

スマート・コンフィグレータ

R01AN4422JJ0130

RH850/F1KM 用サンプル・プロジェクト説明書

Rev.1.30

2023.06.20

要旨

本ドキュメントでは、サンプル・プロジェクトを利用し、各統合開発環境でスマート・コンフィグレータが出力するファイルをビルドする方法について説明します。

対象デバイス

RH850/F1KM-S1, RH850/F1KM-S2, RH850/F1KM-S4

目次

1. 概要.....	3
1.1 本ドキュメントの目的.....	3
1.2 動作環境.....	3
2. サンプル・プロジェクトの概要.....	4
2.1 サンプル・プロジェクト一覧.....	4
2.2 サンプル・プロジェクト注意事項.....	5
3. スマート・コンフィグレータの基本操作フロー.....	6
3.1 CS+環境.....	6
3.2 MULTI 環境.....	7
3.3 IAREW 環境.....	8
4. CS+, CCRH 用サンプル・プロジェクトの説明.....	9
4.1 CS+, CCRH 用サンプル・プロジェクトの構成.....	9
4.2 基本操作フロー.....	10
4.3 デバイス変更フロー.....	13
4.4 CS+, CCRH 用サンプル・プロジェクトでの設定.....	17
5. CS+, GHS CCRH850 用サンプル・プロジェクトの説明.....	18
5.1 CS+, GHS CCRH850 用サンプル・プロジェクトの構成.....	18
5.2 基本操作フロー.....	19
5.3 デバイス変更フロー.....	24
5.4 CS+, GHS CCRH850 用サンプル・プロジェクトでの設定.....	29
6. CS+, CCRH, RI850V4 用サンプル・プロジェクトの説明.....	30
6.1 CS+, CCRH, RI850V4 用サンプル・プロジェクトの構成.....	30
6.2 基本操作フロー.....	31
6.3 デバイス変更フロー.....	34
6.4 CS+, CCRH, RI850V4 用サンプル・プロジェクトでの設定.....	37

7.	MULTI, GHS CCRH850 用サンプル・プロジェクトの説明.....	39
7.1	MULTI, GHS CCRH850 用サンプル・プロジェクトの構成.....	39
7.2	基本操作フロー.....	40
7.3	デバイス変更フロー.....	43
7.4	MULTI, GHS CCRH850 用サンプル・プロジェクトでの設定.....	46
8.	IAREW, IAR ICC 用サンプル・プロジェクトの説明.....	47
8.1	IAREW, IAR ICC 用サンプル・プロジェクトの構成.....	47
8.2	基本操作フロー.....	48
8.3	デバイス変更フロー.....	50
8.4	IAREW, IAR ICC 用サンプル・プロジェクトでの設定.....	54
9.	スマート・コンフィグレータの操作.....	55
9.1	周辺機能(ソフトウェアコンポーネント)設定.....	55
9.2	ドライバの生成.....	58
9.3	ユーザコード領域にアプリケーションコードを追加.....	58
9.4	main()にアプリケーションコードを追加.....	59
	改訂記録.....	1

1. 概要

1.1 本ドキュメントの目的

本ドキュメントでは、サンプル・プロジェクトを利用し、各統合開発環境で RH850/F1KM 用スマート・コンフィグレータが出力するファイルをビルドする方法について説明します。

本アプリケーションノートをマイクロコントローラへ適用する場合、使用するマイクロコントローラの仕様にあわせて変更し、十分評価してください。

1.2 動作環境

サンプル・プロジェクトを利用し、スマート・コンフィグレータで生成したソースを使用し各統合開発環境でプログラムの作成やビルドを行うには、スマート・コンフィグレータおよびご使用になるツールをインストールしてください。

また、各統合開発環境の使い方については、各統合開発環境のユーザズマニュアルを参照してください。

表 1-1 動作環境

種別	名称	本書で使用する略語
IDE	CS+ for CC V8.10.00 以降	CS+
ツールチェーン	RH850 ファミリ用ルネサス C コンパイラ・パッケージ	CCRH
ツールチェーン	RH850 ファミリ用リアルタイム OS [RI850V4] V3	RI850V4
IDE	MULTI v8.1.4 以降	MULTI
ツールチェーン	Green Hills Compiler	GHS CCRH850
IDE	IAR Embedded Workbench for RH850 V3.10.1 以降	IAREW
ツールチェーン	IAR C/C++ Compiler	IAR ICC

Green Hills Software は、米国、およびその他の国における Green Hills Software, Inc.の商標または登録商標です。

IAR Embedded Workbench は、IAR Systems の登録商標です。

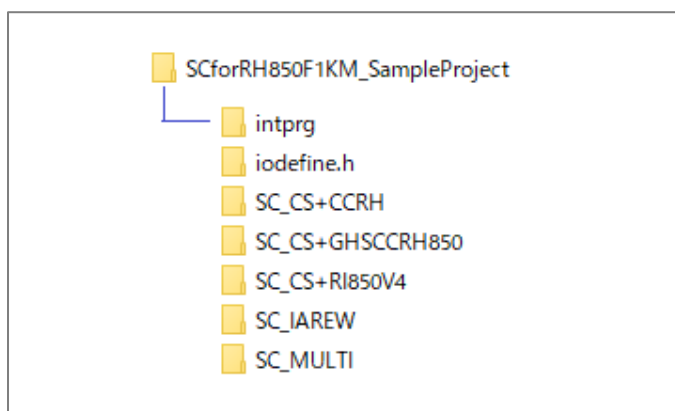
2. サンプル・プロジェクトの概要

RH850/F1KM 用スマート・コンフィグレータは、main 関数とスマート・コンフィグレータのコンポーネントで設定した各周辺機能の初期化を行うソースファイルを出力します。マイクロコントローラをリセットしたあと、main 関数を実行する前に行う初期化処理、main 関数の起動などの処理を行うスタートアップ・ルーチンは出力しません。

そのため、スマート・コンフィグレータで設定した周辺機能とユーザ・アプリケーションをすぐにビルドできるように、サンプルのスタートアップ等を含むサンプル・プロジェクトを用意しています。

2.1 サンプル・プロジェクト一覧

RH850/F1KM 用スマート・コンフィグレータでは、次のサンプル・プロジェクトを用意しています。



サンプル・プロジェクトの詳細は、各説明章を参照してください。

表 2-1 サンプル・プロジェクト

説明箇所	フォルダ名	対象デバイス	説明
2 章	intprg	—	RH850/F1KM-S1, RH850/F1KM-S2, RH850/F1KM-S4 用の EI マスカブル割り込みベクタ・テーブルを定義したファイル
2 章	iodefine.h	—	RH850/F1KM-S1, RH850/F1KM-S2, RH850/F1KM-S4 用のレジスタを定義した Renesas CCRH 用ヘッダ・ファイル
4 章	SC_CS+CCRH	R7F701651 (RH850/F1KM-S4)	CS+ for CC, CCRH コンパイラのパッケージ
5 章	SC_CS+GHSCCRH850	R7F701649 (RH850/F1KM-S4)	CS+ for CC, GHS CCRH850 コンパイラのパッケージ
6 章	SC_CS+RI850V4	R7F701649 (RH850/F1KM-S4)	CS+ for CC, CCRH コンパイラ, RI850V4 のパッケージ
7 章	SC_MULTI	R7F701645 (RH850/F1KM-S4)	MULTI, GHS CCRH850 コンパイラのパッケージ
8 章	SC_IAREW	R7F701645 (RH850/F1KM-S4)	IAREW, IAR ICC コンパイラのパッケージ (ワークスペース)

2.2 サンプル・プロジェクト注意事項

1. 本サンプル・プロジェクトを使用する際は、アクセスが制限されていないディレクトリにコピーして使用してください。アクセスが制限されたディレクトリ*で使用した場合、生成コードなどのファイルが保存されず、エラーとなる場合があります。
※一般に、下記のようなフォルダはアクセスが制限されています。
 - ・プログラムデータフォルダ（「C:¥ProgramData」など）
 - ・プログラムファイルフォルダ（「C:¥Program Files」など）
 - ・システムルートフォルダ（「C:¥Windows」など）お使いの環境については、システム管理者(IT 部門)にお問い合わせください。
2. スマート・コンフィグレータは Renesas CCRH 用の `iodefine.h` に従ってレジスタ記述を出力しています。レジスタ定義のヘッダ・ファイルは、GHS CCRH850 コンパイラ、IAR ICC コンパイラでも用意されていますが、スマート・コンフィグレータ生成のファイルをビルドする際には、Renesas CCRH 用の `iodefine.h` をインクルードしてください。
3. スマート・コンフィグレータでは、割込みハンドラ・アドレスの選択方式として、テーブル参照方式の割り込みを使用しています。サンプル・プロジェクトではテーブルの先頭アドレスを `0x00000200` に設定しています。
4. スマート・コンフィグレータで設定した周辺機能の割り込みベクタ・テーブルの定義は、スマート・コンフィグレータが出力する `smc/general/r_cg_intvector.c` に反映されます。`intprg` フォルダ以下の `sc_intprg-S1.c`、`sc_intprg-S2.c`、`sc_intprg-S4.c` は、スマート・コンフィグレータで設定しない EI マスカブル割り込み要因のベクタ・テーブルを定義しています。
5. サンプル・プロジェクトで提供するファイル、セクション等の設定はサンプルです。使用するマイクロコントローラの仕様およびお客様のシステムに合わせて変更してください。

3. スマート・コンフィグレータの基本操作フロー

RH850/F1KM 用スマート・コンフィグレータが出力するファイルをユーザ・アプリケーションと組み合わせてビルドする際の基本操作フローを説明します。

本章では、統合開発環境ごとに、サンプル・プロジェクトを使用しない場合の基本操作フローについて説明します。サンプル・プロジェクトを使用時の操作フローは、使用するサンプル・プロジェクトにあわせて4章以降を参照してください。

3.1 CS+環境

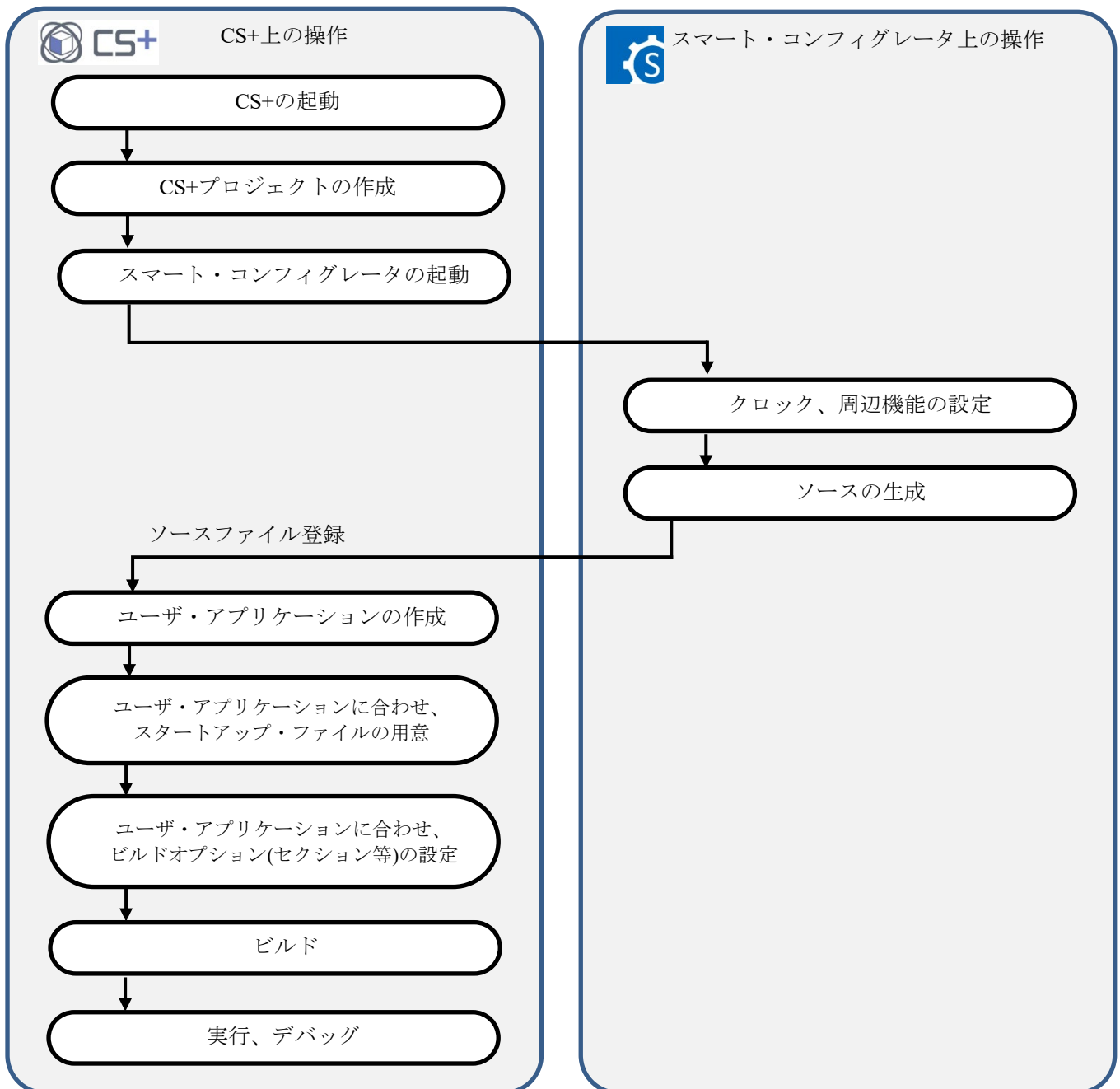


図 3-1 CS+環境 基本操作フロー

3.2 MULTI 環境

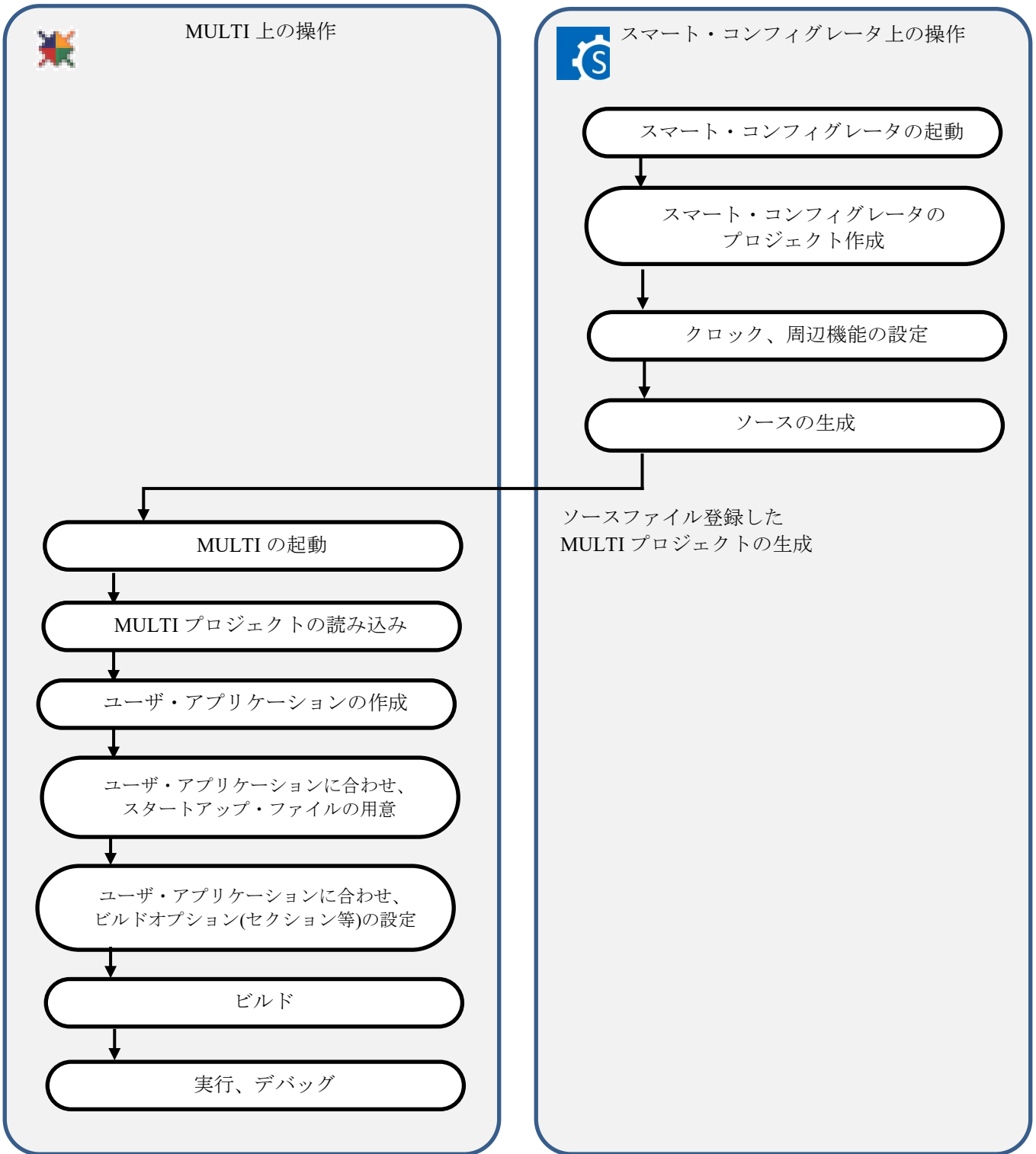


図 3-2 MULTI 環境 基本操作フロー

3.3 IAREW 環境

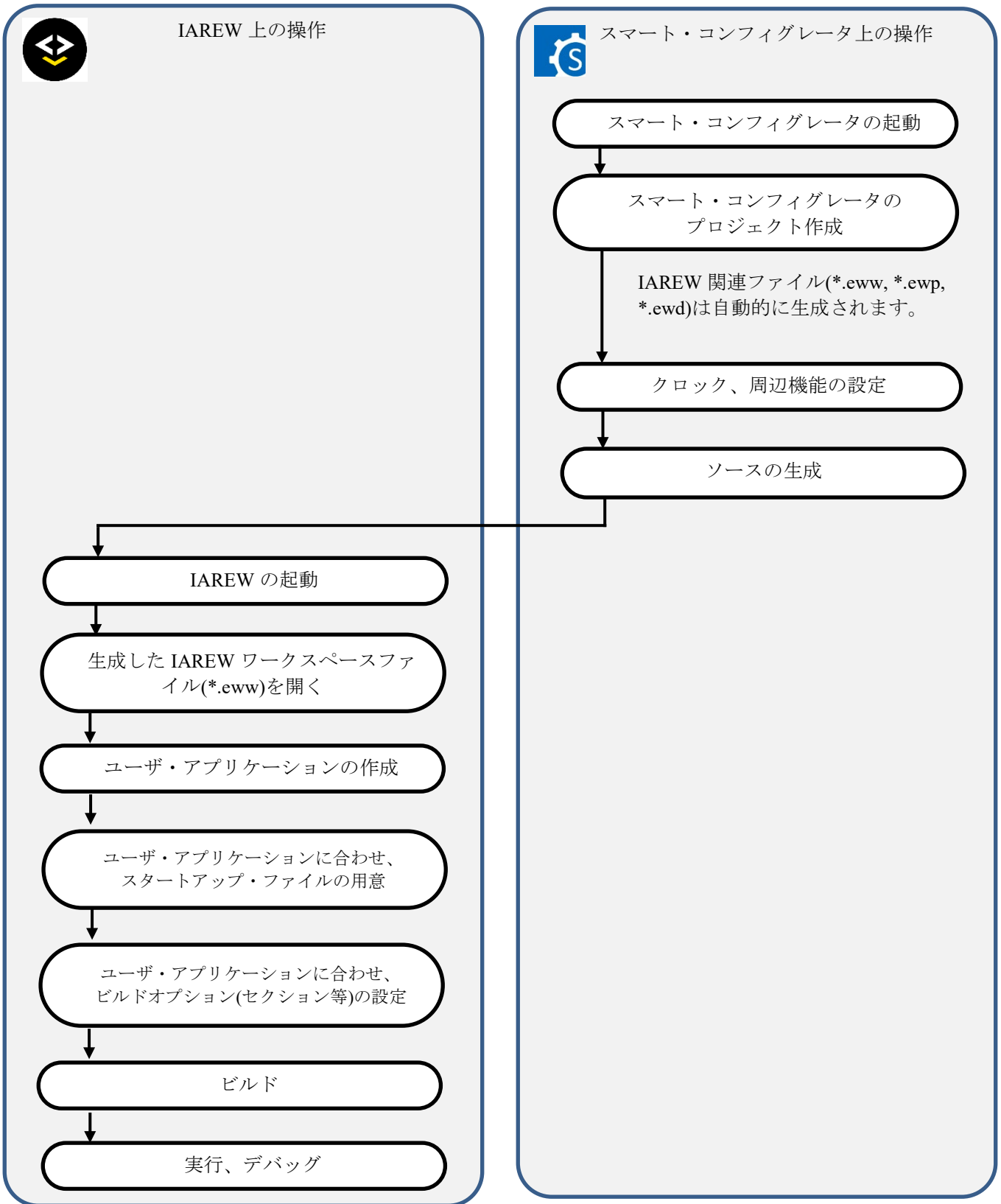


図 3-3 IAREW 環境 基本操作フロー

4. CS+, CCRH 用サンプル・プロジェクトの説明

4.1 CS+, CCRH 用サンプル・プロジェクトの構成

サンプル・プロジェクトの構成は下記です。

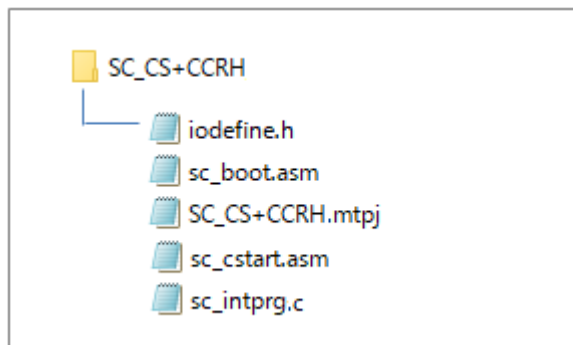


図 4-1 CS+, CCRH 用サンプル・プロジェクトのファイル構成

表 4-1 CS+, CCRH 用サンプル・プロジェクトのファイル概要

ファイル名	ファイルの概要
SC_CS+CCRH.mtpj	CS+のプロジェクト・ファイル
sc_boot.asm	リセットの発生からスタートアップ・ルーチンに分岐するまでの処理の定義と割り込みベクタ・テーブルの定義
sc_cstart.asm	main 関数に分岐するまでのスタートアップ・ルーチンの定義
sc_intprg.c	EI レベルマスカブル割り込みベクタ・テーブルの定義
iodefine.h	レジスタの定義

スマート・コンフィグレータは、上記のファイルを出力しません。

4.2 基本操作フロー

図 4-2 に、サンプル・プロジェクトをベースにスマート・コンフィグレータを使用する際の操作フローを以下に示します。

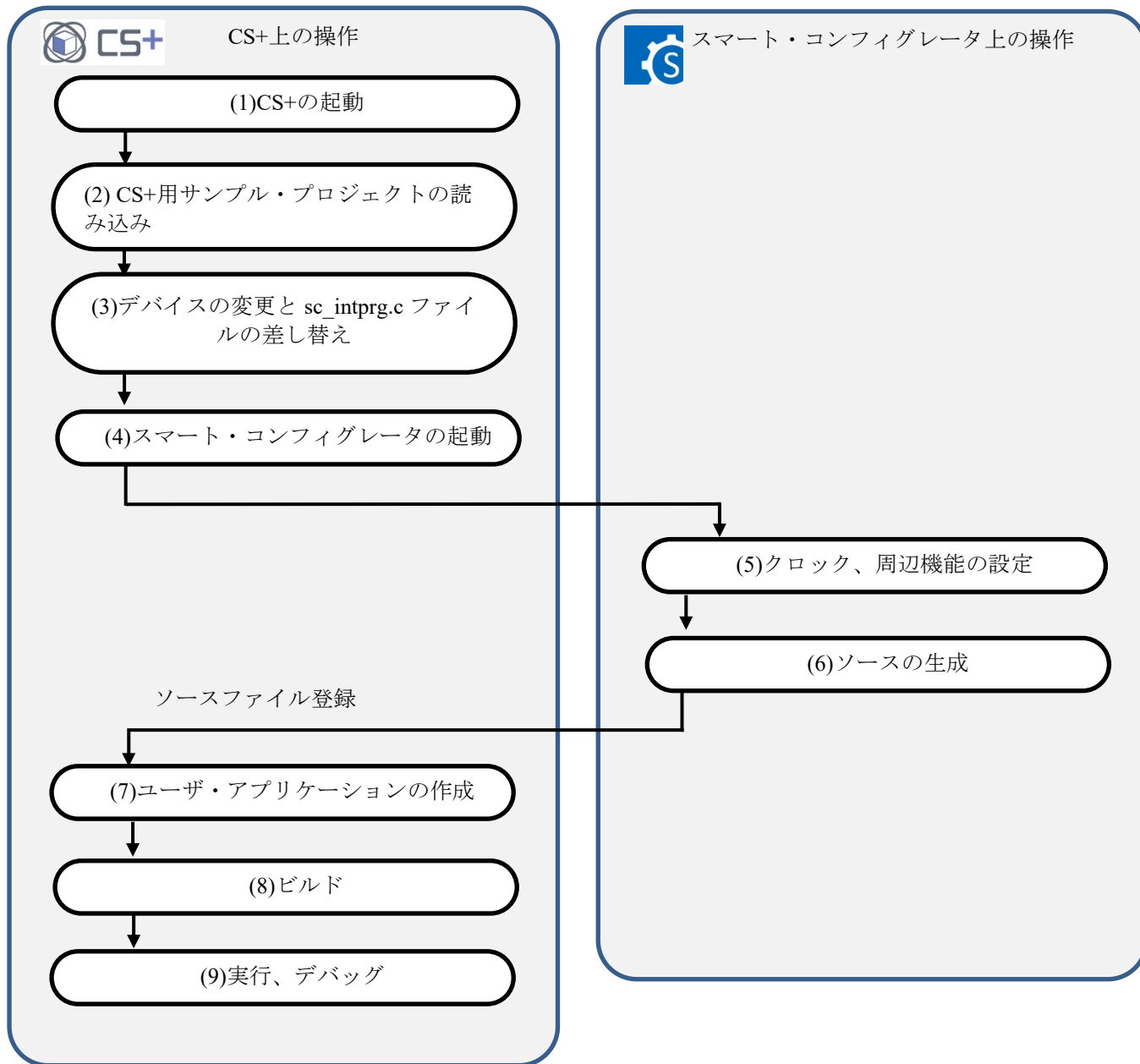


図 4-2 CS+, CCRH 用サンプル・プロジェクト 操作フロー

(1) CS+の起動

Windows のスタートメニューから [Renesas Electronics CS+] → [CS+ for CC(RL78,RX,RH850)] を選択します。

(2) CS+のサンプル・プロジェクトの読み込み

CS+の [ファイル] → [ファイルを開く(O)...] メニューまたは [既存のプロジェクトを開く] で、「SC_CS+CCRH.mtpj」を選択します。

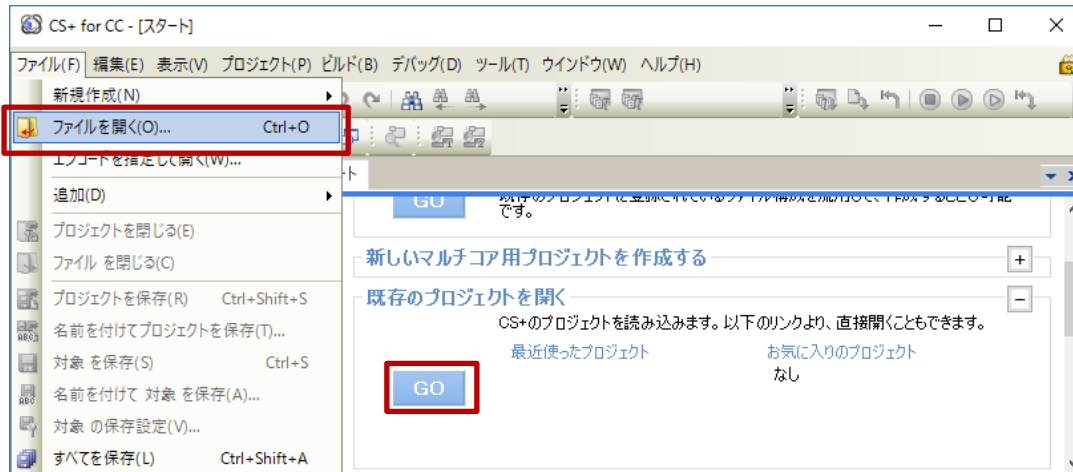


図 4-3 CS+, CCRH 用サンプル・プロジェクトの読み込み

サンプル・プロジェクトは、必ずアクセス制限のない場所で使用してください。(2.2 参照)

(3) デバイスの変更

サンプル・プロジェクトは、対象デバイスとして R7F701651 (RH850/F1KM-S4) を選択しています。ご使用になるデバイスと異なる場合には、「4.3 デバイス変更フロー」を参照し対象デバイスおよび使用するファイルを変更して使用してください。デバイス変更が不要な場合は、(4)へ進んでください。

(4) スマート・コンフィグレータの起動

- (4)-1. スマート・コンフィグレータのパス設定を確認します。 [プロジェクト・ツリー] で [スマート・コンフィグレータ] を選択し [プロパティ] を開きます。 [RH850 用スマート・コンフィグレータの exe ファイルパス] にスマート・コンフィグレータをインストールしたパスが設定されていることを確認します。

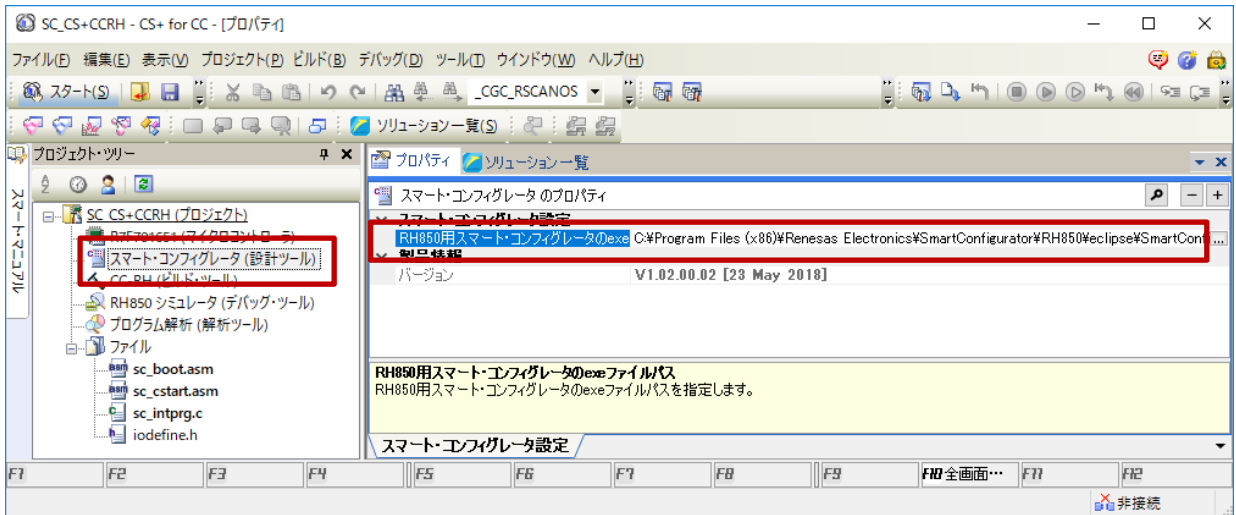


図 4-4 スマート・コンフィグレータのパス設定

(4)-2. スマート・コンフィグレータを起動します。 [プロジェクト・ツリー] で [スマート・コンフィグレータ] をダブルクリックします。

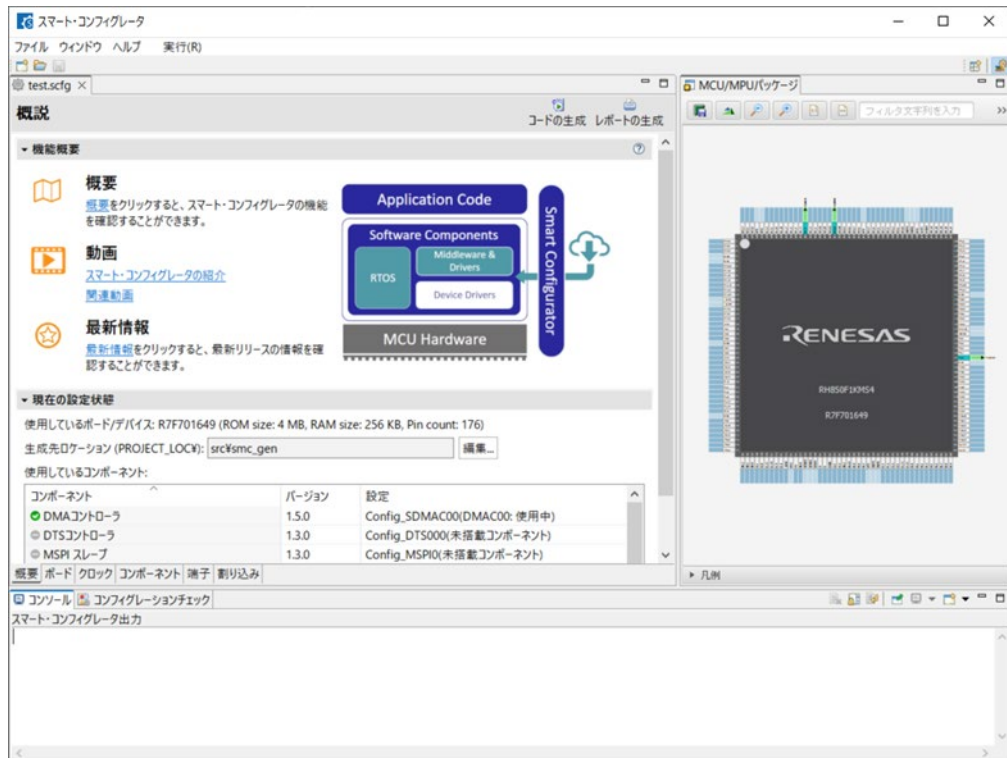



図 4-5 スマート・コンフィグレータの起動

(5)-(7)のスマート・コンフィグレータでのクロック、コンポーネント設定およびドライバ生成の手順については、「9 スマート・コンフィグレータの操作」を参照してください。

(8) ビルド

ドライバ、アプリケーションコードをビルドします。CS+の [ビルド] → [ビルド・プロジェクト] メニューまたはツールバーの [ビルド]  ボタンをクリックします。

(9) 実行、デバッグ

エミュレータでのプログラム実行およびデバッグについては、「CS+ V8.09.00 統合開発環境 ユーザーズマニュアル RH850 デバッグ・ツール編」(R20UT5202)を参照してください。

4.3 デバイス変更フロー

表 2-1 サンプル・プロジェクトに記載したサンプル・プロジェクトの対象デバイスが、ご使用になるデバイスと異なる場合には、次の手順で対象デバイス/使用するファイルを変更して使用してください。

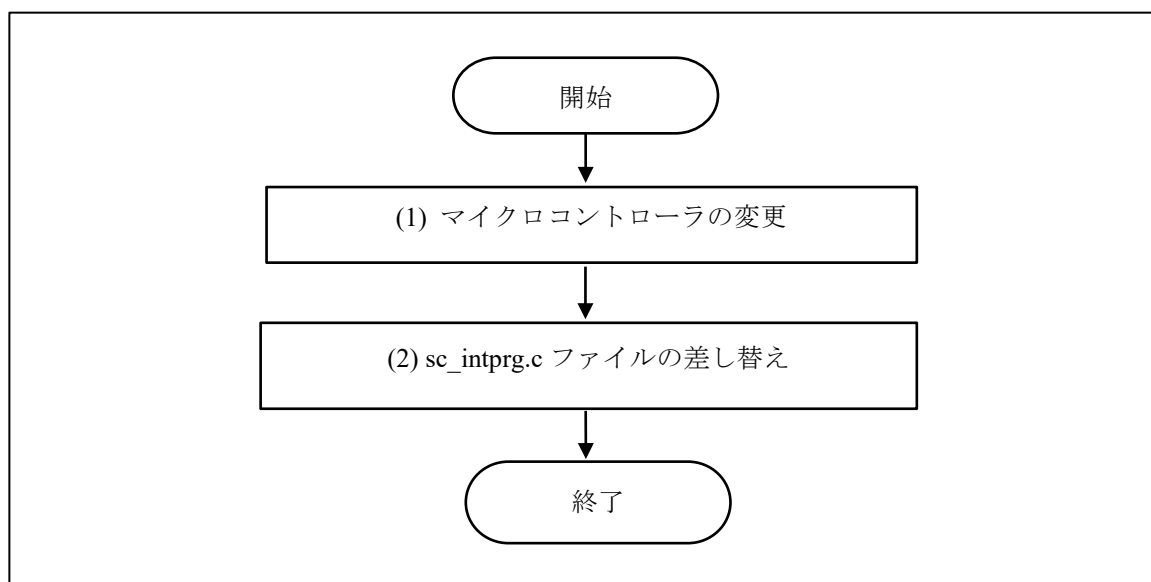


図 4-6 サンプル・プロジェクトの対象デバイスの変更 (CS+プロジェクト)

(1) マイクロコントローラの変更

- (1)-1. 「R7F701651(マイクロコントローラ)」を選択し、コンテキストメニューで [マイクロコントローラを変更...] を選択します。 [質問] ダイアログで「OK」を選択します。

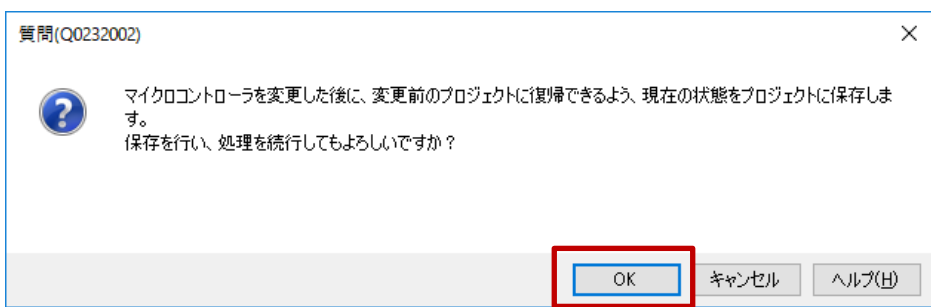
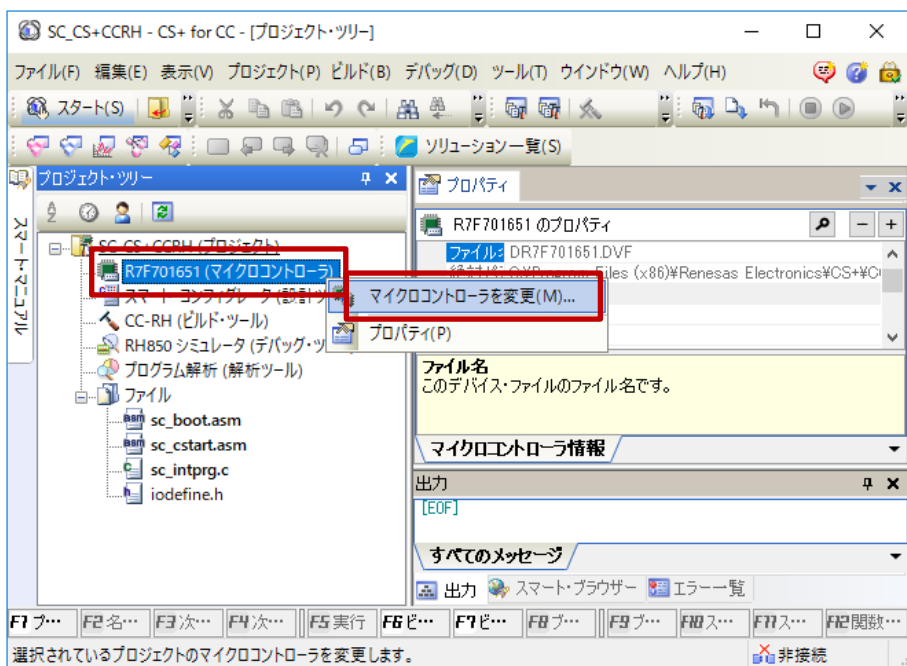


図 4-7 「マイクコントローラを変更」の選択

(1)-2. [マイクコントローラの変更] ダイアログで、使用する RH850/F1KM-S1/S2/S4 のデバイスを選択します。

例：R7F701651(RH850/F1KM-S4)から R7F701684(RH850/F1KM-S1)へ変更

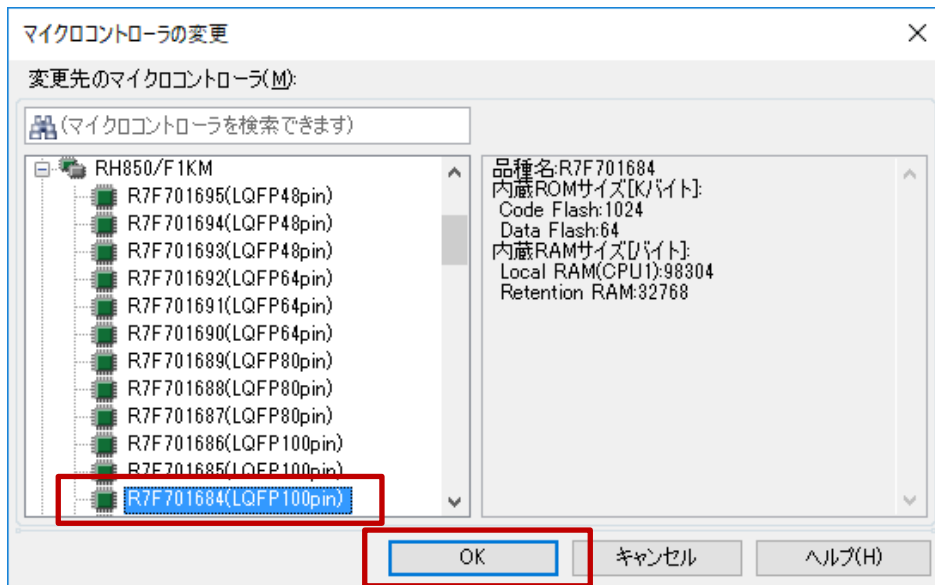


図 4-8 変更先のマイクロコントローラを選択

- (1)-3. プロジェクト・ツリーのマイクロコントローラの表示が、変更後のデバイスであることを確認します。

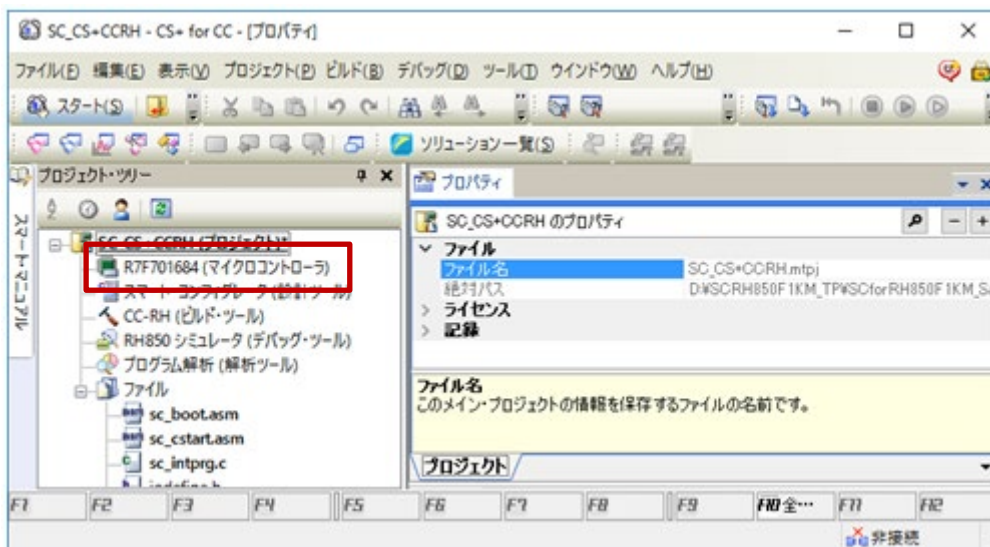


図 4-9 変更後のデバイスの確認

- (1)-4. プロジェクトを保存します。[ファイル] → [プロジェクトを保存] を選択します。

(2) sc_intprg.c ファイルの差し替え

本手順は、(1) で RH850/F1KM-S4 グループから RH850/F1KM-S1/S2 グループへ変更した場合のみ行ってください。

- (2)-1. サンプル・プロジェクトに含まれる sc_intprg.c は、RH850/F1KM-S4 用のため RH850/F1KM-S1 または S2 用のファイルへ差し替えます。Windows エクスプローラで、intprg/sc_intprg-S1.c または sc_intprg-S2.c を「sc_intprg.c」へリネームしサンプル・プロジェクトの sc_intprg.c と差し替えてください。

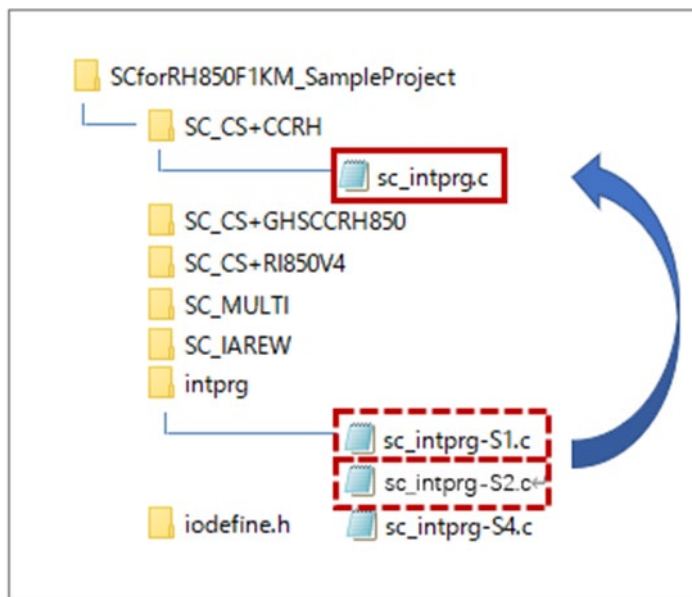


図 4-10 sc_intprg.c ファイルの差し替え

(図では、説明不要のファイルの記載を省略しています。)

4.4 CS+, CCRH 用サンプル・プロジェクトでの設定

サンプル・プロジェクトは、CS+で「空のアプリケーション(CC-RH)」プロジェクトで作成し、インクルード・パスの追加の他に、次のオプションについて設定を変更しています。

- (a) [CC-RH (ビルド・ツール)] のプロパティ → [リンク・オプション] → [セクション] → [セクションの開始アドレス]

[サンプル・プロジェクトでの設定]

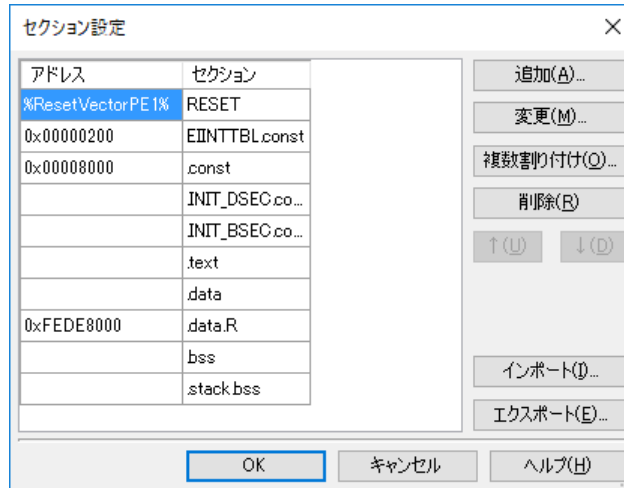


図 4-11 セクションの設定

- (b) [CC-RH (ビルド・ツール)] のプロパティ → [I/O ヘッダ・ファイル生成オプション] → [I/O ヘッダ・ファイル] → [ビルド時に I/O ヘッダ・ファイルを更新する]

[サンプル・プロジェクトでの設定]

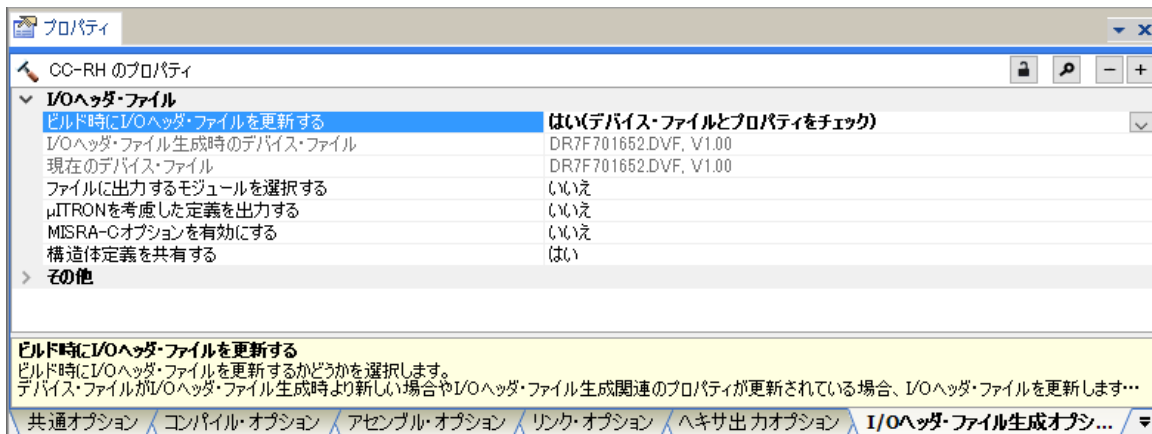


図 4-12 I/O ヘッダ・ファイルの更新設定

5. CS+, GHS CCRH850 用サンプル・プロジェクトの説明

5.1 CS+, GHS CCRH850 用サンプル・プロジェクトの構成

サンプル・プロジェクトの構成は下記です。

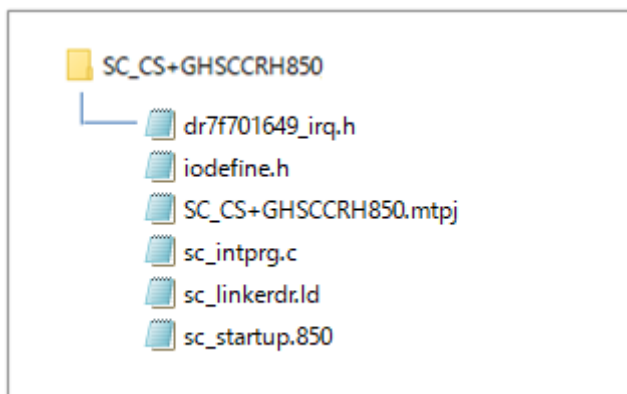


図 5-1 サンプル・プロジェクトの構成

表 5-1 CS+, GHS CCRH850 用サンプル・プロジェクト ファイルの概要

ファイル名	ファイルの概要
SC_CS+GHSCCRH850.mtpj	CS+のプロジェクト・ファイル
sc_startup.850	リセットの発生から各アプリケーション・プロジェクトに分岐するまでの処理の定義、例外割り込みベクタ・テーブルの定義、 main 関数に分岐するまでのスタートアップ・ルーチンの定義
dr7f701649_irq.h	例外割り込みベクタ・テーブルの定義（マクロ定義）
sc_linkerdr.ld	リンク・ディレクティブ・ファイル
sc_intprg.c	EI レベルマスカブル割り込みベクタ・テーブルの定義
iodefine.h	レジスタの定義

スマート・コンフィグレータは、上記のファイルを出力しません。

5.2 基本操作フロー

図 5-2 に、サンプル・プロジェクトをベースにスマート・コンフィグレータを使用する際の操作フローを以下に示します。

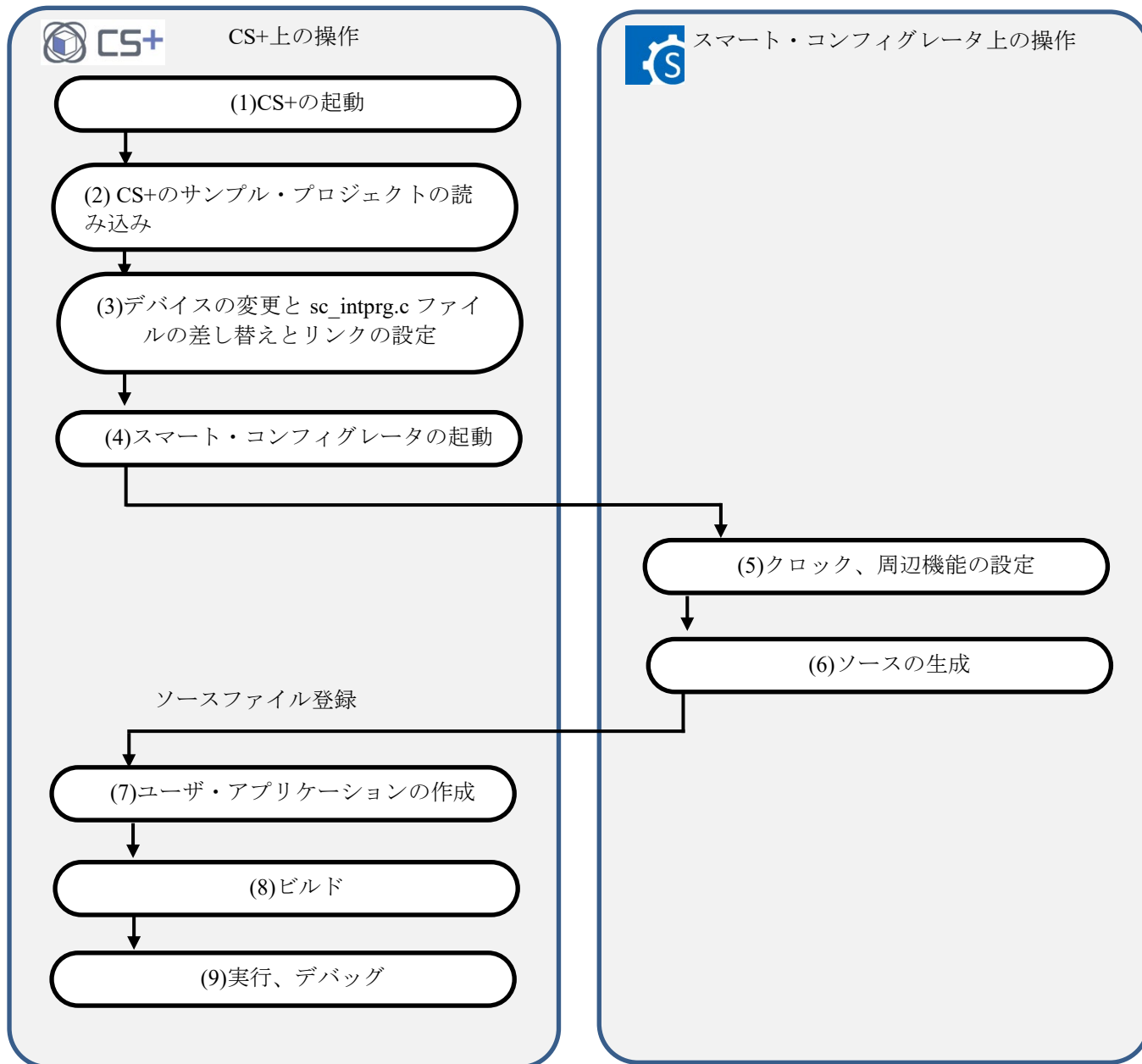


図 5-2 CS+, GHS CCRH850 用サンプル・プロジェクト 操作フロー

(1) CS+の起動

Windows のスタートメニューから [Renesas Electronics CS+] → [CS+ for CC(RL78,RX,RH850)] を選択します。

(2) CS+のサンプル・プロジェクトの読み込み

CS+の [ファイル] → [ファイルを開く(O)...] メニュー、または [既存のプロジェクトを開く] で、「SC_CS+GHSCCRH850.mtpj」を選択します。

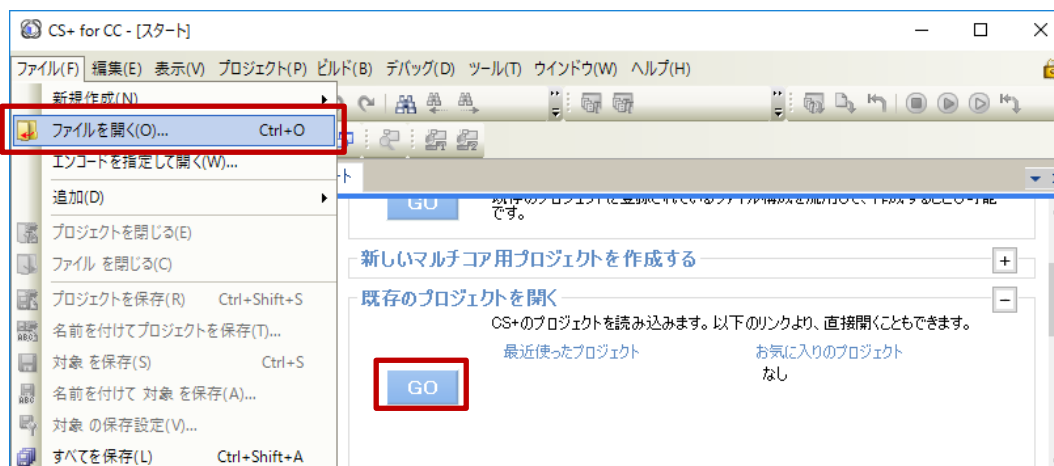


図 5-3 サンプル・プロジェクトの読み込み

サンプル・プロジェクトは、必ずアクセス制限のない場所で使用してください。(2.2 参照)

(3) デバイスの変更

サンプル・プロジェクトは、対象デバイスとして R7F701649(RH850/F1KM-S4)を選択しています。ご使用になるデバイスと異なる場合には、「5.3 デバイス変更フロー」を参照し対象デバイスおよび使用するファイルを変更して使用してください。デバイス変更が不要な場合は、(4)へ進んでください。

(4) スマート・コンフィグレータの起動

- (4)-1. スマート・コンフィグレータのパス設定を確認します。[プロジェクト・ツリー]で[スマート・コンフィグレータ]を選択し[プロパティ]を開きます。[RH850用スマート・コンフィグレータのexeファイルパス]にスマート・コンフィグレータをインストールしたパスが設定されていることを確認します。

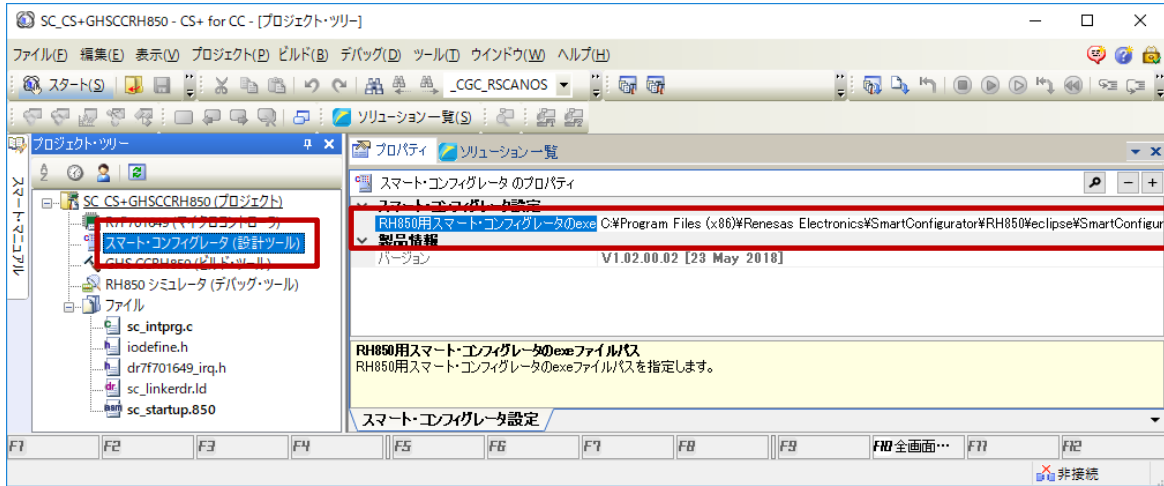


図 5-4 スマート・コンフィグレータのパス設定

- (4)-2. スマート・コンフィグレータを起動します。[プロジェクト・ツリー]で[スマート・コンフィグレータ]をダブルクリックします。



図 5-5 スマート・コンフィグレータの起動

(5)-(7)のスマート・コンフィグレータでのクロック、コンポーネント設定およびドライバ生成の手順については、「9 スマート・コンフィグレータの操作」を参照してください。

(8) ビルド

(8)-1. GHS CCRH850 コンパイラのパスを設定します。[プロジェクト・ツリー] で [GHS CCRH850(ビルド・ツール)] を選択し [プロパティ] を開きます。[共通オプション] - [コンパイラ・パッケージのフォルダ] に GHS CCRH850 コンパイラをインストールしたパスを設定します。

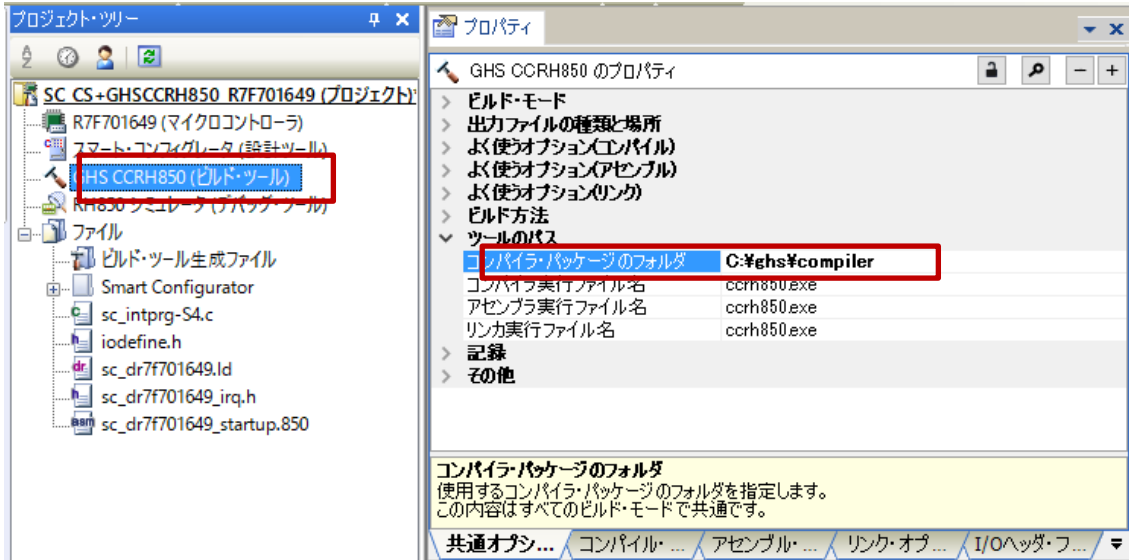
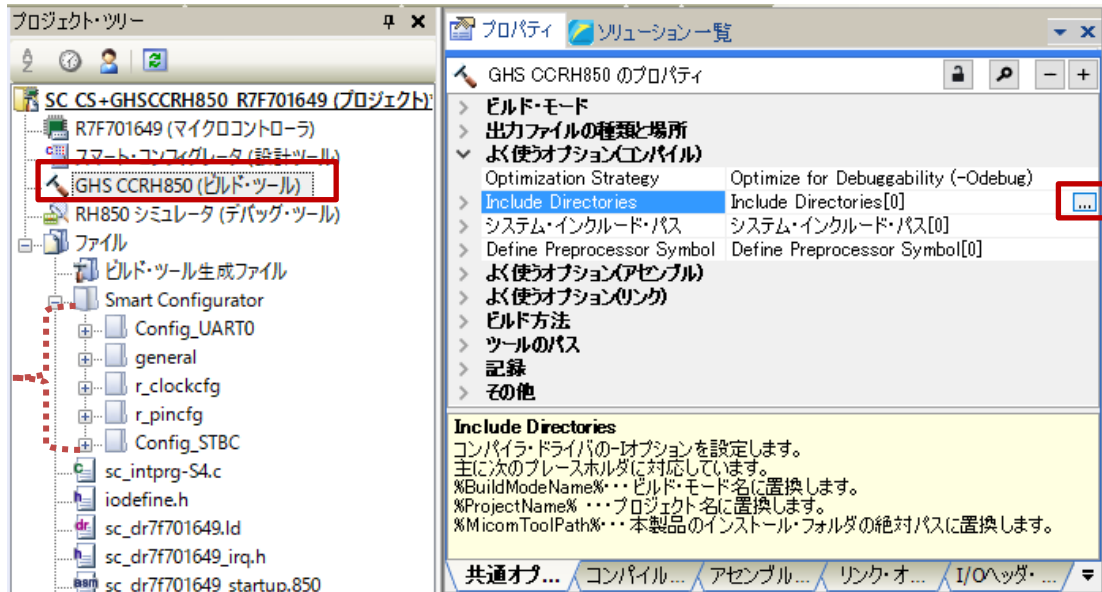


図 5-6 コンパイラのパス設定

- (8)-2. スマート・コンフィグレータが生成したファイルが存在するフォルダをインクルード・パスに追加します。[プロジェクト・ツリー] で [GHS CCRH850(ビルド・ツール)] を選択し [プロパティ] を開きます。[共通オプション] - [Include Directories] で ボタンをクリックします。



[パス編集] ダイアログの [参照] ボタンで、iodefine.h とスマート・コンフィグレータが生成したファイルが存在するフォルダを追加します。

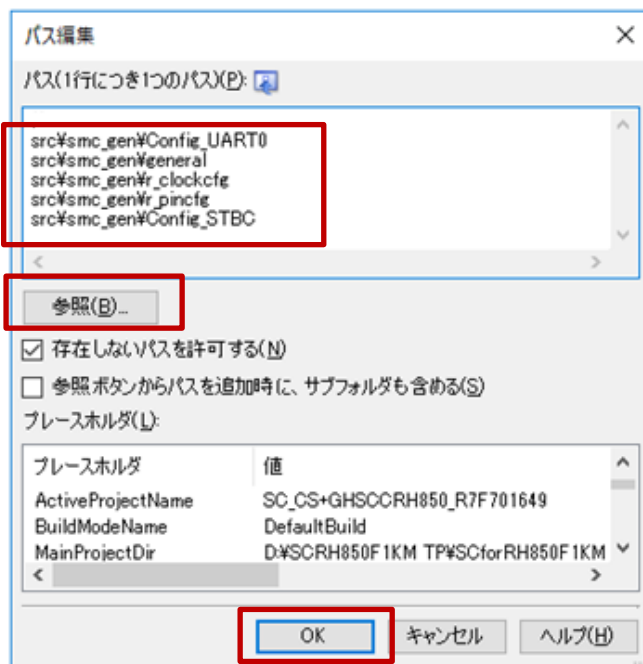



図 5-7 インクルード・パスの追加

備考: スマート・コンフィグレータは、サンプルプロジェクトが存在するフォルダ下に「src/smc_gen」フォルダを作成し、「src/smc_gen」下にファイルを生成します。

[パス編集]ダイアログボックスでパスを追加する際、[参照ボタンからパスを追加時に、サブフォルダも含める]にチェックを入れた後、「src/smc_gen」を指定すると、「src/smc_gen」下の関連するサブフォルダが追加されます。

(8)-3. ドライバ、アプリケーションコードをビルドします。CS+の [ビルド] → [ビルド・プロジェクト] メニューまたはツールバーの [ビルド]  ボタンをクリックします。

(9) 実行、デバッグ

エミュレータでのプログラム実行およびデバッグについては、「CS+ V8.09.00 統合開発環境 ユーザーズマニュアル RH850 デバッグ・ツール編」(R20UT5202)を参照してください。

5.3 デバイス変更フロー

表 2-1 サンプル・プロジェクトに記載したサンプル・プロジェクトの対象デバイスが、ご使用になるデバイスと異なる場合には、次の手順で対象デバイス/使用するファイルを変更して使用してください。

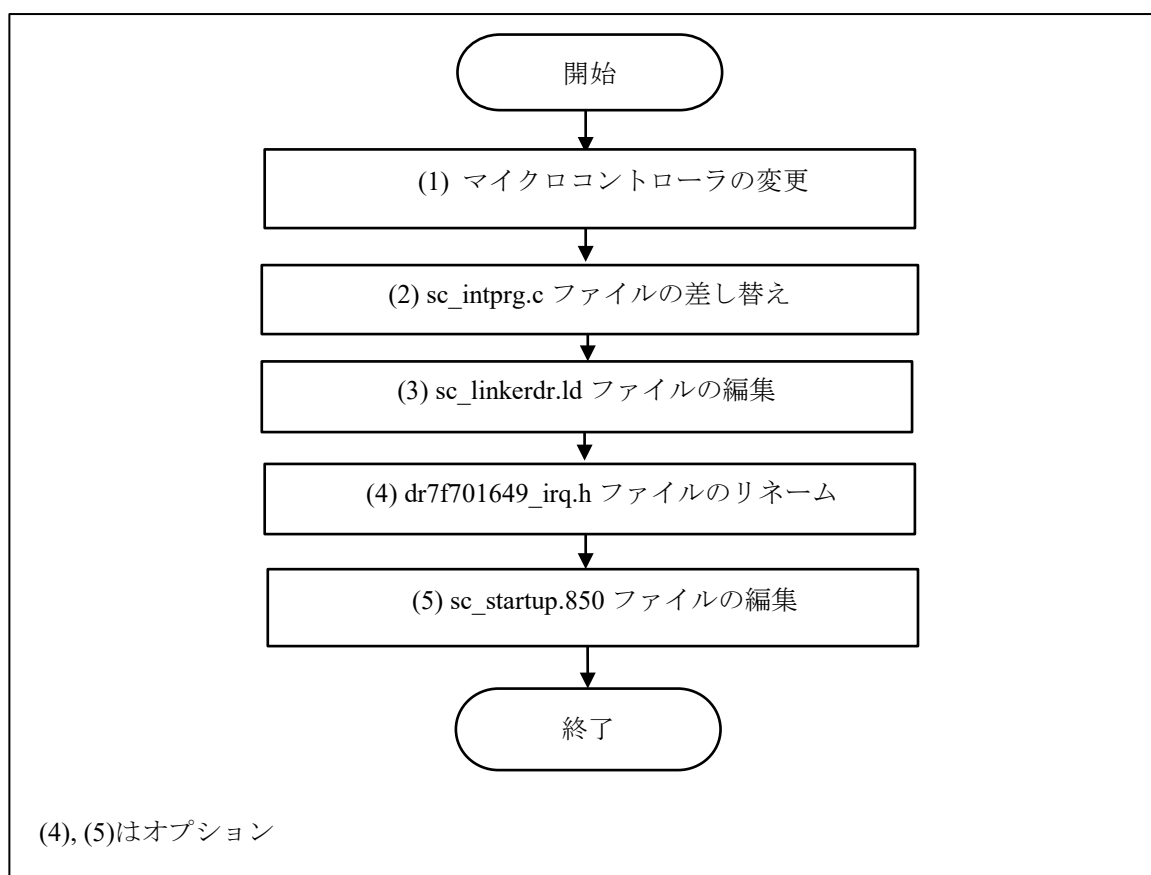


図 5-8 サンプル・プロジェクトの対象デバイスの変更 (CS+プロジェクト)

(1) マイクロコントローラの変更

- (1)-1. 「R7F701649(マイクロコントローラ)」を選択し、コンテキストメニューで [マイクロコントローラを変更...] を選択します。 [質問] ダイアログで「OK」を選択します。

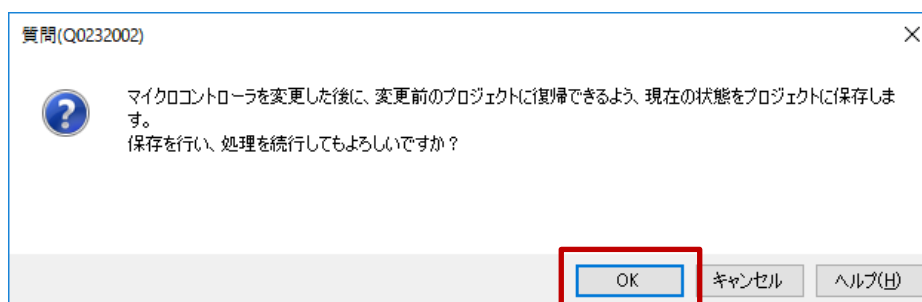
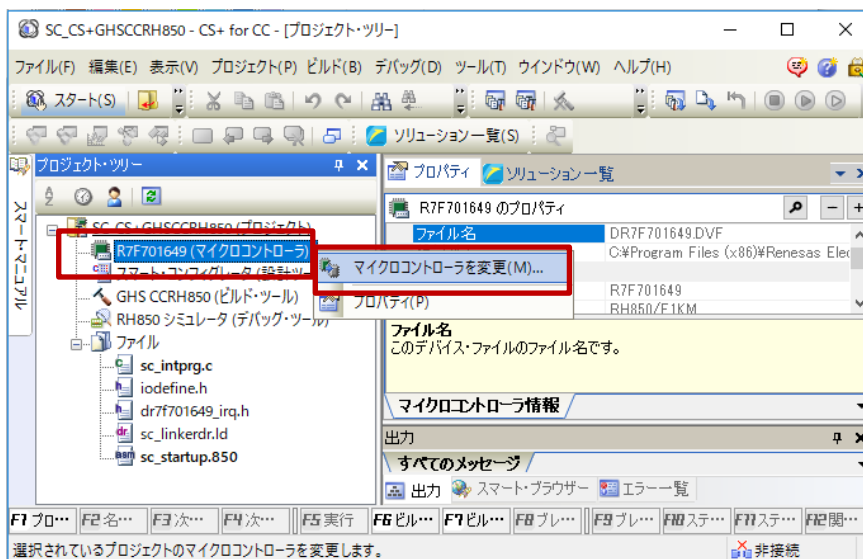


図 5-9 「マイクロコントローラを変更」の選択

- (1)-2. [マイクロコントローラの変更] ダイアログで、使用する RH850/F1KM-S1/S2/S4 のデバイスを選択します。

例：R7F701649(RH850/F1KM-S4)から R7F701684(RH850/F1KM-S1)へ変更

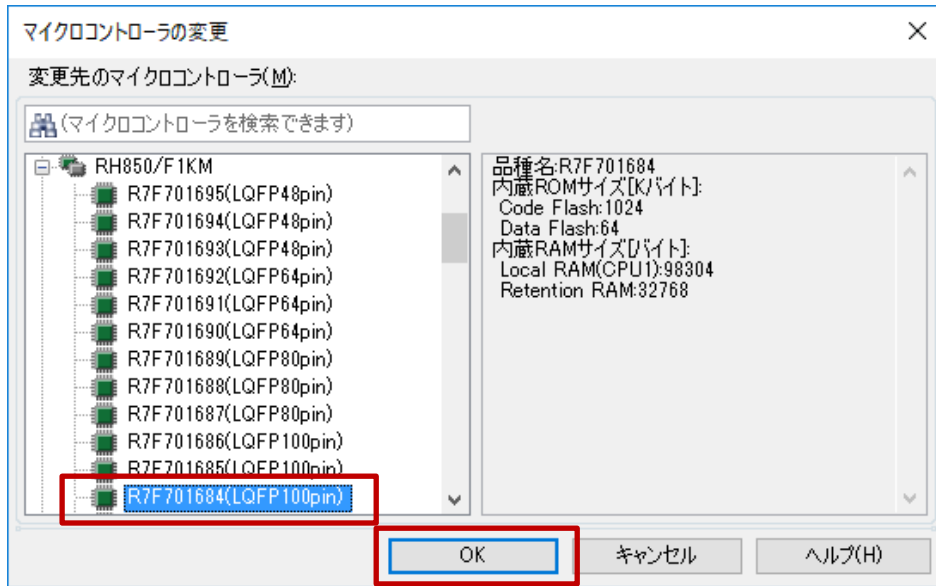


図 5-10 変更先のマイクロコントローラを選択

(1)-3. プロジェクト・ツリーのマイクロコントローラの表示が、変更後のデバイスであることを確認します。

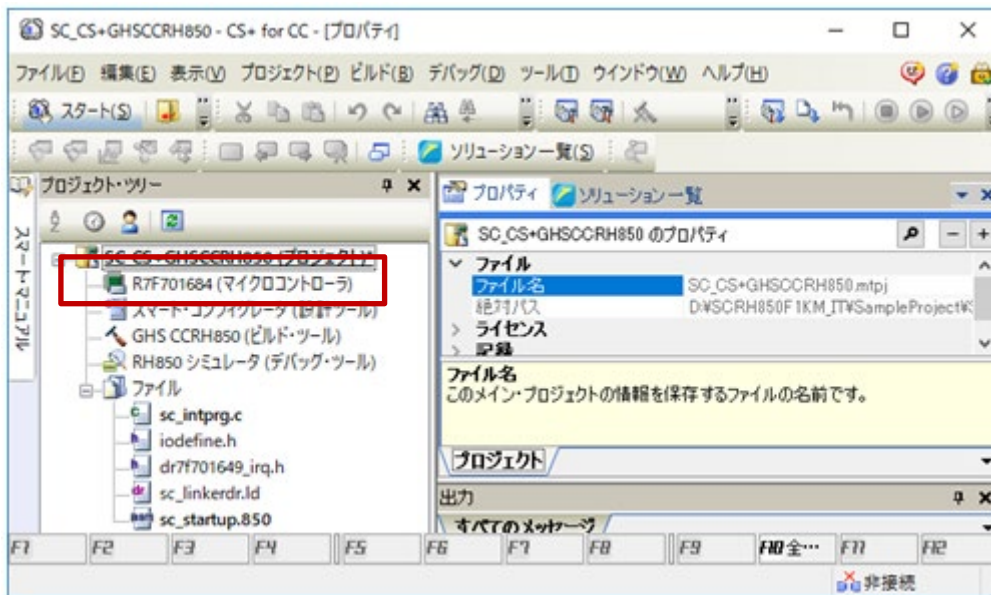


図 5-11 変更後のデバイスの確認

(1)-4. プロジェクトを保存します。[ファイル] → [プロジェクトを保存] を選択します。

(2) sc_intprg.c ファイルの差し替え

本手順は、(1) で RH850/F1KM-S4 グループから RH850/F1KM-S1/S2 グループへ変更した場合のみ行ってください。

- (2)-1. サンプル・プロジェクトに含まれる sc_intprg.c は、RH850/F1KM-S4 用のため RH850/F1KM-S1 または S2 用のファイルへ差し替えます。Windows エクスプローラで、intprg/sc_intprg-S1.c または sc_intprg-S2.c を「sc_intprg.c」へリネームしサンプル・プロジェクトの sc_intprg.c と差し替えてください。

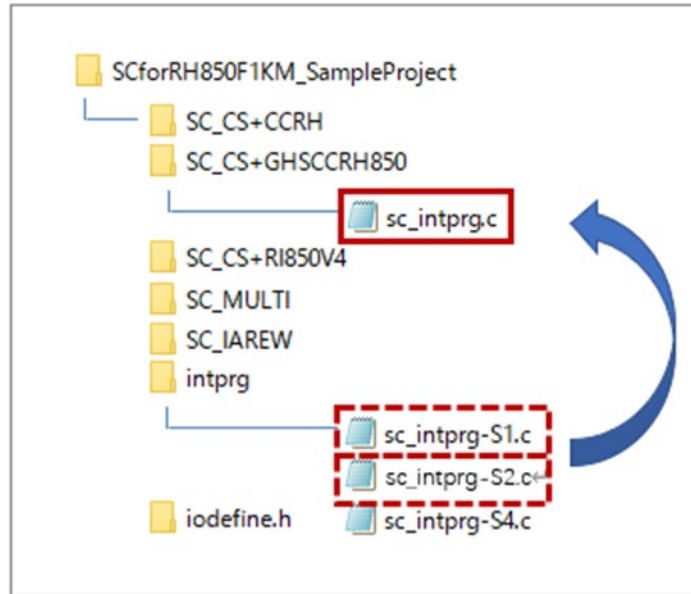


図 5-12 sc_intprg.c ファイルの差し替え

(図では、説明不要のファイルの記載を省略しています。)

(3) sc_linkerdr.ld ファイルの編集

- (3)-1. sc_linkerdr.ld ファイルを任意のエディタで開きます。

- (3)-2. 使用するマイクロコントローラの仕様に合わせてメモリの開始アドレスとサイズを変更してください。(ファイルヘッダー内のマイクロコントローラ名の変更は任意です。)

```

44 /*
45 /*=====
46 /* Environment:
47 /*          Device:      R7F701649
48 /*          IDE:         GHS Multi for V800 V6.xx or late
49 /*=====
50
51 CONSTANTS
52 {
53     zero_start = 0xffff8000
54 }
55
56 MEMORY
57 {
58     iROM_0      : ORIGIN = 0x00000000, LENGTH = 4096k
59     iROM_1      : ORIGIN = 0x01000000, LENGTH = 32k
60     iRAM_0      : ORIGIN = 0xFEBC0000, LENGTH = 256k
61     iRAM_R      : ORIGIN = 0xFEFD0000, LENGTH = 64k
62 }

```

図 5-13 メモリの開始アドレスとサイズの変更

- (3)-3. EIINTTBL セクションの終了アドレスである「EIINTTBL_end」を、使用するマイクロコントローラの仕様に合わせて変更してください。

```
64 SECTIONS
65 {
66
67 /* Start of internal ROM area (iROM_0) */
68
69 /* original */
70 /* .intvect                :>iROM_0 /* start of inter
71 /* .intvect_end 0x07E4      :>. /* end of inter
72 /* */
73
74 /* Changed for sample project */
75 .intvect                :>iROM_0 /* start of interr
76 .intvect_end 0x01FF      :>. /* end of interr
77 EIINTTBL_align (4)      :>iROM_0 /* start of EI mask
78 EIINTTBL_end 0x07E4     :>. /* end of EI mask
79 /* */
```

図 5-14 EIINTTBL_end の値の変更

参考：

RH850/F1KM-S1: 0x0798

RH850/F1KM-S2: 0x07E4

RH850/F1KM-S4: 0x07E4

次の手順 (4) と (5) はオプションです。(4)、(5)の手順を省略してもサンプルプロジェクトのビルドは可能です。

(4) dr7f701649_irq.h ファイルのリネーム

- (4)-1. 使用するマイクロコントローラ名と一致するように、名前を「dr7f01649_irq.h」から「dr7f0xxx_irq.h」に変更します。

(5) sc_startup.850 ファイルの編集

- (5)-1. 任意のエディタで「sc_startup.850」を開いて編集します。

- (5)-2. sc_startup.850 ファイル内のインクルードファイル記述を(4)で変更したファイル名に変更し、保存します。(ファイルヘッダー内のマイクロコントローラ名の変更は任意です。)

```
45 -----
46 Environment:
47 Device: R7F701649
48 IDE: GHS Multi for
49 -----
50
51 -----
52 Selection of external interrupt
53 User modifiable section
54 Please uncomment the required
55 -----
56 #include "dr7f701649_IRQ.h"
57 -----
```

図 5-15 インクルードファイル名の変更

5.4 CS+, GHS CCRH850 用サンプル・プロジェクトでの設定

サンプル・プロジェクトは、CS+で「空のアプリケーション(GHS CCRH850)」プロジェクトで作成し、インクルード・パスの追加の他に、次のオプションについて設定を変更しています。

- (a) [GHS CCRH850 (ビルド・ツール)] のプロパティ → [I/O ヘッダ・ファイル生成オプション] → [I/O ヘッダ・ファイル] → [ビルド時に I/O ヘッダ・ファイルを更新する]

[サンプル・プロジェクトでの設定]

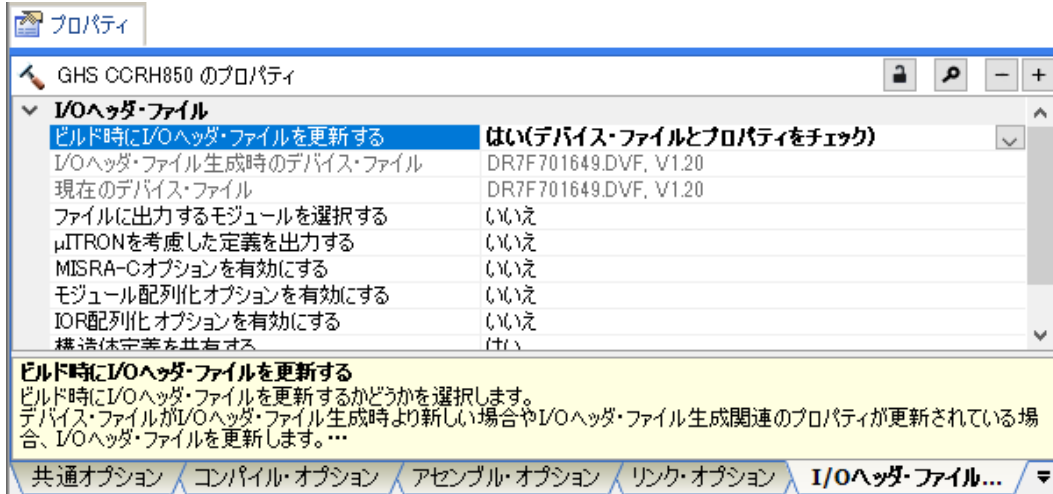


図 5-16 I/O ヘッダ・ファイルの更新設定

6. CS+, CCRH, RI850V4 用サンプル・プロジェクトの説明

6.1 CS+, CCRH, RI850V4 用サンプル・プロジェクトの構成

サンプル・プロジェクトの構成は下記です。

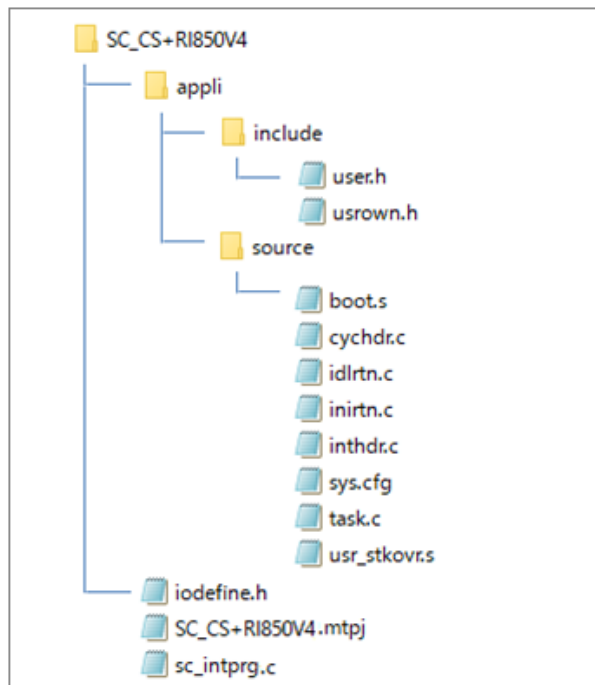


図 6-1 サンプル・プロジェクトの構成

表 6-1 CS+, CCRH, RI850V4 用サンプル・プロジェクト ファイルの概要

ファイル名	ファイルの概要
SC_CS+RI850V4.mtpj	CS+のプロジェクト・ファイル
user.h	システム依存情報(ユーザ・オウン・コーディング部)
userown.h	システム依存情報(ユーザ・オウン・コーディング部)
boot.s	ブート処理
cychdr.c	周期ハンドラ
idlrtn.c	アイドル・ルーチン
inirtn.c	初期化ルーチン
inthdr.c	割り込みハンドラ
task.c	タスク
usr_stkovr.s	オーバーフロー後処理
sys.cfg	システム・コンフィギュレーション・ファイル
iodefine.h	レジスタの定義
sc_intprg.c	EI レベルマスカブル割り込みベクタ・テーブルの定義

スマート・コンフィグレータは、上記のファイルを出力しません。

また、RI850V4 関連のファイルの詳細は、「RI850V4 V2 リアルタイム・オペレーティング・システム ユーザーズマニュアル コーディング編」(R20UT2889)を参照してください。

6.2 基本操作フロー

図 6-2 に、サンプル・プロジェクトをベースにスマート・コンフィグレータを使用する際の操作フローを以下に示します。

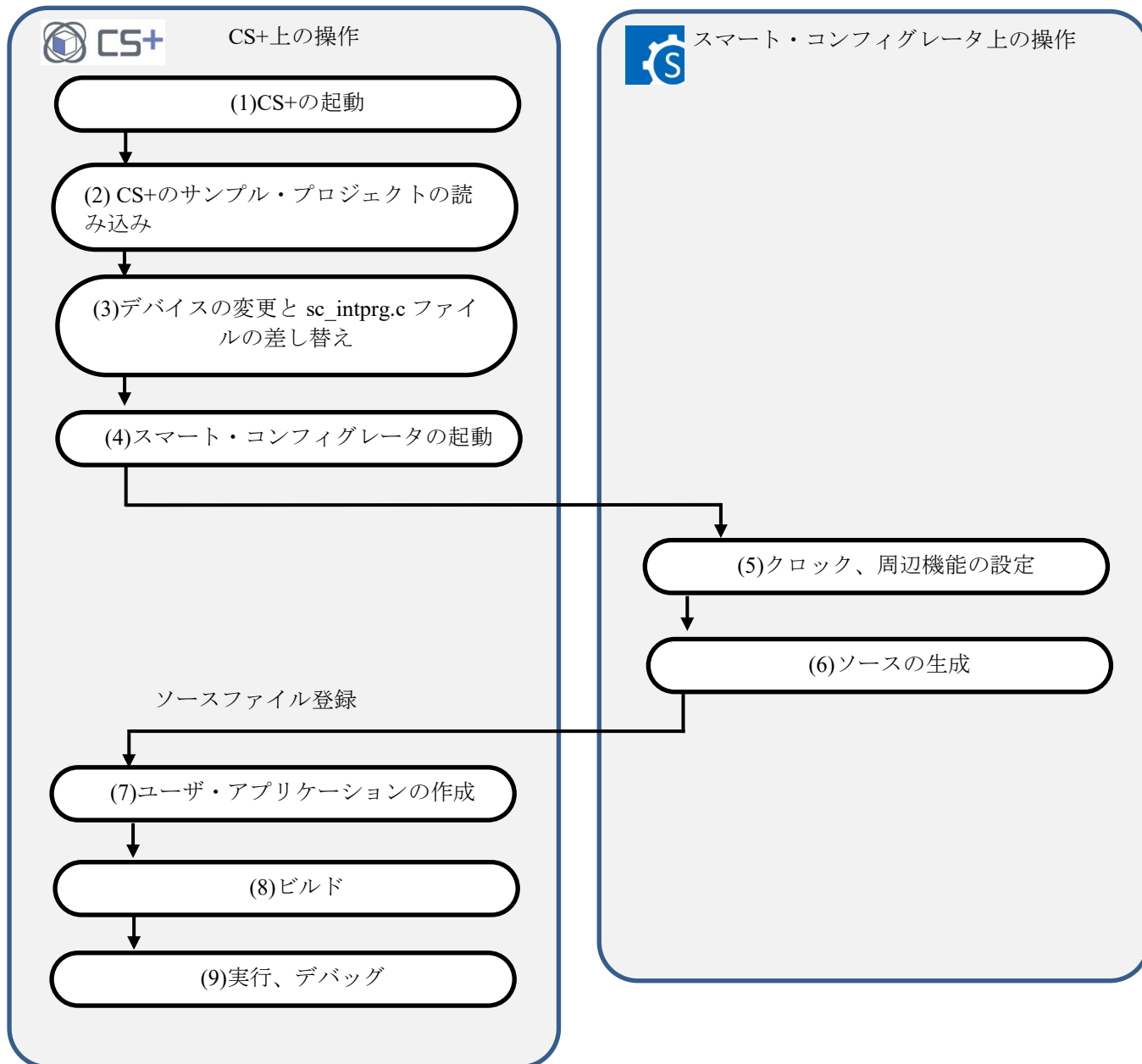


図 6-2 CS+, CCRH,RI850V4 用サンプル・プロジェクト 操作フロー

(1) CS+の起動

Windows のスタートメニューから [Renesas Electronics CS+] → [CS+ for CC(RL78,RX,RH850)] を選択します。

(2) CS+のサンプル・プロジェクトの読み込み

CS+の [ファイル] → [ファイルを開く(O)...] メニューまたは [既存のプロジェクトを開く] で、「SC_CS+RI850V4.mtpj」を選択します。

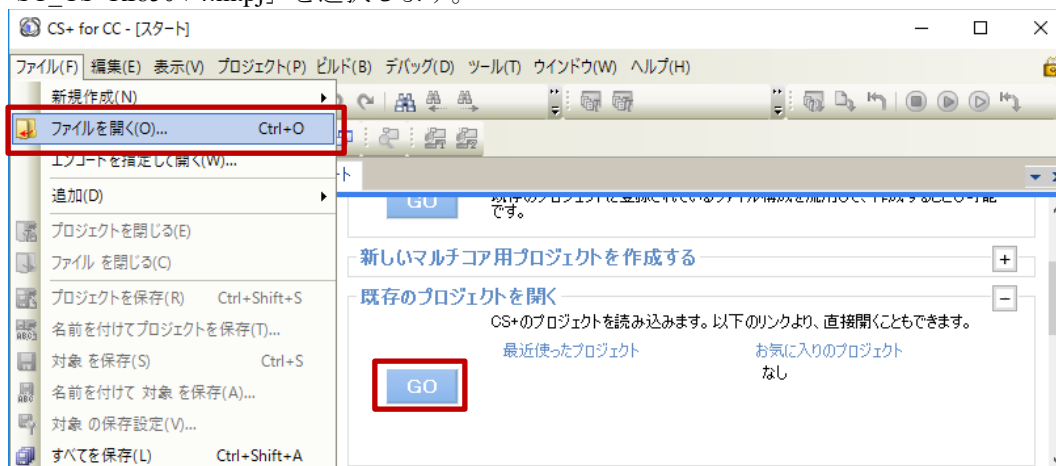


図 6-3 サンプル・プロジェクトの読み込み

サンプル・プロジェクトは、必ずアクセス制限のない場所で使用してください。(2.2 参照)

(3) デバイスの変更

サンプル・プロジェクトは、対象デバイスとして R7F701649 (RH850/F1KM-S4) を選択しています。ご使用になるデバイスと異なる場合には、「6.3 デバイス変更フロー」を参照し対象デバイスおよび使用するファイルを変更して使用してください。デバイス変更が不要な場合は、(4)へ進んでください。

(4) スマート・コンフィグレータの起動

- (4)-1. スマート・コンフィグレータのパス設定を確認します。[プロジェクト・ツリー]で[スマート・コンフィグレータ]を選択し[プロパティ]を開きます。[RH850用スマート・コンフィグレータのexeファイルパス]にスマート・コンフィグレータをインストールしたパスが設定されていることを確認します。

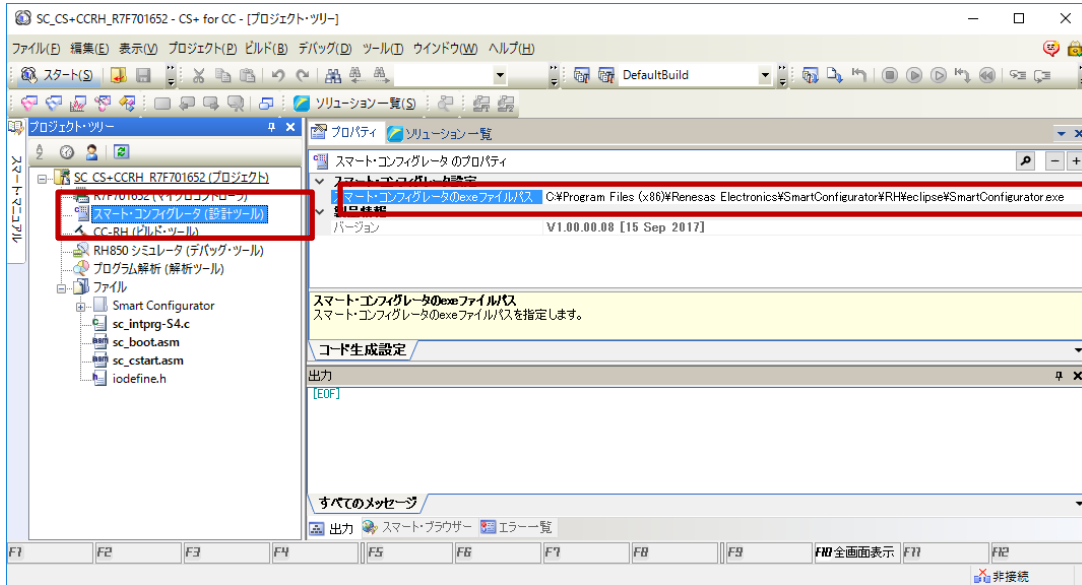


図 6-4 スマート・コンフィグレータのパス設定

- (4)-2. スマート・コンフィグレータを起動します。[プロジェクト・ツリー]で[スマート・コンフィグレータ]をダブルクリックします。

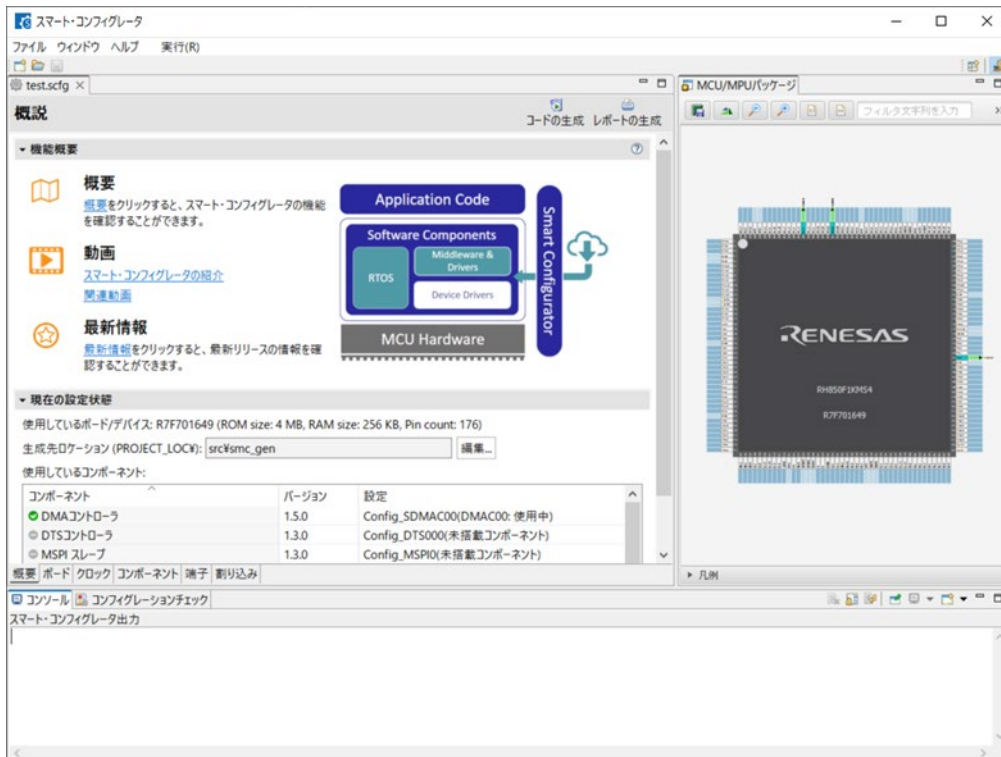



図 6-5 スマート・コンフィグレータの起動

(5)-(7)のスマート・コンフィグレータでのクロック、コンポーネント設定およびドライバ生成の手順については、「9 スマート・コンフィグレータの操作」を参照してください。

注意：スマート・コンフィグレータのコンポーネント設定で、OS タイマ（OSTM）は使用しないでください。OS タイマはサンプル・プロジェクト内の `appli/source/inirtn.c` 内で OS タイマの設定コードが記載されているためです。

(8) ビルド

ドライバ、アプリケーションコードをビルドします。CS+の [ビルド] → [ビルド・プロジェクト] メニューまたはツールバーの [ビルド]  ボタンをクリックします。

(9) 実行、デバッグ

エミュレータでのプログラム実行およびデバッグについては、「CS+ V8.09.00 統合開発環境 ユーザーズマニュアル RH850 デバッグ・ツール編」（R20UT5202）を参照してください。

6.3 デバイス変更フロー

表 2-1 サンプル・プロジェクトに記載したサンプル・プロジェクトの対象デバイスが、ご使用になるデバイスと異なる場合には、次の手順で対象デバイス/使用するファイルを変更して使用してください。

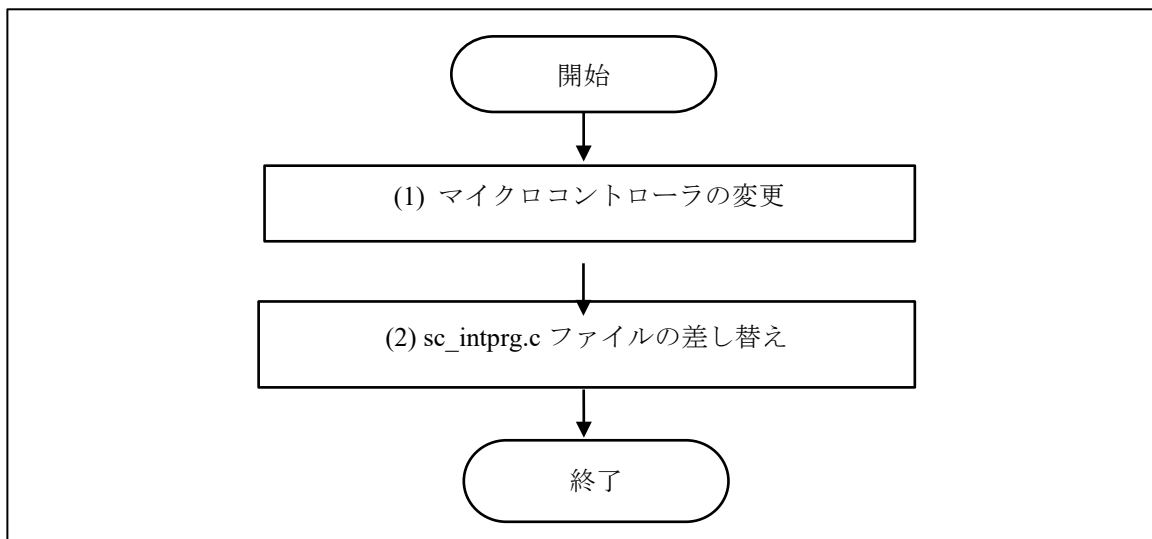


図 6-6 サンプル・プロジェクトの対象デバイスの変更（CS+プロジェクト）

(1) マイクロコントローラの変更

(1)-1. 「R7F701649(マイクロコントローラ)」を選択し、コンテキストメニューで [マイクロコントローラを変更...] を選択します。[質問] ダイアログで「OK」を選択します。

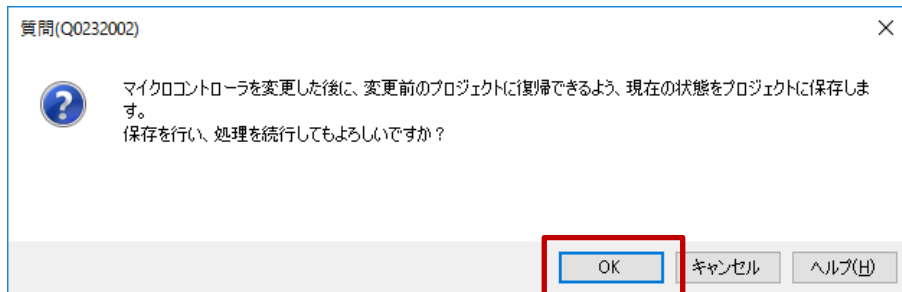
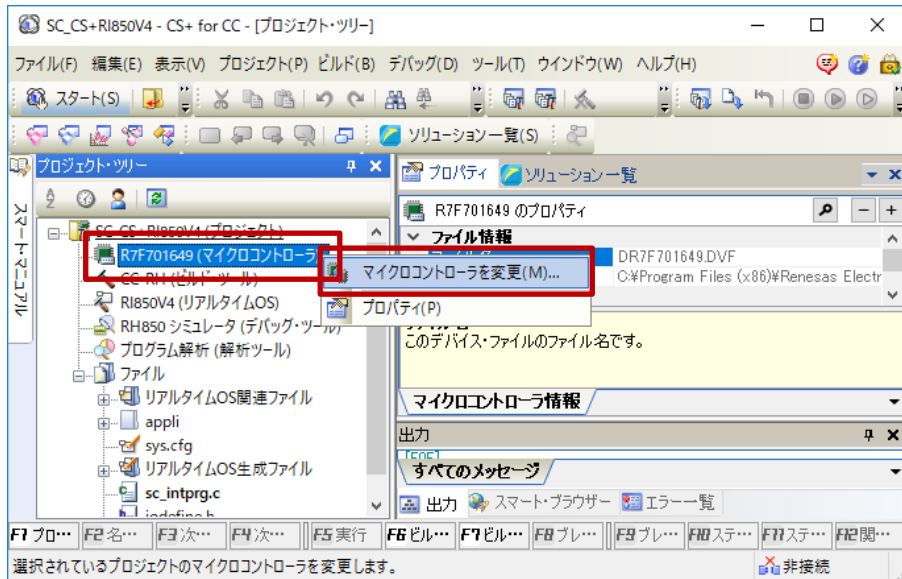


図 6-7 「マイクロコントローラを変更」の選択

(1)-2. [マイクロコントローラの変更] ダイアログで、使用する RH850/F1KM-S1/S2/S4 のデバイスを選択します。

例：R7F701649(RH850/F1KM-S4)から R7F701684(RH850/F1KM-S1)へ変更

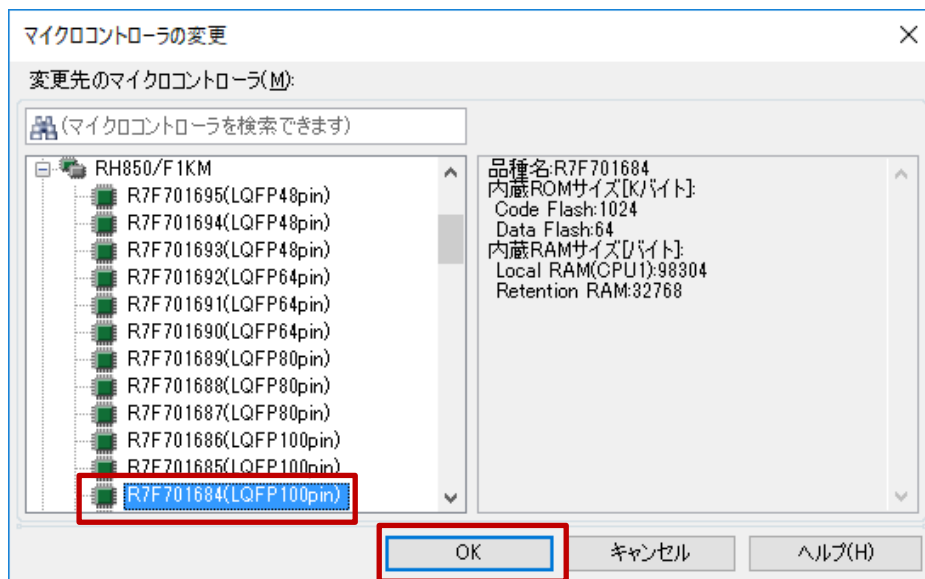


図 6-8 変更先のマイクロコントローラの選択

- (1)-3. プロジェクト・ツリーのマイクロコントローラの表示が、変更後のデバイスであることを確認します。

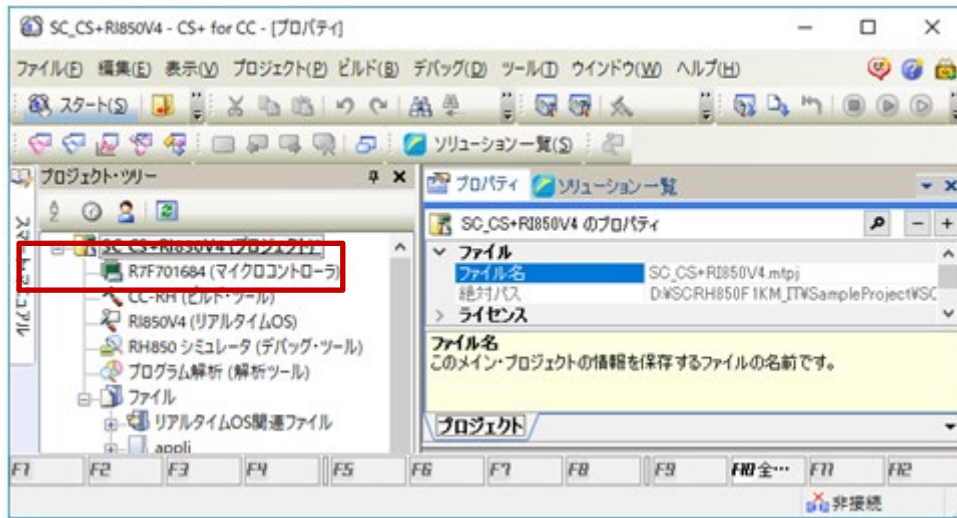


図 6-9 変更後のデバイスの確認

- (1)-4. プロジェクトを保存します。[ファイル] → [プロジェクトを保存] を選択します。

(2) sc_intprg.c ファイルの差し替え

本手順は、(1) で RH850/F1KM-S4 グループから RH850/F1KM-S1 グループへ変更した場合のみ行ってください。

- (2)-1. サンプル・プロジェクトに含まれる sc_intprg.c は、RH850/F1KM-S4 用のため RH850/F1KM-S1 または S2 用のファイルへ差し替えます。Windows エクスプローラで、intprg/sc_intprg-S1.c または sc_intprg-S2.c を「sc_intprg.c」へリネームしサンプル・プロジェクトの sc_intprg.c と差し替えてください。

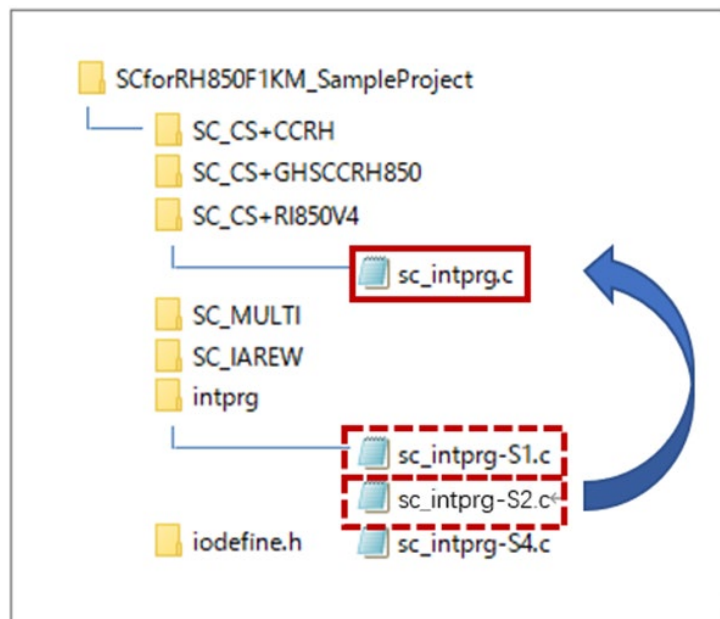


図 6-10 sc_intprg.c ファイルの差し替え

(図では、説明不要のファイルの記載を省略しています。)

6.4 CS+, CCRH, RI850V4 用サンプル・プロジェクトでの設定

サンプル・プロジェクトは、CS+で「アプリケーション(RI850V4,CC-RH)」プロジェクトで作成し、インクルード・パスの追加の他に、次のオプションについて設定を変更しています。

- (a) [CC-RH (ビルド・ツール)] のプロパティ → [共通オプション] → [よく使うオプション(コンパイル)] → [定義マクロ]

[サンプル・プロジェクトでの設定]

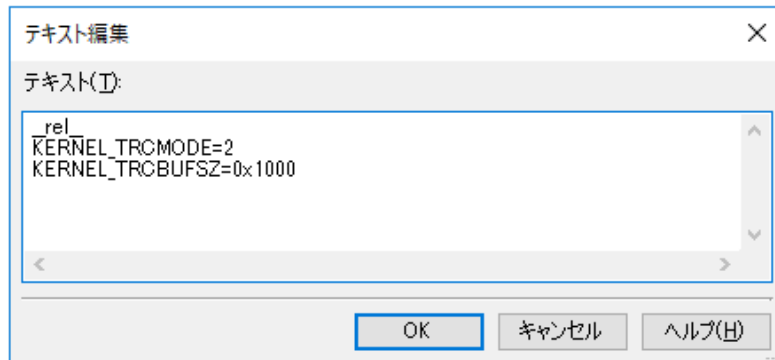


図 6-11 マクロ定義の設定

- (b) [CC-RH (ビルド・ツール)] のプロパティ → [リンク・オプション] → [セクション] → [セクションの開始アドレス]

[サンプル・プロジェクトでの設定]

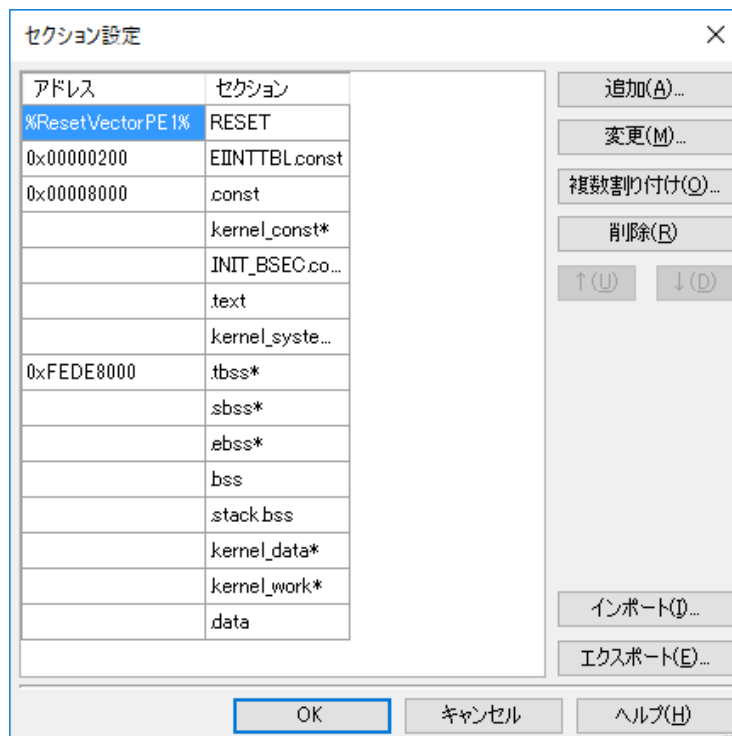


図 6-12 セクションの設定

- (c) [CC-RH (ビルド・ツール)] のプロパティ → [リンク・オプション] → [セクション] → [ROM から RAM へマッピングするセクション]

[サンプル・プロジェクトでの設定]

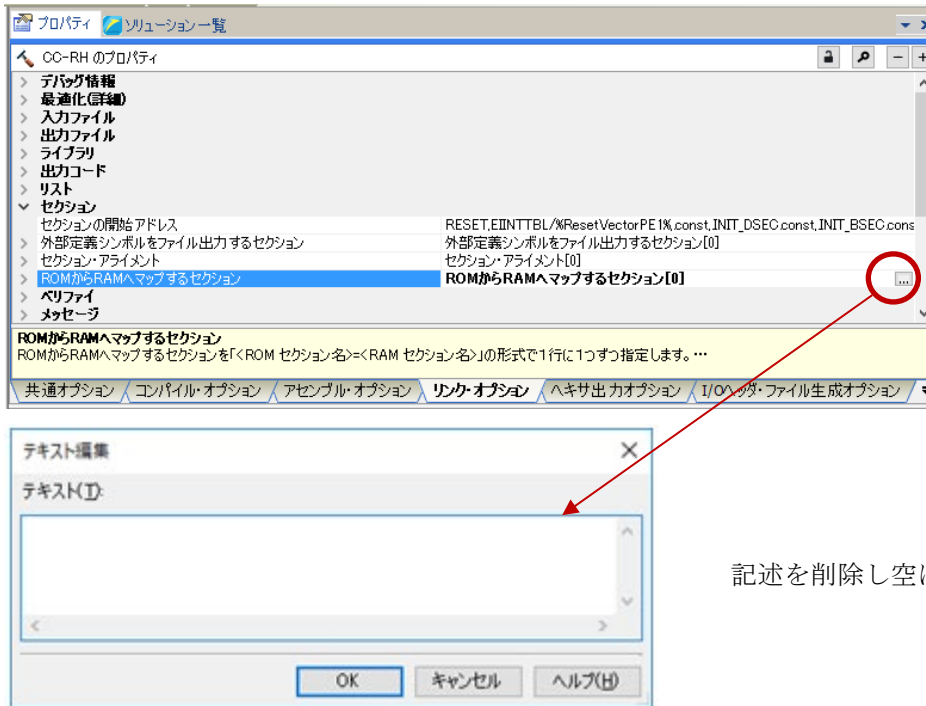


図 6-13 ROM から RAM へマッピングするセクションの設定

- (d) [CC-RH (ビルド・ツール)] のプロパティ → [I/O ヘッダ・ファイル生成オプション] → [I/O ヘッダ・ファイル] → [ビルド時に I/O ヘッダ・ファイルを更新する]

[サンプル・プロジェクトでの設定]

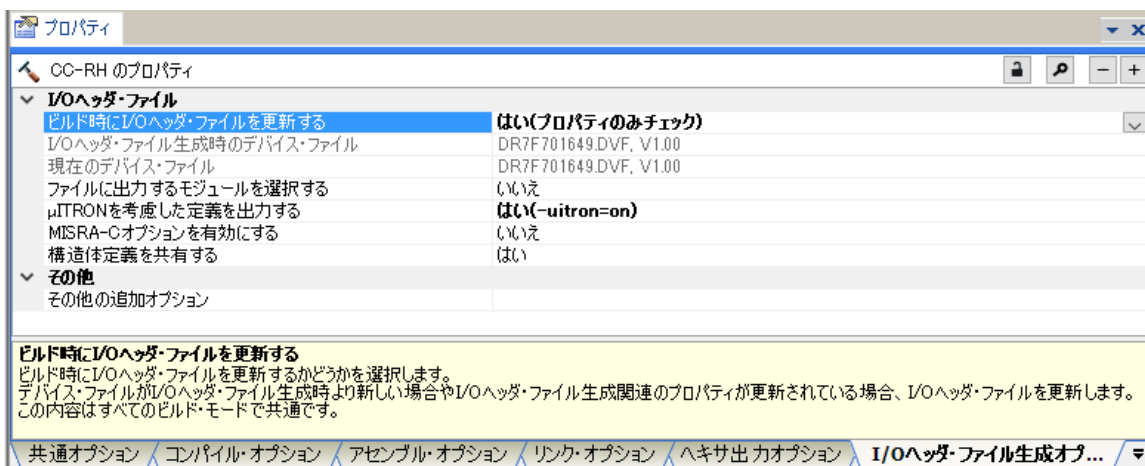


図 6-14 I/O ヘッダ・ファイルの更新設定

7. MULTI, GHS CCRH850 用サンプル・プロジェクトの説明

7.1 MULTI, GHS CCRH850 用サンプル・プロジェクトの構成

サンプル・プロジェクトの構成は下記です。

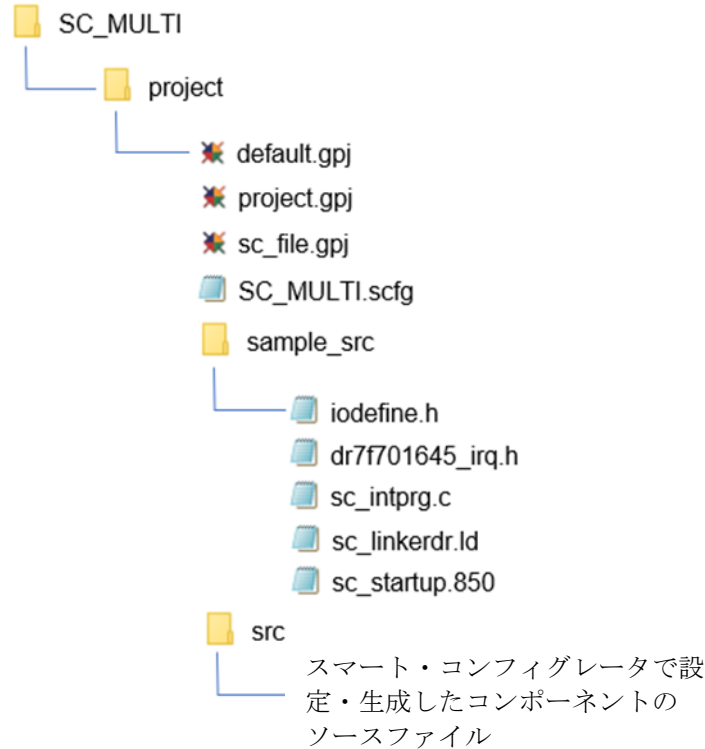


図 7-1 サンプル・プロジェクトのファイル構成

表 7-1 MULTI, GHS CCRH850 用サンプル・プロジェクト ファイル構成

ファイル名	ファイルの概要
default.gpj	MULTI の TOP プロジェクト・ファイル
project.gpj	MULTI のプロジェクト・ファイル
sc_file.gpj	スマート・コンフィグレータの MULTI 用設定ファイル
SC_MULTI.scfg	スマート・コンフィグレータのプロジェクトファイル
iodefine.h(※)	レジスタの定義
dr7f701645_irq.h (※)	例外割り込みベクタ・テーブルの定義 (マクロ定義)
sc_intprg.c (※)	EI レベルマスカブル割り込みベクタ・テーブルの定義
sc_linkerdr.ld (※)	リンク・ディレクティブ・ファイル
sc_startup.850 (※)	リセットの発生から各アプリケーション・プロジェクトに分岐するまでの処理の定義、例外割り込みベクタ・テーブルの定義、main 関数に分岐するまでのスタートアップ・ルーチンの定義

(※) スマート・コンフィグレータは、これらのファイルを出力しません。

7.2 基本操作フロー

図 7-2 に、サンプル・プロジェクトをベースにスマート・コンフィグレータを使用する際の操作フローを以下に示します。

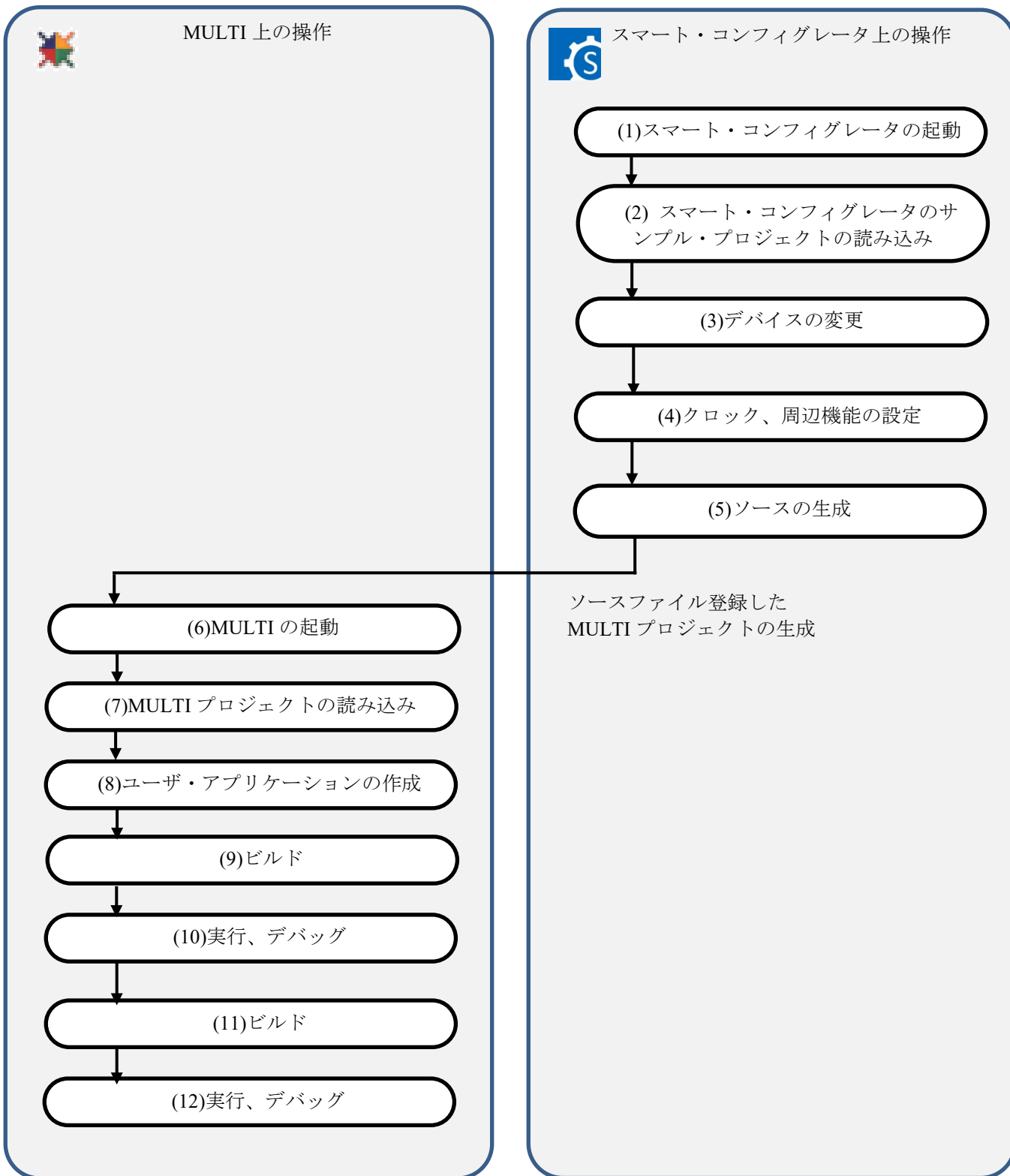


図 7-2 MULTI,GHS CCRH850 用サンプル・プロジェクト 操作フロー

(1) スマート・コンフィグレータの起動

Windows のスタートメニューから [Renesas Electronics Smart Configurator] → [Smart Configurator for RH850] を選択します。

(2) スマート・コンフィグレータのサンプル・プロジェクトの読み込み

スマート・コンフィグレータの [ファイル] → [開く...] メニュー、または [既存コンフィグレーションファイルを開く]  ボタンで、「SC_MULTI.scfg」を選択します。

サンプル・プロジェクトは、必ずアクセス制限のない場所で使用してください。(2.2 参照)

(3) デバイスの変更

サンプル・プロジェクトは、対象デバイスとして R7F701645 (RH850/F1KM-S4) を選択しています。ご使用になるデバイスと異なる場合には、スマート・コンフィグレータで対象デバイスの変更と、「7.3 デバイス変更フロー」を参照し使用するファイルを変更して使用してください。デバイス変更が不要な場合は、(4)へ進んでください。

・スマート・コンフィグレータで対象デバイスの変更

[ボード] ページのデバイス選択で変更します。

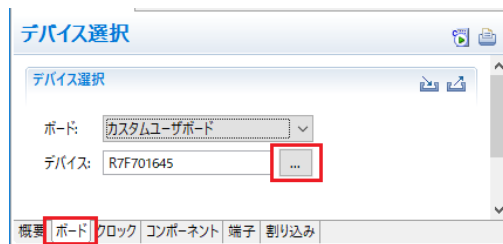


図 7-3 スマート・コンフィグレータのデバイス変更

(4)-(5)のスマート・コンフィグレータでのクロック、コンポーネント設定およびドライバ生成の手順については、「9 スマート・コンフィグレータの操作」を参照してください。

(6) MULTI の起動

Windows のスタートメニューから [Green Hills Software] → [MULTI] を選択します。

(7) MULTI プロジェクトの読み込み

MULTI の [Components] → [Open Project Manager...] メニューを選択します。[Select a project to open] ダイアログで「default.gpj」を選択します。

(8) ユーザ・アプリケーションの作成

ユーザ・アプリケーションの作成手順については、「9 スマート・コンフィグレータの操作」を参照してください。

(9) インクルード・パスの追加

iodefine.h を参照するためのインクルード・パスを追加します。

(9)-1. MULTI 上の「project.gpj」を右クリックし、[Set Build options...]を選択します。

(9)-2. [Build Options for project.gpj]ダイアログボックスの[Basic Options]タブで、[Project]-[Include Directories]をダブルクリックします。

(9)-3. [Edit List Option]ダイアログボックスで、iodefine.h のパスを追加します。

(10) サンプルファイルの追加

サンプルのファイルをビル対象として追加します。

(10)-1. MULTI 内の「project.gpj」を右クリックし、[Add File into project.gpj...]を選択します。

(10)-2. [hoose file(s) to add:]ダイアログボックスで、「sample_src」フォルダ以下のすべてのファイルを追加します。

(11) ビルド

MULTI の [Build] → [Build Program Project] を選択します。

(12) 実行、デバッグ

エミュレータでのプログラム実行およびデバッグについては、MULTI のユーザーズマニュアルを参照してください。

7.3 デバイス変更フロー

表 2-1 サンプル・プロジェクトに記載したサンプル・プロジェクトの対象デバイスが、ご使用になるデバイスと異なる場合には、次の手順で対象デバイス/使用するファイルを変更して使用してください。

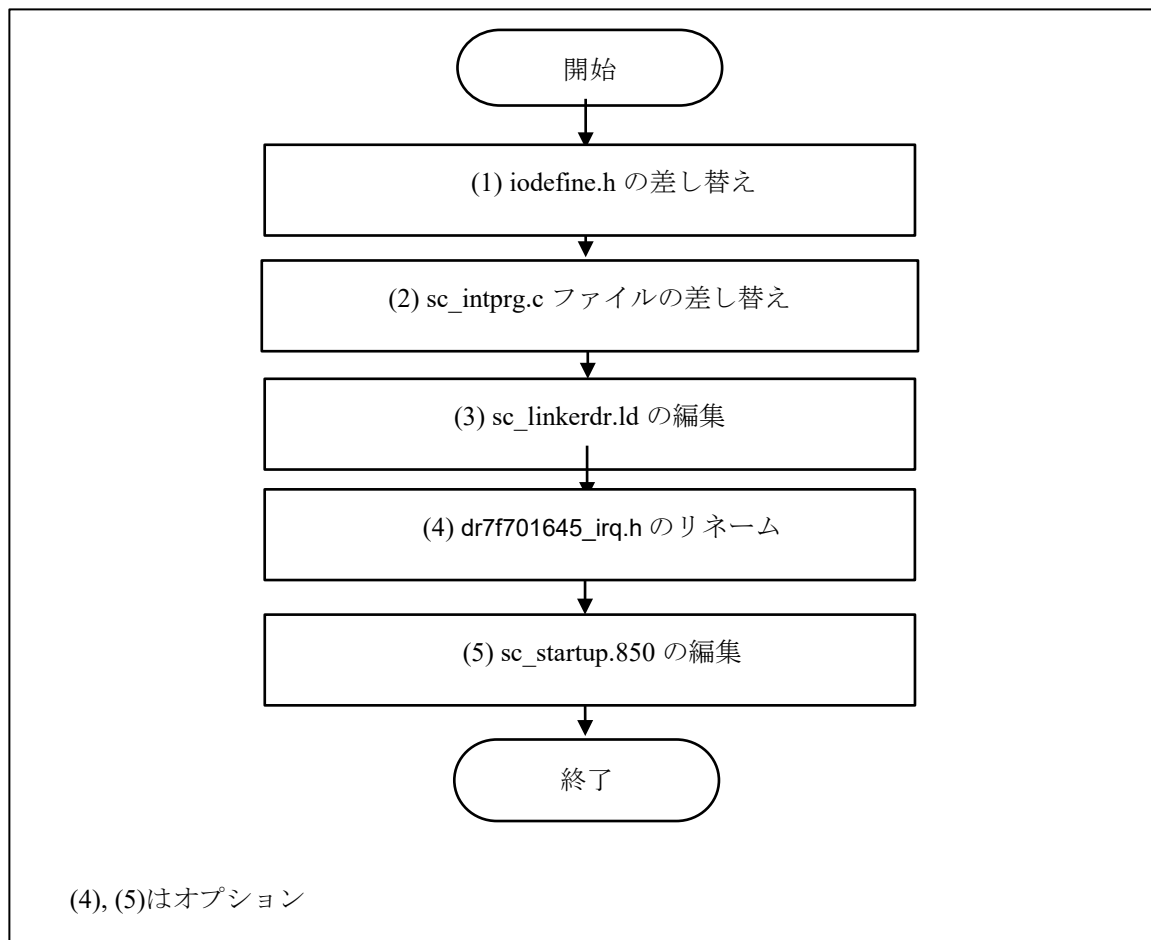


図 7-4 サンプル・プロジェクトの対象デバイスの変更 (MULTI プロジェクト)

以降、サンプル・プロジェクト「SC_MULTI」を使用して説明します。

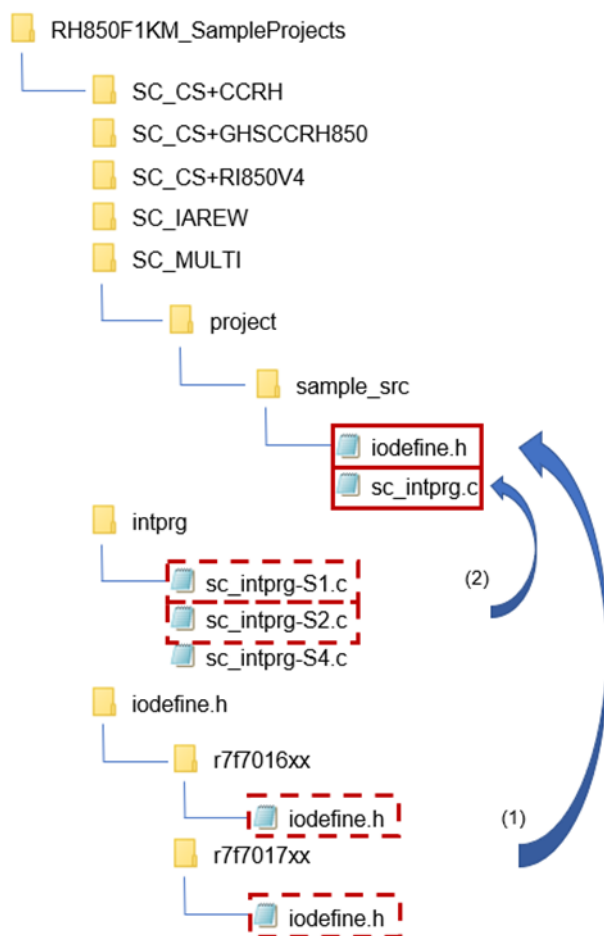


図 7-5 sc_intprg.c と iodefine.h ファイルの入れ替え

(図では、説明不要のファイルの記載を省略しています。)

(1) iodefine.h の差し替え

Windows エクスプローラで、使用するデバイスの iodefine.h へ差し替えます。iodefine.h フォルダから使用するデバイスの「iodefine.h」をサンプル・プロジェクトへコピーします。

(2) sc_intprg.c ファイルの差し替え

本手順は、RH850/F1KM-S4 グループから RH850/F1KM-S1 または S2 グループへ変更した場合のみ行ってください。

(2)-1. サンプル・プロジェクトに含まれる sc_intprg.c は、RH850/F1KM-S4 用のため RH850/F1KM-S1 用のファイルへ差し替えます。Windows エクスプローラで、intprg/sc_intprg-S1.c または sc_intprg-S1.c をサンプル・プロジェクトへコピーします。

(2)-2. サンプル・プロジェクトの「sc_intprg.c」を削除します。

(2)-3. サンプル・プロジェクトの「sc_intprg-S1.c」または「sc_intprg-S2.c」を「sc_intprg.c」へリネームします。

(3) sc_linkerdr.ld ファイルの編集

(3)-1. sc_linkerdr.ld を任意のエディタで開きます。

(3)-2. 使用するマイクロコントローラの仕様に合わせてメモリの開始アドレスとサイズを変更してください。(ファイルヘッダー内のマイクロコントローラ名の変更は任意です。)

```

44 /*
45 /*-----
46 /* Environment:
47 /*      Device:      R7F701645
48 /*      IDE:        GHS Multit for V800 V6.xx or later
49 /*-----
50
51 CONSTANTS
52 {
53     zero_start = 0xffff8000
54 }
55
56 MEMORY
57 {
58     iROM_0      : ORIGIN = 0x00000000, LENGTH = 4096k
59     iROM_1      : ORIGIN = 0x01000000, LENGTH = 32k
60     iRAM_0      : ORIGIN = 0xFEB00000, LENGTH = 256k
61     iRAM_R      : ORIGIN = 0xFE000000, LENGTH = 64k
62 }
63
64 SECTIONS

```

図 7-6 メモリの開始アドレスとサイズの変更

(3)-3. EIINTTBL セクションの終了アドレスである「EIINTTBL_end」を、使用するマイクロコントローラの仕様に合わせて変更してください。

```

64 SECTIONS
65 {
66
67 /* Start of internal ROM area (iROM_0) */
68
69 /* original */
70 /* .intvect          :>iROM_0 /* start of inter
71 /* .intvect_end 0x07E4      :>. /* end of inter
72 /* */
73
74 /* Changed for sample project */
75 .intvect          :>iROM_0 /* start of interr
76 .intvect_end 0x01FF      :>. /* end of interr
77 EIINTTBL_align (4)      :>iROM_0 /* start of EI mask
78 EIINTTBL_end 0x07E4      :>. /* end of EI mask
79 /* */

```

図 7-7 EIINTTBL_end の値の変更

参考：

RH850/F1KM-S1: 0x0798

RH850/F1KM-S2: 0x07E4

RH850/F1KM-S4: 0x07E4

次の手順 (4) と (5) はオプションです。(4)、(5)の手順を省略してもサンプルプロジェクトのビルドは可能です。

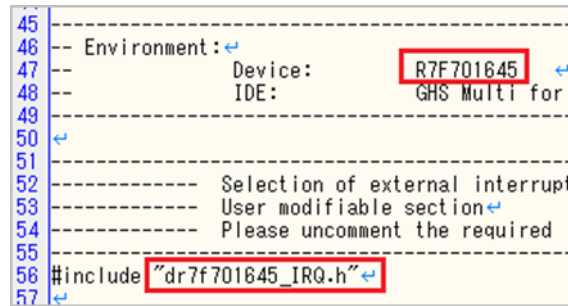
(4) dr7f701649_irq.h ファイルのリネーム

(4)-1. 使用するマイクロコントローラ名と一致するように、名前を「dr7f01649_irq.h」から「dr7f0xxxx_irq.h」に変更します。

(5) sc_startup.850 ファイルの編集

(5)-1. 任意のエディタで「sc_startup.850」を開いて編集します。

(5)-2. sc_startup.850 ファイル内のインクルードファイル記述を(4)で変更したファイル名に変更し、保存します。(ファイルヘッダー内のマイクロコントローラ名の変更は任意です。)



```
45 -----
46 Environment:
47 Device: R7F701645
48 IDE: GHS Multi for
49 -----
50
51
52 ----- Selection of external interrupt
53 ----- User modifiable section
54 ----- Please uncomment the required
55 -----
56 #include "dr7f701645_IRQ.h"
57 -----
```

図 7-8 インクルードファイル名の変更

7.4 MULTI, GHS CCRH850 用サンプル・プロジェクトでの設定

サンプル・プロジェクトは、スマート・コンフィグレータで出力した MULTI プロジェクトにビルド対象とインクルード・パスが追加されています。

備考: サンプルファイルのビルド対象設定と追加された「iodefinc.h」のインクルード・パスは、7.2 の手順(5)で削除されます。

8. IAREW, IAR ICC 用サンプル・プロジェクトの説明

8.1 IAREW, IAR ICC 用サンプル・プロジェクトの構成

サンプル・プロジェクトの構成は下記です。

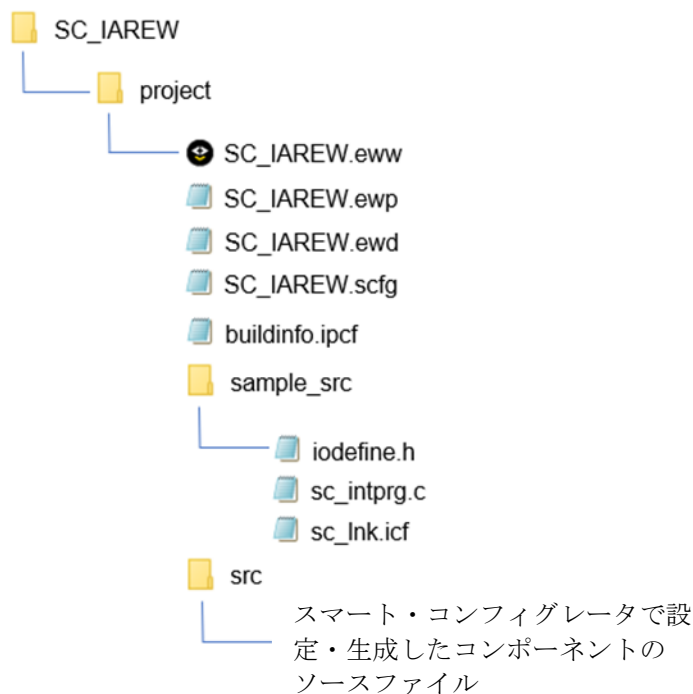


図 8-1 サンプル・プロジェクトの構成

表 8-1 IAREW, IAR ICC 用サンプル・プロジェクト ファイル概要

ファイル名	ファイルの概要
SC_IAREW.eww	IAREW のワークスペース
SC_IAREW.ewp	IAREW のプロジェクト・ファイル
SC_IAREW.ewd	IAREW のデバッグ・ファイル
buildinfo.ipcf	IAREW プロジェクト・コネクションファイル
SC_prj.scfg	スマート・コンフィグレータのプロジェクト・ファイル
sc_lnk.icf (※)	リンカ設定ファイル
sc_intprg.c (※)	EI レベルマスカブル割り込みベクタ・テーブルの定義
iodefine.h (※)	レジスタの定義

(※) スマート・コンフィグレータは、これらのファイルを出力しません。

8.2 基本操作フロー

図 8-2 に、サンプル・プロジェクトをベースにスマート・コンフィグレータを使用する際の操作フローを以下に示します。

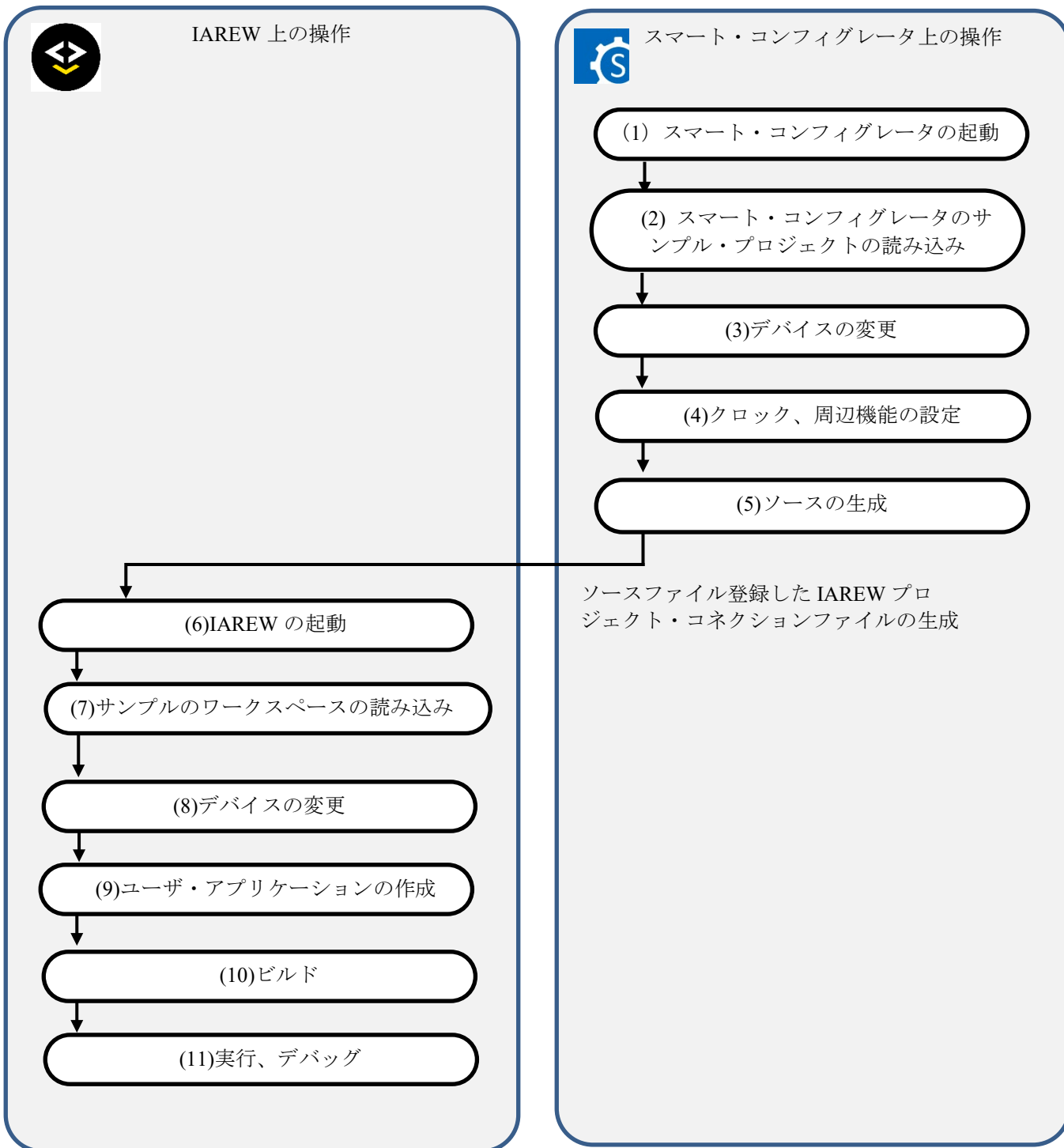


図 8-2 IAREW,IAR ICC 用サンプル・プロジェクト 操作フロー

(1) スマート・コンフィグレータの起動

Windows のスタートメニューから [Renesas Electronics Smart Configurator] → [Smart Configurator for RH850] を選択します。

(2) スマート・コンフィグレータのサンプル・プロジェクトの読み込み

スマート・コンフィグレータの [ファイル] → [開く...] メニュー、または [既存コンフィグレーションファイルを開く]  ボタンで、「SC_IAREW.scfg」を選択します。

サンプル・プロジェクトは、必ずアクセス制限のない場所で使用してください。(2.2 参照)

(3) デバイスの変更

サンプル・プロジェクトは、対象デバイスとして R7F701645 (RH850/F1KM-S4) を選択しています。ご使用になるデバイスと異なる場合には、スマート・コンフィグレータで対象デバイスの変更してください。デバイス変更が不要な場合は、(4)へ進んでください。

- ・スマート・コンフィグレータで対象デバイスの変更
[ボード] ページのデバイス選択で変更します。

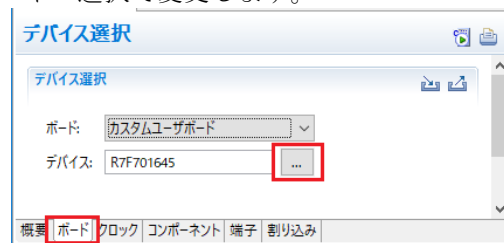


図 8-3 スマート・コンフィグレータのデバイス変更

(4)-(5)のスマート・コンフィグレータでのクロック、コンポーネント設定およびドライバ生成の手順については、「9 スマート・コンフィグレータの操作」を参照してください。

(6) IAREW の起動

Windows のスタートメニューから [IAR Systems] → [IAR Embedded Workbench] を選択します。

(7) サンプルのワークスペースの読み込み

IAREW の [File] → [Open] → [Workspace...] を選択します。[Open Workspace] ダイアログで「SC_IAREW.eww」を選択します。

(8) デバイスの変更

サンプル・プロジェクトは、対象デバイスとして R7F701645 (RH850/F1KM-S4) を選択しています。ご使用になるデバイスと異なる場合には、「8.3 デバイス変更フロー」を参照し使用するファイルを変更して使用してください。デバイス変更が不要な場合は、(9)へ進んでください。

(9) ユーザ・アプリケーションの作成

ユーザ・アプリケーションの作成手順については、「9 スマート・コンフィグレータの操作」を参照してください。

(10) ビルド

IAREW の [Project] → [Rebuild All] を選択します。

(11) 実行、デバッグ

エミュレータでのプログラム実行およびデバッグについては、IAREW のユーザズマニュアルを参照してください。

また、main 関数を実行前に INTBP レジスタの値を 0x00000200 に設定してください。

8.3 デバイス変更フロー

表 2-1 サンプル・プロジェクトに記載したサンプル・プロジェクトの対象デバイスが、ご使用になるデバイスと異なる場合には、次の手順で対象デバイス/使用するファイルを変更して使用してください。

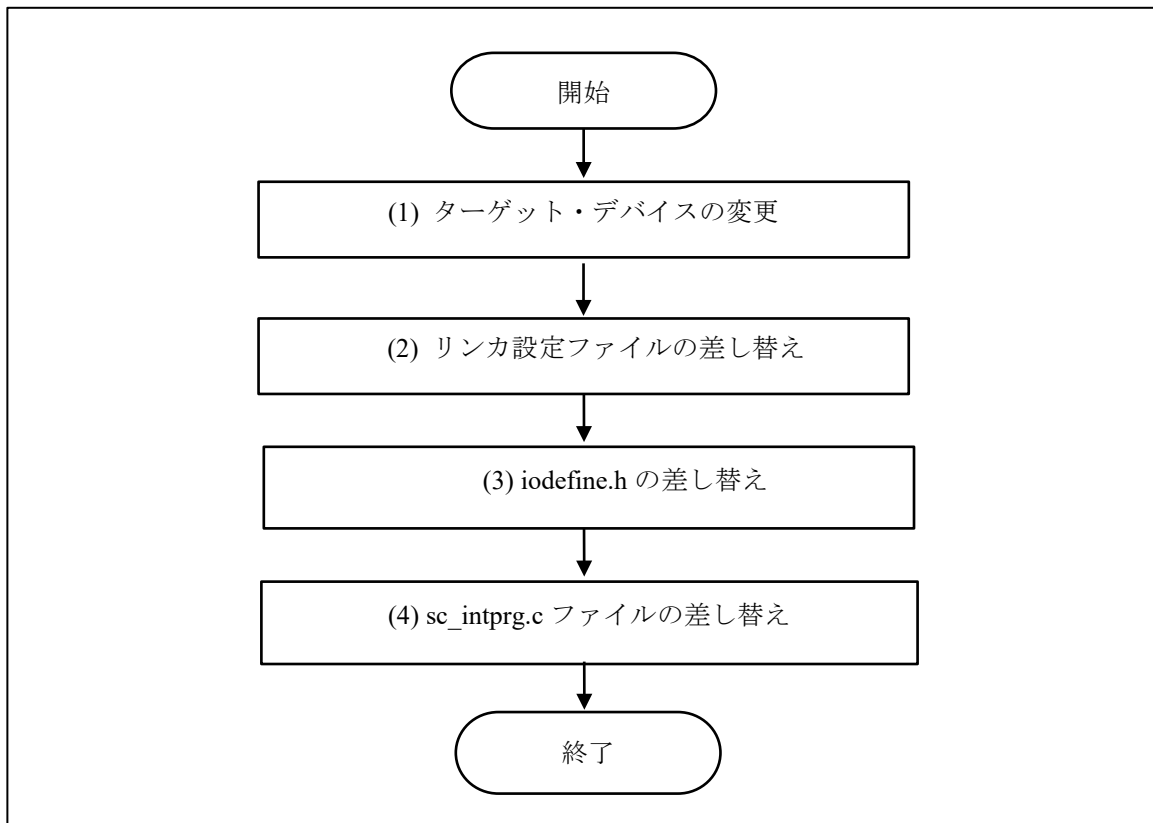


図 8-4 サンプル・プロジェクトの対象デバイスの変更 (IAREW プロジェクト)

以降、サンプル・プロジェクト「SC_IAREW」を使用して説明します。

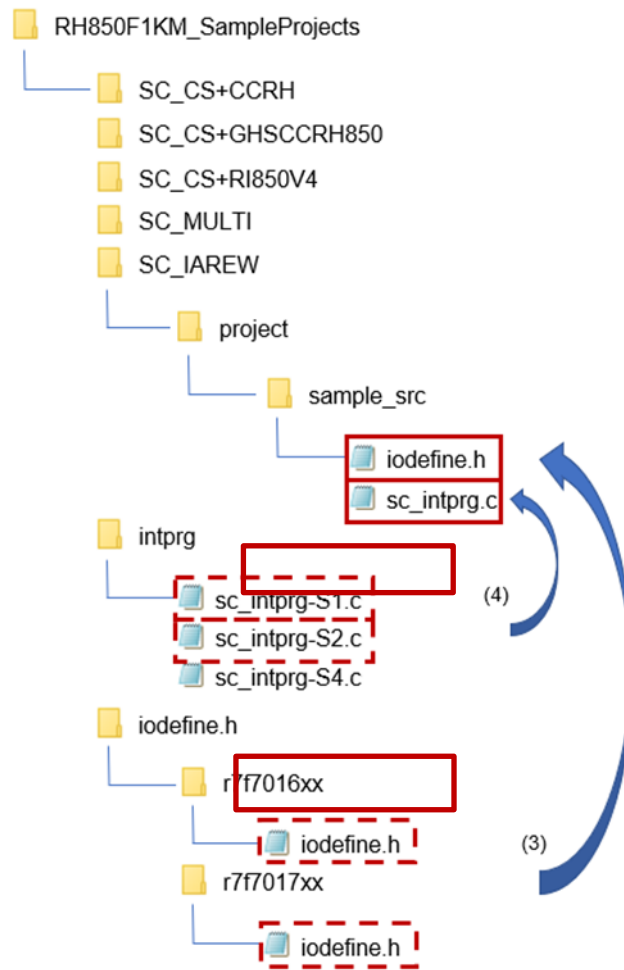


図 8-5 sc_intprg.c と iodefine.h ファイルの入れ替え

(図では、説明不要のファイルの記載を省略しています。)

(1) ターゲット・デバイスを変更

(1)-1. IAREW の [Project] → [Options...] メニューを選択します。

(1)-2. [Options for node "SC_IAREW"] ダイアログで、[General Options] → [Target] を選択します。
[Device] で、使用するデバイスを選択します。

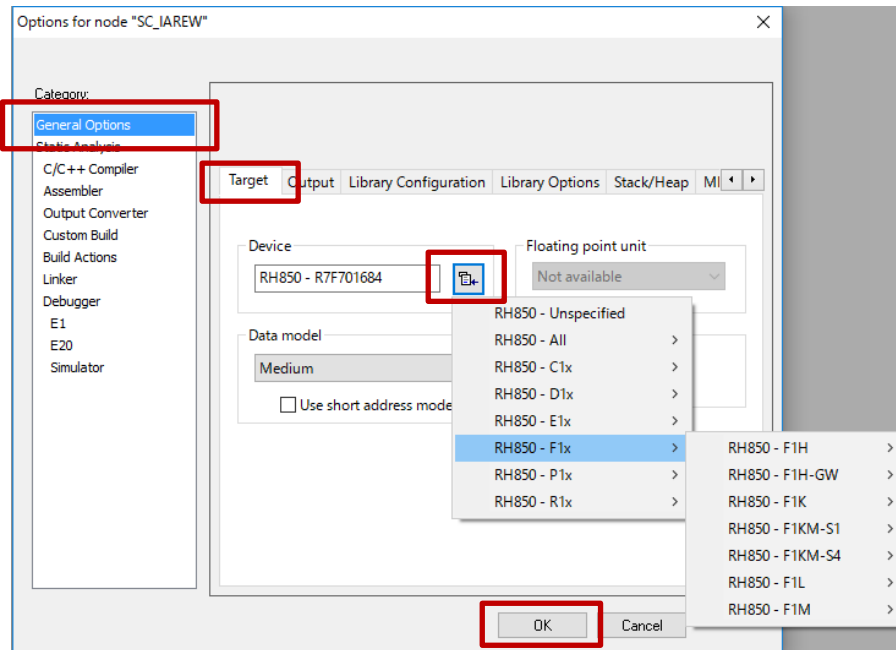


図 8-6 デバイスの変更

(2) リンカ設定ファイルの差し替え

(2)-1. Windows エクスプローラで、IAREW のインストール・フォルダ C:\Program Files (x86)\IAR Systems\Embedded Workbench x.x\rh850\config から、使用するデバイスのファイル「Inkr7f70xxxx.icf」をサンプル・プロジェクトへコピーします。

(2)-2. コピーした「Inkr7f70xxxx.icf」を編集します。任意のエディタで開きます。

(2)-3. サンプル・プロジェクトで使用するセクションを追加します。EIINTTBL セクションを 0x200 から配置するため次のように変更しファイルを保存します。

[変更前]

```

69 //Uncomment and adapt the following section if you want to place the interrupt ta
70 // define block .interrupt_core1 with alignment = 8 {←
71 //   ro section .table.interrupt_core_1 };←
72 //←
73 // "INTERRUPT":place at address mem:0x200←
74 // {←
75 //   block .interrupt_core1←
76 // };←
77 ←
78 include "layout.icf";←
79 ←
80 //-----←
81 //      End of file←
82 //-----←
[EOF]

```

[変更後]

```

69 //Uncomment and adapt the following section if you want to place the interrupt t
70 define block .interrupt_core1 with alignment = 8 {←
71   ro section .table.interrupt_core_1 };←
72 //←
73 // "INTERRUPT":place at address mem:0x200←
74 // {←
75 //   block .interrupt_core1←
76 // };←
77 ←
78 "EIINTTBL":place at address mem:0x200←
79 {←
80   block .interrupt_core1,←
81   section EIINTTBL←
82 };←
83 ←
84 ←
85 include "layout.icf";←
86 ←
87 //-----←
88 //      End of file←
89 //-----←
[EOF]

```

図 8-7 リンカ設定ファイルの変更

(行番号は、使用するファイルによって異なる場合があります。)

(3) iodefne.h の差し替え

Windows エクスプローラで、使用するデバイスの `iodefine.h` へ差し替えます。`iodefine.h` フォルダから使用するデバイスの「`iodefine.h`」をサンプル・プロジェクトへコピーします。

(4) `sc_intprg.c` ファイルの差し替え

本手順は、RH850/F1KM-S4 グループから RH850/F1KM-S1 または S2 グループへ変更した場合のみ行ってください。

(4)-1. サンプル・プロジェクトに含まれる `sc_intprg.c` は、RH850/F1KM-S4 用のため RH850/F1KM-S1 または S2 用のファイルへ差し替えます。Windows エクスプローラで、`intprg/sc_intprg-S1.c` または `sc_intprg-S2.c` をサンプル・プロジェクトへコピーします。

(4)-2. サンプル・プロジェクトの「`sc_intprg.c`」を削除します。

(4)-3. サンプル・プロジェクトの「`sc_intprg-S1.c`」を「`sc_intprg.c`」へリネームします。

8.4 IAREW, IAR ICC 用サンプル・プロジェクトでの設定

サンプル・プロジェクトは、IAREW でプロジェクト作成し、インクルード・パスの追加の他に、次のオプションについて設定を変更しています。

- (a) IAREW の [Project] → [Options...] メニュー → [Options for node "SC_IAREW"] ダイアログ
→ [Linker] → [Linker configuration file]

[サンプル・プロジェクトでの設定]

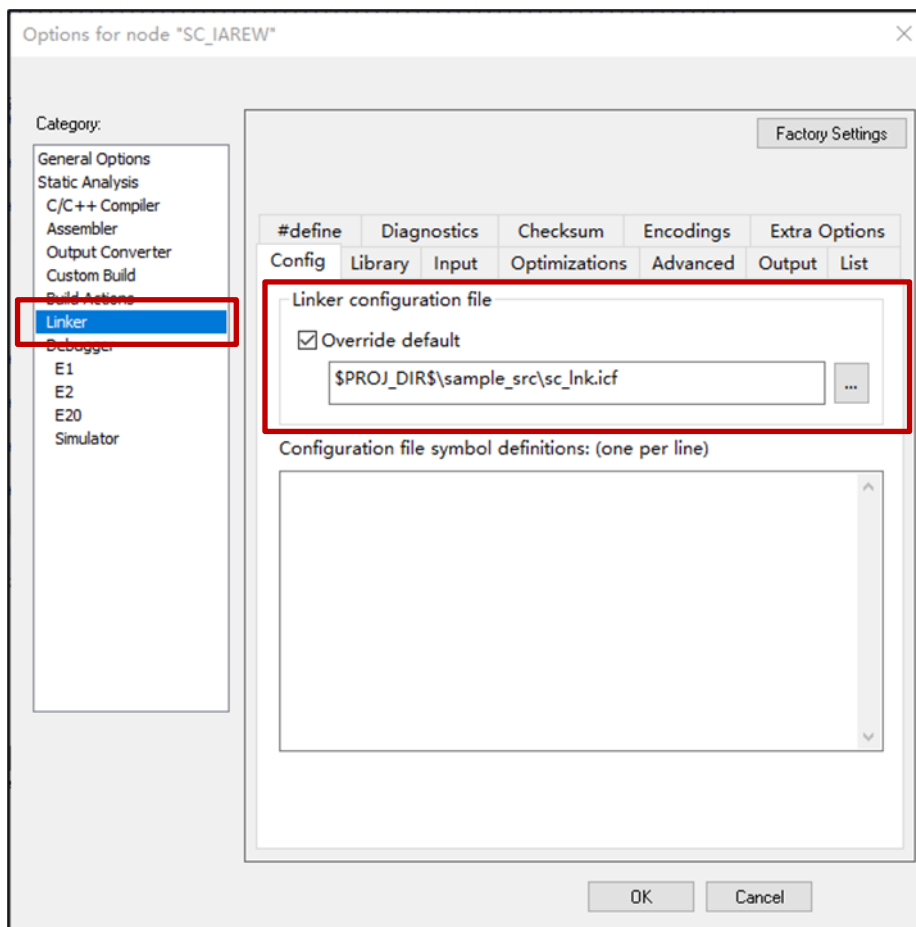


図 8-8 リンカ設定ファイルの設定

9. スマート・コンフィグレータの操作

スマート・コンフィグレータでデバイス周辺機能のドライバを設定およびコード生成する操作に概要について説明します。

詳細は、「スマート・コンフィグレータ ユーザーガイド：CS+編」 (R20AN0516)を参照してください。

9.1 周辺機能(ソフトウェアコンポーネント)設定

- (1) [クロック設定] タブで、デバイスのクロックを構成します。

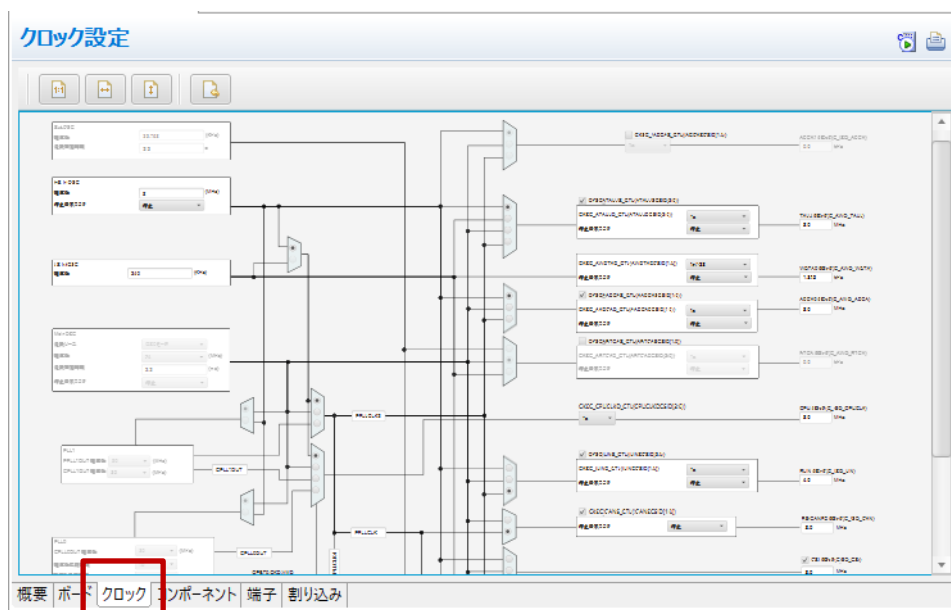



図 9-1 クロック設定

- (2) [コンポーネント] タブで、デバイスの周辺機能の追加・設定をします。周辺機能は、ソフトウェアコンポーネントとして設定します。[コンポーネントの追加]  アイコンをクリックします。

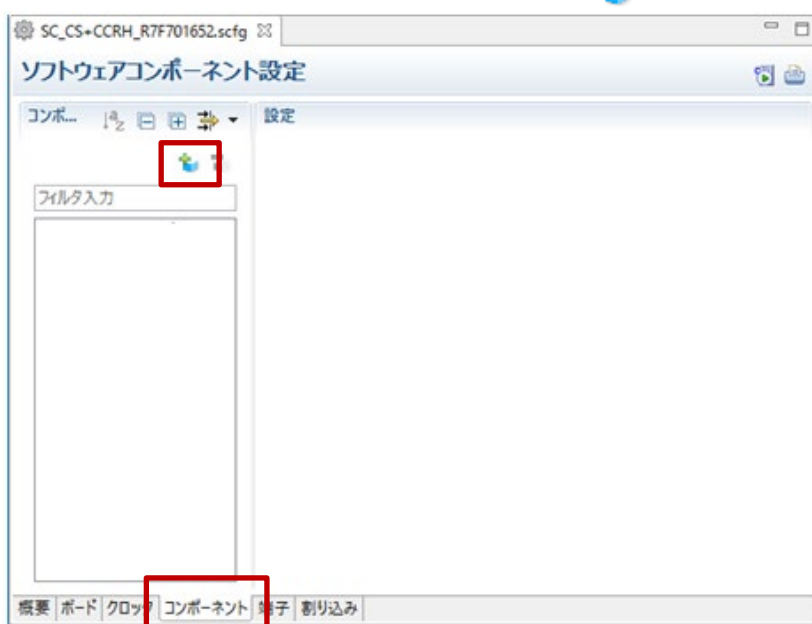


図 9-2 周辺機能の追加

備考：RH850 では、FIT(Firmware Integration Technology)モジュールのミドルウェアのインポートは未サポートです。

- (3) [コンポーネントの追加] ダイアログの [ソフトウェアコンポーネントの選択] ページで、コンポーネントを選択します。リストから使用するコンポーネントを選択し [次へ] をクリックします。

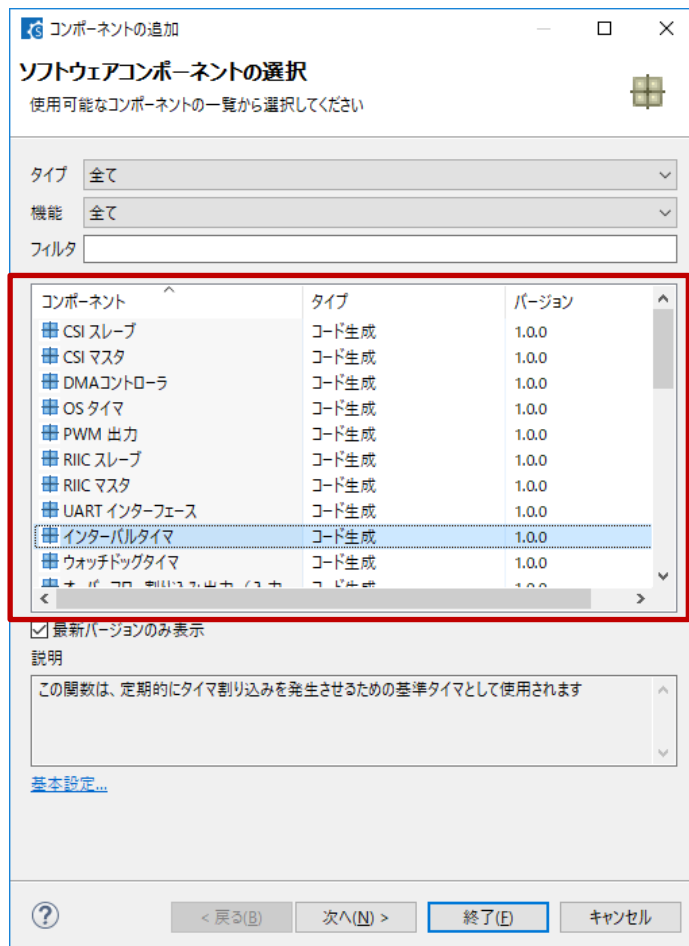


図 9-3 ソフトウェアコンポーネントの選択

- (4) 選択したコンポーネントのコンフィグレーション名とリソースを選択します。[コンポーネントの追加] ダイアログの [選択したコンポーネントのコンフィグレーションを追加します] ページで、適切なコンフィグレーション名を入力、またはデフォルト名を使用します。リソースを選択、またはデフォルトのリソースを使用します。選択後、[終了] をクリックします。

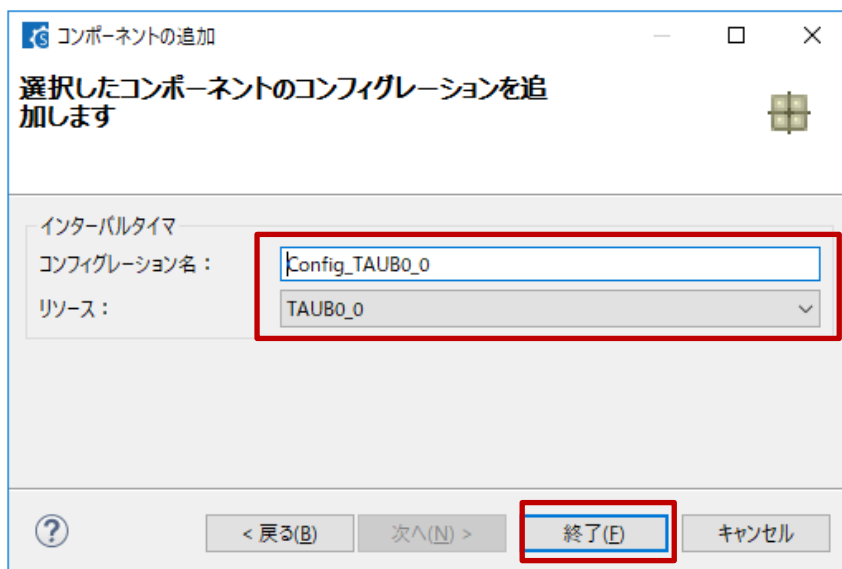


図 9-4 コンフィグレーション名とリソースの設定

- (5) コンポーネントのコンフィグレーションを設定します。コンポーネントツリーにあるコンフィグレーションをクリックします。右側の設定パネルで詳細を設定します。

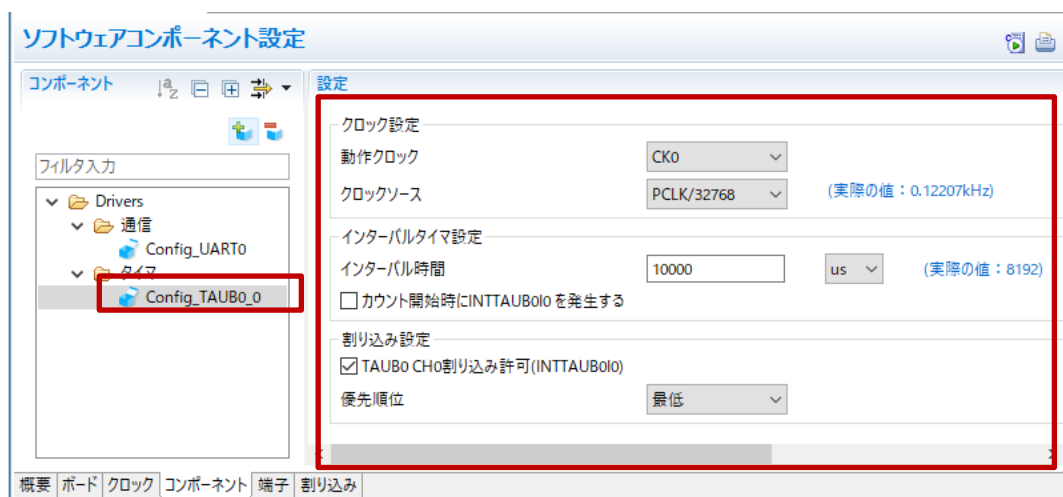


図 9-5 ソフトウェアコンポーネントの設定

- (6) 使用するコンポーネントについて、(2)から(5)の操作を行います。

9.2 ドライバの生成

- (1). [コード生成] ボタンをクリックします。スマート・コンフィグレータによって生成されたソースファイルは、<ProjectDir>\\$src\\$smc_gen フォルダに保管されます。

<ProjectDir>は、スマート・コンフィグレータのプロジェクト・ファイル (scfg) が存在するフォルダです。

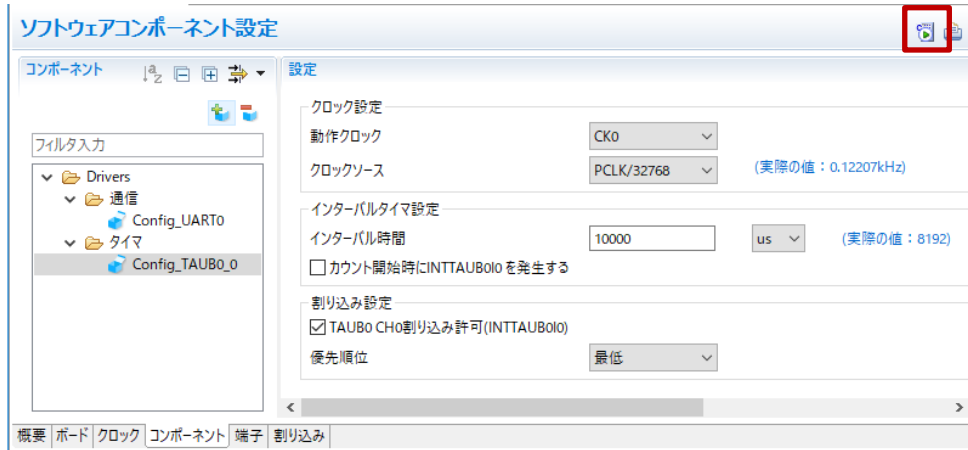


図 9-6 コード生成

9.3 ユーザコード領域にアプリケーションコードを追加

生成されたソースファイルの一部のファイルにはユーザコードの記述が可能なユーザコード領域があります。各統合開発環境のエディタでファイルを開き、割り込み処理の記述など必要に応じてユーザコード領域にアプリケーションコードを追加してください。

(例)CS+環境、インターバル・タイマ コンポーネント用に生成されたファイル

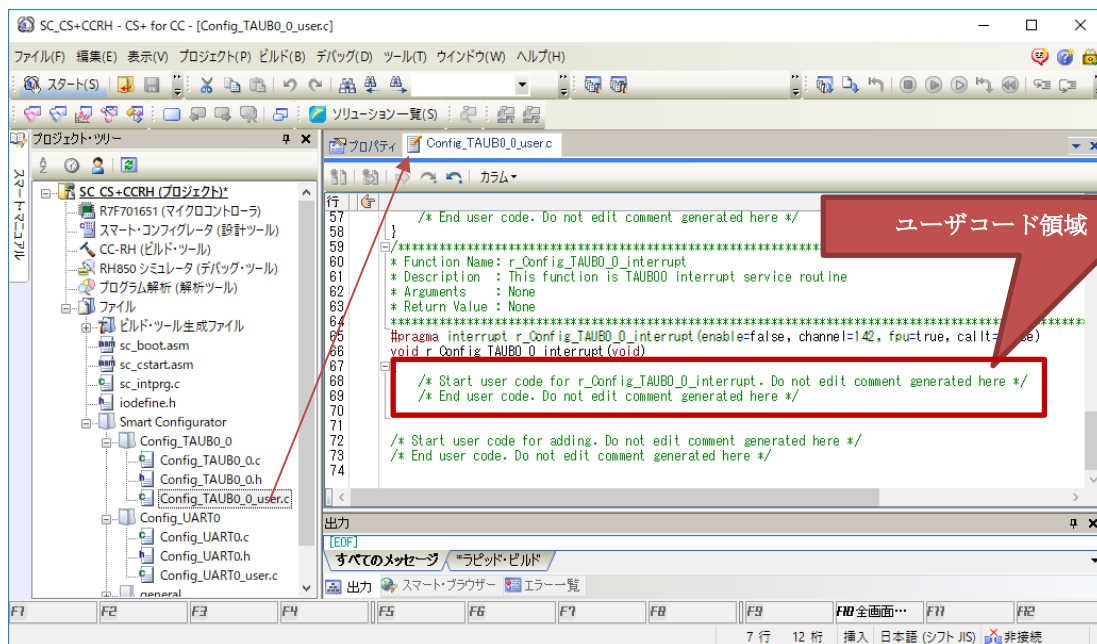


図 9-7 アプリケーションコードの追加

9.4 main()にアプリケーションコードを追加

<ProjectDir>%src%smc_gen %general%r_cg_main.c に main()関数があります。各統合開発環境のエディタでファイルを開き、ユーザコード領域にアプリケーションコードを追加してください。

(例)CS+環境

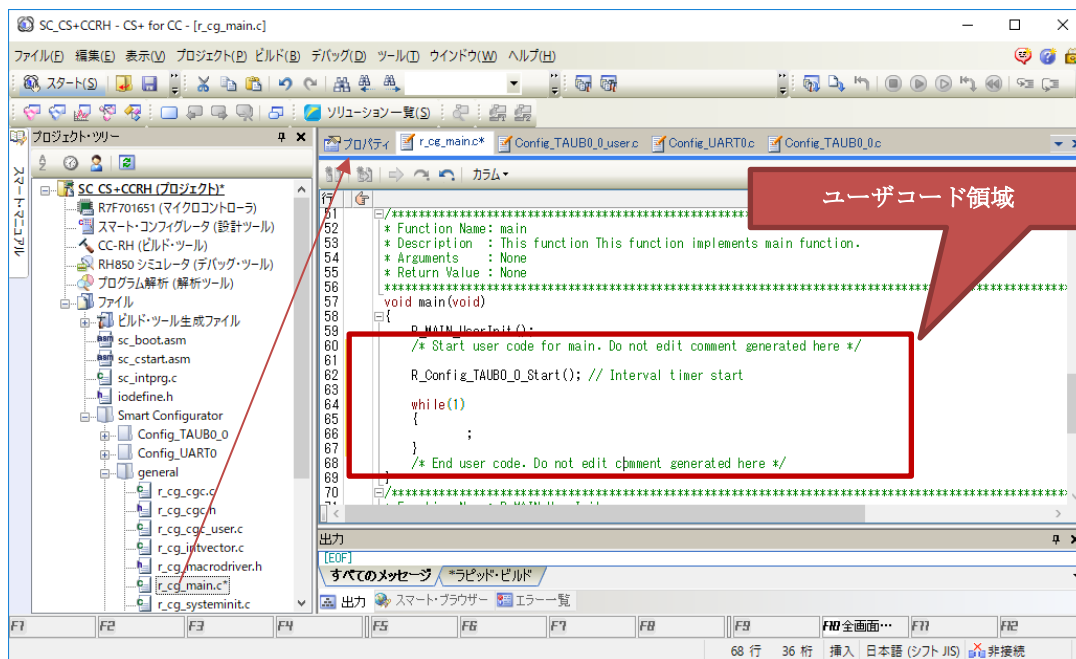


図 9-8 main()関数の編集

改訂記録

Rev.	章番号	改訂内容
1.00	—	新規発行
1.10	—	全項 誤記修正
1.20	5.	図 5-2 に(3)を追加
		(8)-2 に備考を追加
		図 5-8 の(3)(4)(5)を変更
		5.3 の(3)(4)(5)の内容を変更
	7.	図 7-2 に(9)(10)を追加
		7.2 に(9)(10)を追加
		図 7-4 の(3)(4)(5)を変更
		7.3 の(3)(4)(5)の内容を変更
		7.4 の内容を変更
	1.30	要旨
1.		表 1-1 を更新
2.		表 2-1 に RH850/F1KM-S2 を追加
		2.2 の内容を更新
4.		4.3 に RH850/F1KM-S2 を追加
5.		5.3 に RH850/F1KM-S2 を追加
6.		6.3 に RH850/F1KM-S2 を追加
7.		7.3 に RH850/F1KM-S2 を追加
		図 7-5 に RH850/F1KM-S2 を追加
8.		8.3 に RH850/F1KM-S2 を追加
		図 8-1 の内容を更新
	表 8-1 の内容を更新	
	図 8-5 に RH850/F1KM-S2 を追加	

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、 $V_{IL}(\text{Max.})$ 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くと、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）、交通制御（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、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。