

## RL78 ソフトウェア置き換えガイド

### スマート・コンフィグレータで生成されたソフトウェアの移植 (CS+, e2 studio, IAR)

---

#### 要旨

本アプリケーションノートでは、RL78 スマート・コンフィグレータ (SC) で生成されたソフトウェアの移植について説明します。例として RL78/G23 用サンプルコードを RL78/G15 用コードに置き換える手順を説明します。

#### 対象デバイス

RL78/G15

RL78/G22

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

## 目次

1. 概要	4
1.1 移植対象	4
1.2 サンプルコードの構成	5
1.3 移植方法	10
2. 移植手順	12
2.1 事前準備	12
2.2 プロジェクトのコピー	13
2.3 プロジェクトの読み込み	13
2.3.1 CS+ for CC の場合	13
2.3.2 e <sup>2</sup> studio の場合	14
2.3.3 IAR EW for Renesas RL78 の場合	16
2.4 コンパイラのバージョン変更	17
2.4.1 CS+ for CC の場合	17
2.4.1 e <sup>2</sup> studio の場合	18
2.4.2 IAR EW for Renesas RL78 の場合	18
2.5 コード変更の確認	19
2.6 リソースの変更	20
2.6.1 CS+ for CC での SC 起動手順	21
2.6.2 e <sup>2</sup> studio での SC 起動手順	22
2.6.3 IAR EW for Renesas RL78 での SC 起動手順	23
2.6.4 リソースの変更	24
2.7 SC のレポート生成	26
2.8 デバイスの変更	28
2.8.1 CS+ for CC の場合	28
2.8.2 e <sup>2</sup> studio の場合	31
2.8.3 IAR EW for Renesas RL78 の場合	34
2.9 SC 設定の移行結果確認	37
2.10 設定／リソースの変更	40
2.10.1 クロック設定	40
2.10.2 システム設定	42
2.10.3 割り込み設定	43
2.10.4 ソフトウェアコンポーネント設定	44
2.10.5 端子設定	50
2.11 SC の設定保存	63
2.12 コード生成	63
2.13 コード編集	64
2.14 移植先プロジェクトのビルド	69
3. 移植例	70
3.1 事前準備	70
3.2 プロジェクトのコピー	73
3.3 プロジェクトの読み込み	73
3.4 コンパイラのバージョン変更	73
3.5 コード変更の確認	73

3.6	リソースの変更 .....	74
3.7	SC レポートの生成.....	74
3.8	デバイスの変更 .....	74
3.9	SC 設定の移行結果確認 .....	74
3.10	設定／リソースの変更.....	74
3.10.1	クロック設定 .....	74
3.10.2	システム設定 .....	76
3.10.3	割り込み設定 .....	76
3.10.4	ソフトウェアコンポーネント設定 .....	78
3.10.5	端子設定 .....	79
3.11	SC の設定保存.....	81
3.12	コード生成.....	81
3.13	コード編集.....	81
3.14	移植先プロジェクトのビルド.....	82
4.	サンプルコード .....	84
5.	参考ドキュメント .....	84

## 1. 概要

### 1.1 移植対象

本アプリケーションノートでは、スマート・コンフィグレータ (SC) が生成したプログラム (デバイス・ドライバ) を他の RL78 製品用に置き換える手順について説明します。

ただし、移植先の RL78 製品が同じ周辺機能を搭載していない場合は、デバイス・ドライバの置き換えができません。また、同じ周辺機能を搭載していても、サンプルコードで使用している周辺機能の動作モードが移植先の RL78 製品にない場合などは置き換えできません。置き換えができない場合は、移植先の RL78 製品用プログラムを新規に開発する必要があります。

表 1-1 に移植先対象の RL78 製品を示します。

表 1-1 移植先対象の RL78 製品

RL78/G1x	RL78/G15
RL78/G2x	RL78/G22

表 1-2 プログラム移植判定

移植先の RL78 製品	移植判定
移植元と同じ周辺機能が搭載されていない。	不可 (新規プログラム開発)
移植元と同じ周辺機能があり、 移植元と同じチャンネルがある。	可
移植元と同じ周辺機能があるが、 移植元と同じチャンネルがない。	チャンネル変更できれば、可
移植元と同じ周辺機能があるが、 移植元と同じ動作モードがない。	不可 (新規プログラム開発)
移植元と動作電圧が異なる。	動作電圧に応じた設定に変更できれば、可 <sup>注</sup>
移植元と動作周波数が異なる。	動作周波数に応じた設定に変更できれば、可 <sup>注</sup>
移植元で使用しているクロックソースがない。	不可 (新規プログラム開発)

注. 移植先の RL78 製品の電気的特性を満たすように回路設計してください。

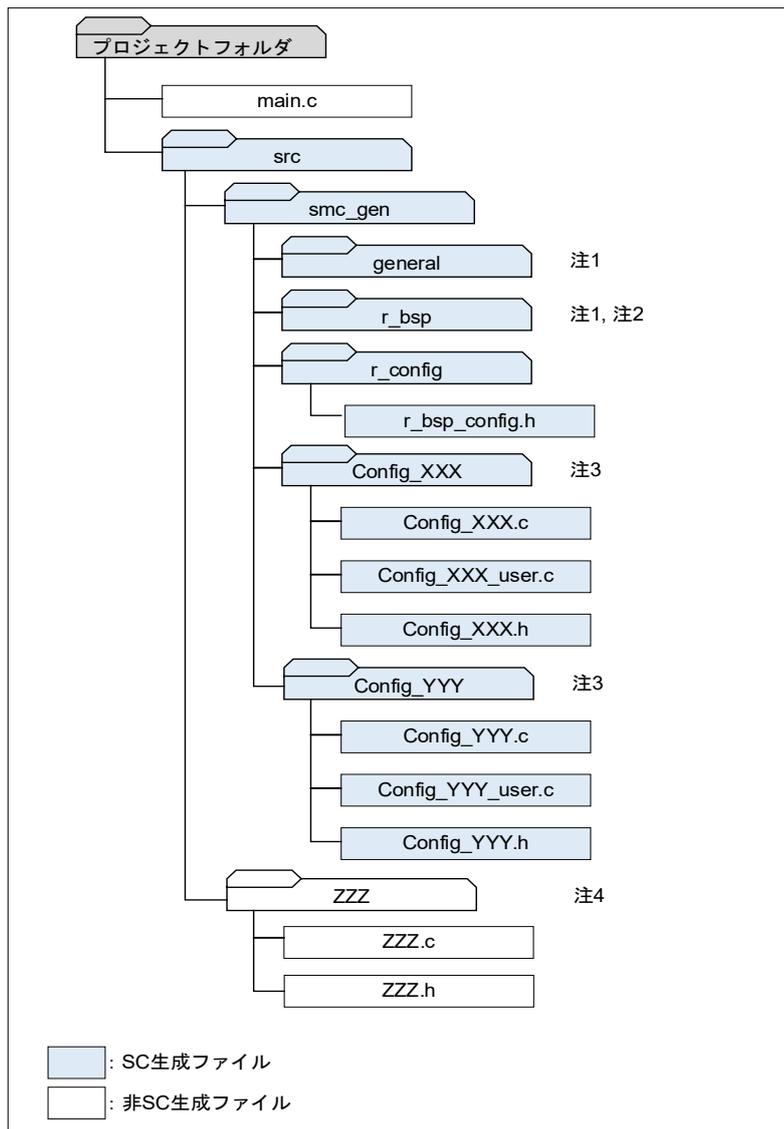
## 1.2 サンプルコードの構成

RL78 製品のサンプルコードは、スマート・コンフィグレータ (SC) が生成するファイル (SC 生成ファイル) とそれ以外のファイル (非 SC 生成ファイル) で構成されています。SC 生成ファイル内には、プロジェクトの対象 RL78 製品に合わせたボードサポートパッケージ (BSP) と周辺機能の設定コードがあります。

図 1-1 にサンプルコードのプロジェクト・フォルダ構成例を示し、表 1-3 に SC 生成フォルダおよびファイルの概要を示します。

図 1-3～図 1-8 にデバイスの CPU リセット発生から main 関数までの処理イメージを示します。

図 1-1 サンプルコードのプロジェクト例



注 1. 各フォルダ以下に生成されるファイルの記載を省略しています。

注 2. ボードサポートパッケージ (BSP) です。

注 3. XXX, YYY には、周辺機能名を表す省略文字が入ります。サンプルコードにより存在するフォルダ数は異なります。

注 4. ZZZ は任意の文字です。ZZZ で示すフォルダ/ファイルは、サンプルコードにより存在しない、または複数存在する場合があります。

表 1-3 SC 生成フォルダおよびファイルの概要

フォルダ名	ファイル名	内容
general	(省略)	SC 生成ファイル内で共通で使用するマクロ定義および Typedef 定義
r_bsp	(省略)	ボードサポートパッケージ (BSP) 注1
r_config	r_bsp_config.h	プラットフォームの設定 注1 (BSP で提供されるスタートアップ、BSP の API 関数が参照するファイル)
Config_XXX 注2, 3	Config_XXX.c Config_XXX_user.c Config_XXX.h	周辺機能の初期化、動作開始、動作停止の関数定義 周辺機能の割り込み関数 周辺機能用マクロ定義および Typedef 定義
ZZZ 注4	(省略)	非 SC 生成ファイル (サンプルコードで必要な処理関数を記載)

注 1. ボードサポートパッケージ (BSP) およびプラットフォームの設定の詳細は、RL78 ファミリ ボードサポートパッケージモジュール アプリケーションノート (R01AN5522) を参照してください。

注 2. スマート・コンフィグレータ (SC) が生成するファイルおよび関数の詳細は、スマート・コンフィグレータ RL78 API レファレンス編 ユーザーズマニュアル (R20UT4852) を参照してください。

注 3. XXX には、周辺機能名を表す省略文字が入ります。

注 4. ZZZ は任意の文字です。

備考. SC 生成ファイル内のコードは、SC の設定内容に従って生成されるため、基本的に変更せずに使用します。各ファイル内にあるユーザコード記述用コメント内にはユーザコードを追記することができます。このコメント内に記述したコードは、SC で再度コード生成しても上書きされずに保護されます。

図 1-2 ユーザコード用コメント例

```

#include
=====
#include "r_cg_macrodriver.h"
#include "r_cg_userdef line.h"
#include "Config_TAU0_0.h"
/* Start user code for include. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */

Pragma directive
=====
#pragma interrupt r_Config_TAU0_0_channel0_interrupt(vect=INTTMO0)
/* Start user code for pragma. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */

Global variables and functions
=====
/* Start user code for global. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */

=====
* Function Name: R_Config_TAU0_0_Create_UserInit
* Description : This function adds user code after initializing the TAU0 channel0.
* Arguments : None
* Return Value : None
=====
void R_Config_TAU0_0_Create_UserInit(void)
/* Start user code for user init. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */

```

図 1-3 CPU リセット発生から main 関数までの処理イメージ : スタートアップ関数

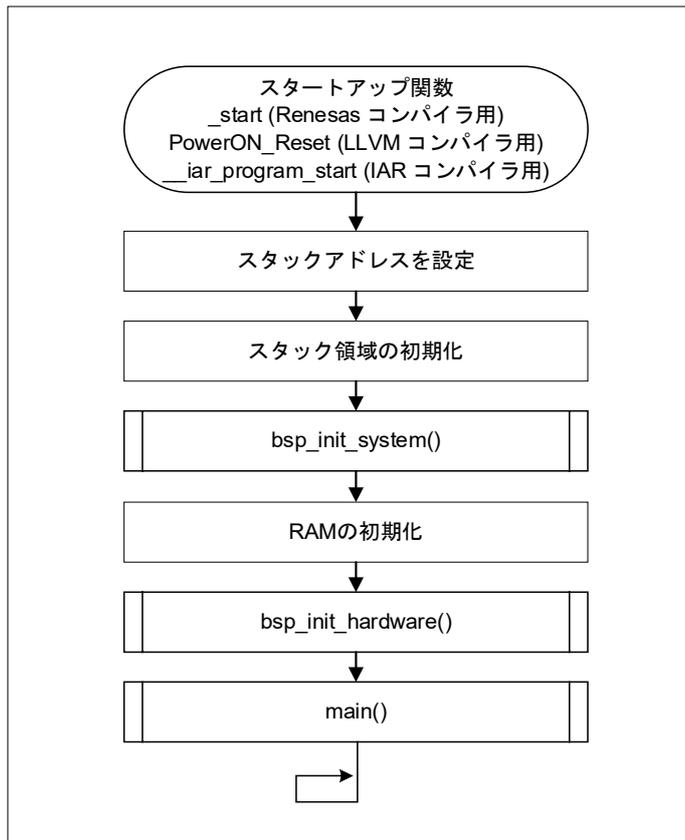
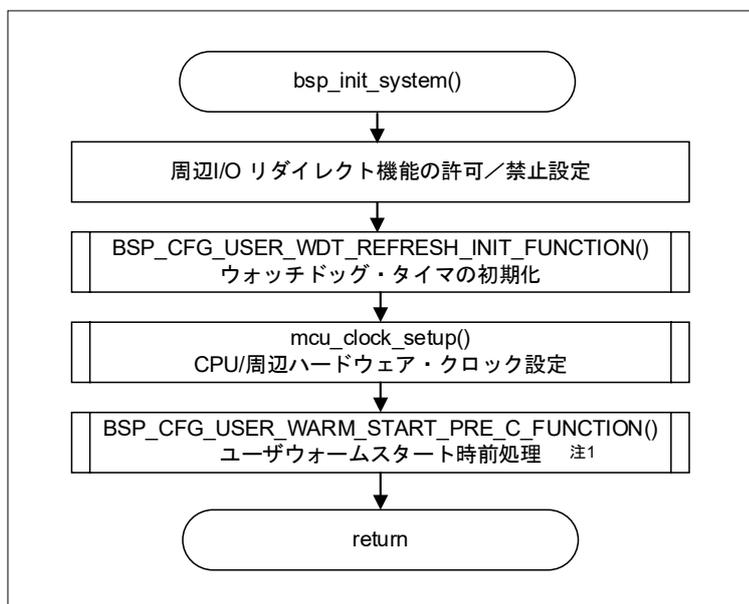
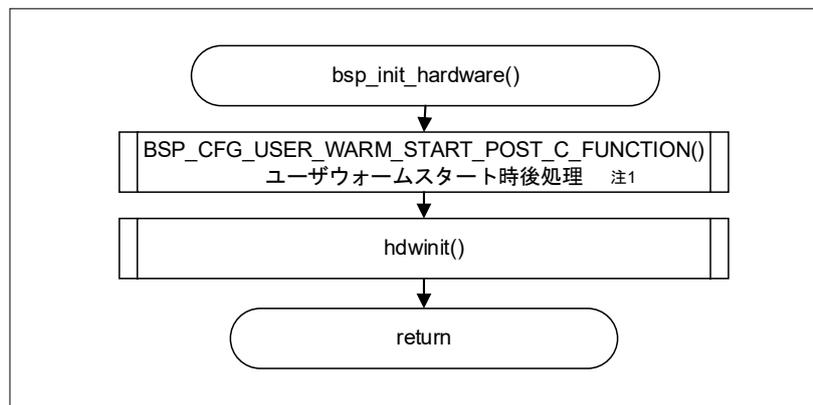


図 1-4 CPU リセット発生から main 関数までの処理イメージ : bsp\_init\_system()



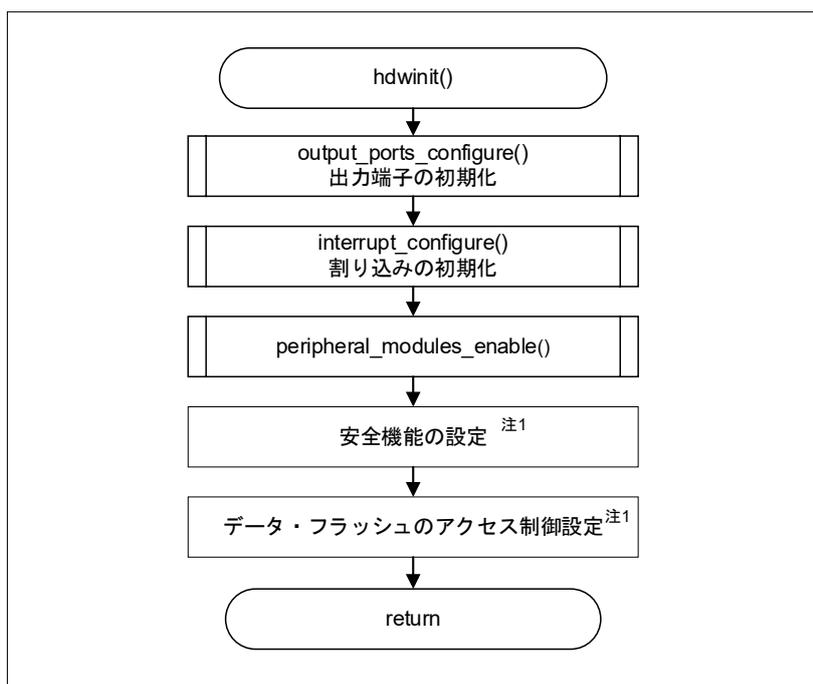
注 1. r\_bsp\_config.h 内のコンフィグレーションオプションの設定によって動作は異なります。スマート・コンフィグレータ (SC) を使用する場合、ソフトウェアコンポーネント設定画面でコンフィグレーションオプションを設定できます。SC で設定したコンフィグレーションオプションは、コードを生成時に自動で r\_bsp\_config.h に反映されます。

図 1-5 CPU リセット発生から main 関数までの処理イメージ : bsp\_init\_hardware()



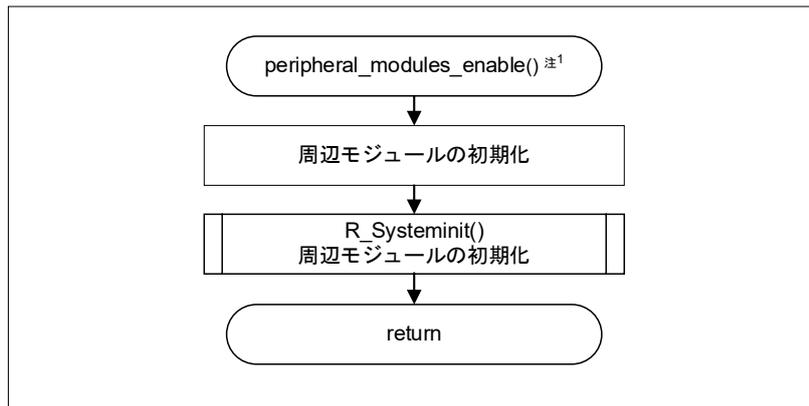
注 1. `r_bsp_config.h` 内のコンフィグレーションオプションの設定によって動作は異なります。スマート・コンフィグレータ (SC) を使用する場合、ソフトウェアコンポーネント設定画面でコンフィグレーションオプションを設定できます。SC で設定したコンフィグレーションオプションは、コードを生成時に自動で `r_bsp_config.h` に反映されます。

図 1-6 CPU リセット発生から main 関数までの処理イメージ : hdwinit()



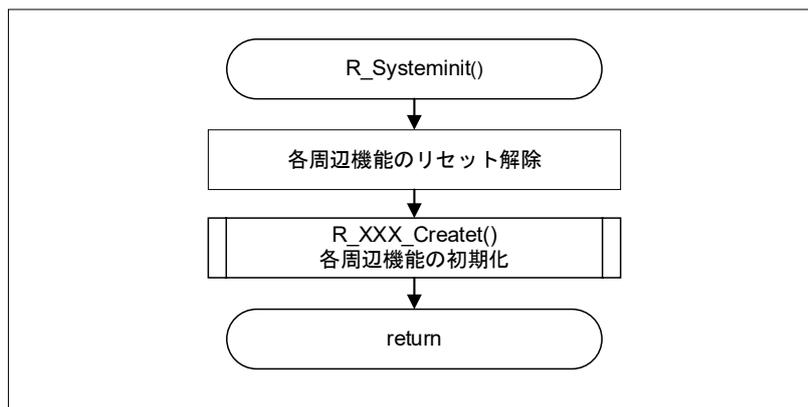
注 1. `r_bsp_config.h` 内のコンフィグレーションオプションの設定によって動作は異なります。スマート・コンフィグレータ (SC) を使用する場合、ソフトウェアコンポーネント設定画面でコンフィグレーションオプションを設定できます。SC で設定したコンフィグレーションオプションは、コードを生成時に自動で `r_bsp_config.h` に反映されます。

図 1-7 CPU リセット発生から main 関数までの処理イメージ : peripheral\_modules\_enable()



注 1. `r_bsp_config.h` 内のコンフィグレーションオプションの設定によって動作は異なります。スマート・コンフィグレータ (SC) を使用する場合、ソフトウェアコンポーネント設定画面でコンフィグレーションオプションを設定できます。SC で設定したコンフィグレーションオプションは、コードを生成時に自動で `r_bsp_config.h` に反映されます。

図 1-8 CPU リセット発生から main 関数までの処理イメージ : R\_Systeminit()



備考. 詳細は、RL78 ファミリ ボードサポートパッケージモジュール アプリケーションノート (R01AN5522) を参照してください。

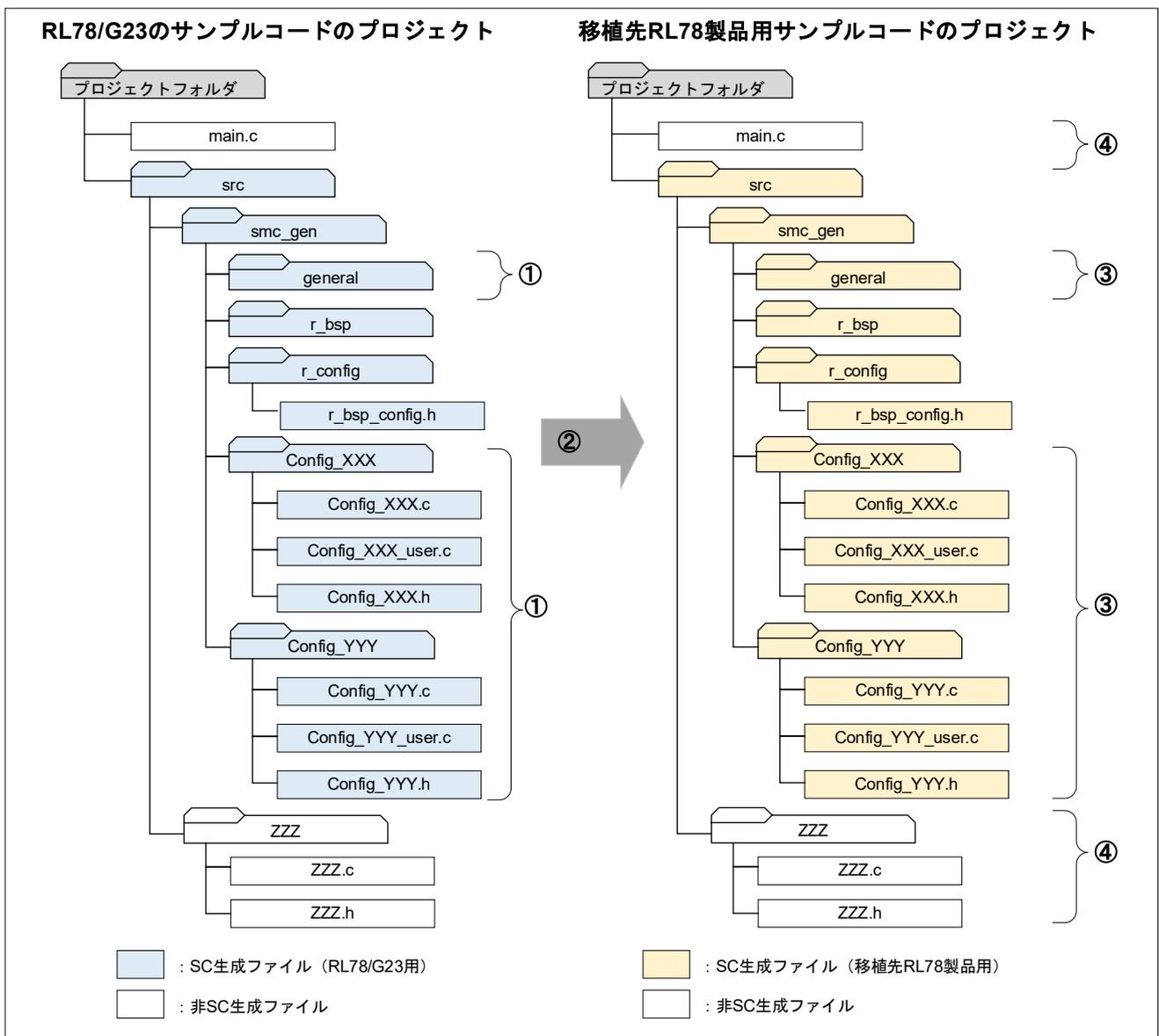
### 1.3 移植方法

サンプルコードを置き換える場合、移植先 RL78 製品用サンプルコードのプロジェクトに対してデバイス変更を実施し、移植先 RL78 製品に応じたリソース変更を実施します。

サンプルコードによっては、スマート・コンフィグレータ (SC) 生成コードを部分的にコメントアウトする等の変更しているものがあります。移植先のプロジェクトをリビルドする時に同じ変更を加える必要があるため、移植元の RL78 製品用サンプルコードに対して、SC 生成コードが変更されているかを確認してください。また、移植先の RL78 製品では、移植元のプロジェクトと同じリソースを使用できない場合があります。その場合は、移植先のプロジェクトで別のリソースへ変更する必要があります。

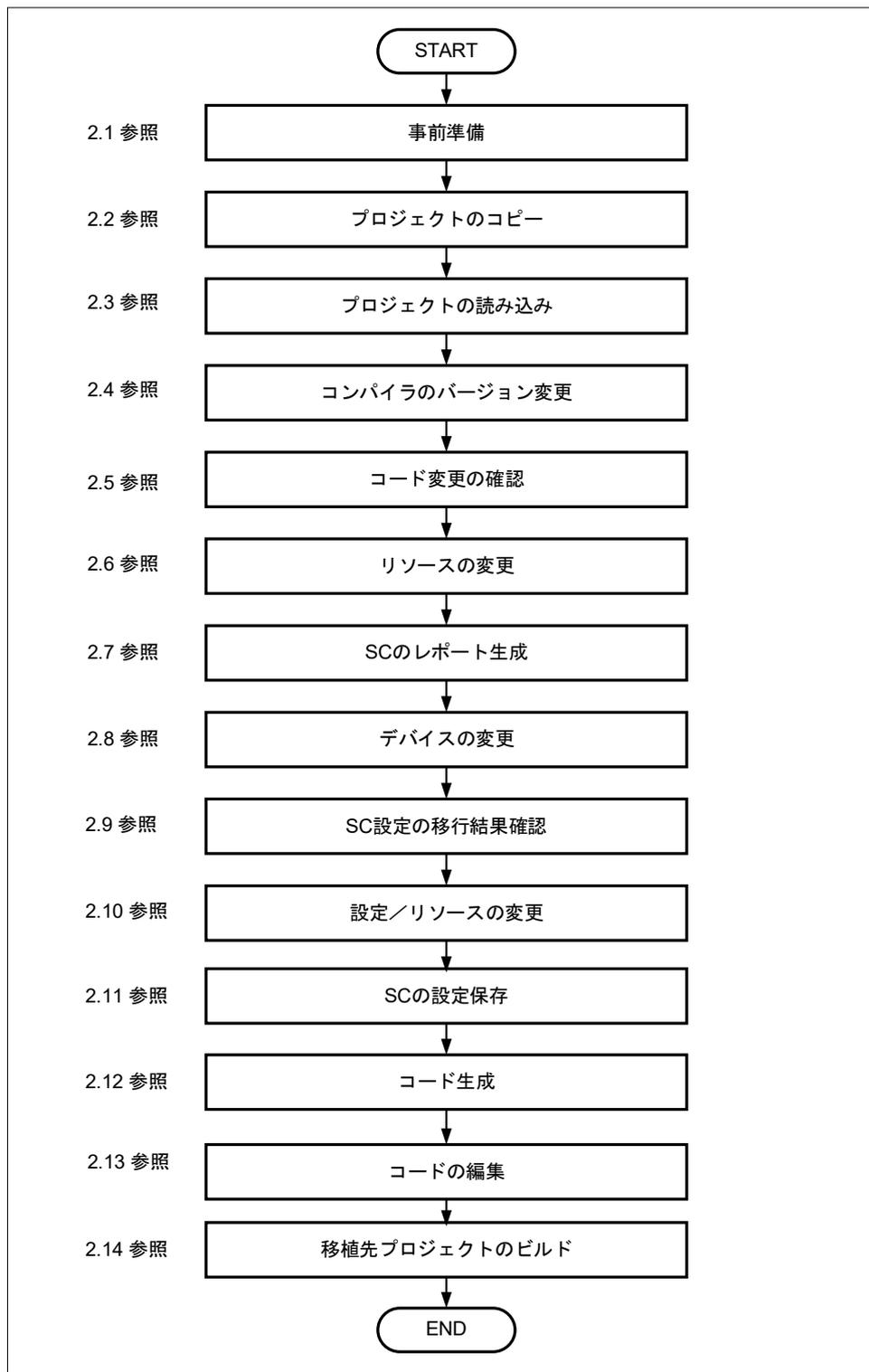
図 1-9 に移植のイメージを示し、図 1-10 に移植フローの概要を示します。

図 1-9 移植のイメージ



- ① 移植元の SC 生成コードに対して、プログラムを追加または削除されていないか確認する。
- ② 移植元プロジェクトの対象デバイスを移植先 RL78 製品に変更してコード生成する。
- ③ ①で SC 生成コードが変更されていた場合、同じ変更を③のコードに対して実施する。
- ④ リソース変更がある場合、適切にコードを変更する。

図 1-10 移植フローの概要



## 2. 移植手順

### 2.1 事前準備

サンプルコードで使用している周辺機能と端子を、サンプルコードに添付されているドキュメントで確認します。次に、移植先 RL78 製品のユーザーズマニュアルを利用して移植方法を検討します。

なお、表 1-2 で「不可」に該当する場合、新規にプログラムを開発してください。

表 2-1、表 2-2、表 2-3、表 2-4 に、それぞれ動作電圧の確認内容、動作周波数の確認内容、周辺機能の確認内容、使用端子の確認内容を示します。

表 2-1 動作電圧の確認内容

移植先の RL78 製品	移植方法
移植元と同じ電圧で動作できる。	同じ動作電圧を使用する。
移植元と同じ電圧で動作できない。	別の動作電圧に変更する。 <sup>注</sup>

注. 移植先の RL78 製品の電気的特性を満たすように回路設計してください。

表 2-2 動作周波数の確認内容

移植先の RL78 製品	移植方法
移植元と同じ動作周波数で動作できる。	同じ動作周波数を使用する。
移植元と同じ動作周波数で動作できない。	別の動作周波数に変更する。 <sup>注</sup>

注. 移植先の RL78 製品の電気的特性を満たすように回路設計してください。

表 2-3 周辺機能の確認内容

移植先の RL78 製品	移植方法
移植元と同じチャンネルがある。	同じチャンネルを使用する。
移植元と同じチャンネルがない。	別のチャンネルに変更する。

表 2-4 使用端子の確認内容

移植先の RL78 製品	移植方法
移植元と同じポート端子があり、同じ兼用機能が割り付けられている。	同じ端子を使用する。
移植元と同じポート端子があるが、同じ兼用機能が割り付けられていない。	兼用機能を使用せず、ポート機能を使用する場合： 同じポート端子を利用する。 兼用機能を使用する場合： 同じ兼用機能がある端子に変更する。
移植元と同じポート端子がない。	兼用機能を使用せず、ポート機能を使用する場合： 別のポート端子に変更する。 兼用機能を使用する場合： 同じ兼用機能がある端子に変更する。

## 2.2 プロジェクトのコピー

移植元のサンプルコードのプロジェクトを、任意のフォルダへコピーし移植先プロジェクトとします。

## 2.3 プロジェクトの読み込み

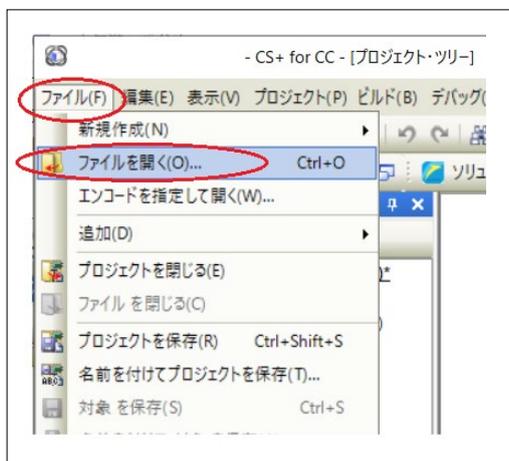
統合開発環境 (IDE) 毎にプロジェクトの読み込み手順について説明します。

### 2.3.1 CS+ for CC の場合

CS+ for CC を起動し、移植先プロジェクトを読み込みます。

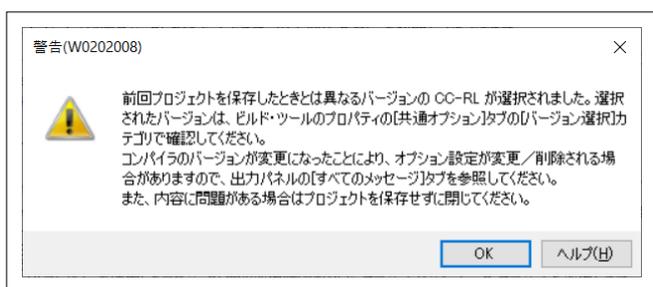
CS+ for CC を起動し、「ファイル」－「ファイルを開く」メニューを選択し、移植先プロジェクトの mtpj ファイルを読み込みます。

図 2-1 mtpj ファイルの読み込み



備考. サンプルコード作成時のツールと現在使用しているツールのバージョンが異なる場合、バージョンが異なることを知らせる警告メッセージが出ます。内容を確認し「OK」をクリックしてください。

図 2-2 「警告」ダイアログ

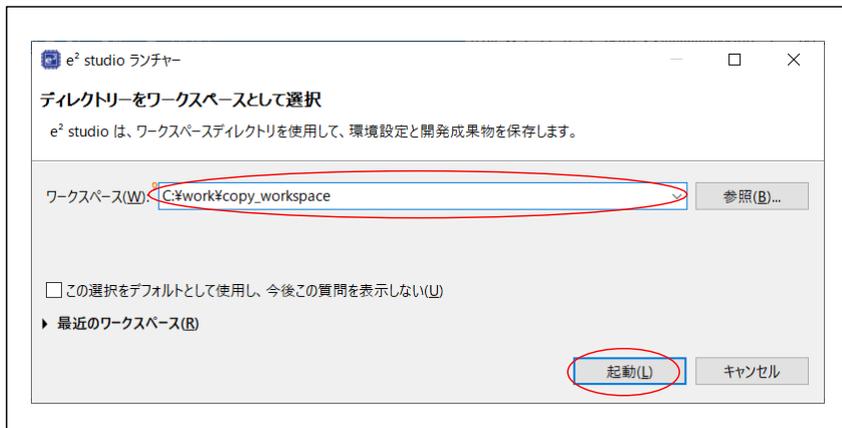


### 2.3.2 e<sup>2</sup> studio の場合

e<sup>2</sup> studio を起動し、移植先プロジェクトを読み込みます。

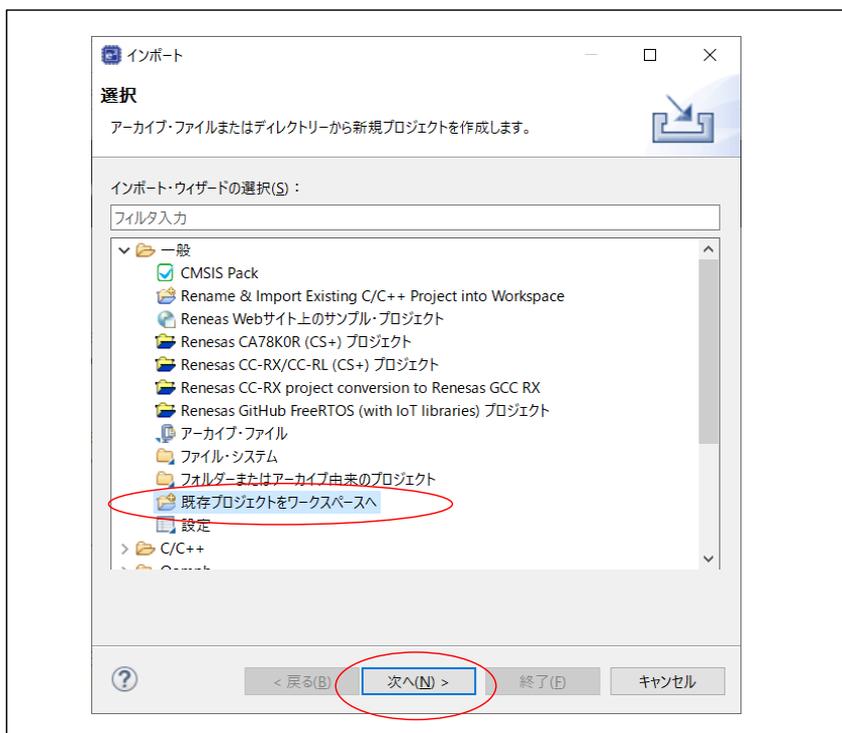
e<sup>2</sup> studio を起動し、e<sup>2</sup> studio ランチャーのワークスペースで任意のフォルダを指定し、「起動」をクリックします。

図 2-3 e<sup>2</sup> studio ランチャー



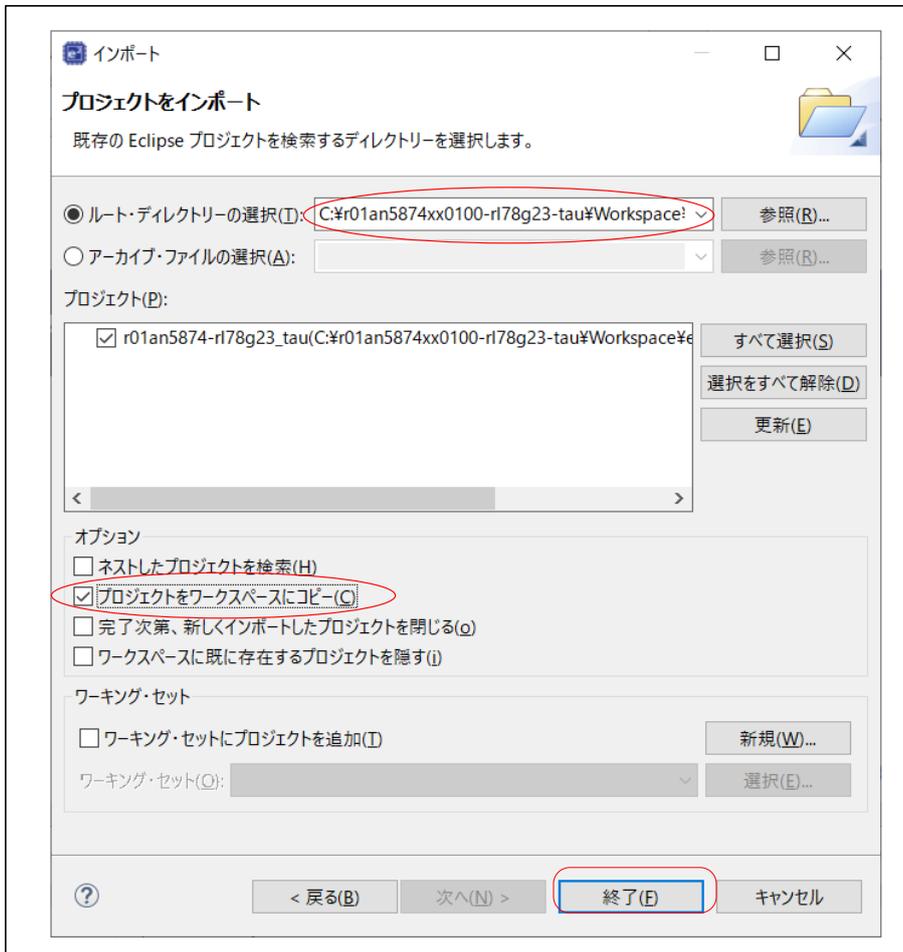
e<sup>2</sup> studio の「ファイル」－「インポート」メニューを選択します。インポートダイアログで、「一般」－「既存プロジェクトをワークスペースへ」を選択し、「次へ」をクリックします。

図 2-4 「インポート」ダイアログ (1/2)



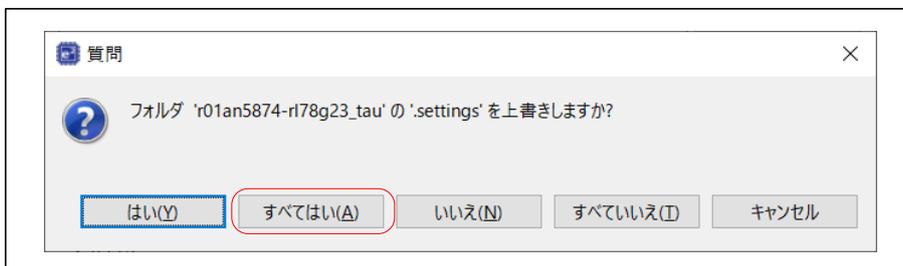
インポートダイアログで、「ルート・ディレクトリーの選択」に移植先プロジェクトが存在するフォルダを指定後、オプションの「プロジェクトをワークスペースにコピー」にチェックをし、「終了」をクリックします。

図 2-5 「インポート」ダイアログ (2/2)



「質問」ダイアログで、「すべてはい」をクリックします。

図 2-6 「質問」ダイアログ

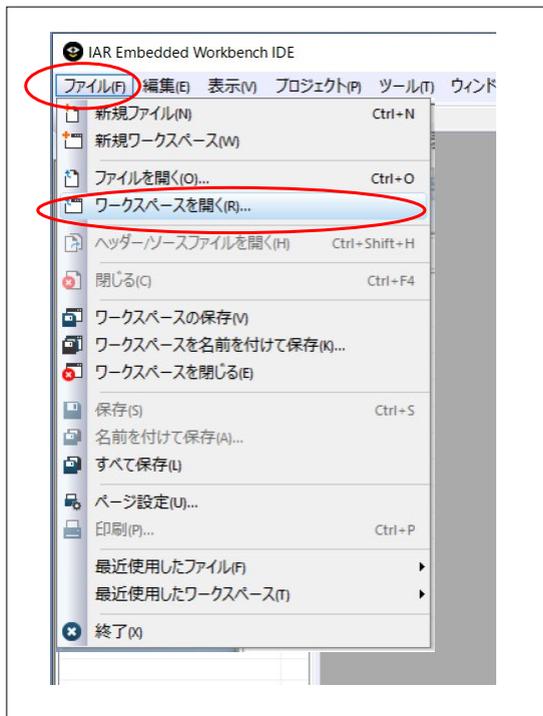


### 2.3.3 IAR EW for Renesas RL78 の場合

IAR EW for Renesas RL78 を起動し、移植先プロジェクトを読み込みます。

IAR EW for Renesas RL78 を起動し、「ファイル」－「ワークスペースを開く」メニューを選択し、移植先プロジェクトの eww ファイルを読み込みます。

図 2-7 eww ファイルの読み込み



## 2.4 コンパイラのバージョン変更

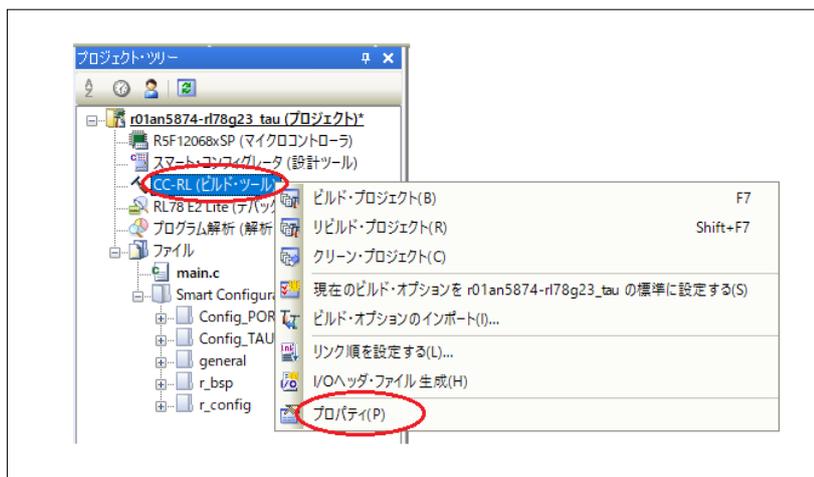
IDE 毎にコンパイラのバージョン変更について説明します。

### 2.4.1 CS+ for CC の場合

CS+ for CC で使用するコンパイラのバージョンを確認し、使用するバージョンに変更します。

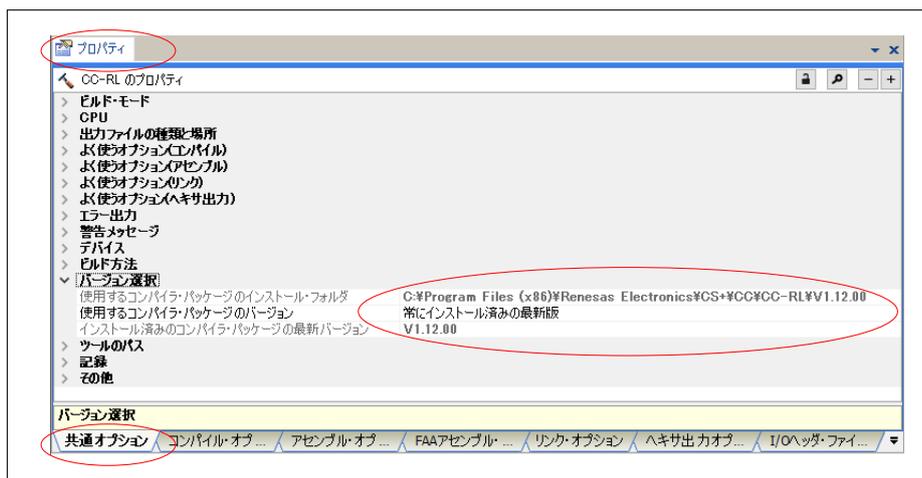
CS+ for CC のプロジェクト・ツリーで、「CC-RL (ビルド・ツール)」を右クリックし「プロパティ」をクリックします。

図 2-8 CC-RL プロパティ



「プロパティ」の「共通オプション」タブで、現在使用しているインストール済の CC-RL コンパイラのバージョンを指定します。

図 2-9 CC-RL プロパティバージョン選択

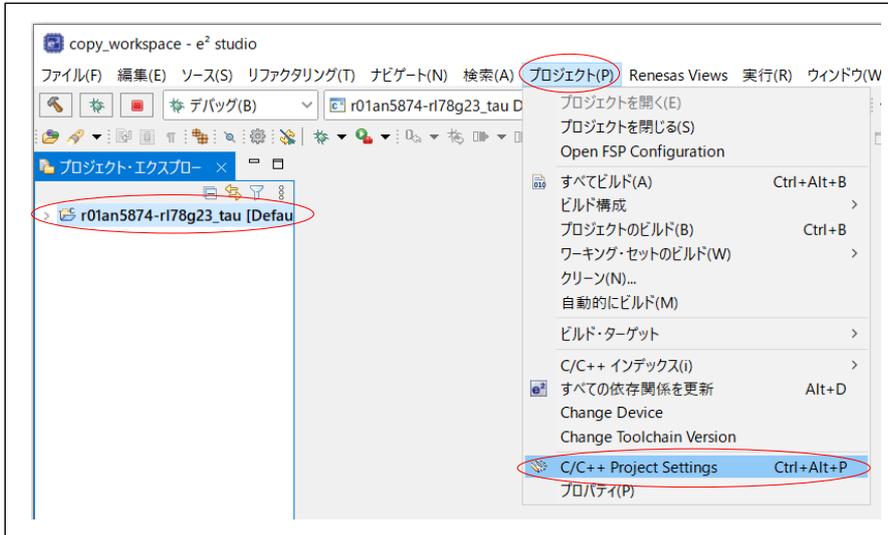


### 2.4.1 e<sup>2</sup> studio の場合

e<sup>2</sup> studio で使用するコンパイラのバージョンを確認し、使用するバージョンに変更します。

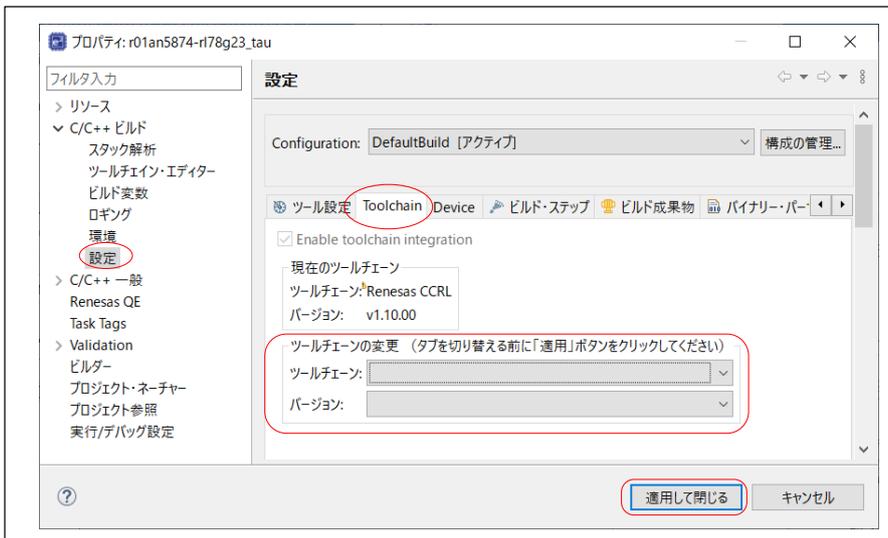
e<sup>2</sup> studio のプロジェクト・エクスプローラーでプロジェクト名をクリックで選択し、「プロジェクト」－「C/C++ Project Settings」メニューをクリックします。

図 2-10 Toolchain 設定 (1/2)



プロパティ・ダイアログの「C/C++ ビルド」－「設定」の Toolchain タブで、使用するバージョンを指定し、「適用して閉じる」をクリックします。

図 2-11 Toolchain 設定 (2/2)



### 2.4.2 IAR EW for Renesas RL78 の場合

「IAR EW for Renesas RL78」を変更してください。「IAR EW for Renesas RL78」に含まれるコンパイラが使用されます。

## 2.5 コード変更の確認

デバイス変更前のプロジェクトに対して、スマート・コンフィグレータ (SC) が生成したコードを変更していないか確認します。

対象ファイルは、図 1-9 の①で示すファイルです。エディタで対象ファイルを開き、SC 生成コードに変更を加えたことを示すコメントがないか確認してください。変更があった場合には、同じ変更を図 1-9 の③で示すファイルに実施します。

プロジェクト・ツリーのイメージについて、図 2-12、図 2-13、図 2-14 に、それぞれ CS+ for CC の場合、e<sup>2</sup> studio の場合、IAR EW for Renesas RL78 の場合を示します。

各図の例では、赤枠で示す Config\_XXX および general フォルダ以下のファイルが確認対象となります。

図 2-12 CS+ for CC の場合

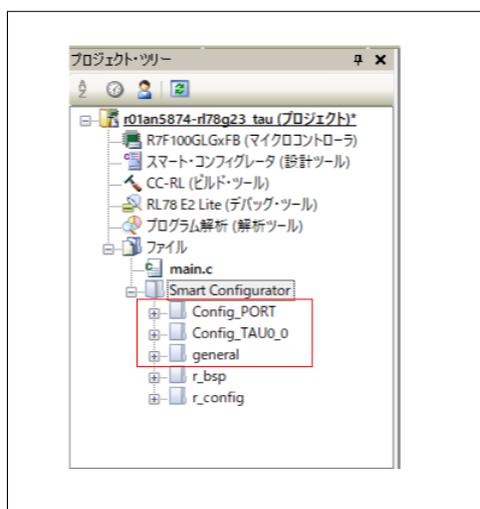


図 2-13 e<sup>2</sup> studio の場合

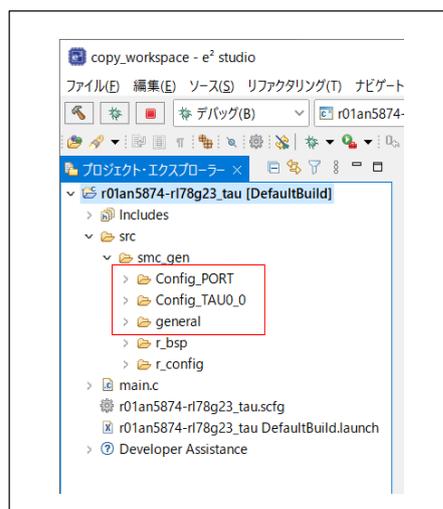
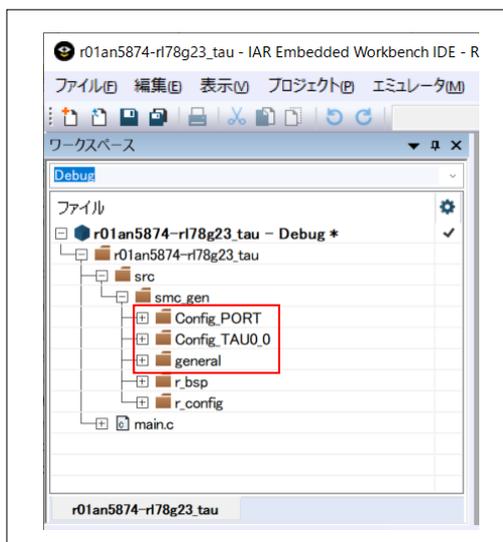


図 2-14 IAR EW for Renesas RL78 の場合



## 2.6 リソースの変更

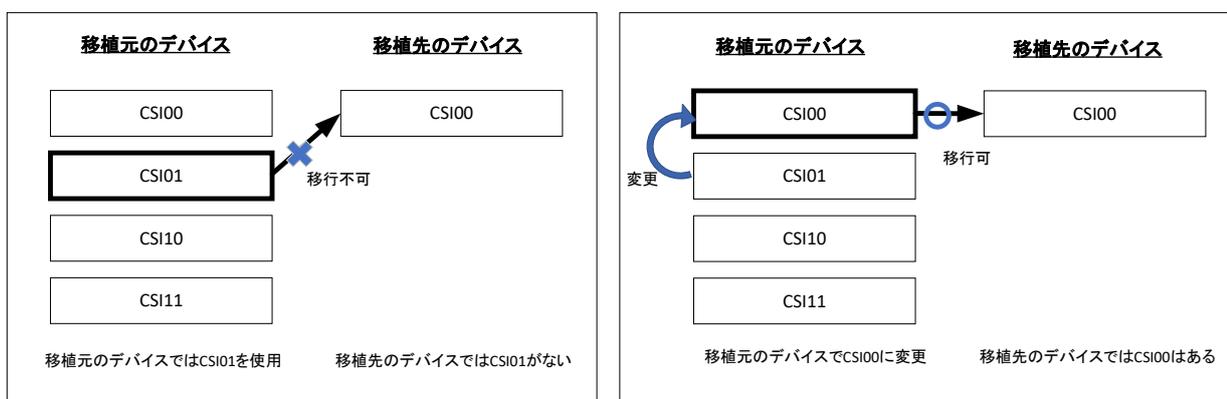
「2.1 事前準備」で、周辺機能の移植方法として「別のチャンネルに変更する。」を選択した場合、デバイス変更前にスマート・コンフィグレータ (SC) でチャンネル (リソース) の変更を行います。「同じチャンネルを使用する。」を選択した場合は、この作業は不要です。また、使用端子については「2.10.5 端子」で最後に見直しを行います。

別のチャンネルに変更する例として、下記の場合を考えます。

- ・移植元のデバイスは CSI00 と CSI01 を搭載している。CSI01 が使用されている。
- ・移植先のデバイスは CSI00 を搭載している。

この場合、移植元のデバイスで CSI01 から CSI00 にリソースを変更した後、移植先のデバイスに変更します。

図 2-15 リソース変更のイメージ



備考. 「2.8 デバイスの変更」を実施した後もリソースの変更は可能です。しかし、上記の場合、デバイス変更時に CSI01 の設定内容が削除されるため、デバイス変更後に CSI00 の設定内容を手動で設定する必要があります。

各 IDE で SC の起動手順が異なります。しかし、すべての IDE で SC の基本操作は同じです。最初に IDE 毎に SC 起動手順を説明します。その後、リソースの変更について説明します。

表 2-5 SC (Smart Configurator) の起動手順

IDE	起動手順
CS+ for CC	CS+ for CC から SC を起動する
e2 studio	e2 studio 内 (プラグイン) で起動する
IAR EW for Renesas RL78	IAR EW を使用せず、SC を単独で起動する

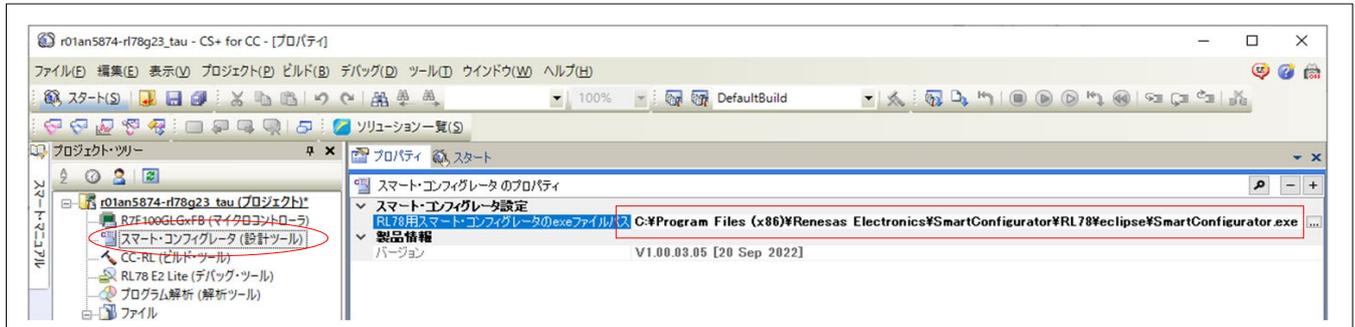
## RL78 ソフトウェア置き換えガイド

### スマート・コンフィグレータで生成されたソフトウェアの移植 (CS+, e2 studio, IAR)

#### 2.6.1 CS+ for CC での SC 起動手順

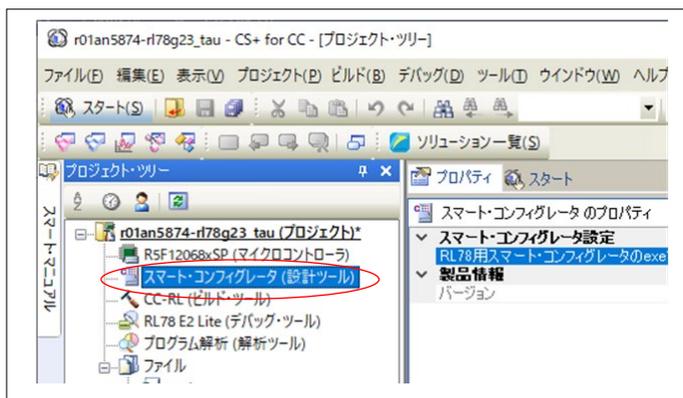
CS+ for CC のプロジェクト・ツリーで「スマート・コンフィグレータ (設計ツール)」をクリックし、プロパティで使用する SC の exe ファイルがあるパスを指定します。

図 2-16 スマート・コンフィグレータのパス指定



CS+ for CC のプロジェクト・ツリーで「スマート・コンフィグレータ (設計ツール)」をダブルクリックします。

図 2-17 スマート・コンフィグレータの起動



SC が起動します。

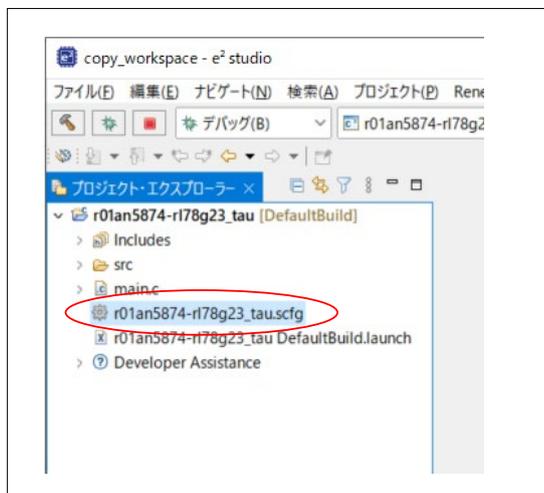
図 2-18 スマート・コンフィグレータ



## 2.6.2 e<sup>2</sup> studio での SC 起動手順

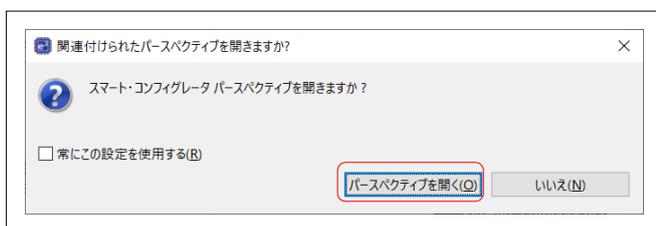
e<sup>2</sup> studio のプロジェクト・エクスプローラーで、scfg ファイルをダブルクリックします。

図 2-19 スマート・コンフィグレータの起動



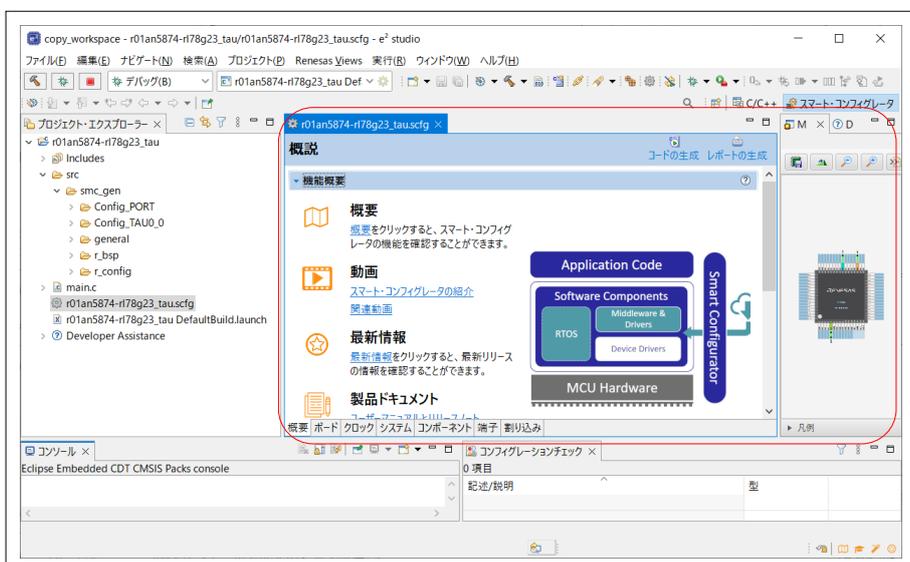
「関連付けられたパースペクティブを開きますか？」ダイアログで「パースペクティブを開く」をクリックします。

図 2-20 「関連付けられたパースペクティブを開きますか？」ダイアログ



SC が起動します。

図 2-21 スマート・コンフィグレータ (プラグイン)

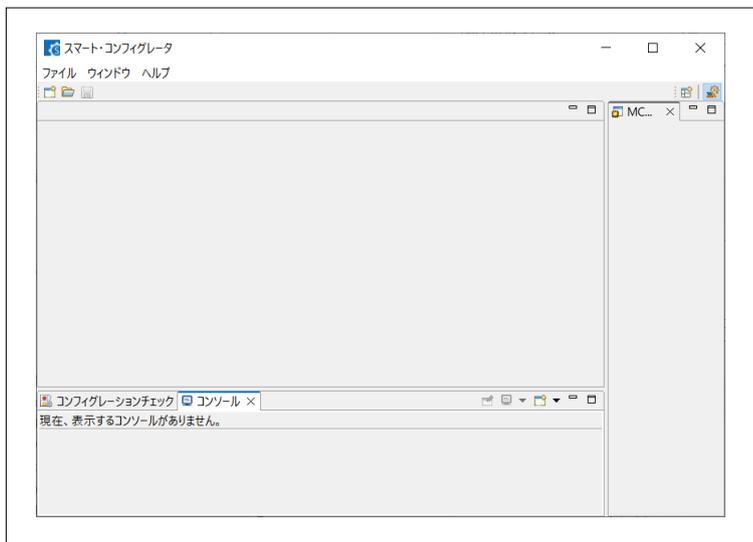


### 2.6.3 IAR EW for Renesas RL78 での SC 起動手順

Windows のスタートメニューから「Renesas Electronics Smart Configurator」 – 「Smart Configurator for RL78 Vx.x.x」をクリックします。(x.x.x はバージョン番号)

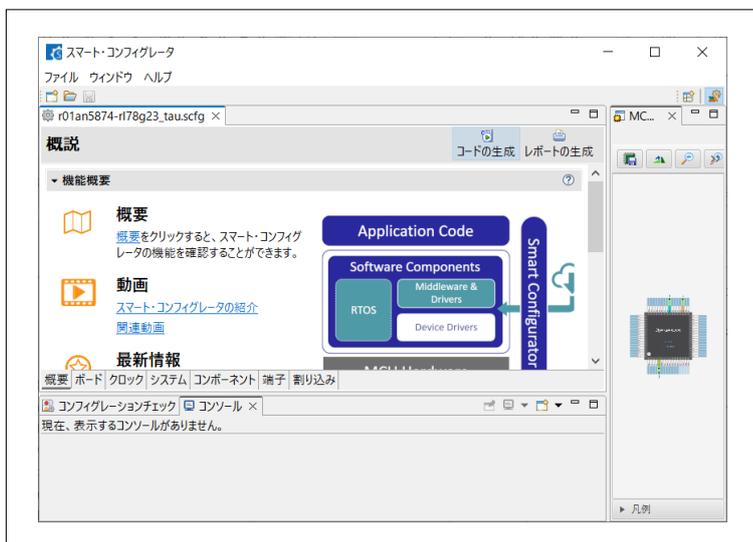
SC が起動します。

図 2-22 スマート・コンフィグレータの起動



「ファイル」 – 「開く」メニューで、移植先プロジェクトの scfg ファイルを読み込みます。

図 2-23 scfg 読み込み後のスマート・コンフィグレータ



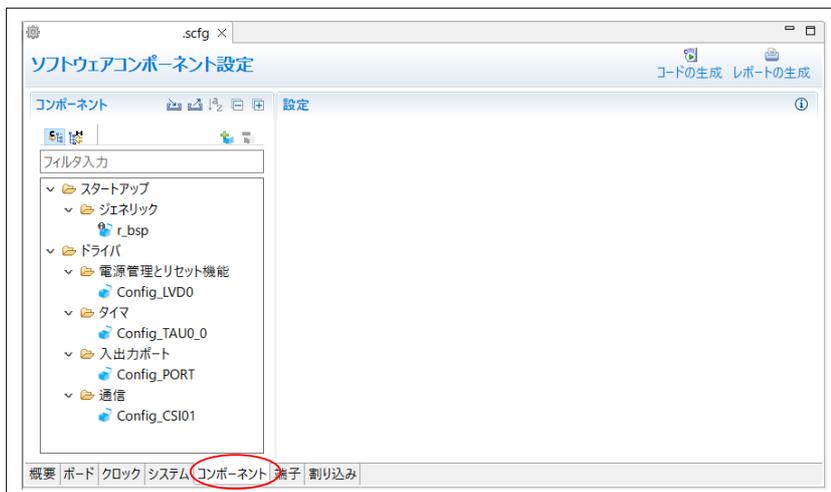
備考. IAR EW for Renesas RL78 の「ツール」 – 「ツールの設定」メニューで、SC の exe ファイルを登録すると、「ツール」メニューに SC 起動の項目が追加され IAR EW for Renesas RL78 からの起動ができます。

## 2.6.4 リソースの変更

CSI01 から CSI00 へ変更する場合を例として説明します。

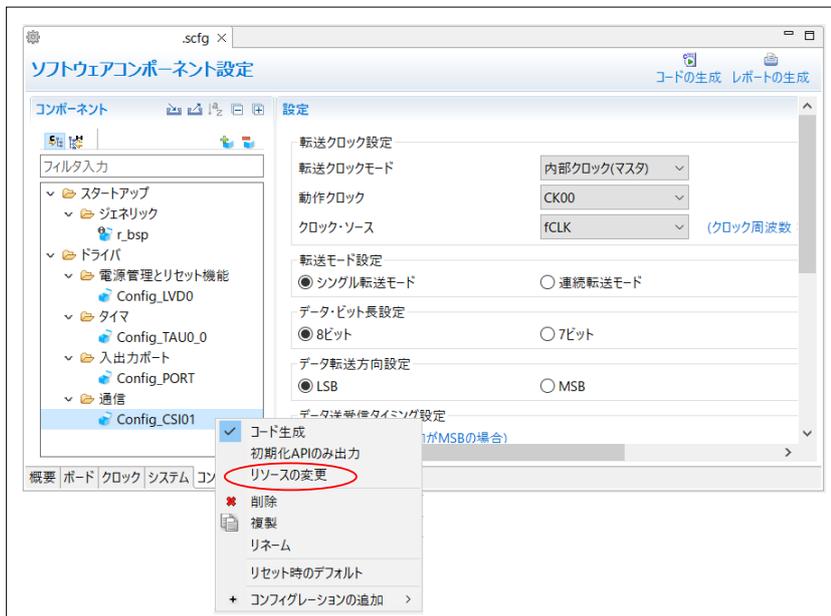
SC の「コンポーネント」タブを表示します。

図 2-24 「コンポーネント」タブの選択



リソースを変更します。「Config\_CSI01」を右クリックし、「リソースの変更」をクリックします。

図 2-25 「リソースの変更」メニュー



「リソースの選択」ダイアログでリソースを変更します。リソースを変更後、「終了」をクリックします。

図 2-26 「リソースの選択」ダイアログ (1/4)

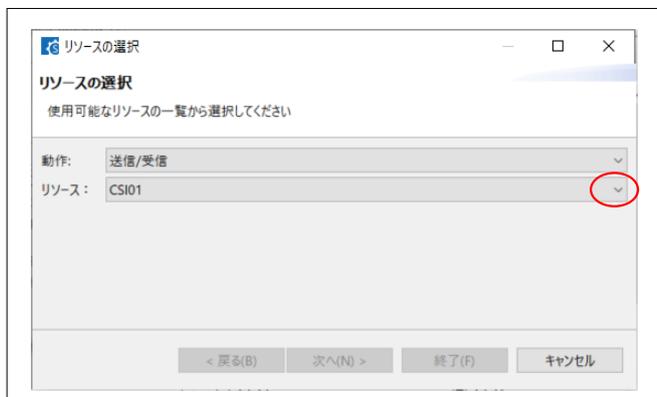


図 2-27 「リソースの選択」ダイアログ (2/4)

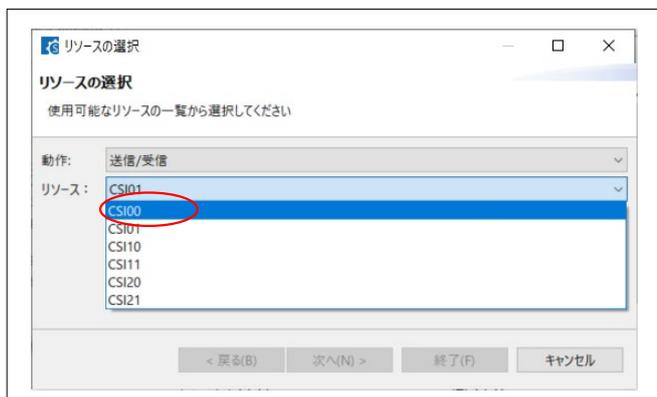


図 2-28 「リソースの選択」ダイアログ (3/4)

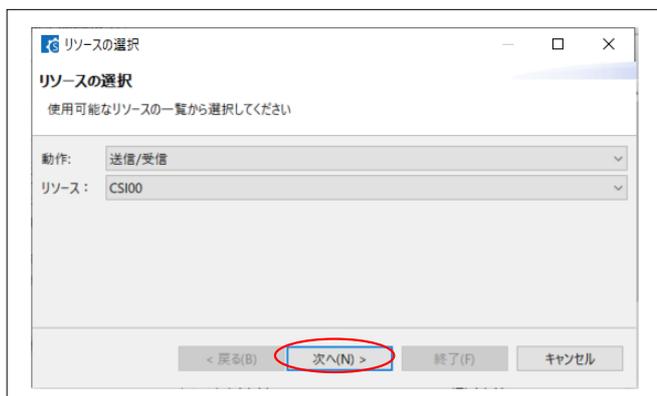
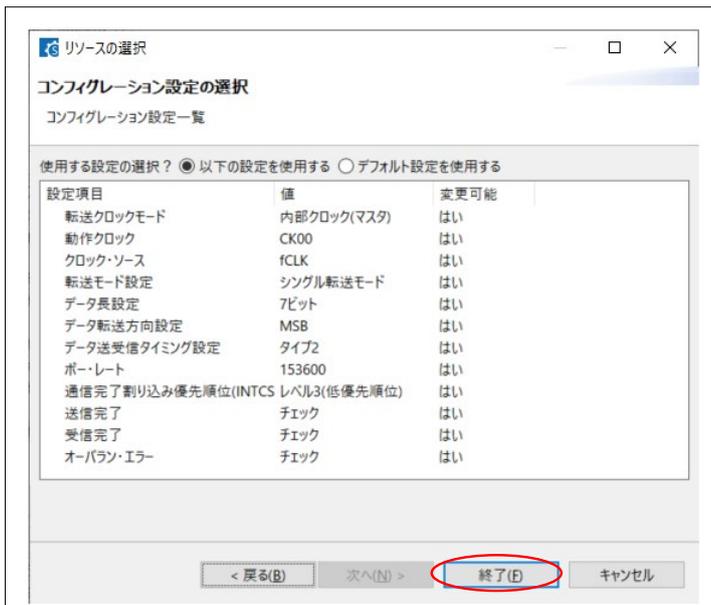


図 2-29 「リソースの選択」ダイアログ (4/4)



## 2.7 SC のレポート生成

スマート・コンフィグレータ (SC) でデバイスを変更する前に、SC の設定内容をレポートとして保存します。レポートは、「2.10 設定／リソースの変更」でデバイス変更前の設定内容の確認をする際に参照します。

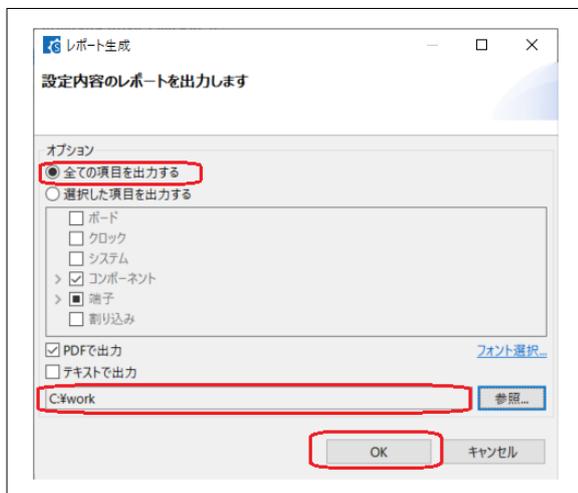
SC の「レポートの生成」をクリックします。

図 2-30 スマート・コンフィグレータ レポートの生成



「レポート生成」ダイアログで、「全ての項目を出力する」の選択とレポートの出力先を指定し、「OK」をクリックします。

図 2-31 「レポート生成」ダイアログ



指定したフォルダにレポートが生成されます。また、「コンソール」タブのリンクからレポートを開くことができます。

図 2-32 レポートファイル



## 2.8 デバイスの変更

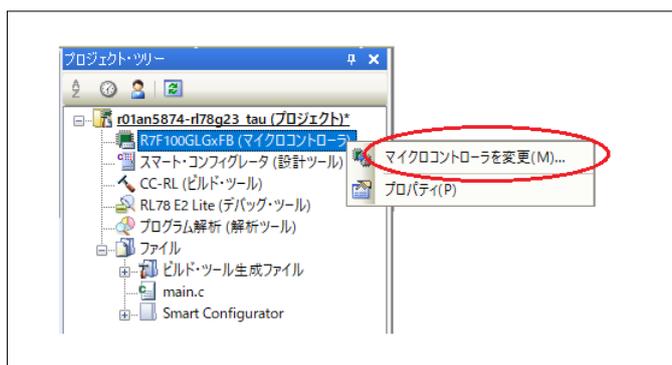
IDE 毎にデバイスの変更について説明します。

### 2.8.1 CS+ for CC の場合

CS+ for CC とスマート・コンフィグレータ (SC) で対象デバイスを変更します。

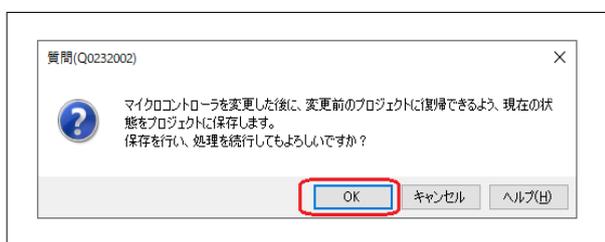
CS+ for CC のプロジェクト・ツリーで、マイクロコントローラ名を右クリックし、「マイクロコントローラを変更」をクリックします。

図 2-33 マイクロコントローラの変更



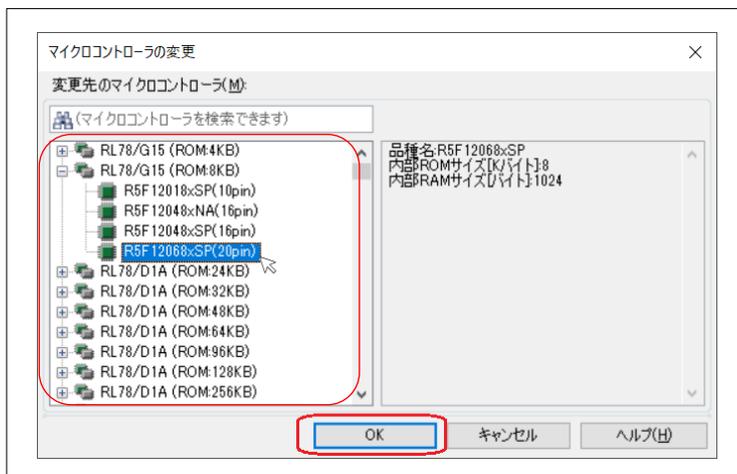
「質問」ダイアログで、「OK」をクリックします。

図 2-34 「質問」ダイアログ



「マイクロコントローラの変更」ダイアログで移植先のデバイスを選択し、「OK」をクリックします。

図 2-35 「マイクロコントローラの変更」



備考. 移植元と移植先のデバイス仕様の差分により、CC-RLのオプション設定を移行できない場合があります。移行できなかった内容は、CS+ for CCの「出力」タブに表示されます。「2.12 コード生成」でSCにてコード生成時にユーザ・オプション・バイト等の設定は変更後のデバイスに合わせて再設定されるため、ここでは内容の確認のみ行います。

図 2-36 CS+ for CC デバイス変更時のメッセージ例



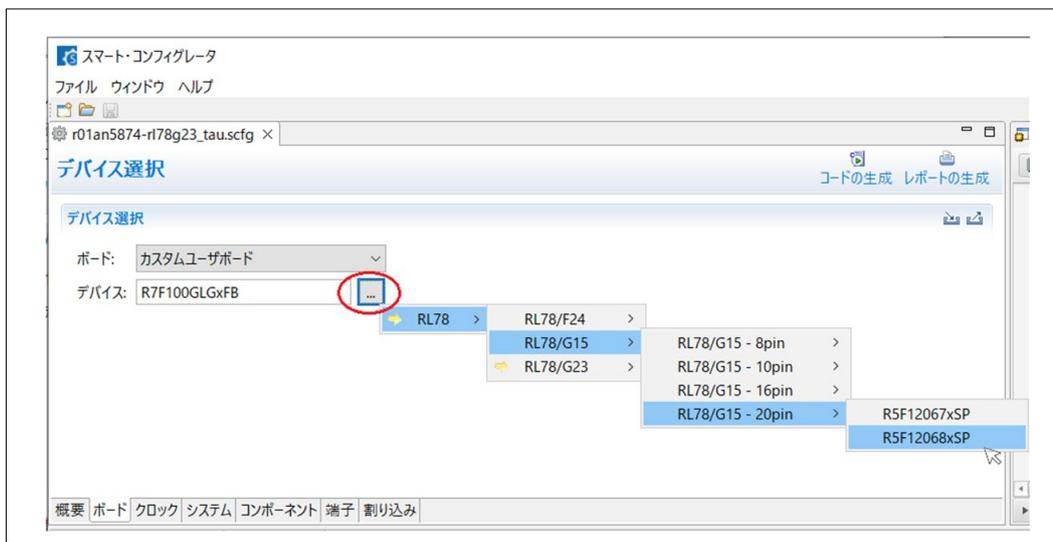
SCの「ボード」タブを表示します。

図 2-37 「ボード」タブの選択



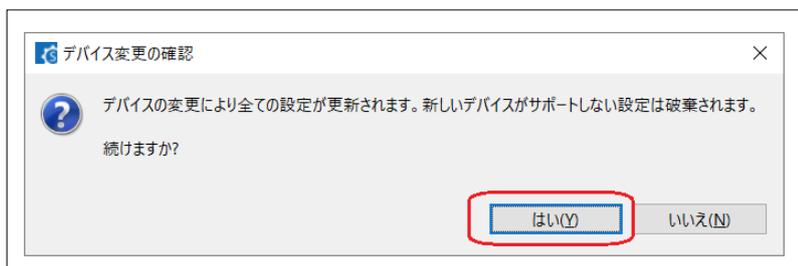
「デバイス」の「...」をクリックし、移植先のデバイス名をクリックします。

図 2-38 デバイスの選択



「デバイス変更の確認」ダイアログが表示されます。「はい」をクリックします。

図 2-39 「デバイス変更の確認」ダイアログ

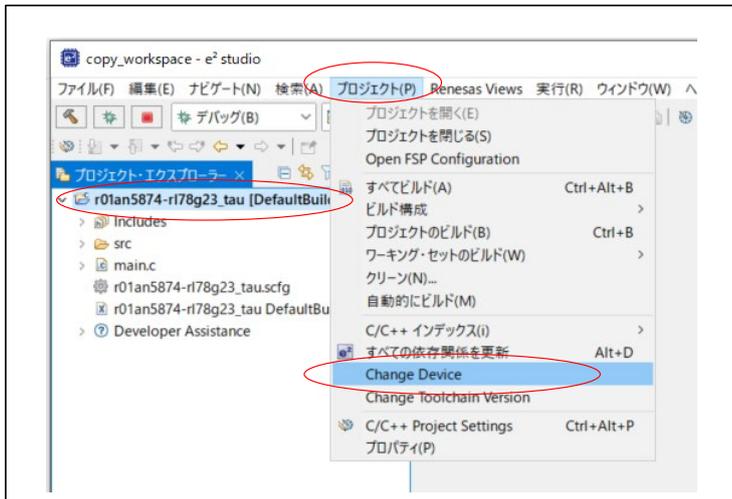


## 2.8.2 e<sup>2</sup> studio の場合

e<sup>2</sup> studio で対象デバイスを変更します。

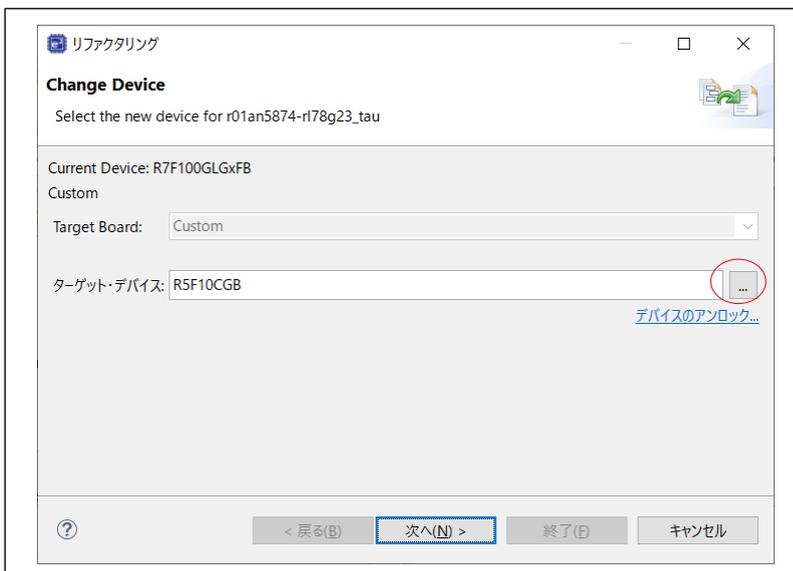
e<sup>2</sup> studio のプロジェクト・エクスプローラーでプロジェクトをクリックし、「プロジェクト」 - 「Change Device」メニューをクリックします。

図 2-40 「プロジェクト」 - 「Change Device」メニュー



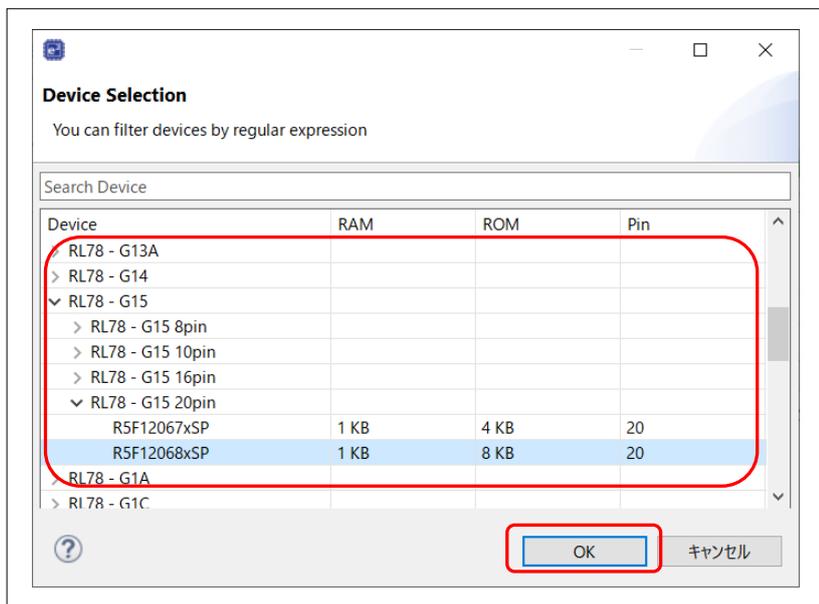
「リファクタリング」ダイアログで、「...」をクリックします。  
 (「ターゲット・デバイス」のテキストボックスに、移植先のデバイス名を入力することもできます。)

図 2-41 「リファクタリング」ダイアログ (1/4)



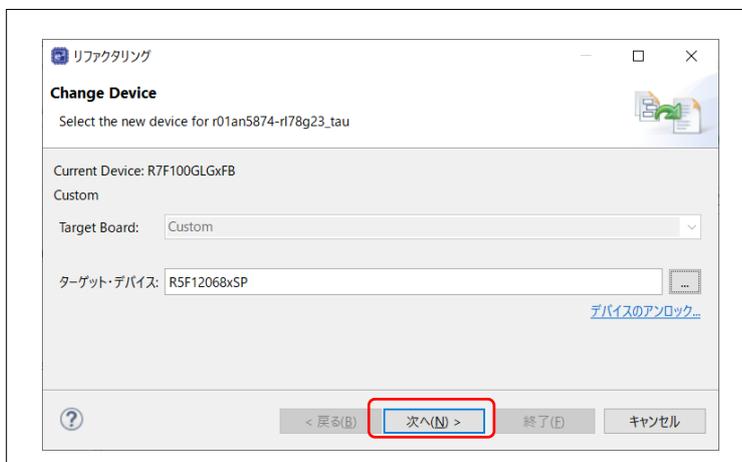
「Device Selection」ダイアログで移植先のデバイスを選択し、「OK」をクリックします。

図 2-42 「Device Selection」ダイアログ



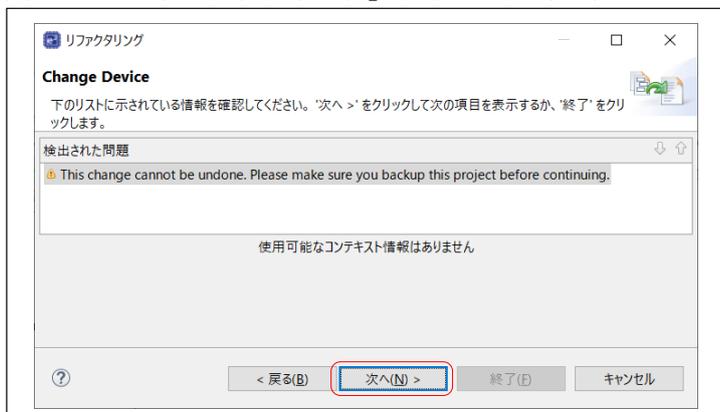
「リファクタリング」ダイアログで、「次へ」をクリックします。

図 2-43 「リファクタリング」ダイアログ (2/4)



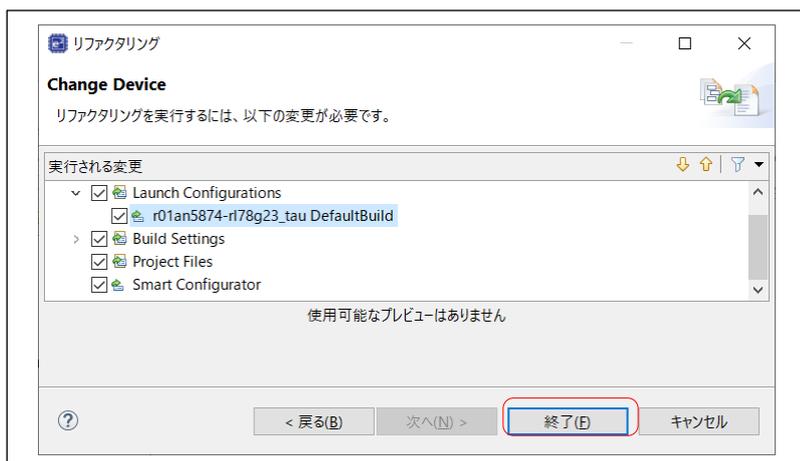
表示された情報を確認し、「次へ」をクリックします。

図 2-44 「リファクタリング」ダイアログ (3/4)



表示された情報を確認し、「終了」をクリックします。チェックボックスのチェックの変更は不要です。

図 2-45 「リファクタリング」ダイアログ (4/4)

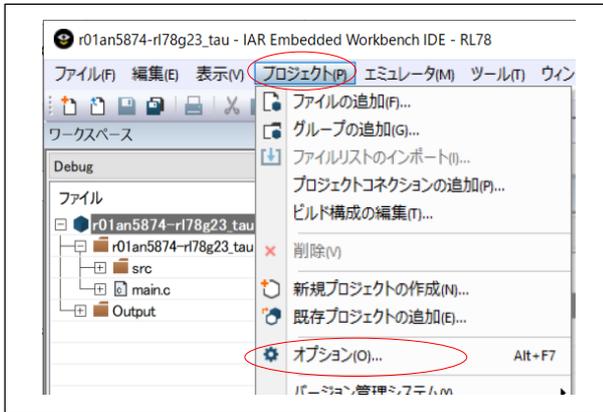


### 2.8.3 IAR EW for Renesas RL78 の場合

IAR EW for Renesas RL78 とスマート・コンフィグレータ (SC) で対象デバイスを変更します。

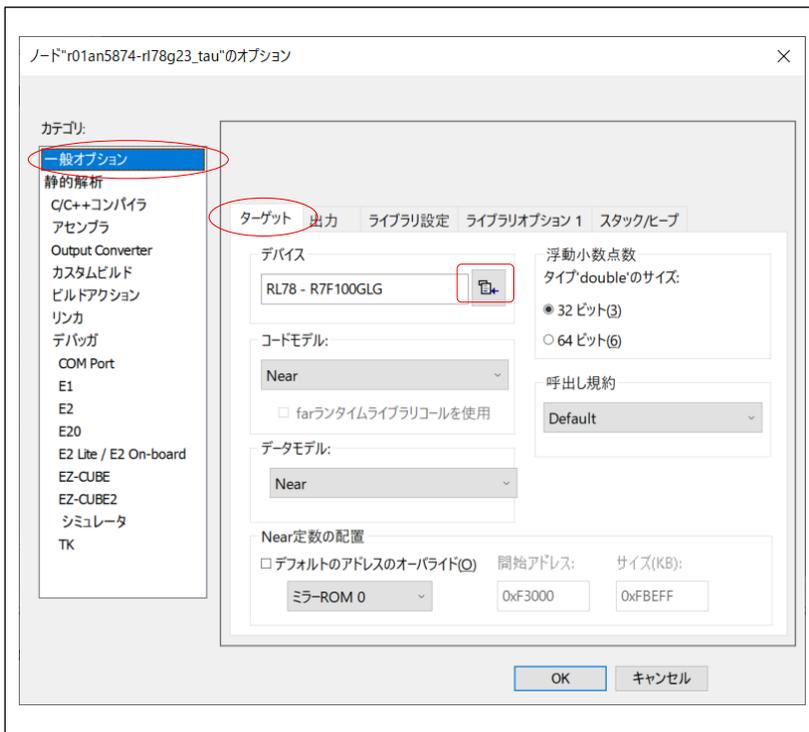
IAR EW for Renesas RL78 の「プロジェクト」－「オプション」メニューをクリックします。

図 2-46 「プロジェクト」－「オプション」メニュー



「オプション」ダイアログで、「一般オプション」－「ターゲット」タブを表示し、「デバイス」のボタンをクリックします。

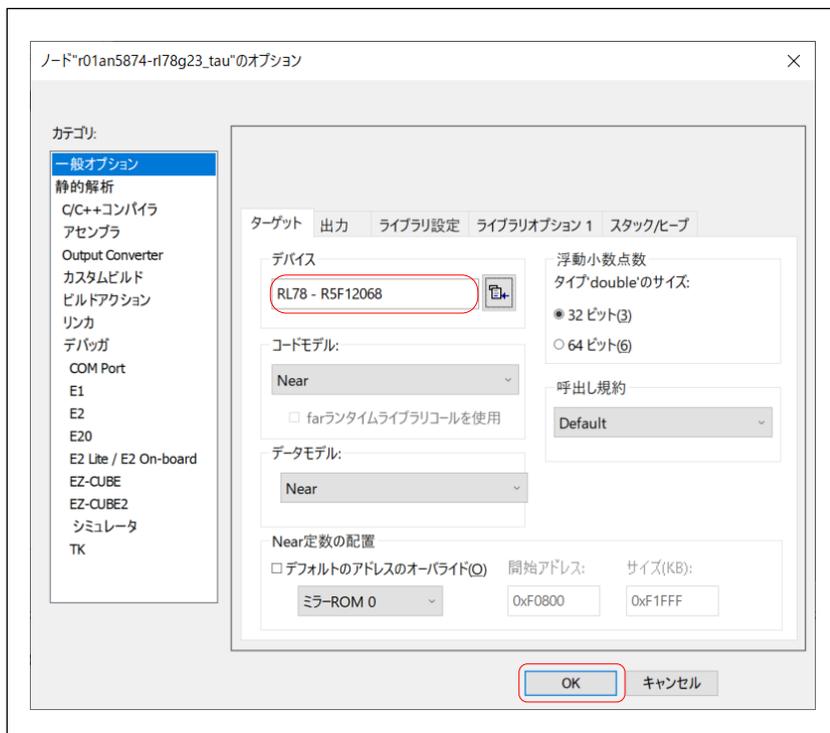
図 2-47 「オプション」ダイアログ (1/2)



表示されたリストから移植先のデバイスを選択します。

「OK」をクリックします。

図 2-48 「オプション」ダイアログ (2/2)



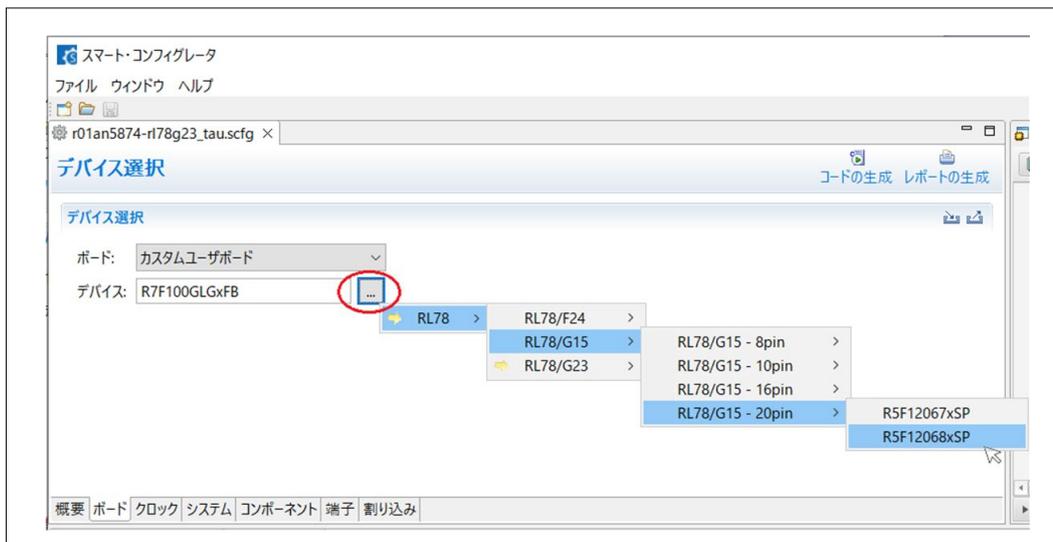
SCの「ボード」タブを表示します。

図 2-49 「ボード」タブの選択



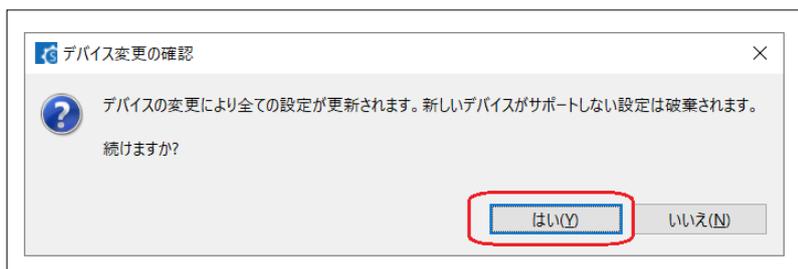
「デバイス」の「...」をクリックし、移植先のデバイス名をクリックします。

図 2-50 デバイスの選択



「デバイス変更の確認」ダイアログが表示されます。「はい」をクリックします。

図 2-51 「デバイス変更の確認」ダイアログ



## 2.9 SC 設定の移行結果確認

「2.8 デバイスの変更」で、スマート・コンフィグレータ (SC) がデバイス変更レポートを生成します。デバイス変更レポートを参照して移行結果を確認します。移行結果に従って必要な処置を行います。

「コンソール」タブに表示されているリンクをクリックし、デバイス変更レポートを表示します。

図 2-52 デバイス変更レポート

スマート・コンフィグレータデバイス変更レポート

変更前のデバイス: R7F100GLGxFB.  
 変更後のデバイス: R5F12068xSP.

### 1 クロック設定

移行済み。一部の設定項目が移行されました  
 コンフィグレーションの移行状況を示す表です。

Table 1-1 ボードの供給クロック周波数の移行状況

設定	値(変更前のデバイス)	値(変更後のデバイス)	移行状況
高速オンチップ・オシレータクロック	32	16	移行なし。値が存在しません

Table 1-2 クロック設定値の移行状況

設定	値(変更前のデバイス)	値(変更後のデバイス)	移行状況
fIL	0.032768	0.032768	移行済み。
fSXL	0.032768	-	移行なし。設定が存在しない
fIHP	32	-	移行なし。設定が存在しない
fMAIN	32	32	移行済み。
fCLK	32	32	移行済み。

表 2-6, 表 2-7 に、デバイス変更レポートの記載について示します。

表 2-6 デバイス変更レポートの移行状況 (1/2)

対象	移行／変換状況	原因／処置	参照章
クロック 設定	移行済み。	[原因] 変更前のデバイスの設定内容が、変更後のデバイスで設定されました。 [処置] 必要な処置はありません。	—
	移行なし。値が存在しません	[原因] 変更前のデバイスで設定していた周波数が、変更後のデバイスで設定できません。 [処置] 該当項目の周波数を確認し、値を適宜変更します。	2.10.1
	移行なし。設定が存在しない	[原因] 変更前のデバイスで存在していた設定項目が、変更後のデバイスには存在しません。 [処置] 該当項目がサンプルコードで使用しているクロックの場合、サンプルコードの移植はできません。	—
割り込み	移行済み。	[原因] 変更前のデバイスの設定内容が、変更後のデバイスで設定されました。 [処置] 必要な処置はありません。	—
	移行なし。設定が存在しない	[原因] 変更前のデバイスで存在していた割り込み要因が、変更後のデバイスには存在しません。 [処置] 該当項目がサンプルコードで使用していない割り込みの場合、必要な処置はありません。サンプルコードで使用している割り込みの場合、サンプルコードの移植はできません。	2.10.3
端子	変換成功	[原因] 変更前のデバイスで使用しているポートおよび兼用機能が、変更後のデバイスで設定されました。 [処置] デバイス変更後に必ず端子の設定状況の確認をします。	2.10.5
	変換なし、端子機能が使用できません	[原因] 変更前のデバイスで使用しているポートおよび兼用機能が、変更後のデバイスには存在しません。 [処置] 他の端子を使用します。同機能を持つ他の端子へ変更できない場合は、サンプルコードの移植はできません。	2.10.5

表 2-7 デバイス変更レポートの移行状況 (2/2)

対象	移行／変換状況	原因／処置	参照
ソフトウェア コンポー ネント <small>注1</small>	移行済み。すべての設定項目が移行されました	[原因] 変更前のデバイスの設定内容が、変更後のデバイスで設定されました。 [処置] 必要な処置はありません。	—
	移行済み。一部の設定項目が移行されました	[原因] 変更前のデバイスの設定内容の一部が、変更後のデバイスで設定されました。 [処置] 該当コンポーネントの設定内容を確認し、サンプルコードに影響がないか確認します。影響がある場合、サンプルコードの移植はできません。	2.10.4(2)
	移行なし。変更後のデバイスでチャンネルが存在しません	[原因] 変更前のデバイスで使用しているチャンネルが、変更後のデバイスには存在しません。 [処置] 使用可能なチャンネルがあれば、手動で別のチャンネルを追加し、変更前と同じ設定をします。 使用可能なチャンネルがなければ、サンプルコードの移植はできません。 補足：変更前と変更後のデバイスで共通のチャンネルがある場合、SCでのデバイス変更前に、使用チャンネルを変更することで移行ができます。(参照：2.6 リソース変更 (SC))	—
	移行なし。変更後のデバイスでコンポーネントがサポートされていません	[原因] 変更前のデバイスでサポートされているコンポーネントが、変更後のデバイスには存在しません。 [処置] 該当コンポーネントがサンプルコードで使用しているコンポーネントの場合、サンプルコードの移植はできません。 ただし、RL78/G23 から RL78/G15 への変更で、移行なしのコンポーネントが電源電圧回路 (Config_LVD) の場合は、システム設定で再設定することで移植ができます。	2.10.4(1)
	移行なし。最新バージョンにアップグレードしてください	[原因] 変更前のプロジェクトで使用している BSP のバージョンが、変更後のデバイスに対応していません。 [処置] BSP を最新バージョンに更新します。	2.10.4(3)
各コン ポーネ ント <small>注1</small>	移行済み。	[原因] 変更前のデバイスの設定内容が、変更後のデバイスで設定されました。 [処置] 必要な処置はありません。	—
	移行なし。設定が存在しない	[原因] 変更前のデバイスで存在していた設定項目が、変更後のデバイスには存在しません。 [処置] 該当項目がサンプルコードに影響がないか確認します。影響がある場合、サンプルコードの移植はできません。	2.10.4(1)
システム 設定	移行済み。	[原因] 変更前のデバイスの設定内容が、変更後のデバイスで設定されました。 [処置] 必要な処置はありません。	—
	移行なし。設定が存在しない	[原因] 変更前のデバイスで存在していた設定項目が、変更後のデバイスには存在しません。 [処置] 該当項目がサンプルコードに影響がないか確認します。デバッグに関する設定のため、多くの場合サンプルコードの動作に影響なく移植できます。	2.10.4(1)

注 1. Config\_PORT (ポート) は、「2.10.5 端子設定」で確認を行ってください。

## 2.10 設定／リソースの変更

スマート・コンフィグレータ (SC) 設定の移行結果から、処置が必要な内容について説明します。クロック設定、システム設定、割り込み設定、ソフトウェアコンポーネント設定、端子設定の場合を例として説明します。

### 2.10.1 クロック設定

移植先のデバイスに応じた動作周波数および動作電圧を設定します。

移植元のデバイスでは「fCLK = 32MHz」を使用していたが、移植先のデバイスでは「fCLK = 16MHz」を使用する場合について変更手順を説明します。

図 2-53 デバイス変更レポート例：クロック設定 (1/2)

### 1 クロック設定

移行済み。一部の設定項目が移行されました  
 コンフィグレーションの移行状況を示す表です。

Table 1-1 ボードの供給クロック周波数の移行状況

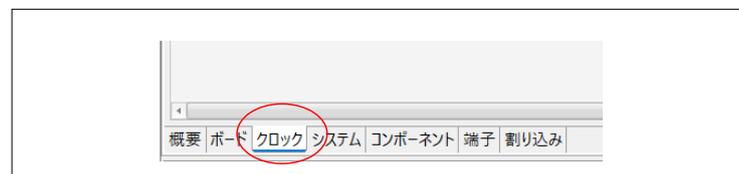
設定	値(変更前のデバイス)	値(変更後のデバイス)	移行状況
高速オンチップ・オシレータクロック	32	16	移行なし。値が存在しません

Table 1-2 クロック設定値の移行状況

設定	値(変更前のデバイス)	値(変更後のデバイス)	移行状況
fIL	0.032768	0.015	移行なし。値が存在しません
fSXL	0.032768	-	移行なし。設定が存在しない
fIHP	32	-	移行なし。設定が存在しない
fMAIN	32	16	移行なし。値が存在しません
fCLK	32	16	移行なし。値が存在しません

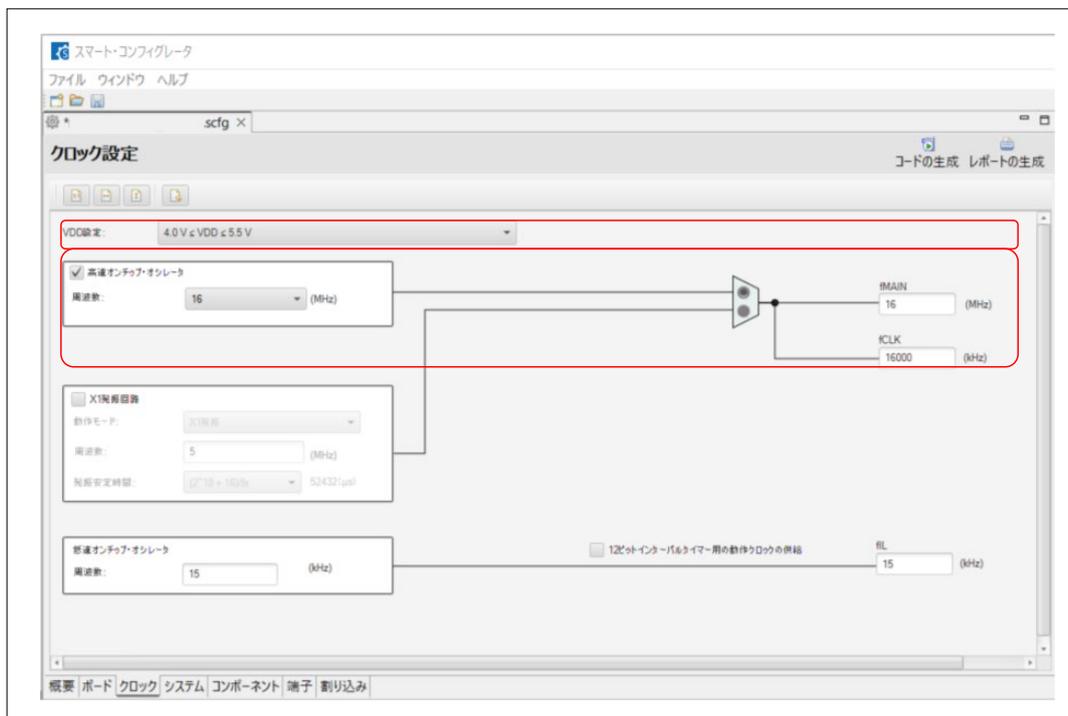
SC の「クロック」タブを表示します。

図 2-54 「クロック」タブの選択



動作周波数および VDD 設定を移植先のデバイスに合わせて設定します。

図 2-55 「クロック」タブ



注意. 移植先の RL78 製品の電気的特性を満たすように回路設計してください。

Table 1-3 で示される設定は BSP (ボードサポートパッケージ) の設定項目です。BSP の設定項目はデバイスごとに異なります。BSP の設定項目を変更する場合には、SC の「コンポーネント」タブの「r\_bsp」で設定を変更してください。

図 2-56 デバイス変更レポート例：クロック設定 (2/2)

Table 1-3 マクロ定義値の移行状況

設定	値(変更前のデバイス)	値(変更後のデバイス)	移行状況
BSP_CFG_SUBCLK_SOURCE	0	-	移行なし。設定が存在しない
BSP_CFG_HOCO_DIVIDE	0	-	移行なし。値が存在しません
BSP_CFG_SUBSYSCLK_SOURCE	1	-	移行なし。設定が存在しない
BSP_CFG_FCLK_SOURCE	0	-	移行なし。設定が存在しない
BSP_CFG_MOCO_DIVIDE	0	-	移行なし。設定が存在しない
BSP_CFG_MOCO_SOURCE	0	-	移行なし。設定が存在しない
BSP_CFG_XT1_OSCMODE	0	-	移行なし。設定が存在しない
BSP_CFG_HISYSCLK_OPERATION	1	1	移行済み。

## 2.10.2 システム設定

デバイス変更後のデバッグ環境に応じて設定します。

図 2-57 デバイス変更レポート例：システム設定

### 5 システム設定

移行済み。一部の設定項目が移行されました  
コンフィギュレーションの移行状況を示す表です。

Table 5-1 オンチップ・デバッグ設定 設定移行状況

設定	値(変更前のデバイス)	値(変更後のデバイス)	移行状況
オンチップ・デバッグ動作設定	使用しない	使用しない	移行済み。
エミュレータ設定	-	-	移行済み。
疑似RRM/DMM機能設定	-	-	移行済み。
Start/Stop関数機能設定	-	-	移行済み。
通過ポイント機能設定	-	-	移行済み。
トレース機能設定	-	-	移行なし。設定が存在しない
セキュリティIDを設定する	チェック	チェック	移行済み。
セキュリティID	0x000000000000000000000000	0x000000000000000000000000	移行済み。
セキュリティID認証失敗時の設定	-	-	移行なし。設定が存在しない

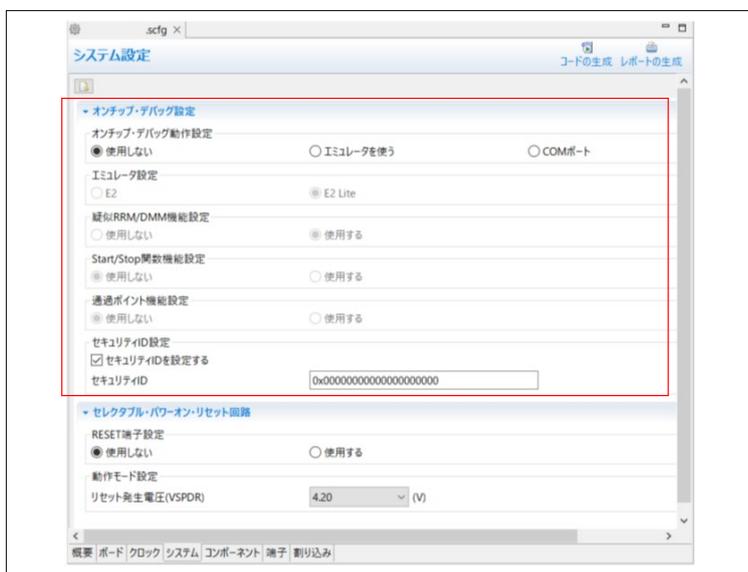
SCの「システム」タブを表示します。

図 2-58 「システム」タブの選択



使用するデバッグ環境に合わせて設定します。

図 2-59 例：「システム」タブ



### 2.10.3 割り込み設定

サンプルコードで使用している割り込みが移行できているか確認します。サンプルコードで使用している割り込みは、「2.7 SCのレポート生成」で生成したレポートで確認します。サンプルコードで使用していない割り込みが移行できていない場合は問題ありません。なお、「2.6 リソースの変更」でリソース変更している場合は、リソース変更後の割り込みが移行されていることを確認します。

「2.7 SCのレポート生成」で生成したレポートを開きます。割り込み設定で使用している割り込みを確認します。

図 2-60 SC のレポート例：割り込み設定

## 7. 割り込み設定

割り込み設定情報:

ベクタ番号	割り込み	優先レベル	バンク指定
14	INTTM00	レベル3(低優先順位)	なし

デバイス変更レポートで割り込みの移行状況を確認します。サンプルコードで使用している割り込み INTTM00 が「状態＝使用中」で移行されていることを確認します。

備考. 割り込みの移行状況は、ベクタ番号ごとに表示されます。そのため、割り込み名が同じでもデバイス変更前／後のベクタ番号が異なる場合、同じ行に表示されません。

図 2-61 デバイス変更レポート例：割り込み設定

## 2 割り込み

デバイス変更による割り込み設定の移行結果

Table 2-1 割り込みの移行結果

ベクタ番号	変換前の割り込み名	変換後の割り込み名	設定(デバイス変更前)	設定(デバイス変更後)	割り込みの移行結果
11	INTELCL	-	状態＝未使用 バンク指定＝なし 優先順位＝レベル3(低優先順位)	-	移行なし。設定が存在しない
13	INTIIC00	INTIIC00	状態＝未使用 バンク指定＝なし 優先順位＝レベル3(低優先順位)	状態＝未使用 バンク指定＝なし 優先順位＝レベル3(低優先順位)	移行済み。
14	INTTM00	INTTM00	状態＝使用中 バンク指定＝なし 優先順位＝レベル3(低優先順位)	状態＝使用中 バンク指定＝なし 優先順位＝レベル3(低優先順位)	移行済み。
15	INTSRE0	INTSRE0	状態＝未使用 バンク指定＝なし 優先順位＝レベル3(低優先順位)	状態＝未使用 バンク指定＝なし 優先順位＝レベル3(低優先順位)	移行済み。

### 2.10.4 ソフトウェアコンポーネント設定

ソフトウェアコンポーネントには r\_bsp (BSP: Board Support Package) と周辺機能が含まれます。r\_bsp コンポーネントを変更するとコード生成が必要となるため、周辺機能の処置を先に行い、r\_bsp コンポーネントは最後に処置します。

図 2-62 デバイス変更レポート例：ソフトウェアコンポーネント設定

## 4 ソフトウェアコンポーネント設定

コンフィグレーションの移行状況を示す表です。

Table 4-1 コンフィグレーションの移行状況

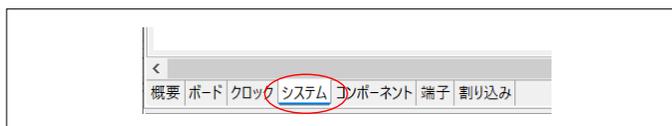
コンフィグレーション	移行状況
(3) r_bsp	移行済み。一部の設定項目が移行されました
(2) Config_COMP0	移行済み。一部の設定項目が移行されました
(1) Config_LVD0	移行なし。変更後のデバイスでコンポーネントがサポートされていません
Config_TAU0_1	移行済み。すべての設定項目が移行されました
(2) Config_WDT	移行済み。一部の設定項目が移行されました

#### (1) 移行状況で「移行なし」の場合

移植先のデバイスで LVD0 がサポートされていないため、サンプルコードの移植はできません。移植先のデバイスで類似の機能がないか確認してください。例えば、RL78/G15 では、セレクトラブル・パワーオン・リセット回路 (SPOR) で電源検出を行います。

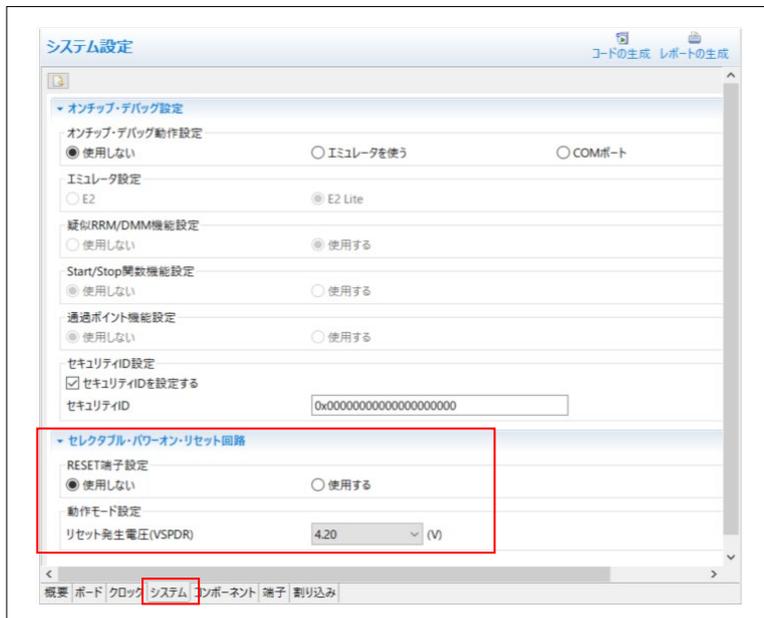
SC の「システム」タブを表示します。

図 2-63 「システム」タブの選択



リセット発生電圧を変更後のデバイスに合わせて設定します。

図 2-64 「システム」タブ (RL78/G15 の場合)



注意. 移植先の RL78 製品の電気的特性を満たすように回路設計してください。

(2) 移行状況で一部の設定項目が移行された場合

一部の設定項目が移行されたコンポーネントに対して移行状況の詳細を確認します。移行できなかった設定がサンプルコードの動作に影響がないか確認します。設定値の変更等で対応できる場合には、該当するコンポーネントの設定画面で設定値を見直します。サンプルコードの動作に影響がある場合、サンプルコードの移植はできません。新規にプログラムを開発してください。

図 2-65 一部の設定項目が移行されたコンポーネント例 : Config\_COMP0

4.2 Config\_COMP0

移行済み。一部の設定項目が移行されました

Table 4-2-1 設定移行状況

設定	値(変更前のデバイス)	値(変更後のデバイス)	移行状況
速度設定	低速	低速	移行済み。
基準電圧	IVREF0	IVREF0	移行済み。
電圧範囲	0 ~ VDD-1.4V	-	移行なし。設定が存在しない
エッジ 設定	立ち上がりエッジ	立ち上がりエッジ	移行済み。
デジタル・フィルタ許可	チェックを外す	チェックを外す	移行済み。
サンプリング・クロック	fCLK	fCLK	移行済み。
出力許可 (VCOUT0)	チェックを外す	チェックを外す	移行済み。
出力極性	通常	通常	移行済み。
コンパレータ 0 割り込み許可 (INTCMP0)	チェック	チェック	移行済み。
優先順位	レベル3(低優先順位)	レベル3(低優先順位)	移行済み。

図 2-66 一部の設定項目が移行されたコンポーネント例 : Config\_WDT

**4.4 Config\_WDT**

移行済み。一部の設定項目が移行されました

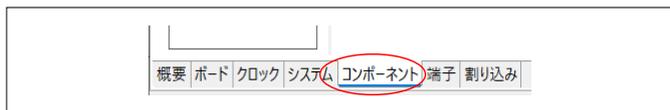
Table 4-4-1 設定移行状況

設定	値(変更前のデバイス)	値(変更後のデバイス)	移行状況
HALT/STOPモード時の動作設定	使用する	使用する	移行済み。
オーバーフロー時間	2 <sup>17</sup> /fIL	(2 <sup>16</sup> -1)/fIL	移行なし。値が存在しません
ウインドウ・オープン期間	100	-	移行なし。設定が存在しない
オーバーフロー時間の75%到達時にインターバル割り込みを発生する + 3/(4 x fIL)(INTWDTI)	チェック	チェック	移行済み。
優先順位	レベル3(低優先順位)	レベル3(低優先順位)	移行済み。

各コンポーネントの設定画面の表示方法を説明します。

SCの「コンポーネント」タブを表示します。

図 2-67 「コンポーネント」タブの選択



左側に使用しているコンポーネント（周辺機能）がツリーに表示されます。ツリー内のコンポーネントをクリックすると左側に設定画面が表示されます。

図 2-68 コンポーネントの設定画面の表示

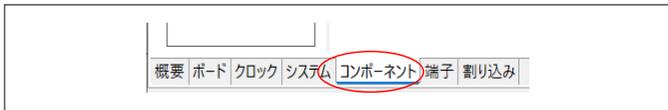


(3) BSP の更新

SC で使用するボードサポートパッケージ (BSP) を最新バージョンに更新します。

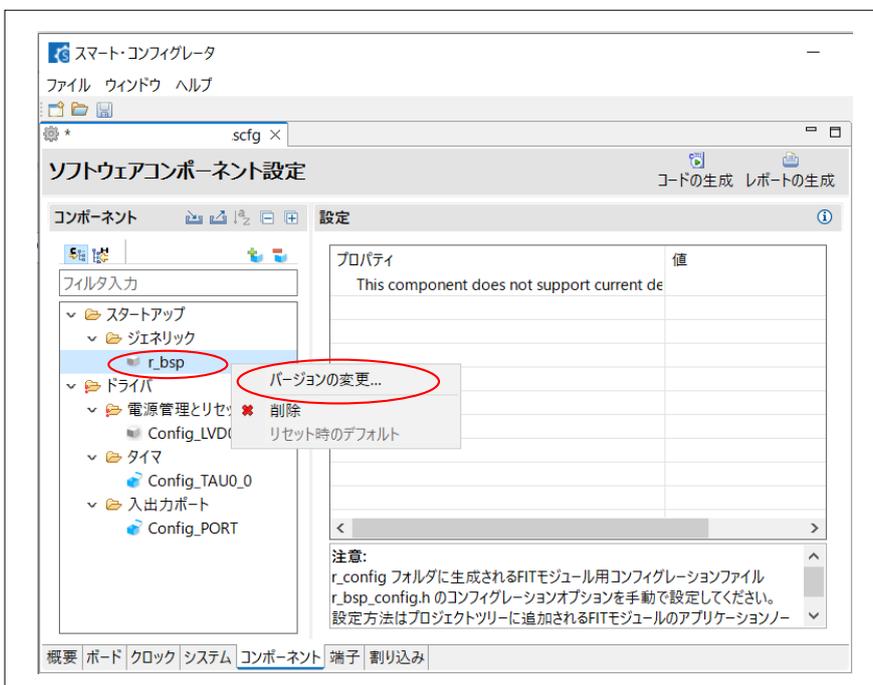
SC の「コンポーネント」タブを表示します。

図 2-69 「コンポーネント」タブの選択



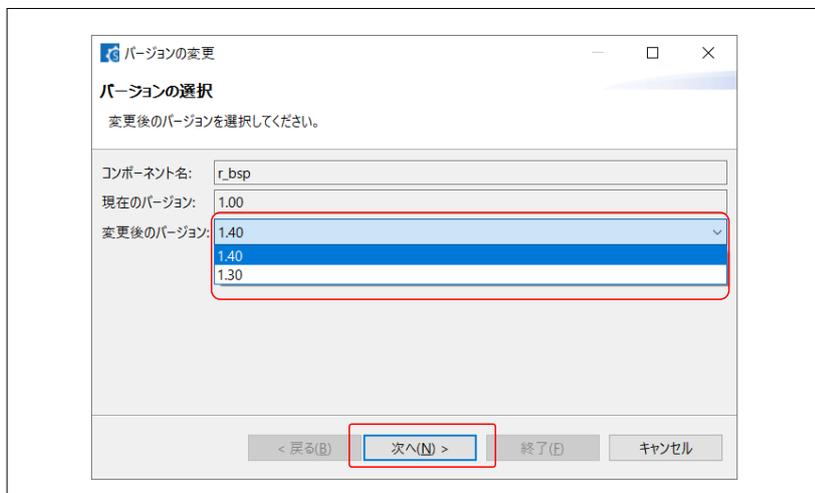
「r\_bsp」を右クリックし、「バージョンの変更」をクリックします。

図 2-70 BSP のバージョン変更



「バージョン変更」ダイアログで、「変更後のバージョン」のドロップダウンリストに表示される最新のバージョンを選択し、「次へ」をクリックします。

図 2-71 「バージョンの変更」ダイアログ：バージョンの選択



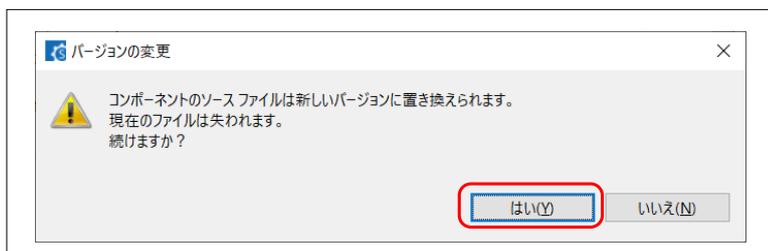
「終了」をクリックします。

図 2-72 「バージョンの変更」ダイアログ：設定の変更内容



「はい」をクリックします。最新バージョンの BSP のファイルが生成されます。

図 2-73 「バージョンの変更」ダイアログ：確認



CS+ for CC の場合、デバイス変更後の SC 設定内容に合わせて CC-RL のオプション設定が変更されます。「OK」をクリックします。e2 studio の場合、CC-RL のオプションが変更されますが下記のダイアログは表示されません。また、IAR EW for Renesas RL78 環境の場合、下記のダイアログは表示されず、設定の変更は BSP ファイルに反映されます。

図 2-74 「リンカオプション変更の確認」ダイアログ



### 2.10.5 端子設定

デバイス変更レポートの変換状況の「成功／変換なし」にかかわらず、使用するすべての端子について、SCの「端子」－「端子番号」タブで設定状況を確認してください。

図 2-75 デバイス変更レポート例：端子設定

## 3 端子

端子割り当てと変換状況

Table 3-1 端子マイグレーション状況

端子機能	端子番号 (変換前)	端子番号(変換後)	変換状況
TOOL0	5	1	変換成功
INTP0	9	8	変換成功
(1) P50	33	-	変換なし、端子機能が使用できません
(2) TxD0	44	6	変換成功
(2) TI00	62	6	変換成功

#### (1) 変換状況で「変換なし」の場合

図 2-75 は、移行先のデバイスには P50 が存在しないため、サンプルコードを移植していないことを示しています。移植先のデバイスで別の端子を使用してください。

移植元のデバイスでは P50 を使用していたが、移植先のデバイスでは P04 を使用する場合について変更手順を説明します。

「2.7 SC のレポート生成」で生成したレポートを開きます。Config\_PORT に表示されているポート設定の内容を確認します。P50 は、「出力」、「1 を出力」に設定されています。

図 2-76 SC のレポート例：ポート設定

### 5.5. Config\_PORT

コンポーネント: ポート

使用状況: 使用

バージョン: 1.3.0

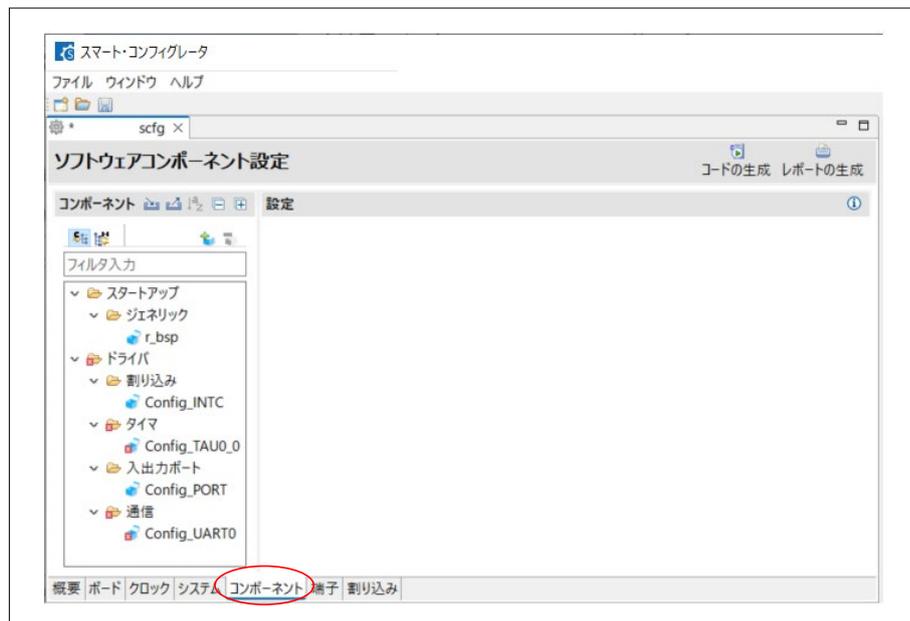
---

P50:

設定名	値
使用状況	出力
内蔵プルアップ	チェックを外す
入力バッファオフ	チェックを外す
N-ch	チェックを外す
1 を出力	チェック
出力電流設定	Hi-Z

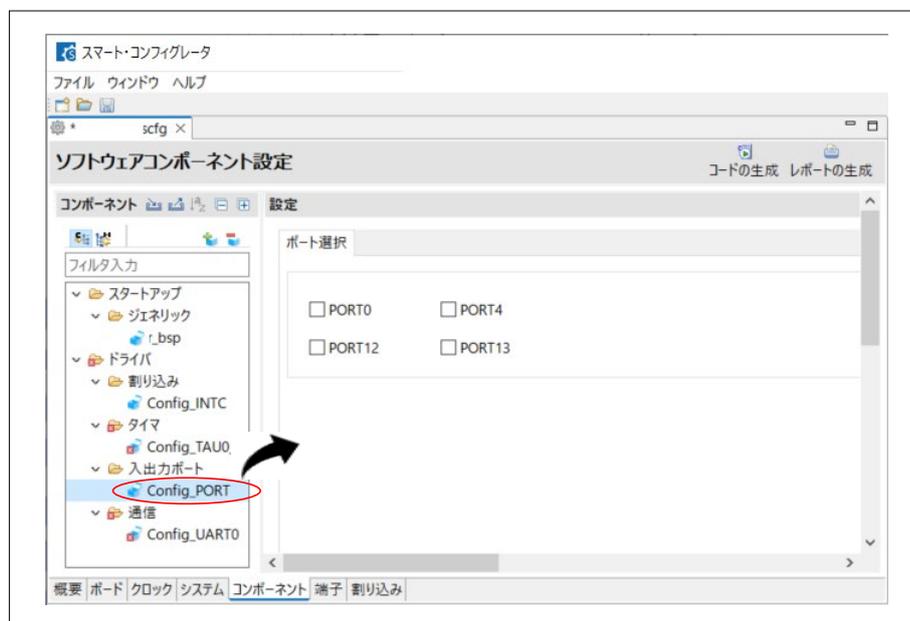
SC の「コンポーネント」タブを表示します。

図 2-77 「コンポーネント」タブ



「Config\_PORT」をクリックします。

図 2-78 「ポート」コンポーネント



「PORT0」をチェックし、P04 を「出力」、「1 を出力」に設定します。

図 2-79 ポートの設定 (1/2)

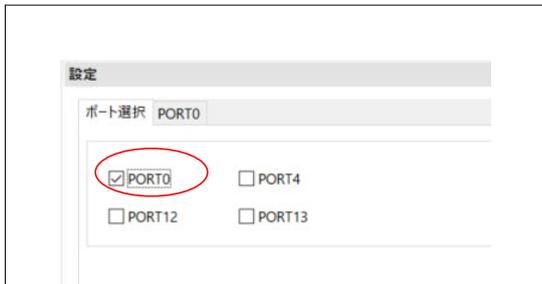
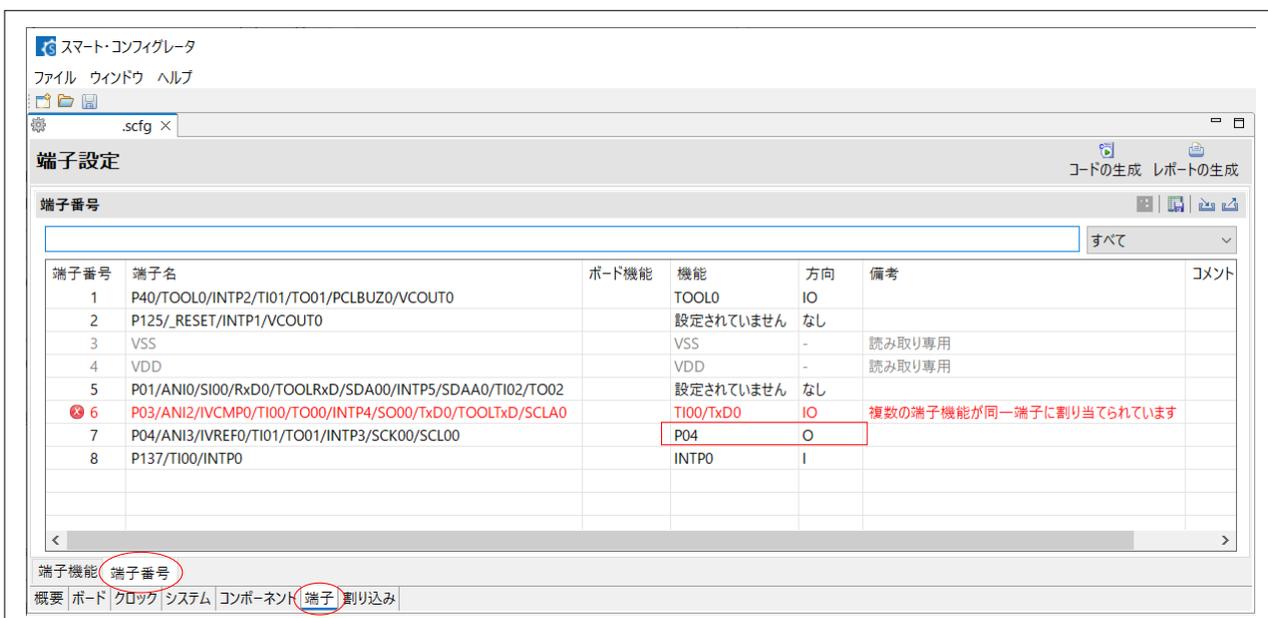


図 2-80 ポートの設定 (2/2)



「端子」 - 「端子番号」タブで、P04 が割り当てられていることを確認します。

図 2-81 「端子」 - 「端子番号」タブ



(2) 変換状況で「変換成功」の場合

変換が成功した端子で、同じ端子番号（重複）になっていないか確認してください。端子が重複している場合は、別の端子に割り当てを変更します。別の端子に割り当てができない場合は、サンプルコードの移植はできません。システムの仕様変更または移植先のデバイス変更が必要になります。なお、重複がない場合は、以下の設定は不要です。

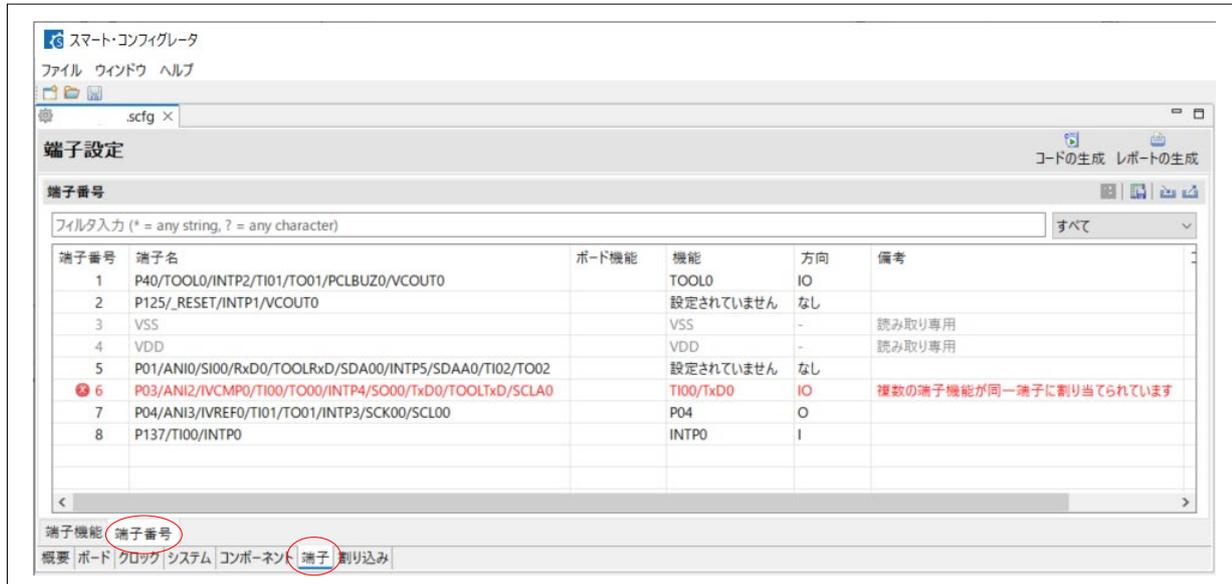
図 2-82 デバイス変更レポート例：端子設定（同じ端子番号）

Table 3-1 端子マイグレーション状況

端子機能	端子番号 (変換前)	端子番号(変換後)	変換状況
TOOL0	5	1	変換成功
INTP0	9	8	変換成功
P50	33	-	変換なし、端子機能が使用できません
TxD0	44	6	変換成功
TI00	62	6	変換成功

SCの「端子」 - 「端子番号」タブを表示します。

図 2-83 「端子」 - 「端子番号」タブ：重複あり



TI00 または TxD0 を別の端子に変更できるか確認します。フィルタ入力に TI00 または TxD0 を入力して使用できる端子番号を調べます。

図 2-84 端子検索 (1)

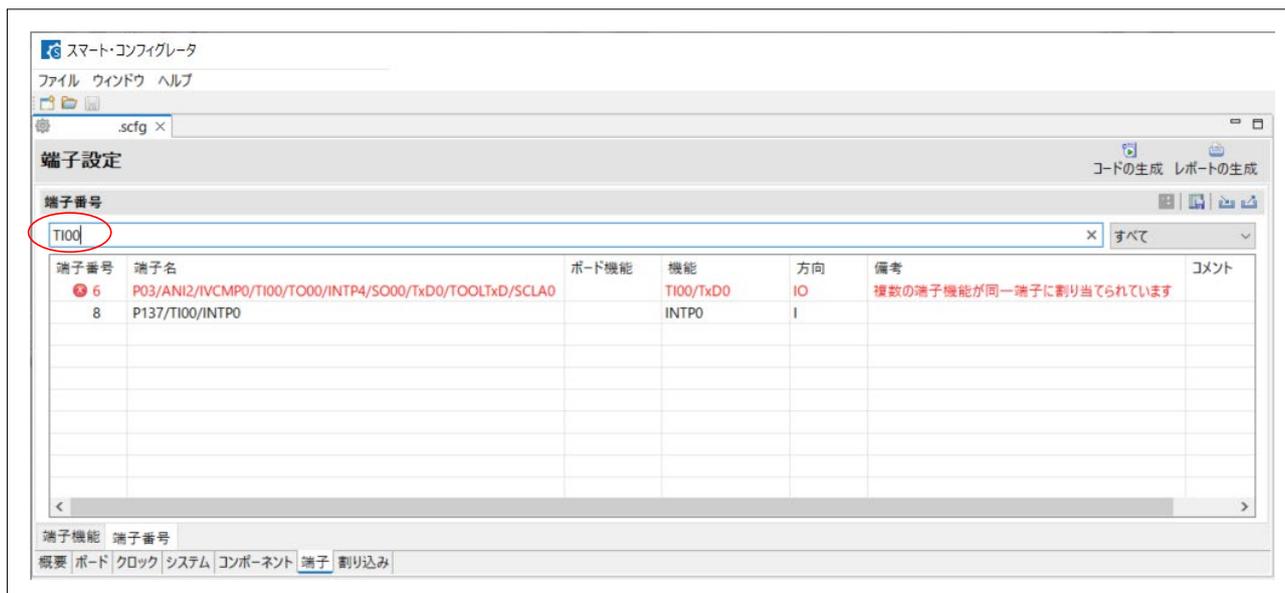
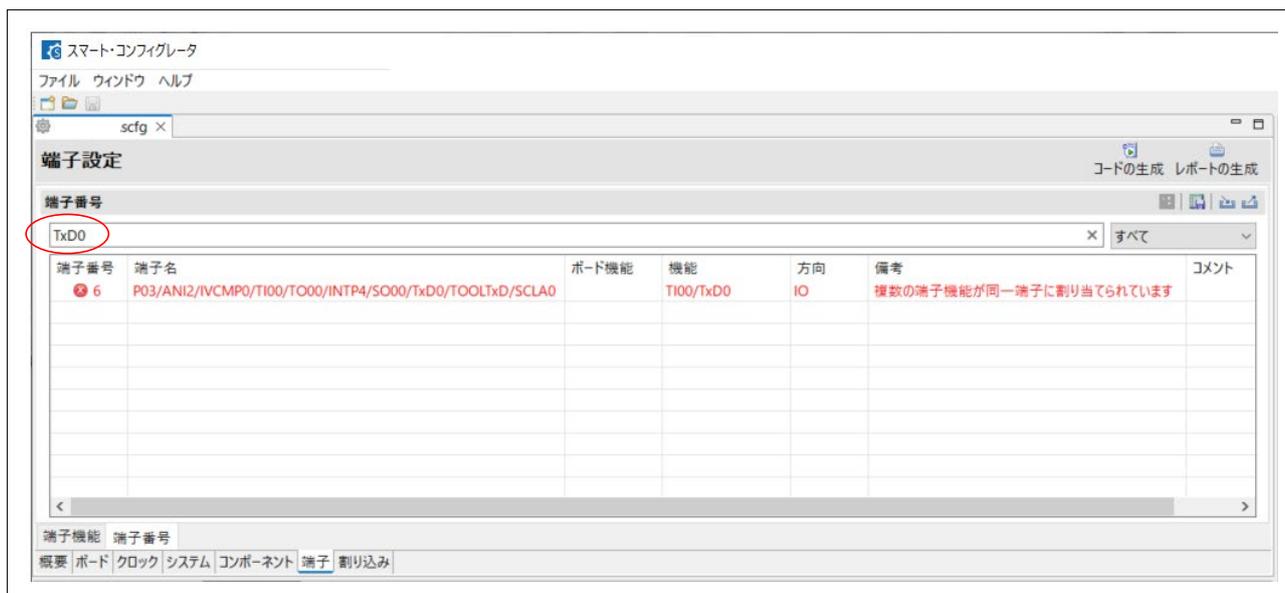


図 2-85 端子検索 (2)



TI00 は、端子番号 6 と 8 に割り当てできますが、端子番号 8 は既に INTPO で使用されています。TxD0 は、端子番号 6 のみ割り当てできます。TI00、TxD0、INTPO を同時に使用することができないため、同じ機能がある端子に変更します。

- ・ 変更例 1 : INTPO を別の INTPx に変更する。
- ・ 変更例 2 : TI00 を別の Tlxx に変更する。

(a) 変更例 1

INTP0 を別の INTPx に変更できるか確認します。「端子」 - 「端子番号」タブのフィルタ入力に「INTP」を入力します。端子番号 2 に INTP1 (未使用) があり、端子番号 5 に INTP5 (未使用) があることがわかります。

図 2-86 端子検索 (3)



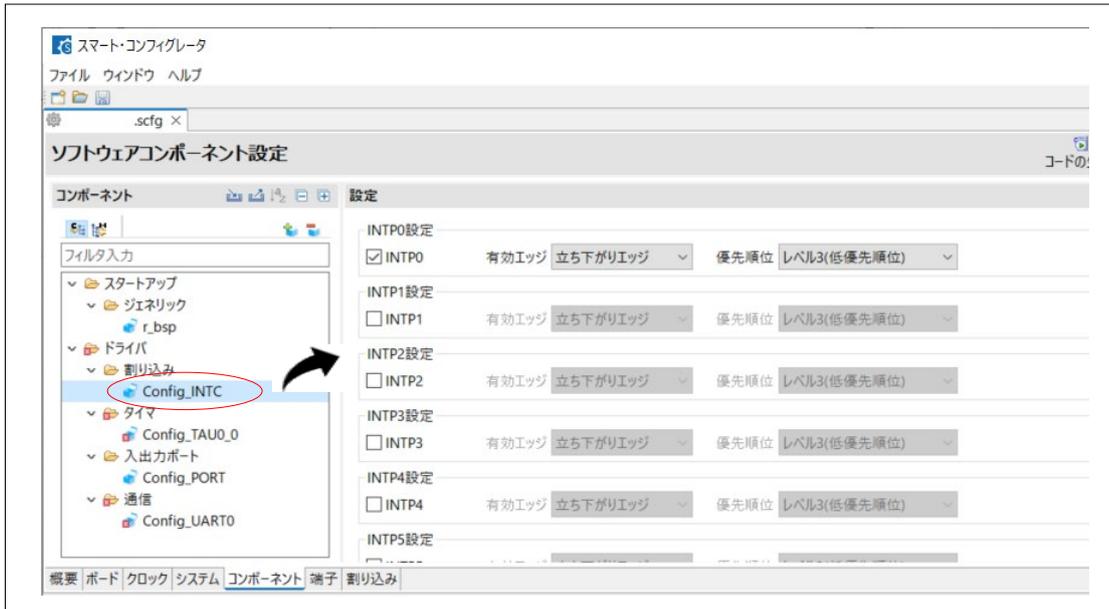
INTP0 を INTP5 に変更します。「コンポーネント」タブを表示します。

図 2-87 「コンポーネント」タブの選択



「Config\_INTC」をクリックします。

図 2-88 「外部割り込み」コンポーネント



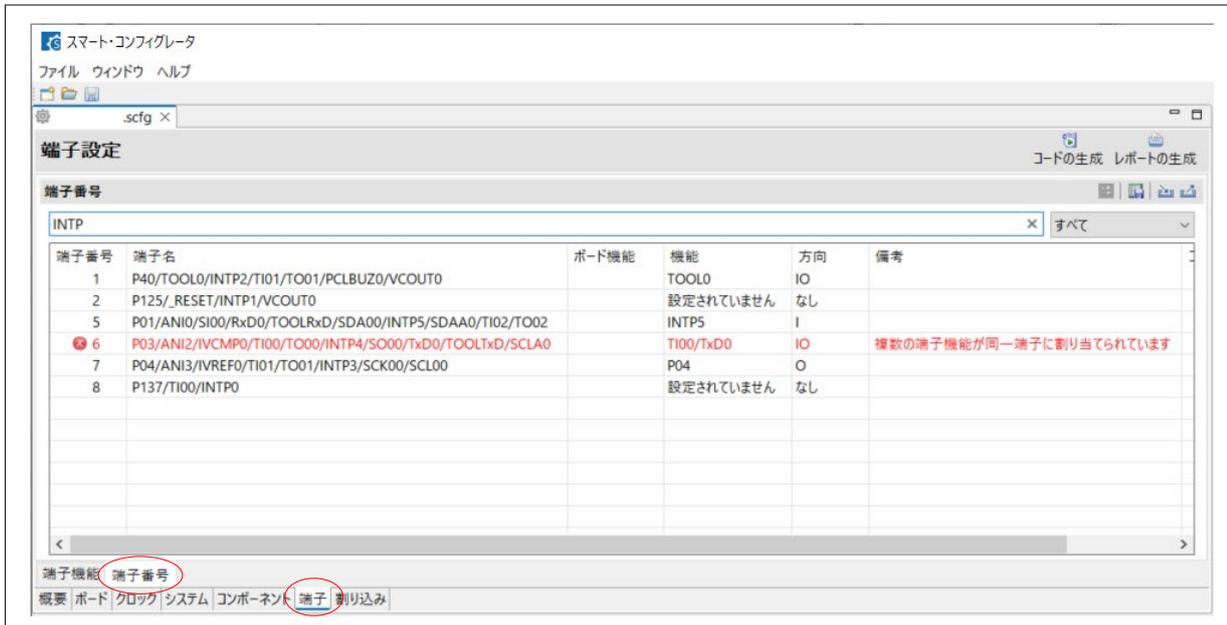
INTP0 のチェックを外し、INTP5 をチェックして INTP0 と同じ設定をします。

図 2-89 外部割り込み設定の変更



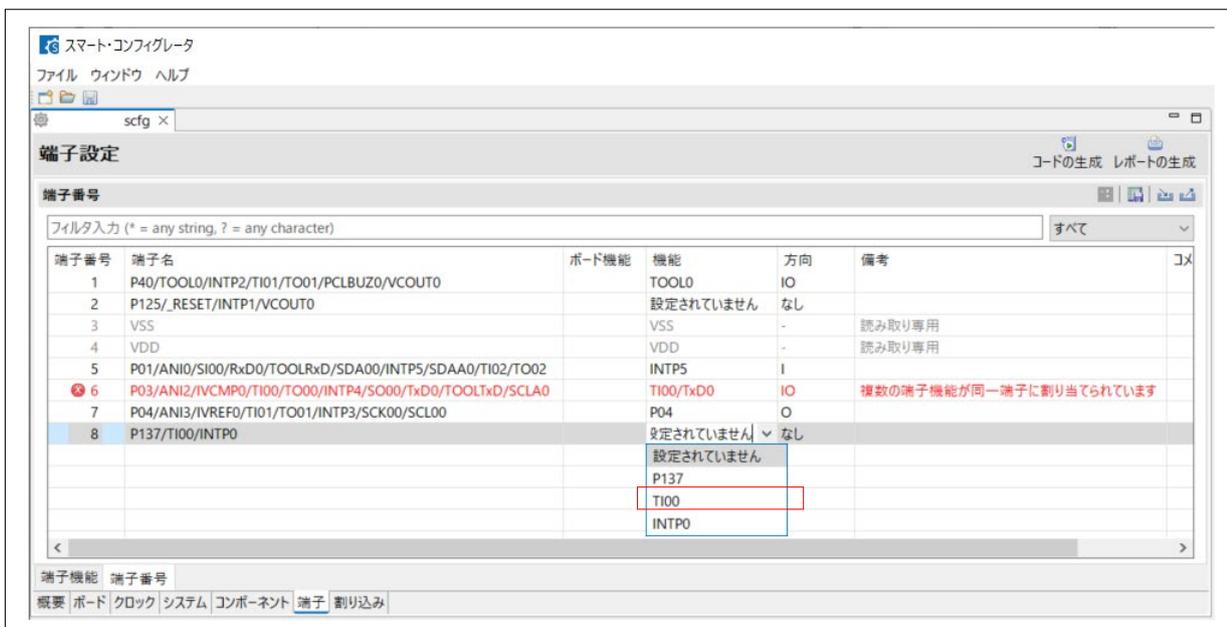
「端子」 - 「端子番号」 タブを表示します。

図 2-90 「端子」 - 「端子番号」 タブ：重複あり



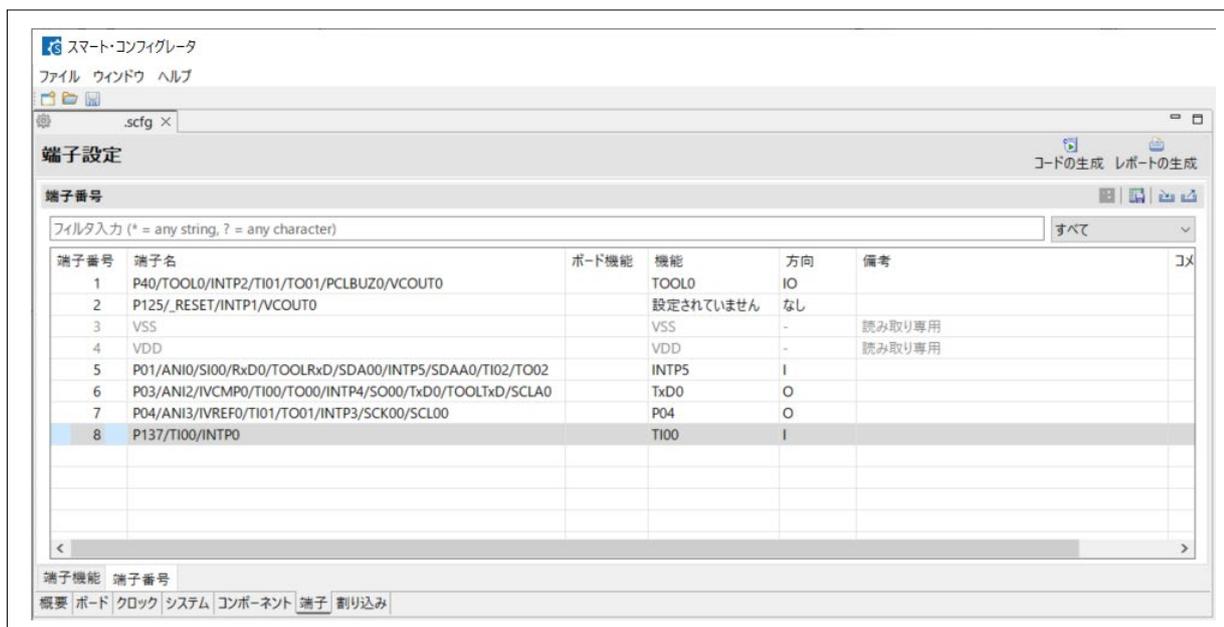
未使用に変更された端子番号 8 に TI00 を割り当てます。端子番号 8 の「機能」欄で「TI00」を選択します。

図 2-91 「端子」 - 「端子番号」 タブ：端子番号の機能変更



重複がなくなったことを確認します。(フィルタ入力欄は空欄にします。)

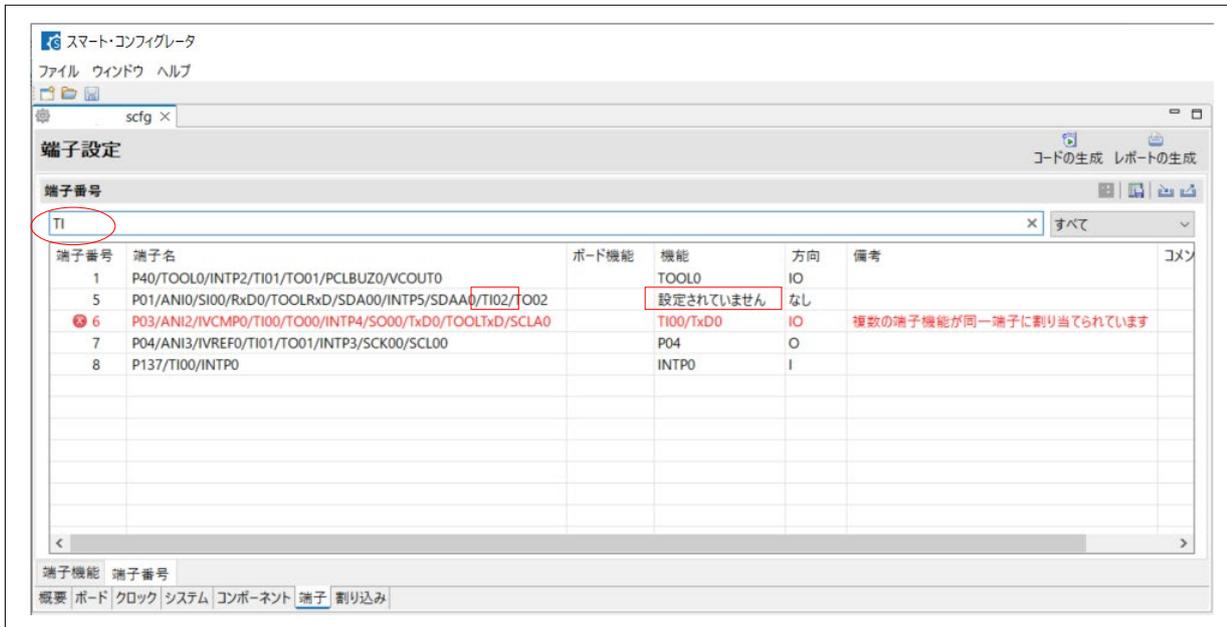
図 2-92 「端子」 - 「端子番号」タブ: 重複なし



(b) 変更例 2

TI00 を別の TIxx に変更できるか確認します。「端子」 - 「端子番号」タブのフィルタ入力に「TI」を入力します。端子番号 5 に TI02 (未使用) があることがわかります。

図 2-93 端子検索 (4)



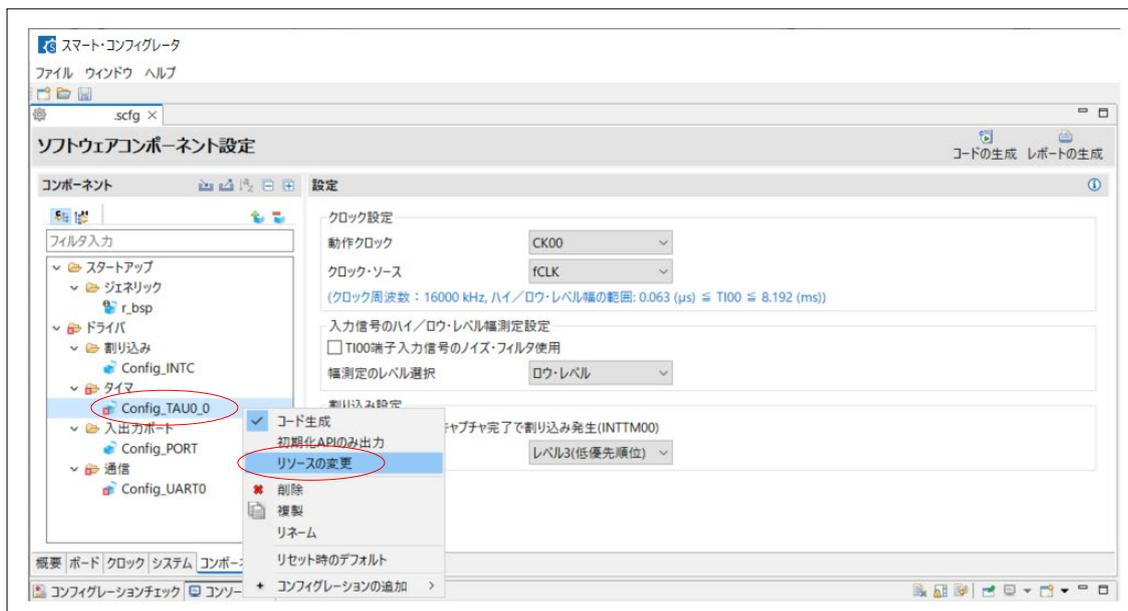
TI00 を TI02 に変更します。「コンポーネント」タブを表示します。

図 2-94 「コンポーネント」タブの選択



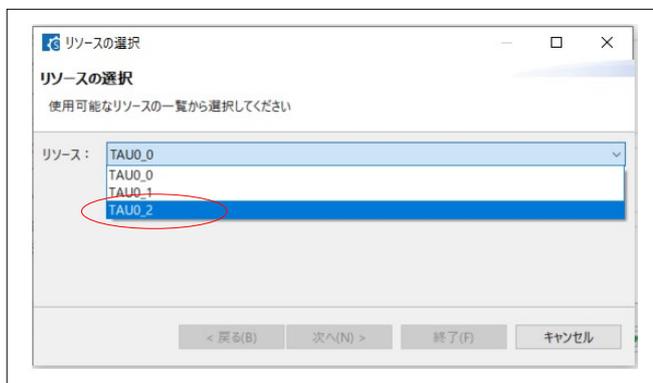
「Config\_TAU0」を右クリックし、「リソースの変更」をクリックします。

図 2-95 リソースの変更



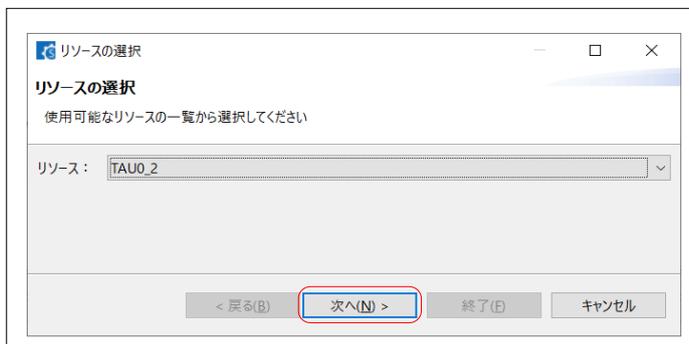
「リソース選択」ダイアログの「リソース:」で「TAU0\_2」を選択します。

図 2-96 「リソースの選択」ダイアログ (1/3)



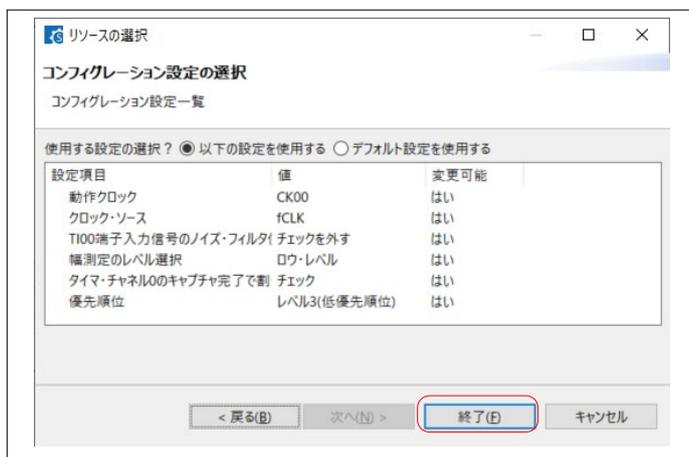
「次へ」をクリックします。

図 2-97 「リソースの選択」ダイアログ (2/3)



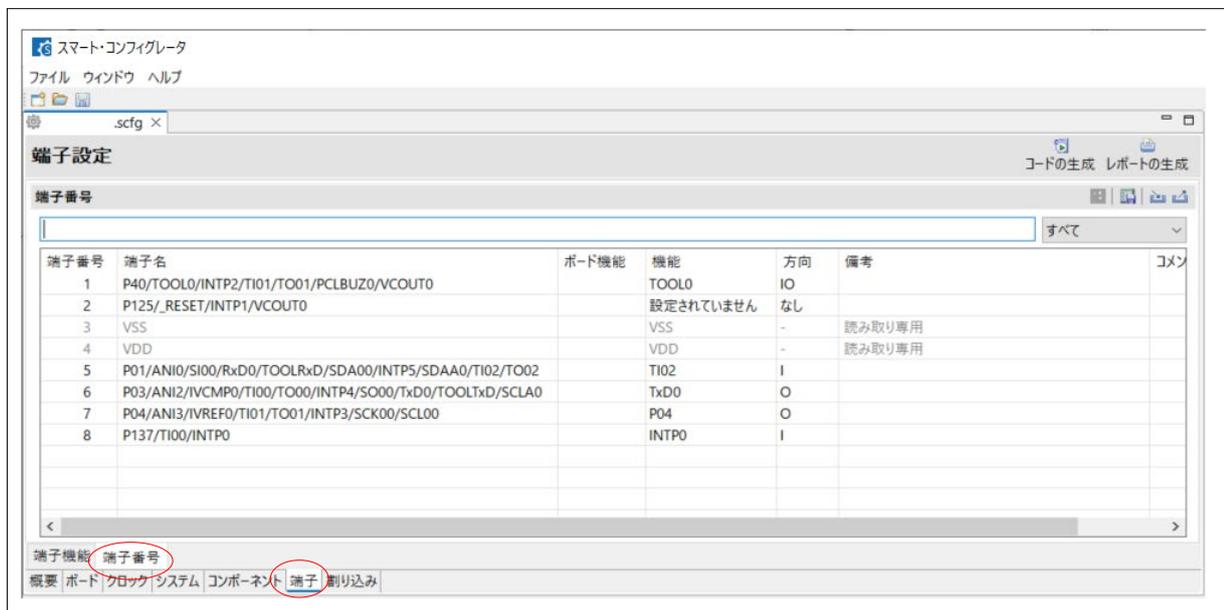
設定を変更せずに「終了」をクリックします。

図 2-98 「リソースの選択」ダイアログ (3/3)



「端子」 - 「端子番号」タブで、重複がなくなったことを確認します。(フィルタ入力欄は空欄にします。)

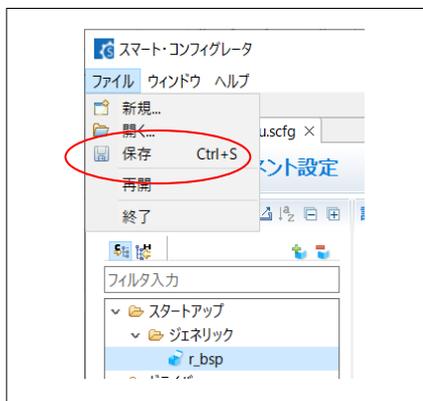
図 2-99 「端子」 - 「端子番号」タブ：重複なし



## 2.11 SC の設定保存

スマート・コンフィグレータ (SC) の変更内容を保存します。SC の「ファイル」 - 「保存」メニューをクリックします。(e<sup>2</sup> studio 環境の場合、e<sup>2</sup> studio の「ファイル」 - 「保存」メニュー)

図 2-100 スマート・コンフィグレータの保存



## 2.12 コード生成

デバイス変更、設定/リソース変更後、スマート・コンフィグレータ (SC) でコードを生成します。SC の「コードの生成」をクリックします。

図 2-101 コードの生成



## 2.13 コード編集

「2.5 コード変更の確認」でスマート・コンフィグレータ (SC) の生成コードが変更されている場合、もしくは「2.10 設定/リソースの変更」で変更した場合、生成コードの修正が必要です。

<編集が必要な箇所>

- (1) 2.5 で、SC 生成コードをコメントアウトするなど変更している箇所  
→ 移植先プロジェクトの該当ファイルで、移植前と同様の変更をする。
- (2) 2.10 で、使用する I/O ポートを変更した場合、I/O ポートの入出力値を扱っているコード  
→ 移植プロジェクトのファイル内で、変更後の I/O ポート名に変更する。
- (3) 2.10 で、使用する周辺機能のチャンネルを変更した場合、周辺機能の動作許可/停止などの SC 生成の API 関数を呼び出している箇所  
→ 移植先プロジェクトの生成ファイル内で、変更後のチャンネル用 API 関数を呼び出しに変更する。  
ただし、API 関数本体は「2.12 コード生成」で変更後のチャンネルに対応したコードに置き換えられているので編集不要です。

ただし、(2)と(3)の編集対象は、main.c または SC 生成ファイルのうち、図 1-2 で示すユーザコード用コメント内のコードに限ります。

以降に、(2)、(3)の処置例を説明します。

## (2) I/O ポートを変更した場合

変更例：変更前のデバイスでは P52 を使用、変更後のデバイスで P04 を使用する変更を行った。

移植先プロジェクトのファイルで、P52 の入出力を行っている箇所を確認します。検索文字列を「P5」として検索します。

検索結果のうち、main.c または SC 生成ファイルのユーザコード用コメント内の P52 に対するコードを P04 に合わせて編集します。

図 2-102 コードの修正：ポート

```

static void __near r_Config_TAU0_0_channel0_interrupt(void)
{
  /* Start user code for r_Config_TAU0_0_channel0_interrupt. Do not edit comment generated here */
  static uint8_t s_tm00_count = 0U; /* INTTMO0 interrupt times counter */
  uint16_t temp_duty = 0U; /* Duty factor calculation */

  if (250 == (++s_tm00_count)) /* Update the duty cycle per 500ms */
  {
    s_tm00_count = 0U; /* Interrupt counter reset */
    P5 ^= 0x04; /* Invert LED2 */

    temp_duty = TDR01; /* Read the current duty cycle (duty factor) setting */
    if (temp_duty <= (_E100_TAU_TDR01_VALUE / 9))
    {
      temp_duty = _E100_TAU_TDR01_VALUE; /* If current duty cycle is 90% (duty factor is 10%), */
      /* Set duty cycle to 10% (duty factor to 90%) */
    }
    else
    {
      /* Increase duty cycle by 20% (decrease duty factor by 20%) */
      temp_duty -= ((_E100_TAU_TDR01_VALUE / 9) * 2);
    }
    TDR01 = temp_duty; /* Update duty cycle (duty factor) */
  }
  else
  {
    ;
  }
}

/* End user code. Do not edit comment generated here */
}

```

↓

```

static void __near r_Config_TAU0_0_channel0_interrupt(void)
{
  /* Start user code for r_Config_TAU0_0_channel0_interrupt. Do not edit comment generated here */
  static uint8_t s_tm00_count = 0U; /* INTTMO0 interrupt times counter */
  uint16_t temp_duty = 0U; /* Duty factor calculation */

  if (250 == (++s_tm00_count)) /* Update the duty cycle per 500ms */
  {
    s_tm00_count = 0U; /* Interrupt counter reset */
    P0 ^= 0x10; /* Invert LED2 */

    temp_duty = TDR01; /* Read the current duty cycle (duty factor) setting */
    if (temp_duty <= (_E100_TAU_TDR01_VALUE / 9))
    {
      temp_duty = _E100_TAU_TDR01_VALUE; /* If current duty cycle is 90% (duty factor is 10%), */
      /* Set duty cycle to 10% (duty factor to 90%) */
    }
    else
    {
      /* Increase duty cycle by 20% (decrease duty factor by 20%) */
      temp_duty -= ((_E100_TAU_TDR01_VALUE / 9) * 2);
    }
    TDR01 = temp_duty; /* Update duty cycle (duty factor) */
  }
  else
  {
    ;
  }
}

/* End user code. Do not edit comment generated here */
}

```

編集後、ファイルを保存します。

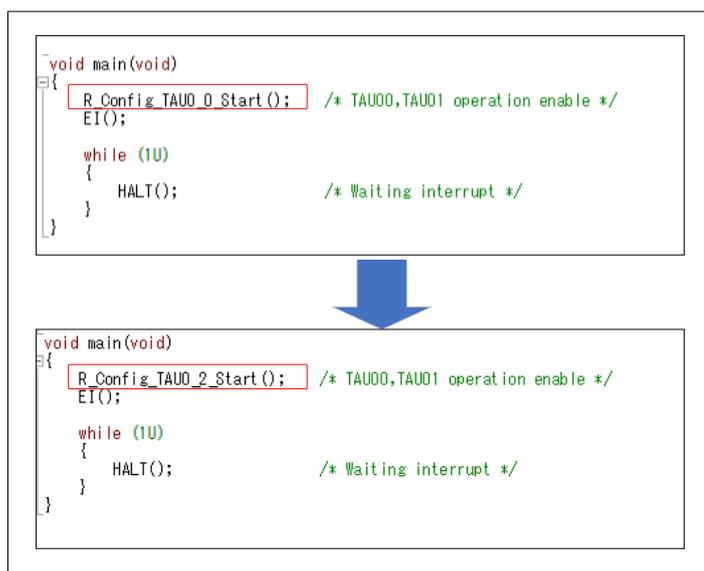
## (3) 周辺機能のチャンネルを変更した場合

変更例：変更前のデバイスでは TAU0\_0 (TAU のチャンネル 0) を使用、変更後のデバイスで TAU0\_2 を使用する変更を行った。

移植先プロジェクトのファイルで、TAU0\_0 の API をコールしている箇所を確認します。検索文字列を「R\_Config\_TAU0\_0\_」として検索します。検索時、大文字/小文字の区別および完全一致のオプションは外します。

検索結果のうち、main.c または SC 生成ファイルのユーザコード用コメント内の R\_Config\_TAU0\_0\_ から始まる API 関数コールを TAU0\_2 に合わせて編集します。

図 2-103 コードの修正 : main()



編集後、ファイルを保存します。

**参考： IDE での文字列検索方法**

CS+ for CC の場合

CS+ for CC の「編集」－「検索」メニューをクリックします。

「検索・置換」ダイアログの「一括検索」タブで、下記の指定し「検索」をクリックします。

図 2-104 CS+ for CC 「検索・置換」ダイアログ



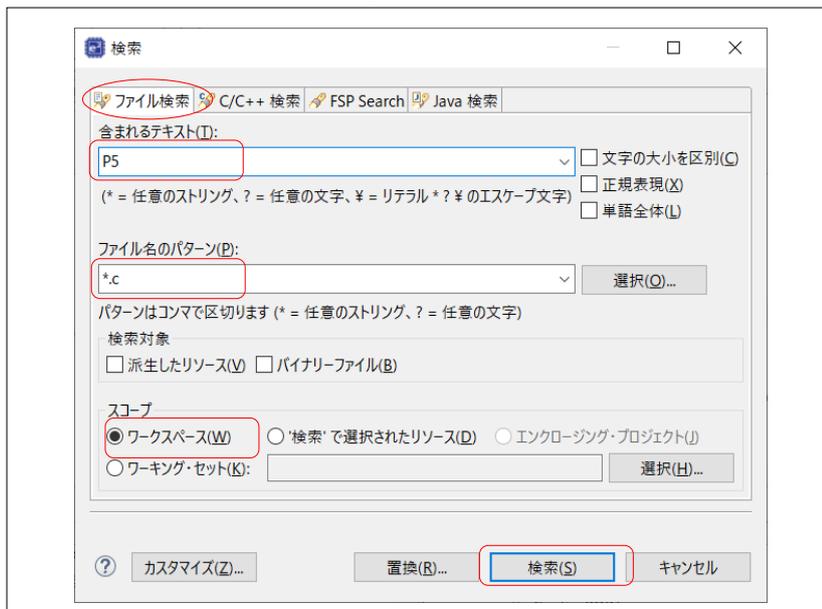
「出力」タブに結果が表示されます。

e<sup>2</sup> studio の場合

e<sup>2</sup> studio の「検索」－「検索」メニューをクリックします。

「検索」ダイアログの「ファイル検索」タブで、下記の指定をし「検索」をクリックします。

図 2-105 e<sup>2</sup> studio 「検索」ダイアログ

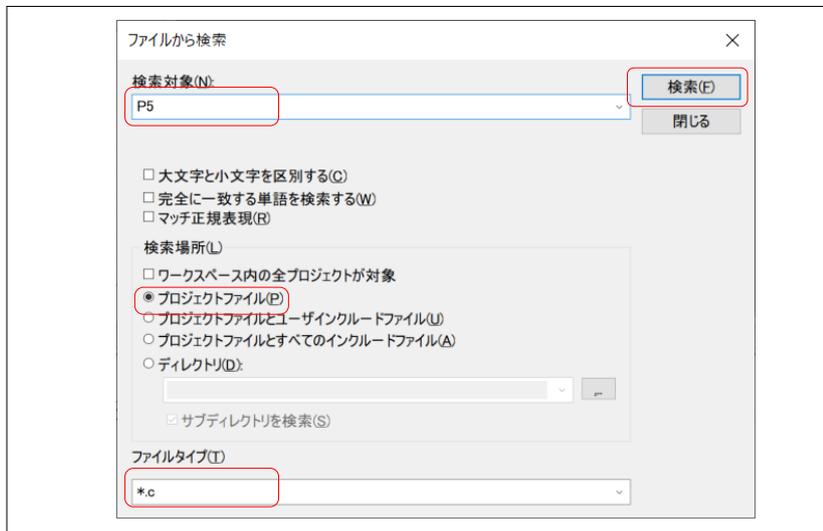


「検索」タブに結果が表示されます。

## IAR EW for Renesas RL78 の場合

IAR EW for Renesas RL78 の「編集」－「検索と置換」－「ファイルから検索」メニューをクリックします。「検索」ダイアログの「ファイル検索」タブで、下記の指定をし「検索」をクリックします。

図 2-106 IAR EW for Renesas RL78 「ファイルから検索」ダイアログ



「ファイルから検索」タブに結果が表示されます。

## 2.14 移植先プロジェクトのビルド

移植先プロジェクトをビルドします。

CS+ for CC の場合 :

「ビルド」→「リビルド・プロジェクト」メニューを選択

e<sup>2</sup> studio の場合 :

「プロジェクト」→「プロジェクトのビルド」メニューを選択

IAR EW for Renesas RL78 の場合 :

「プロジェクト」→「すべてを再ビルド」メニューを選択

注意. エラーが出た場合は、エラーメッセージを元に対応してください。

### 3. 移植例

この章では、サンプルコード移植の具体例を説明します。以下に移植例に使用するアプリケーションノートを示します。

表 3-1 移植例

項目	内容
アプリケーションノート (サンプルコード)	RL78/G23 タイマ・アレイ・ユニット (PWM 出力) (R01AN5874JJ0100)
移植先デバイス	RL78/G15 (R5F12068)
統合開発環境	ルネサス エレクトロニクス製 CS+ for CC V8.09.00
C コンパイラ	ルネサス エレクトロニクス製 CC-RL V1.12.00
Smart Configurator	ルネサス エレクトロニクス製 Smart Configurator for RL78 V1.5.0
移植時の作業概要	コンポーネントの変更、I/O ポートの変更を伴う移植例

#### 3.1 事前準備

RL78/G23 用アプリケーションノートの「2. 動作確認条件」、「表 1-1 使用する周辺機能と用途」、「3.2 使用端子一覧」および「3.1 ハードウェア構成例」を参照し使用しているリソースを確認します。RL78/G15 へ移植時の構成例を以下に示します。

- 動作確認条件

差分を赤字で示します。RL78/G15 は 32MHz 動作できないため、動作周波数を 16MHz に変更します。動作周波数の変更による各種レジスタ設定値の変更はスマート・コンフィグレータ (SC) が対応するため、使用するサンプルコードの動作については影響ありません。また、RL78/G15 では電圧検出回路 (LVD) がないため、セレクトラブル・パワーオン・リセット回路 (SPOR) に変更します。

表 3-2 動作確認条件

項目	RL78/G23	RL78/G15
使用マイコン	RL78/G23 (R7F100GLG)	RL78/G15 (R5F12068)
使用ボード	RL78/G23 Fast Prototyping Board (RTK7RLG230CLG000BJ)	RL78/G15-20p Fast Prototyping Board (RTK5RLG150CLG000BJ)
動作周波数	高速オンチップ・オシレータ・クロック : 32MHz CPU/周辺ハードウェア・クロック : 32MHz	高速オンチップ・オシレータ・クロック : <b>16MHz</b> CPU/周辺ハードウェア・クロック : <b>16MHz</b>
動作電圧	3.3V (1.8V~5.5V で動作可能) LVD0 検出電圧: リセット・モード 立ち上がり時 TYP. 1.90 V (1.84 V ~ 1.95 V) 立ち下がり時 TYP. 1.86 V (1.80 V ~ 1.91 V)	<b>5.0V (2.4V~5.5V で動作可能)</b> <b>SPOR 検出電圧</b> 立ち上がり時 TYP. <b>2.57 V (2.44 V ~ 2.68 V)</b> 立ち下がり時 TYP. <b>2.52 V (2.40 V ~ 2.62 V)</b>

- 使用する周辺機能と用途

RL78/G23 用サンプルコードで使用している周辺機能およびチャンネルをそのまま RL78/G15 でも使用します。

表 3-3 使用する周辺機能

## RL78/G23

周辺機能	用途
タイマ・アレイ・ユニット 0 チャンネル 0, チャンネル 1	チャンネル 0 とチャンネル 1 を連動させて PWM 機能に設定し、TO01 端子から PWM 出力する

## RL78/G15

周辺機能	用途
タイマ・アレイ・ユニット 0 チャンネル 0, チャンネル 1	チャンネル 0 とチャンネル 1 を連動させて PWM 機能に設定し、TO01 端子から PWM 出力する

- 使用端子

差分を赤字で示します。RL78/G15 (R5F12068) には P52 端子が存在しません。そのため、P52 を P21 に変更します。

表 3-4 使用端子

## RL78/G23

端子名	入出力	内容
P16 / TO01	出力	PWM 出力ポート (LED1 の点灯制御)
P52	出力	LED 表示用出力ポート (LED2 の点灯制御)

## RL78/G15

端子名	入出力	内容
P04 / TO01	出力	PWM 出力ポート (LED1 の点灯制御)
<b>P21</b>	出力	LED 表示用出力ポート (LED2 の点灯制御)

● ハードウェア構成

差分を赤字で示します。RL78/G15 (R5F12068) には P52 端子が存在しません。そのため、P52 を P21 に変更します。

図 3-1 ハードウェア構成例 (RL78/G23)

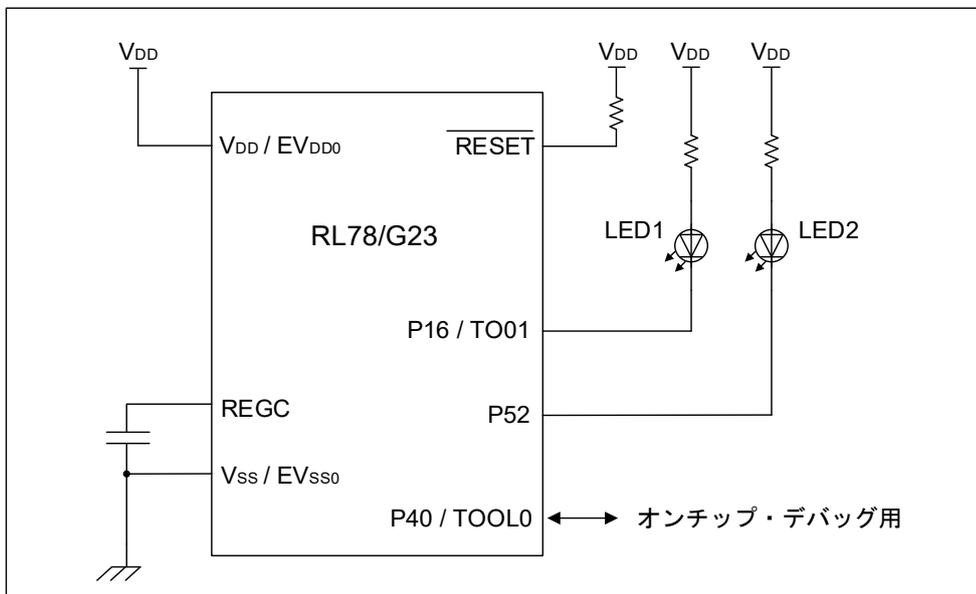
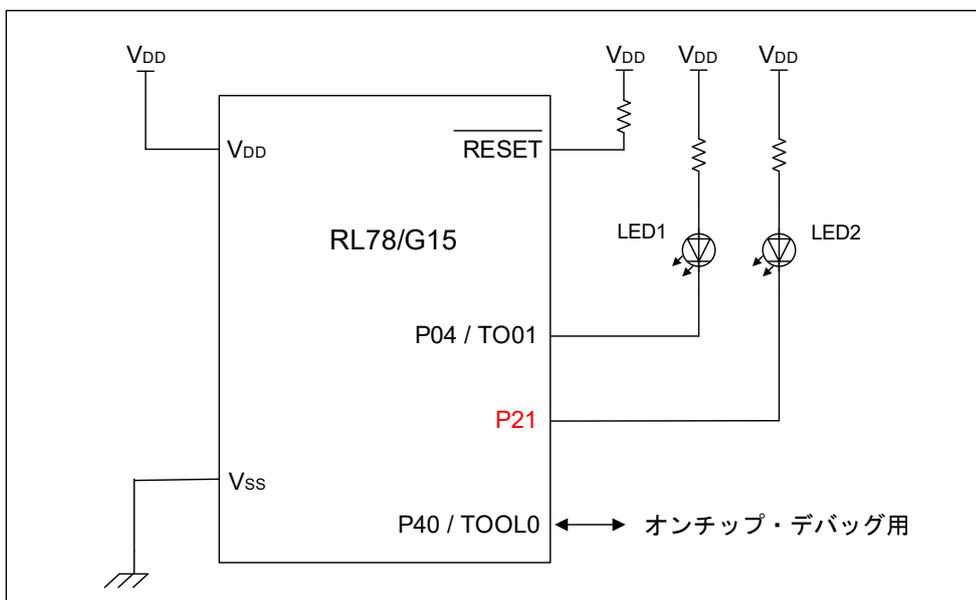


図 3-2 ハードウェア構成例 (RL78/G15)



注意 1 この回路イメージは接続の概要を示す為に簡略化しています。実際に回路を作成される場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください (入力専用ポートは個別に抵抗を介して VDD 又は VSS に接続して下さい)。

2 EVSS で始まる名前のある端子がある場合には VSS に、EVDD で始まる名前のある端子がある場合には VDD にそれぞれ接続してください。

3 VDD は LVD にて設定したリセット解除電圧 (V<sub>LVD</sub>) 以上にしてください。

### 3.2 プロジェクトのコピー

「2.2 プロジェクトのコピー」を参照し、移植プロジェクトを作成します。

### 3.3 プロジェクトの読み込み

「2.3 プロジェクトの読み込み」を参照し、使用する開発環境で移植プロジェクトを読み込みます。

### 3.4 コンパイラのバージョン変更

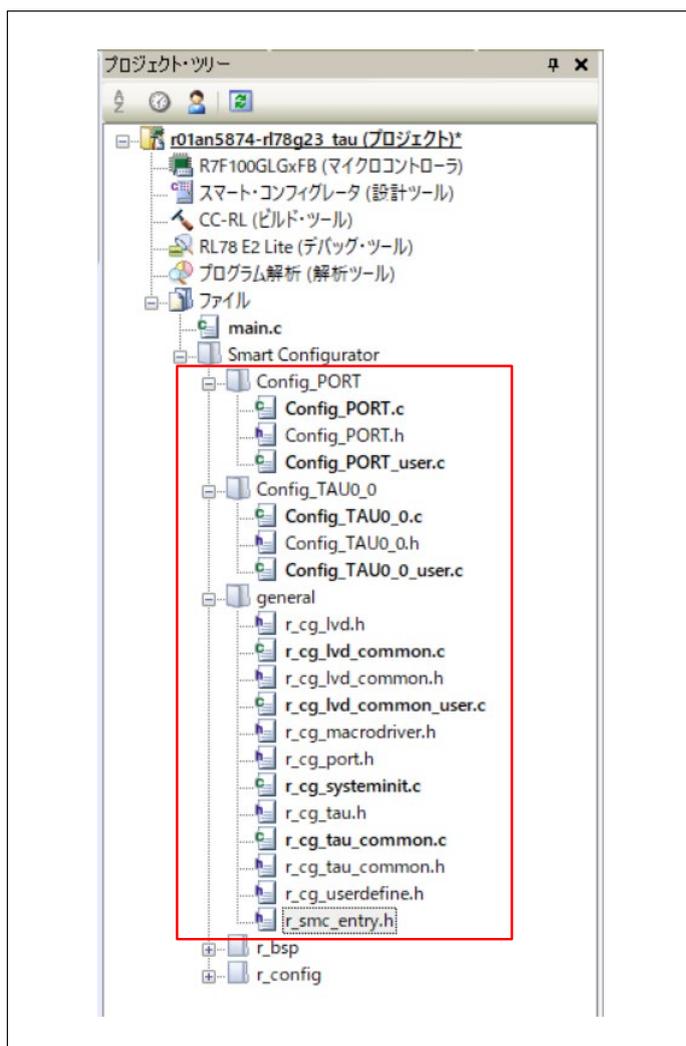
「2.4 コンパイラのバージョン変更」を参照し、使用するコンパイラのバージョンを変更します。

### 3.5 コード変更の確認

スマート・コンフィグレータ (SC) が生成したファイル内でコードを変更していないか確認します。対象のファイルは、図 3-3 で示すファイルです。エディタで対象ファイルを開き、SC 生成コードに変更を加えたことを示すコメントがないか確認します。

本サンプルコードでは、変更した箇所はありません。

図 3-3 コード変更の確認 (CS+ for CC の画面)



### 3.6 リソースの変更

本サンプルコードでは、移植元と移植先で同じ周辺機能の同じチャンネルを使用するため、リソースの変更はしません。

### 3.7 SC レポートの生成

「2.7 SC のレポート生成」を参照し、移植元の SC の設定内容をレポートとして保存します。

### 3.8 デバイスの変更

「2.8 デバイスの変更」を参照し、移植先のデバイスとして「R5F12068xSP」へ変更します。

### 3.9 SC 設定の移行結果確認

「2.9 SC 設定の移行結果確認」を参照し、デバイス変更レポートを表示します。

### 3.10 設定／リソースの変更

デバイス変更レポートで移植元から移植先への SC の設定移行状況を確認し、「3.1 事前準備」に記載の移植の構成例に合わせて処置を行います。

#### 3.10.1 クロック設定

図 3-4 デバイス変更レポート：クロック設定 (1)

スマート・コンフィグレータデバイス変更レポート			
変更前のデバイス: R7F100GLGxFB.			
変更後のデバイス: R5F12068xSP.			
<b>1 クロック設定</b>			
移行済み。一部の設定項目が移行されました			
コンフィギュレーションの移行状況を示す表です。			
Table 1-1 ボードの供給クロック周波数の移行状況			
設定	値(変更前のデバイス)	値(変更後のデバイス)	移行状況
高速オンチップ・オシレータクロック	32	16	移行なし。値が存在しません
Table 1-2 クロック設定値の移行状況			
設定	値(変更前のデバイス)	値(変更後のデバイス)	移行状況
fIL	0.032768	0.015	移行なし。値が存在しません
fSXL	0.032768	-	移行なし。設定が存在しない
fIHP	32	-	移行なし。設定が存在しない
fMAIN	32	16	移行なし。値が存在しません
fCLK	32	16	移行なし。値が存在しません

- 本サンプルコードでは、fIL、fSXL、fIHP、fMAIN を使用していないため、「移行なし」で問題ありません。
- fCLK が 32MHz から 16MHz に変更されましたが、本サンプルコードの動作に影響はありません。
- レポートに記載はありませんが、動作電圧を移植の構成例に合わせて変更します。

SC の「クロック」タブを表示します。「VDD 設定」を「2.4V ≤ VDD ≤ 5.5V」に設定します。

図 3-5 「クロック」タブ : VDD 設定

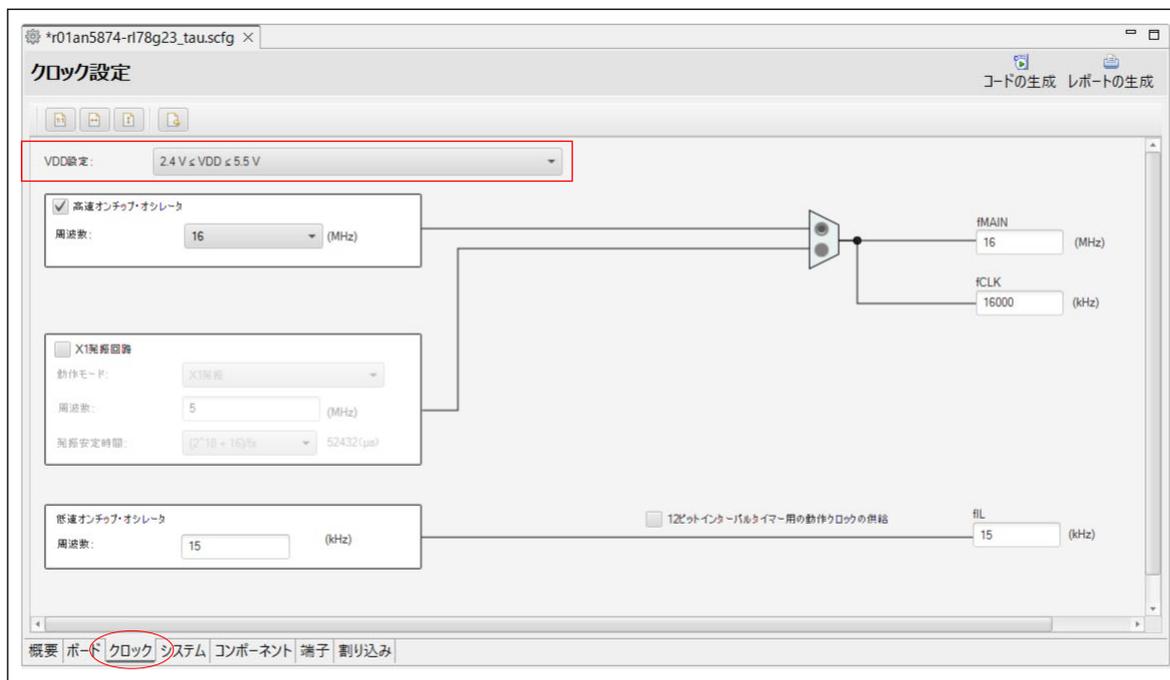


図 3-6 デバイス変更レポート : クロック設定 (2)

Table 1-3 マクロ定義値の移行状況

設定	値(変更前のデバイス)	値(変更後のデバイス)	移行状況
BSP_CFG_SUBCLK_SOURCE	0	-	移行なし。設定が存在しない
BSP_CFG_HOCO_DIVIDE	0	-	移行なし。値が存在しません
BSP_CFG_SUBSYSCLK_SOURCE	1	-	移行なし。設定が存在しない
BSP_CFG_FCLK_SOURCE	0	-	移行なし。設定が存在しない
BSP_CFG_MOCO_DIVIDE	0	-	移行なし。設定が存在しない
BSP_CFG_MOCO_SOURCE	0	-	移行なし。設定が存在しない
BSP_CFG_XT1_OSCMODE	0	-	移行なし。設定が存在しない
BSP_CFG_HISYSCLK_OPERATION	1	1	移行済み。
BSP_CFG_SUBCLK_OPERATION	1	-	移行なし。設定が存在しない
BSP_CFG_MOSC_DIVIDE	0	-	移行なし。設定が存在しない
BSP_CFG_OCOCLK_SOURCE	0	-	移行なし。設定が存在しない
BSP_CFG_HISYSCLK_SOURCE	0	0	移行済み。
BSP_CFG_FMX_START_ON_STARTUP	0	-	移行なし。設定が存在しない
BSP_CFG_WAKEUP_MODE	0	-	移行なし。設定が存在しない
BSP_CFG_FSX_START_ON_STARTUP	0	-	移行なし。設定が存在しない
BSP_CFG_ALLOW_FSUB_IN_STOPHALT	0	-	移行なし。設定が存在しない
BSP_CFG_FIH_START_ON_STARTUP	1	1	移行済み。
BSP_CFG_RTC_OUT_CLK_SOURCE	1	-	移行なし。設定が存在しない
BSP_CFG_FIM_START_ON_STARTUP	0	-	移行なし。設定が存在しない
BSP_CFG_MAINCLK_SOURCE	0	-	移行なし。値が存在しません

- Table1-3 で示される設定は BSP (ボードサポートパッケージ) の設定項目です。本移植例では、3.10.4(3) r\_bsp」で BSP を最新バージョンに変更するためここでは処置しません。

### 3.10.2 システム設定

図 3-7 デバイス変更レポート：システム設定

## 5 システム設定

移行済み。一部の設定項目が移行されました

コンフィグレーションの移行状況を示す表です。

Table 5-1 オンチップ・デバッグ設定 u79fb行状況

設定	値(変更前のデバイス)	値(変更後のデバイス)	移行状況
オンチップ・デバッグ動作設定	エミュレータを使う	エミュレータを使う	移行済み。
エミュレータ設定	E2 Lite	E2 Lite	移行済み。
疑似RRM/DMM機能設定	使用する	使用する	移行済み。
Start/Stop関数機能設定	使用しない	使用しない	移行済み。
通過ポイント機能設定	-	-	移行済み。
トレース機能設定	使用する	-	移行なし。設定が存在しない
セキュリティIDを設定する	チェック	チェック	移行済み。
セキュリティID	0x000000000000000000000000	0x000000000000000000000000	移行済み。
セキュリティID認証失敗時の設定	フラッシュ・メモリのデータを消さない	-	移行なし。設定が存在しない

- 「トレース機能」と「セキュリティ ID 認証失敗時」の設定は、RL78/G15 ではサポートされていない機能です。サンプルコードの動作には影響がないため問題ありません。

### 3.10.3 割り込み設定

サンプルコードで使用している割り込みが移行できているか確認します。サンプルコードで使用している割り込みは、「3.7 SC レポートの生成」で生成したレポートで確認します。

図 3-8 SC のレポート：割り込み設定

## 7. 割り込み設定

割り込み設定情報:

ベクタ番号	割り込み	優先レベル	バンク指定
14	INTTM00	レベル3(低優先順位)	なし

- 移植元では「INTTM00」を使用しています。

図 3-9 デバイス変更レポート：割り込み

## 2 割り込み

デバイス変更による割り込み設定の移行結果

Table 2-1 割り込みの移行結果

ベクタ番号	変換前の割り込み名	変換後の割り込み名	設定(デバイス変更前)	設定(デバイス変更後)	割り込みの移行結果
0	INTWDTI	INTWDTI	状態 = 未使用 バンク指定 = なし 優先順位 = レベル3(低優先順位)	状態 = 未使用 バンク指定 = なし 優先順位 = レベル3(低優先順位)	移行済み。
11	INTELCL	INTTM00	状態 = 未使用 バンク指定 = なし 優先順位 = レベル3(低優先順位)	状態 = 使用中 バンク指定 = なし 優先順位 = レベル3(低優先順位)	移行済み。
12	INTSMSE	INTTM01	状態 = 未使用 バンク指定 = なし 優先順位 = レベル3(低優先順位)	状態 = 未使用 バンク指定 = なし 優先順位 = レベル3(低優先順位)	移行済み。
13	INTST0	INTAD	状態 = 未使用 バンク指定 = なし 優先順位 = レベル3(低優先順位)	状態 = 未使用 バンク指定 = なし 優先順位 = レベル3(低優先順位)	移行済み。
13	INTCSI00	INTAD	状態 = 未使用 バンク指定 = なし 優先順位 = レベル3(低優先順位)	状態 = 未使用 バンク指定 = なし 優先順位 = レベル3(低優先順位)	移行済み。
13	INTIIC00	INTAD	状態 = 未使用 バンク指定 = なし 優先順位 = レベル3(低優先順位)	状態 = 未使用 バンク指定 = なし 優先順位 = レベル3(低優先順位)	移行済み。
14	INTTM00	INTP6	状態 = 使用中 バンク指定 = なし 優先順位 = レベル3(低優先順位)	状態 = 未使用 バンク指定 = なし 優先順位 = レベル3(低優先順位)	移行済み。

- 移植元のベクタ番号 14 の「INTTM00」が、移植先のベクタ 11 「INTTM00」に移行済みです。

### 3.10.4 ソフトウェアコンポーネント設定

サンプルコードで使用しているソフトウェアコンポーネントが移行できているか確認します。

図 3-10 デバイス変更レポート：ソフトウェアコンポーネントの設定

## 4 ソフトウェアコンポーネント設定

コンフィグレーションの移行状況を示す表です。

Table 4-1 コンフィグレーションの移行状況

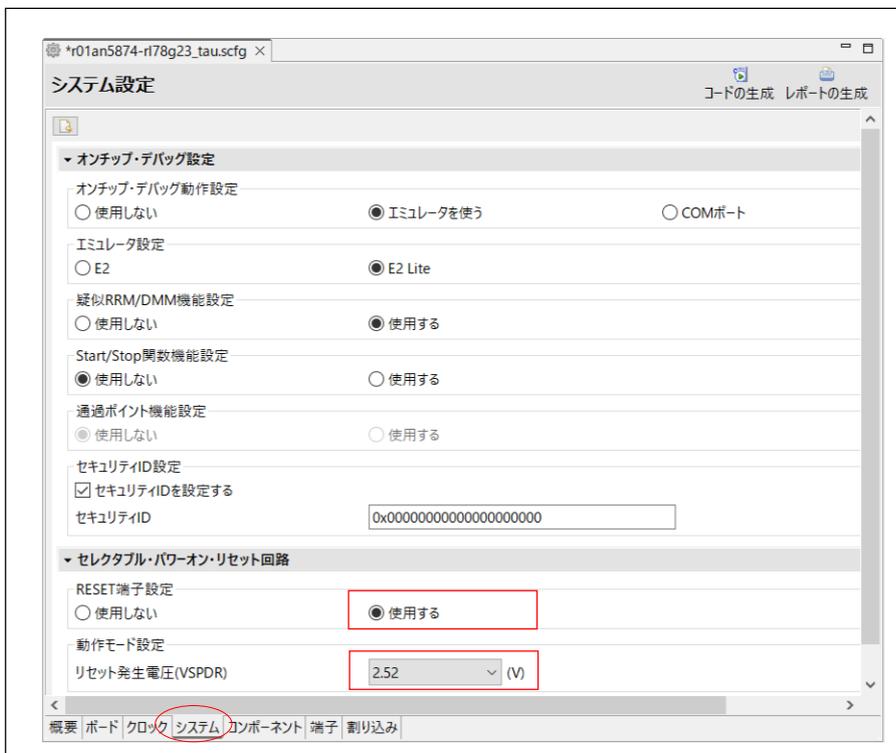
コンフィグレーション	移行状況
(3) r_bsp	移行なし。最新バージョンにアップグレードしてください
(1) Config_LVD0	移行なし。変更後のデバイスでコンポーネントがサポートされていません
Config_TAU0_0	移行済み。すべての設定項目が移行されました
(2) Config_PORT	移行済み。一部の設定項目が移行されました

#### (1) Config\_LVD0

RL78/G15 では、セレクトابل・パワーオン・リセット回路 (SPOR) で電源検出を行います。

SC の「システム」タブを表示します。

図 3-11 「システム」タブ：セレクトابل・パワーオン・リセット回路設定



- 「RESET 端子設定」で「使用する」に設定します。
- 「リセット発生電圧 (VSPDR)」に「2.52V」を設定します。

(2) Config\_PORT

ポートについては「3.10.5 端子設定」で確認を行います。

(3) r\_bsp

「2.10.4(3) BSP の更新」を参照し、BSP を最新バージョンに変更します。

3.10.5 端子設定

図 3-12 デバイス変更レポート：端子

**3 端子**

端子割り当てと変換状況

Table 3-1 端子マイグレーション状況

端子機能	端子番号 (変換前)	端子番号(変換後)	変換状況
TOOL0	5	4	変換成功
P52	35	-	変換なし、端子機能が使用できません
TO01	40	15	変換成功

- 移植の構成例に合わせて、P52（出力ポート）を P21（出力ポート）の設定に変更します。

P52 の設定内容を「3.7 SC レポートの生成」で生成したレポートで確認します。

図 3-13 SC のレポート：Config\_PORT

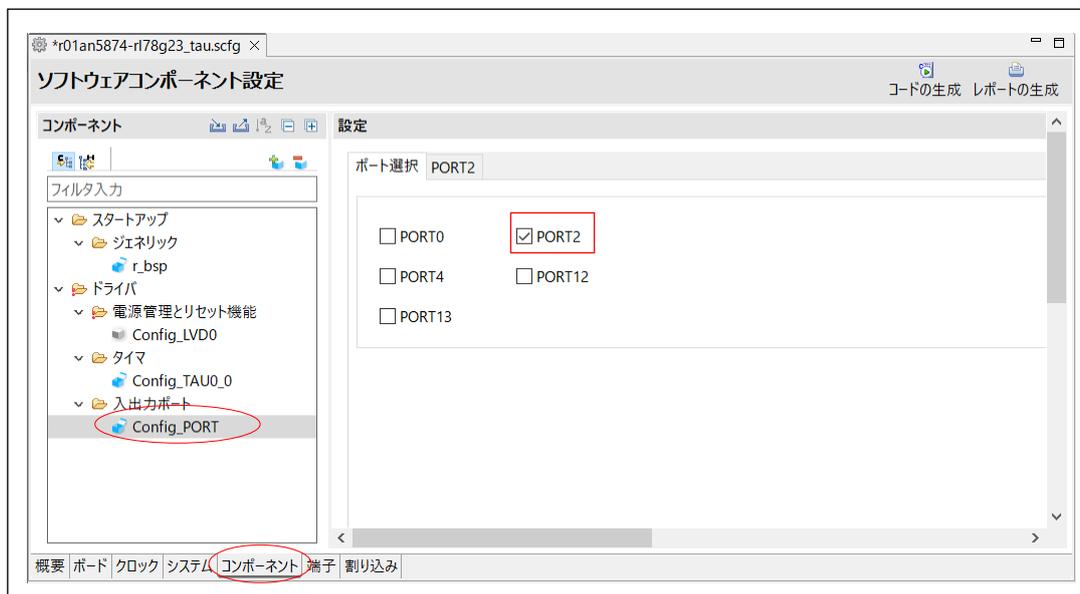
P52:

設定名	値
使用状況	出力
内蔵プルアップ	チェックを外す
1を出力	チェック

- 移植元では、出力ポート、初期値 1 の設定です。

SC の「コンポーネント」タブを表示します。「Config\_PORT」をクリックし、「PORT2」にチェックします。

図 3-14 「コンポーネント」タブ：Config\_PORT



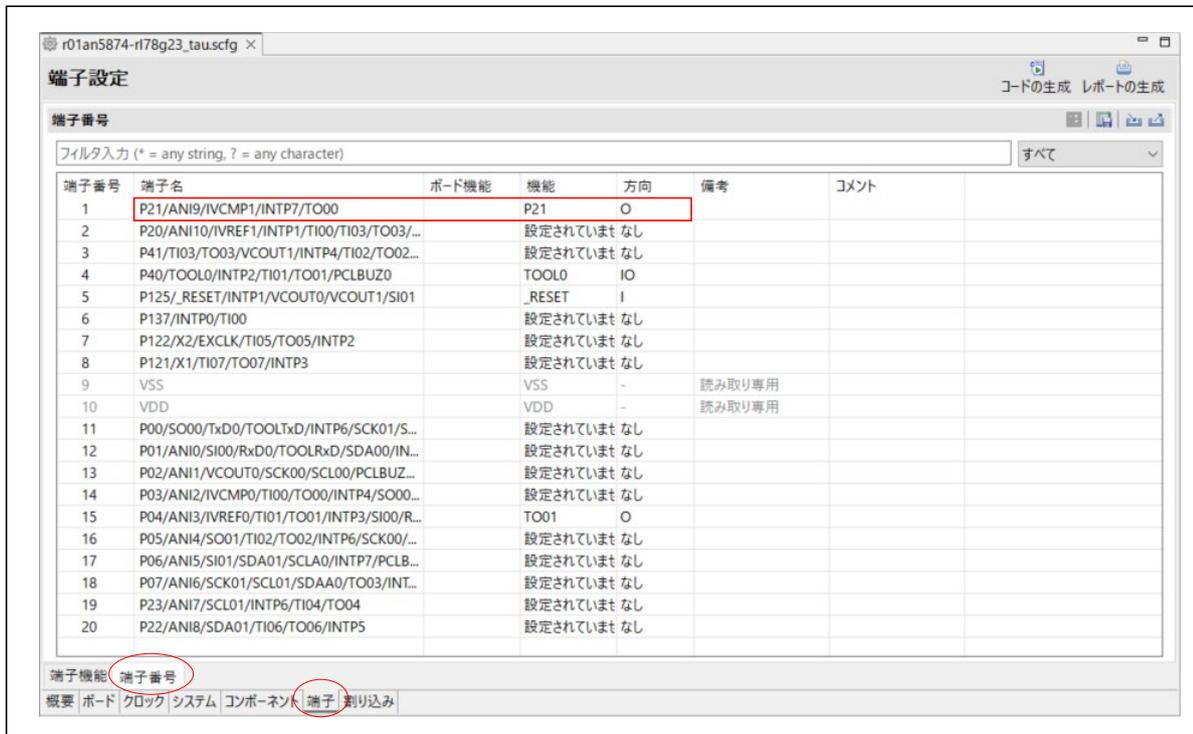
「PORT2」タブを表示し、P21 を「出力」に設定し、「1 を出力」にチェックします。

図 3-15 「コンポーネント」タブ：PORT2



「端子」 - 「端子番号」タブで、P21 が割り当てられていることを確認します。

図 3-16 「端子」 - 「端子番号」タブ



### 3.11 SC の設定保存

「2.11 SC の設定保存」を参照し、SC の設定内容を保存します。

### 3.12 コード生成

「2.12 コード生成」を参照し、SC でコードを生成します。

### 3.13 コード編集

「3.10.5 端子設定」で使用するポートを P52 から P21 へ変更したので、P52 の出力値を制御しているコードを編集します。それ以外の変更は、SC が設定に合わせて生成するためコード変更の必要はありません。

プロジェクトのファイルに対して「P5」で文字列検索を行い、P52 の出力値を制御している箇所を検索します。(参考: IDE での文字列検索方法)

検索の結果、Config\_TAU0\_0\_user.c 内にコードがあるため編集します。

図 3-17 Config\_TAU0\_0\_user.c (CS+ for CC の画面) : 編集 1

```

60
61 /* Function Name: r_Config_TAU0_0_channel0_interrupt
62 * Description : This function is INTTM00 interrupt service routine.
63 * Arguments : None
64 * Return Value : None
65 */
66 static void __near r_Config_TAU0_0_channel0_interrupt(void)
67 {
68 /* Start user code for r_Config_TAU0_0_channel0_interrupt. Do not edit comment generated here */
69 static uint8_t s_tm00_count = 0U; /* INTTM00 interrupt times counter */
70 uint16_t temp_duty = 0U; /* Duty factor calculation */
71
72 if (250 == (++s_tm00_count)) /* Update the duty cycle per 500ms */
73 {
74 s_tm00_count = 0U; /* Interrupt counter reset */
75 P5 ^= 0x04; /* Invert LED2 */
76
77 temp_duty = TDR01; /* Read the current duty cycle (duty factor) setting */
78 if (temp_duty <= (_E100_TAU_TDR01_VALUE / 9))
79 { /* If current duty cycle is 90% (duty factor is 10%). */
80 temp_duty = _E100_TAU_TDR01_VALUE; /* Set duty cycle to 10% (duty factor to 90%) */
81 }
82 else
83 { /* Increase duty cycle by 20% (decrease duty factor by 20%) */
84 temp_duty -= ((_E100_TAU_TDR01_VALUE / 9) * 2);
85 }
86 TDR01 = temp_duty; /* Update duty cycle (duty factor) */
87 }
88 else
89 {
90 :
91 }
92
93 /* End user code. Do not edit comment generated here */
94 }
95
96 /* Start user code for adding. Do not edit comment generated here */
97 /* End user code. Do not edit comment generated here */
98

```

- 変更前 : P5 ^= 0x04;      変更後 : P2 ^= 0x02;

### 3.14 移植先プロジェクトのビルド

「2.14 移植先プロジェクトのビルド」を参照し、移植先プロジェクトをビルドします。

ビルドの結果、Config\_TAU0\_0\_user.c の 78 行目に記載の「\_E100\_TAU\_TDR01\_VALUE」が未定義エラーとなります。本マクロは、タイマ・アレイ・ユニット 0 チャンネル 0, チャンネル 1 の PWM 出力のデューティ比の初期値です。(TRD01 レジスタの設定値)

移植先の初期値に合わせてコードを編集します。初期値は SC 生成の Config\_TAU0\_0\_h 内で定義されています。

図 3-18 Config\_TAU0\_0.h (CS+ for CC の画面)

```

31
32 #ifndef CFG_Config_TAU0_0_H
33 #define CFG_Config_TAU0_0_H
34
35 .....
36 Macro definitions (Register bit)
37 .....
38
39 .....
40 Macro definitions
41 .....
42 #define _7CFF_TAU_TDR00_VALUE (0x7CFFU) /* 16-bit timer data register 00 (TDR00) */
43 #define _7080_TAU_TDR01_VALUE (0x7080U) /* 16-bit timer data register 01 (TDR01) */
44 .....
45
46 Typedef definitions
47 .....
48
49 .....
50 Global functions
51 .....
52 void R_Config_TAU0_0_Create (void);
53 void R_Config_TAU0_0_Start (void);
54 void R_Config_TAU0_0_Stop (void);
    
```

図 3-19 Config\_TAU0\_0\_user.c (CS+ for CC の画面) : 編集 2

```

60
61 * Function Name: r_Config_TAU0_0_channel0_interrupt
62 * Description : This function is INTTM00 interrupt service routine.
63 * Arguments : None
64 * Return Value : None
65 .....
66 static void __near r_Config_TAU0_0_channel0_interrupt(void)
67 {
68 /* Start user code for r_Config_TAU0_0_channel0_interrupt. Do not edit comment generated here */
69 static uint8_t s_tm00_count = 0U; /* INTTM00 interrupt times counter */
70 uint16_t temp_duty = 0U; /* Duty factor calculation */
71
72 if (250 == (++s_tm00_count)) /* Update the duty cycle per 500ms */
73 {
74 s_tm00_count = 0U; /* Interrupt counter reset */
75 P2 ^= 0x02; /* Invert LED2 */
76
77 temp_duty = TDR01; /* Read the current duty cycle (duty factor) setting */
78 if (temp_duty <+ (_E100_TAU_TDR01_VALUE / 9))
79 { /* If current duty cycle is 90% (duty factor is 10%). */
80 temp_duty = _E100_TAU_TDR01_VALUE; /* Set duty cycle to 10% (duty factor to 90%) */
81 }
82 else
83 { /* Increase duty cycle by 20% (decrease duty factor by 20%) */
84 temp_duty -= (_E100_TAU_TDR01_VALUE / 9) * 2;
85 }
86 TDR01 = temp_duty; /* Update duty cycle (duty factor) */
87 }
88 else
89 {
90 :
91 }
92
93 /* End user code. Do not edit comment generated here */
94 }
95
96 /* Start user code for adding. Do not edit comment generated here */
97 /* End user code. Do not edit comment generated here */
    
```

- 変更前 : `_E100_TAU_TDR01_VALUE`      変更後 : `_7080_TAU_TDR01_VALUE`

再度ビルドするとエラーがなくなります。これで、サンプルコードの移植は完了です。

#### 4. サンプルコード

サンプルコードは、ルネサスエレクトロニクスホームページから入手してください。

#### 5. 参考ドキュメント

RL78/G23 ユーザーズマニュアル ハードウェア編 (R01UH0896)

RL78 ファミリユーザーズマニュアルソフトウェア編 (R01US0015)

スマート・コンフィグレータ ユーザーズマニュアル RL78 API リファレンス編 (R20UT4852)

RL78 スマート・コンフィグレータ ユーザーガイド : CS+編 (R20AN0580)

RL78 スマート・コンフィグレータ ユーザーガイド : e<sup>2</sup> studio 編 (R20AN0579)

RL78 スマート・コンフィグレータ ユーザーガイド : IAR 編 (R20AN0581)

RL78 ファミリ ボードサポートパッケージモジュール Software Integration System (R01AN5522)

(最新版をルネサスエレクトロニクスホームページから入手してください。)

テクニカルアップデート

(最新の情報をルネサスエレクトロニクスホームページから入手してください。)

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。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2023.04.7	—	初版発行

## 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、 $V_{IL}$  (Max.) から  $V_{IH}$  (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、 $V_{IL}$  (Max.) から  $V_{IH}$  (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まれます。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、これに限りません。）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た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いたします。
  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な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（豊洲フォレストシア）

[www.renesas.com](http://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/](http://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。