戻す

## Cソース上でのセクション指定(2/3)

### ■ コンパイラ出力セクション名

- 通常プロジェクト、ブート用プロジェクトのセクション名

説明	変数セクション名	初期値格納セクション名
near変数(初期値あり)	@@INIT	@@R_INIT
far変数(初期値あり)	@@INITL	@@RLINIT
sreg変数(初期値あり)	@@INIS	@@R_INIS
near変数(初期値なし)	@@DATA	—
far変数(初期値なし)	@@DATA_L	—
sreg変数(初期値なし)	@@DATS	—
boolean型変数, bit型変数(初期値なし)	@@BITS	—



## Cソース上でのセクション指定(3/3)

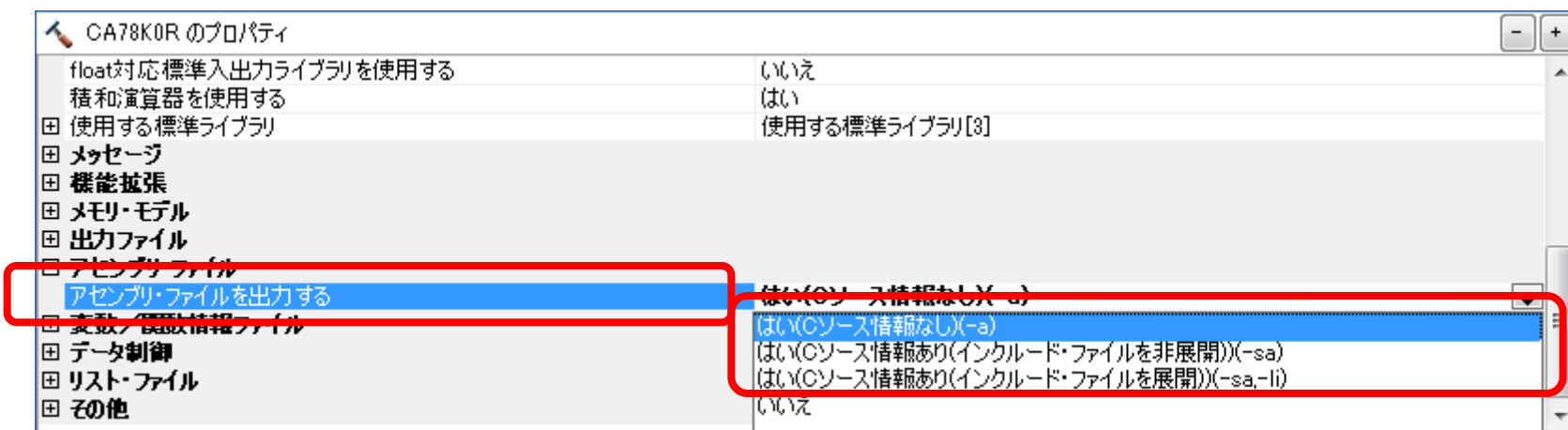
### ■ コンパイラ出力セクション名

#### ● フラッシュ用プロジェクトのセクション名

説明	変数セクション名	初期値格納セクション名
near変数(初期値あり)	@EINIT	@ER_INIT
far変数(初期値あり)	@EINITL	@ERLINIT
sreg変数(初期値あり)	@EINIS	@ER_INIS
near変数(初期値なし)	@EDATA	—
far変数(初期値なし)	@EDATAL	—
sreg変数(初期値なし)	@EDATS	—
boolean型変数, bit型変数(初期値なし)	@EBITS	—

## Cソース上でのセクション指定の注意

- コンパイラで直接オブジェクト(\*.rel)を生成できない場合には、次のワーニングが出力されます。
  - CC78K0R warning W0837: Output assembler source file , not object file
- コンパイラのオプションで、アセンブリ・ファイルを出力するように指定してください。
  - [コンパイルオプション]タブの「アセンブリ・ファイル」
    - 「アセンブリ・ファイルを出力」で「はい」を含むものを選択



# スタートアップルーチン、rom.asmの編集の準備

- 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
  
- ファイルの編集
  - コピー先のファイルを編集する

# rom.asmの変更

## ■ セクション最後のシンボル名の定義

- コピー先の「cc78k0r¥src¥rom.asm」をエディタで編集
- **モジュール名 @rom は変更しない**
- Cソースで指定したセクションを作成し、セクションの最後のシンボルを定義し、他ファイルから参照できるようにPUBLIC宣言する
- 元のセクション定義をコピーして編集すると、再配置属性はそのまま使用可

(例)

変更しないでください

PUBLIC宣言

```
NAME      @rom
PUBLIC    _R_UINIT_E,_UINIT_E,_UDATA_E
R_UINIT   CSEG      UNIT64KP
_R_UINIT_E:
UINIT     DSEG      BASEP
_UINIT_E:
UDATA     DSEG      BASEP
_UDATA_E:
```

再配置属性

セクション名

最後のシンボル

# スタートアップルーチンの変更(1/5)

## ■ 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の通り

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)

## スタートアップルーチンの変更(2/5)

### ■ セクション最後のシンボルのEXTRN宣言

- rom.asmで定義したセクションの最後のシンボルを、初期化処理で使用できるように、EXTRN宣言する

(例)

```
EXTRN    _R_UINIT_E, _UINIT_E, _UDATA_E:
```

# スタートアップルーチンの変更(3/5)

## ■ セクション先頭のシンボル名の定義

- Cソースで指定したセクションを作成し、セクションの先頭のシンボルを定義する
- セクションの最後シンボルの定義をコピーして編集すると、シンボル名のみ変更で済む

(例)

```
R_UINIT CSEG UNIT64KP  
_R_UINIT_S:  
UINIT DSEG BASEP  
_UINIT_S:  
UDATA DSEG BASEP  
_UDATA_S:
```

再配置属性

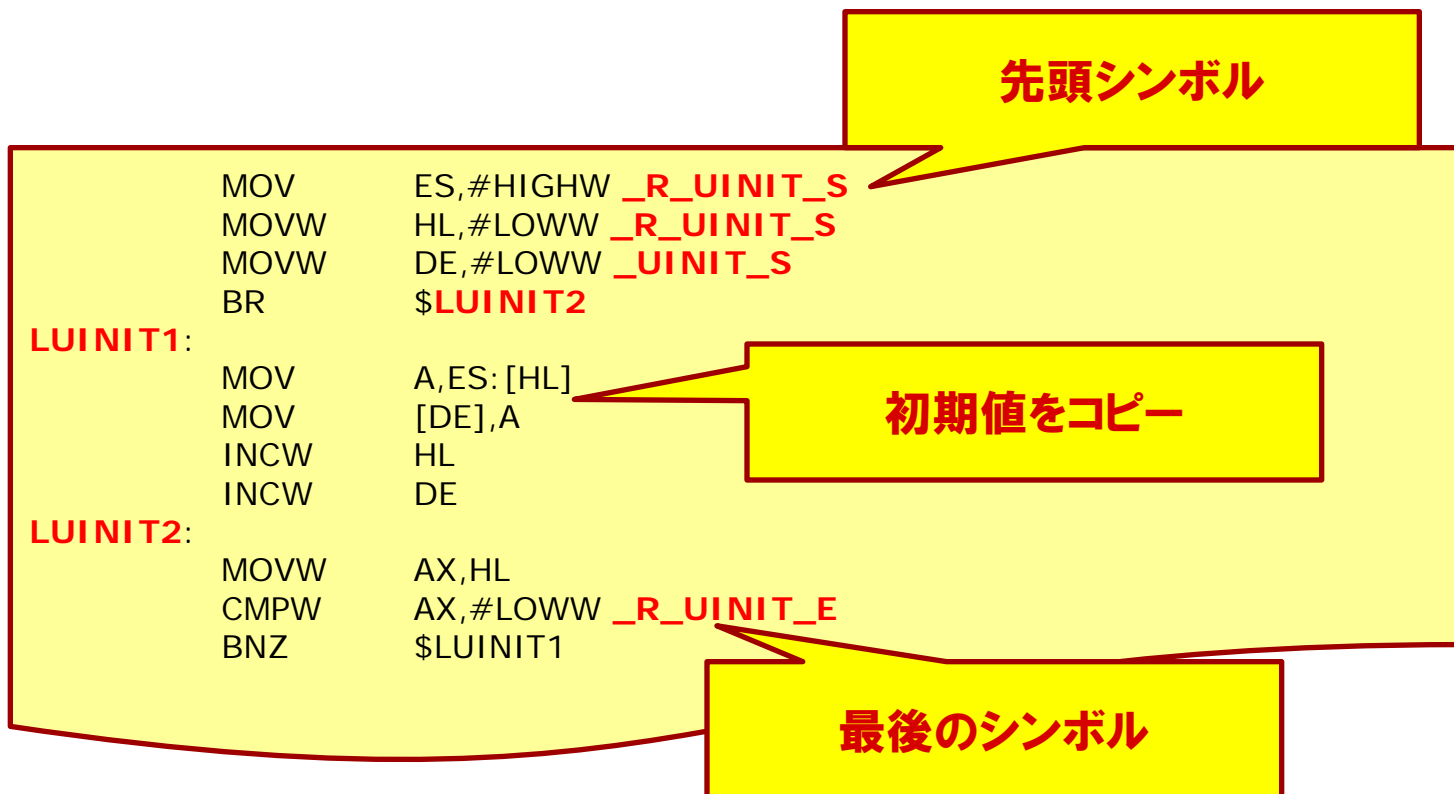
セクション名

先頭シンボル

# スタートアップルーチンの変更(4/5)

- 初期値をコピーするルーチンを追加(初期値あり変数の場合)
  - 元のセクションの初期化するルーチンをコピーして変数することを推奨

(例)



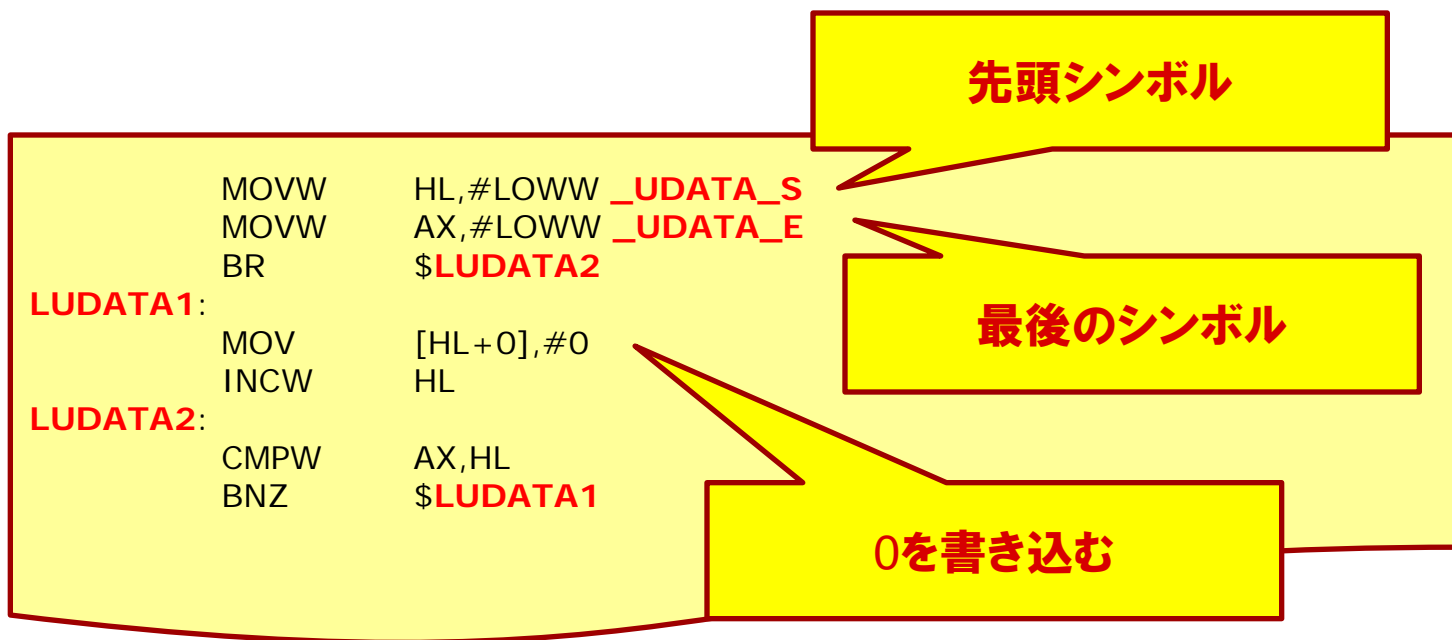


# スタートアップルーチンの変更(5/5)

## ■ 0で初期化するルーチンを追加(初期値なし変数の場合)

- sreg変数は、スタートアップルーチン上で、saddr領域を0初期化しているので処理を追加する必要なし
- 元のセクションの0初期化するルーチンをコピーして変数することを推奨

(例)



# スタートアップルーチン、ライブラリの再構築(1/3)

- 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

## スタートアップルーチン、ライブラリの再構築(2/3)

- 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

## スタートアップルーチン、ライブラリの再構築(3/3)

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

(例)

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

品種指定名

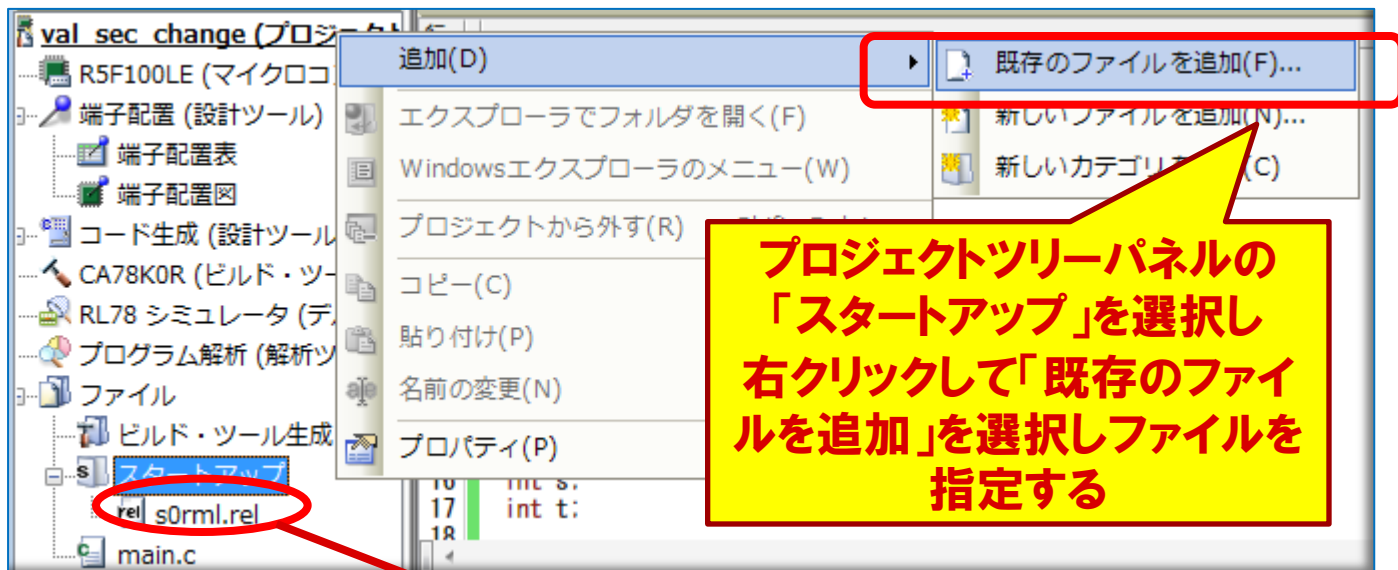
CubeSuite+の  
インストールフォルダ

RL78  
(78K0Rの場合は78K0R)

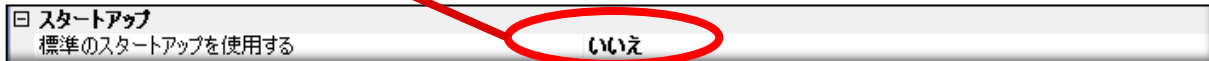
- 「¥cc78k0r¥lib」にライブラリのファイルが生成される

# スタートアップルーチン、ライブラリのプロジェクト登録(1/4)

## ■ スタートアップの登録



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



# スタートアップルーチン、ライブラリのプロジェクト登録(2/4)

## ■ 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

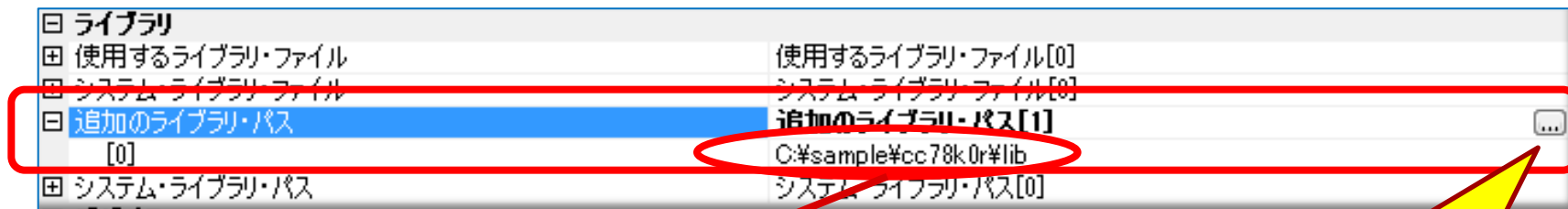
e : フラッシュ領域用

# スタートアップルーチン、ライブラリのプロジェクト登録(3/4)

## ■ ライブラリパスの指定

- [リンクオプション]タブの「ライブラリ」

- 「追加のライブラリ・パス」に新たに作成したライブラリのパスを指定



登録完了するとファイルが表示される

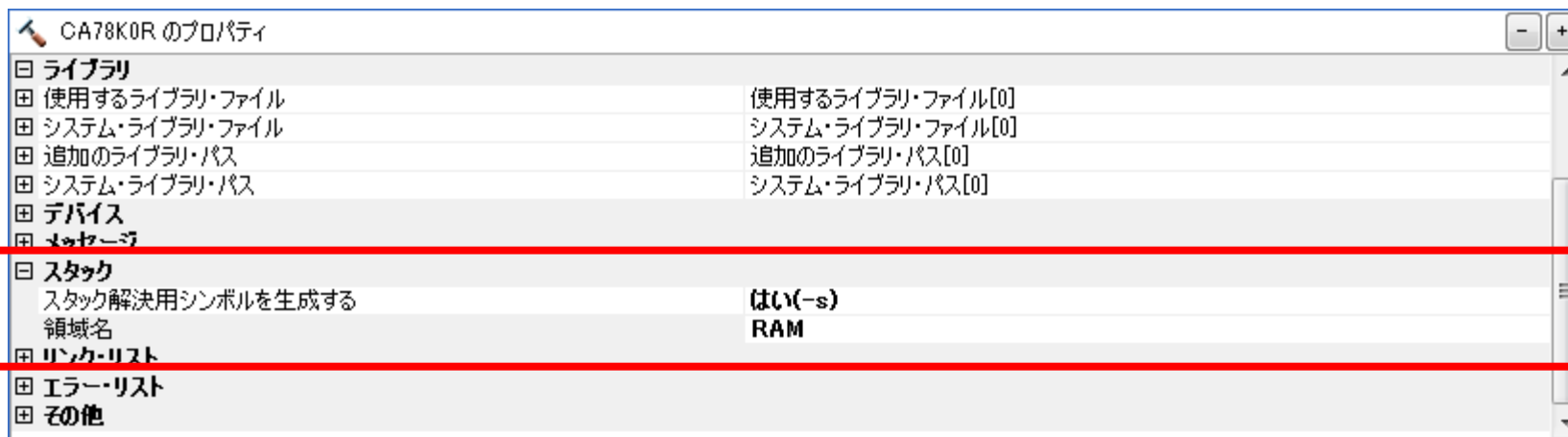
ボタンを押すことによりパスを入力するダイアログが開く

# スタートアップルーチン、ライブラリのプロジェクト登録(4/4)

## ■ スタック解決用のシンボルの指定

### ● [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」を「はい(-s)」を選択
- 領域名にRAMに配置される領域名を指定
  - デフォルトのリンクディレクティブでは、領域名はRAMになります





**RENEASAS**

**ルネサス ソリューションズ株式会社**

© 2014 Renesas Solutions Corp.