RL78 ファミリ

Renesas Flash Driver RL78 Type01 SC 対応版 (Extra Area)

要旨

本アプリケーションノートでは RL78/G2x グループにて、スマート・コンフィグレータ(SC)を使用し Renesas Flash Driver RL78 Type01（以下、RFD）のエクストラ領域用ドライバを組み込む方法、およびサンプルプログラムを使用してエクストラ領域を書き換える方法について説明します。

また、SC 対応仕様の RFD と、従来の RFD とを比較する内容を含むため、便宜的に「単体版」、「SC 版」との記載があり、それぞれ以下の内容を示します。

単体版：従来のプロジェクトに直接 RFD のドライバを組み込むもの

SC 版：スマート・コンフィグレータを使用しプロジェクトに RFD のドライバを組み込むもの

動作確認デバイス

RL78/G22 グループ

RL78/G23 グループ

RL78/G24 グループ

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. 仕様 3
1.1 動作環境 3
1.2 サンプルプログラムフォルダ構成 4
1.3 RFD ドライバの各ファイル構成 5
1.3.1 RFD 共通ドライバ（r_rfd_rl78_common）のファイル構成 5
1.3.2 RFD エクストラ領域ドライバ（r_rfd_rl78_extraarea）のファイル構成 6
1.4 サンプルプログラムでのエクストラ領域の書き換え処理内容 7

2. エクストラ領域書き換えサンプルプロジェクトの作成 8
2.1 サンプルプロジェクト作成例 8
2.1.1 CS+の場合 8
2.1.2 e2 studio(CC-RL)の場合 8
2.1.3 IAR EW for Renesas RL78 の場合 8
2.1.4 e2 studio(LLVM)の場合 9
2.2 ソースコードの登録例 10
2.2.1 CS+の場合 10
2.2.1.1 r_rfd_rl78_codeflash と r_rfd_rl78_extraarea を同時に使用する場合の注意事項 13
2.2.2 e2 studio(CC-RL)の場合 14
2.2.2.1 r_rfd_rl78_codeflash と r_rfd_rl78_extraarea を同時に使用する場合の注意事項 17
2.2.3 IAR EW for Renesas RL78 の場合 18
2.2.3.1 r_rfd_rl78_codeflash と r_rfd_rl78_extraarea を同時に使用する場合の注意事項 23
2.2.4 e2 studio(LLVM)の場合 24
2.2.4.1 r_rfd_rl78_codeflash と r_rfd_rl78_extraarea を同時に使用する場合の注意事項 27
2.3 サンプルプログラムをプロジェクトに登録 28
2.4 サンプルプログラムの動作確認 42
2.4.1 CS+の場合 42
2.4.2 e2 studio(CC-RL)の場合 43
2.4.3 IAR EW for Renesas RL78 の場合 44
2.4.4 e2 studio(LLVM)の場合 45

3. 注意事項 46

4. 関連文書 47

5. 改訂記録 48

1. 仕様

RFD に含まれるサンプルプログラムでは、フラッシュ・シールド・ウインドウ(FSW)を制御する4byte(32bit)の領域を書き換えます。

1.1 動作環境

- C コンパイラ

表 1-1 対象の C コンパイラ・パッケージ

パッケージ	メーカー	バージョン
CC-RL(CS+,e^2 studio 用)	Renesas Electronics	V1.10 以降
IAR Embedded Workbench	IAR システムズ	V4.21 以降
LLVM(e^2 studio)	オープンソースソフトウェア	V10.0.0.202312 以降

※ IAR Systems、IAR Embedded Workbench、IAR および IAR システムズのロゴタイプは、IAR Systems AB が所有権を有する商標または登録商標です。

- エミュレータ

動作確認したエミュレータを表 1-2 に示します。

表 1-2 動作確認したエミュレータ

エミュレータ	メーカー
E2 エミュレータ Lite	Renesas Electronics

- ターゲット MCU

RL78/G22

RL78/G23

RL78/G24

1.2　サンプルプログラムフォルダ構成

サンプルプログラムの構成を図 1.1 に示します。

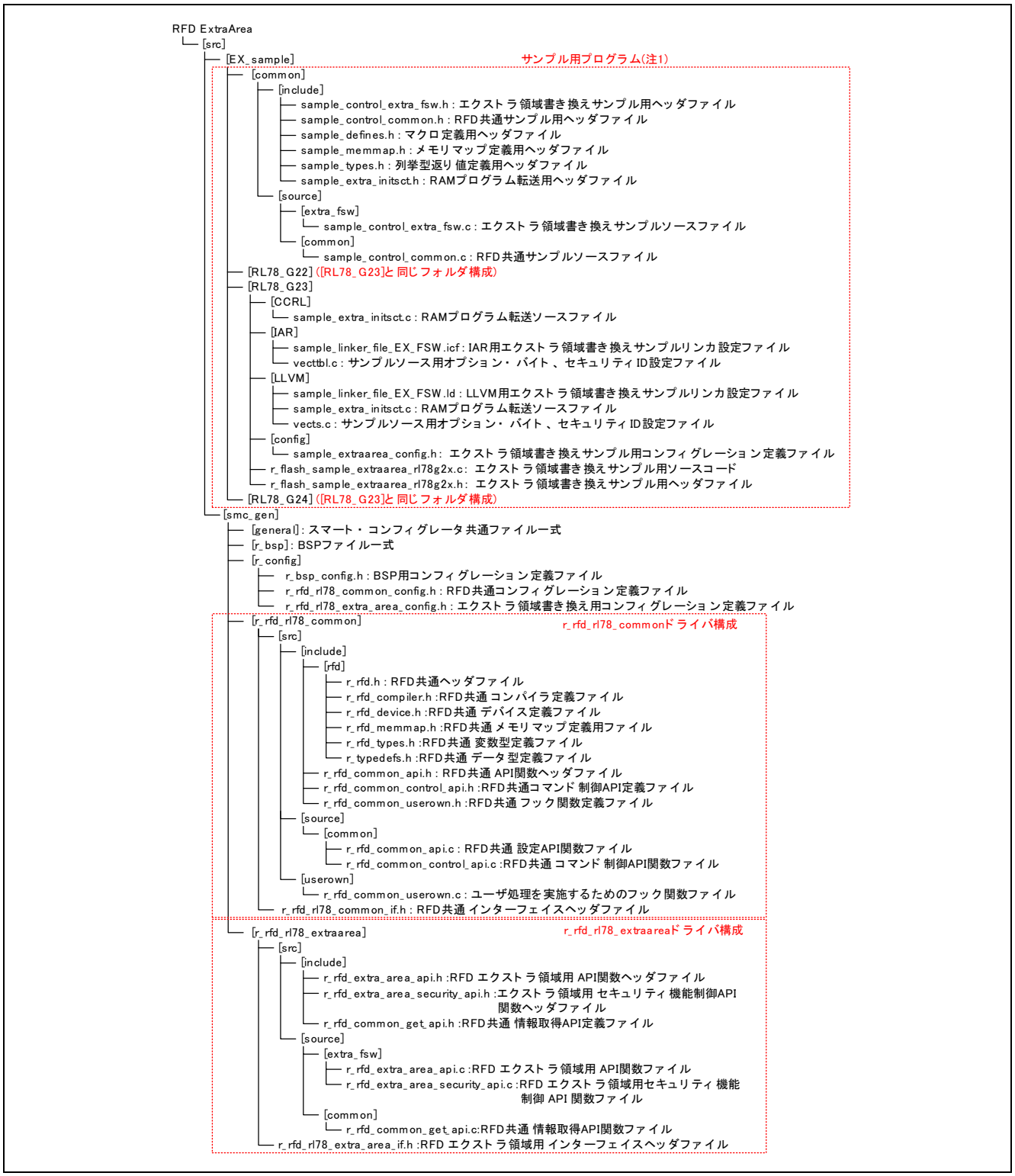


図 1.1　サンプルプログラム構成

注1.　zip ファイル形式のサンプルプログラムをスマート・コンフィグレータでダウンロードします。"demo"フォルダに出力された圧縮ファイル(EX_sample.zip)を解凍し、[EX_sample]フォルダを[src]フォルダの下に移動します。プロジェクト登録の詳細は「2.3 サンプルプログラムをプロジェクトに登録」を参照してください。

1.3　RFD ドライバの各ファイル構成

1.3.1　RFD 共通ドライバ（r_rfd_rl78_common）のファイル構成

RFD 共通ドライバの SC 対応版と単体版の差分を示す。

なお、RFD 共通ドライバの詳細仕様については、「RL78 ファミリ Renesas Flash Driver RL78 Type01 ユーザーズマニュアル（R20UT4830）」を参照ください。

表 1-3　RFD SC 対応版と単体版のファイル内容相違点
（共通 API：r_rfd_rl78_common\src\source\common フォルダ）

ファイル名	単体版	SC 版
r_rfd_common_api.c	変更なし	
r_rfd_common_control_api.c	変更なし	

表 1-4　RFD SC 対応版と単体版のファイル内容相違点
（共通 API：r_rfd_rl78_common\src\userown フォルダ）

ファイル名	単体版	SC 版
r_rfd_common_userown.c	変更なし	

表 1-5　RFD SC 対応版と単体版のファイル内容相違点
（共通ヘッダ：r_rfd_rl78_common\src\include フォルダ）

ファイル名	単体版	SC 版
r_rfd_common_api.h	変更なし	
r_rfd_common_control_api.h	変更なし	
r_rfd_common_userown.h	変更なし	

表 1-6　RFD SC 対応版と単体版のファイル内容相違点
（共通ヘッダ： r_rfd_rl78_common\src\include\rfd フォルダ）

ファイル名	単体版	SC 版
r_rfd.h	変更なし	
r_rfd_compiler.h	変更なし	
r_rfd_device.h	変更なし	
r_rfd_memmap.h	変更なし	
r_rfd_types.h	変更なし	
r_typedefs.h	変更なし	

表 1-7　RFD SC 対応版と単体版のファイル内容相違点
（共通インタフェースヘッダ：r_rfd_rl78_common\）

ファイル名	単体版	SC 版
r_rfd_rl78_common_if.h	ー	新規作成 共通 API 用ヘッダファイルをインクルードする

1.3.2　RFD エクストラ領域ドライバ（r_rfd_rl78_extraarea）のファイル構成

RFD エクストラ領域ドライバの SC 対応版と単体版の差分を示す。

なお、RFD エクストラ領域ドライバの詳細仕様については、「RL78 ファミリ Renesas Flash Driver RL78 Type01 ユーザーズマニュアル（R20UT4830）」を参照ください。

表 1-8　RFD SC 対応版と単体版のファイル内容相違点
（エクストラ領域 API：r_rfd_rl78_extraarea\src\source\extra_fsw フォルダ）

ファイル名	単体版	SC 版
r_rfd_extra_area_api.c	変更なし	
r_rfd_extra_area_security_api.c	変更なし	

表 1-9　RFD SC 対応版と単体版のファイル内容相違点
（エクストラ領域 API：r_rfd_rl78_extraarea\src\source\common フォルダ）

ファイル名	単体版	SC 版
r_rfd_common_get_api.c	変更なし	

表 1-10　RFD SC 対応版と単体版のファイル内容相違点
（エクストラ領域 API ヘッダ：r_rfd_rl78_extraarea \src\include フォルダ）

ファイル名	単体版	SC 版
r_rfd_extra_area_api.h	変更なし	
r_rfd_common_get_api.h	変更なし	
r_rfd_extra_area_security_api.h	変更なし	

表 1-11　RFD SC 対応版と単体版のファイル内容相違点
（エクストラ領域 インタフェースヘッダ：r_rfd_rl78_extraarea\）

ファイル名	単体版	SC 版
r_rfd_rl78_extra_area_if.h	―	新規作成 エクストラ領域 API 用ヘッダファイルをインクルードする

1.4　サンプルプログラムでのエクストラ領域の書き換え処理内容

図 1.2 にサンプルプログラムのフローチャートを示します。

sample_extraarea_main 関数を実行することで、RAM 上で実行するプログラムを ROM 領域からコピーし、RAM 上からエクストラ領域を書き換える処理を行います。

Sample_ExtraFSWControl 関数の処理内容については処理内容に変更がないため Renesas Flash Driver RL78 Type01 ユーザーズマニュアル(R20UT4830) 「Sample_ExtraFSWControl 関数」を参照ください。

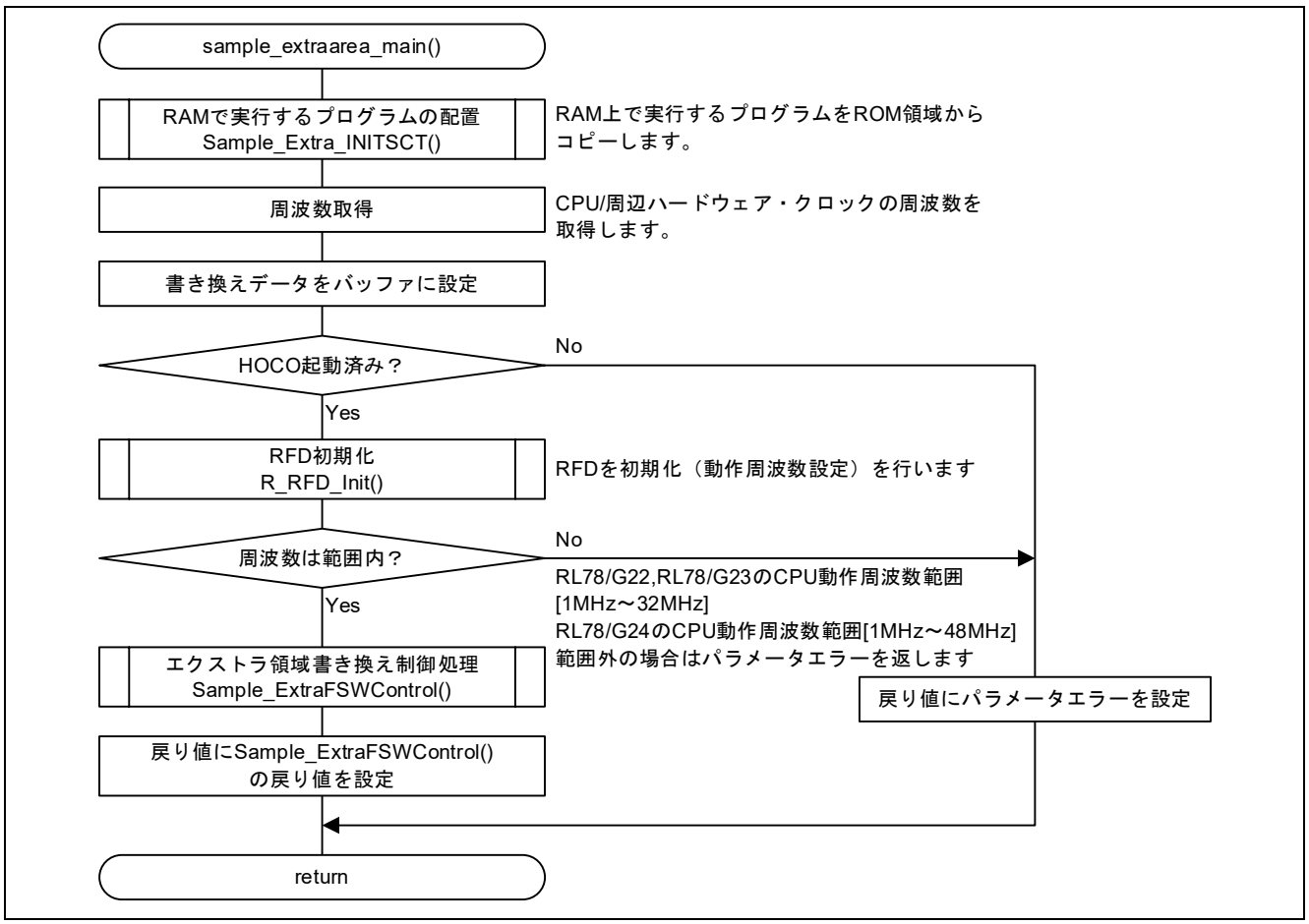


図 1.2　サンプルプログラム動作

注1.　エクストラ領域書き換え時のコード・フラッシュ・プログラミング・モード中は、コード・フラッシュ上のデータを参照できないため、Sample_ExtraFSWControl 関数、およびその内部で参照するデータは、事前に RAM へコピーして、RAM 上で参照する必要があります。

注2.　CPU/周辺ハードウェア・クロックの"周波数取得"は、ボード・サポート・パッケージの関数を使用しています。

2.　エクストラ領域書き換えサンプルプロジェクトの作成

2.1　サンプルプロジェクト作成例

2.1.1　CS+の場合

プロジェクトの作成は Renesas Flash Driver RL78 Type01 ユーザーズマニュアル(R20UT4830JJ) 「サンプル・プロジェクト作成例」を参照ください。

2.1.2　e^2 studio(CC-RL)の場合

プロジェクトの作成は Renesas Flash Driver RL78 Type01 ユーザーズマニュアル(R20UT4830) 「サンプル・プロジェクト作成例」を参照ください。

今回はスマート・コンフィグレータを使用するため、ターゲット・デバイス、デバッグ・ツールを選択後、"終了"ボタンを押さず"次へ"ボタンを押し、以下の手順を実行してください。

"Use Smart Configurator"にチェックを入れ、"終了"ボタンを押します。

2.1.3　IAR EW for Renesas RL78 の場合

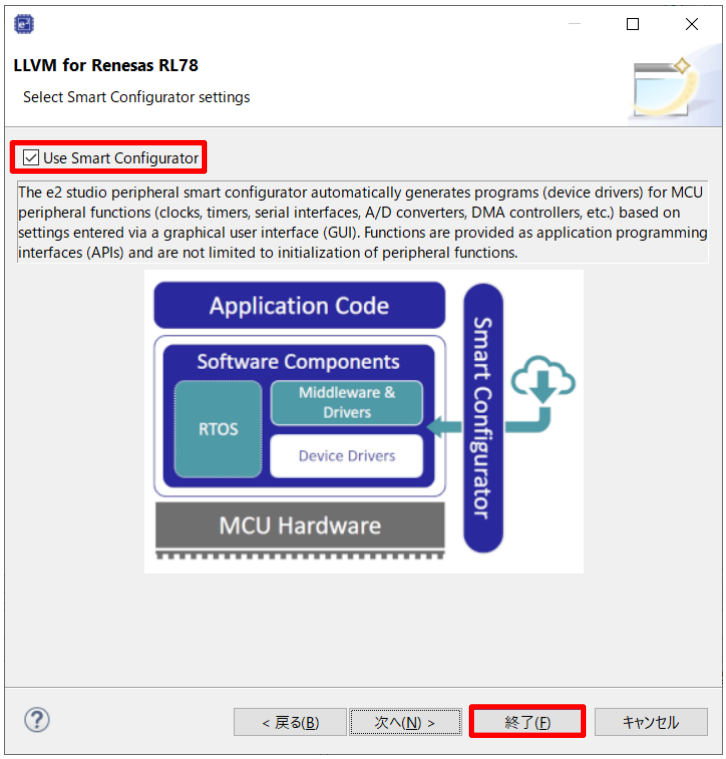
プロジェクトの作成は Renesas Flash Driver RL78 Type01 ユーザーズマニュアル(R20UT4830) 「サンプル・プロジェクト作成例」を参照ください。

2.1.4　e² studio(LLVM)の場合

プロジェクトの作成は Renesas Flash Driver RL78 Type01 ユーザーズマニュアル(R20UT4830) 「サンプル・プロジェクト作成例」を参照ください。

今回はスマート・コンフィグレータを使用するため、ターゲット・デバイス、デバッグ・ツールを選択後、"終了"ボタンを押さず"次へ"ボタンを押し、以下の手順を実行してください。

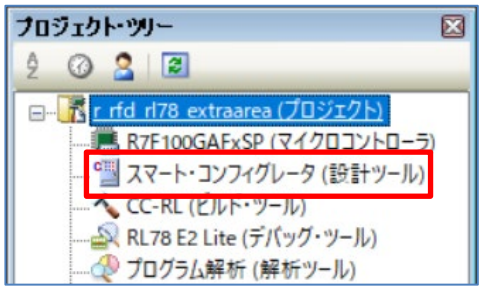
"Use Smart Configurator"にチェックを入れ、"終了"ボタンを押します。



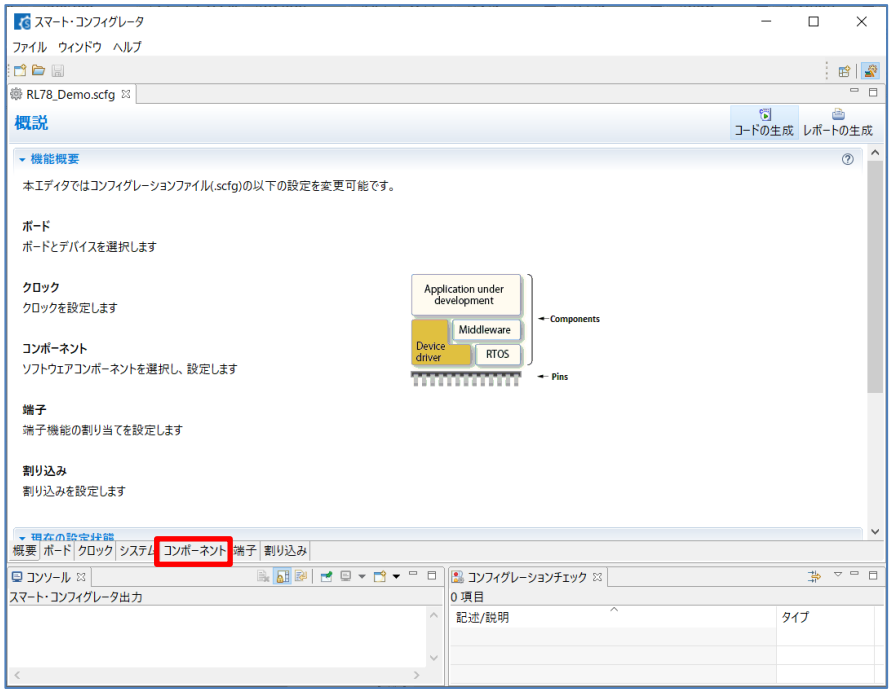
2.2　ソースコードの登録例

2.2.1　CS+の場合

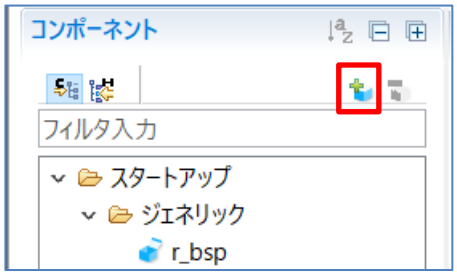
① プロジェクトツリーの"スマート・コンフィグレータ(設計ツール)"をダブルクリックし、スマート・コンフィグレータを起動します。



② コンポーネントタブを選択します。



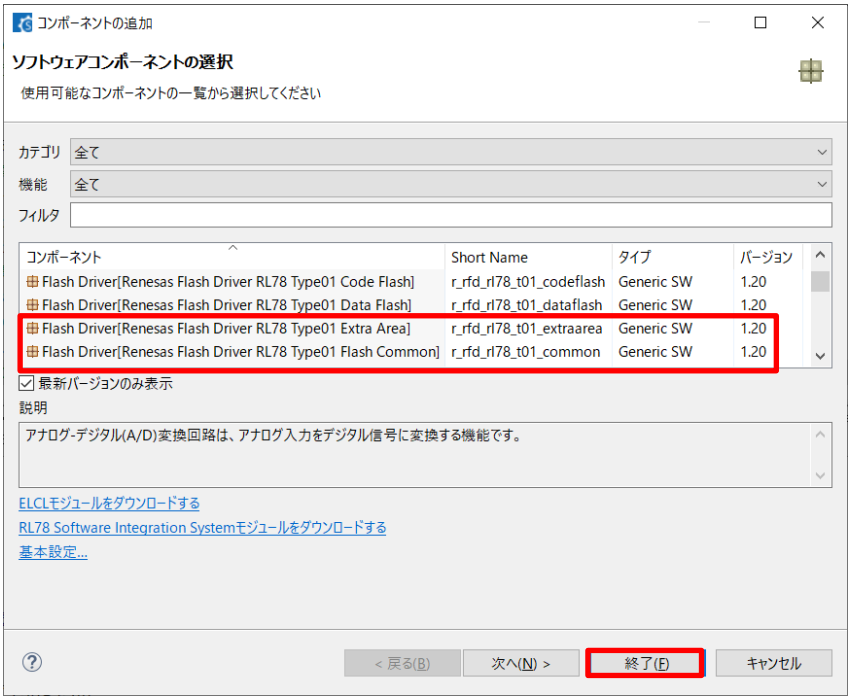
③ コンポーネントの追加ボタンを押し、"コンポーネントの追加"ダイアログを開きます。



④ 以下のコンポーネントを選択し、"終了"を押します。

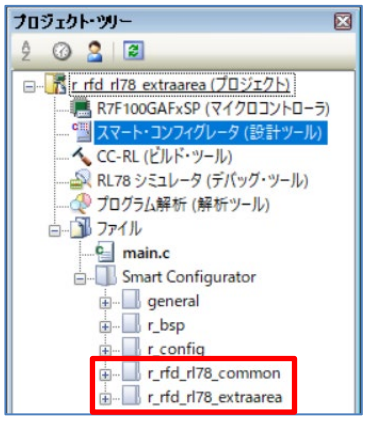
・Flash Driver[Renesas Flash Driver RL78 Type01 Extra Area](r_rfd_rl78_t01_extraarea)

・Flash Driver[Renesas Flash Driver RL78 Type01 Flash Common](r_rfd_rl78_t01_common)



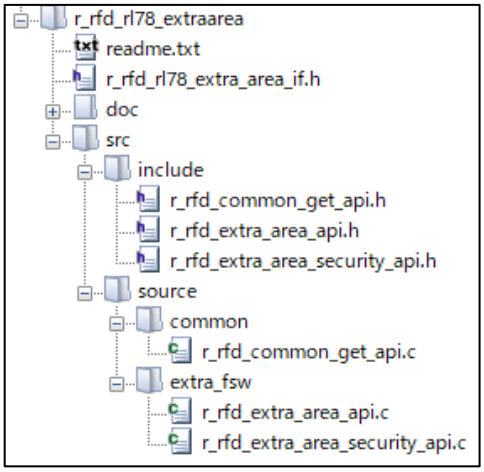
⑤ コード生成ボタンを押し、コードを生成完了後スマート・コンフィグレータを閉じます。

⑥ プロジェクトツリーに"r_rfd_rl78_common"フォルダ、"r_rfd_rl78_extraarea"フォルダが追加されます。

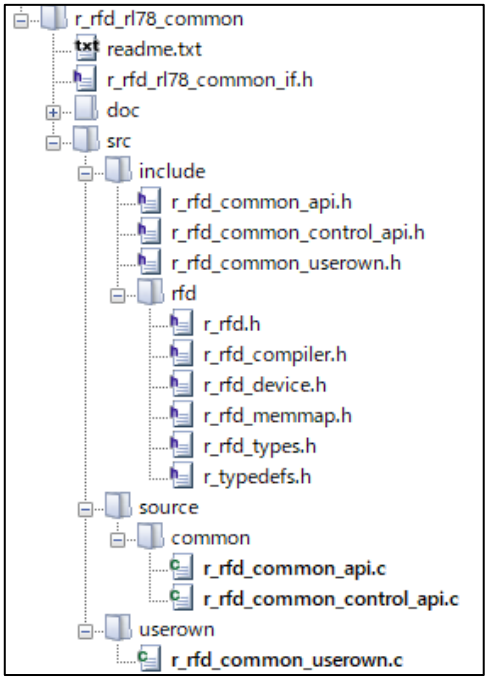


各フォルダはそれぞれ以下のように展開されます。

展開時の r_config フォルダ



展開時の r_rfd_rl78_extraarea フォルダ

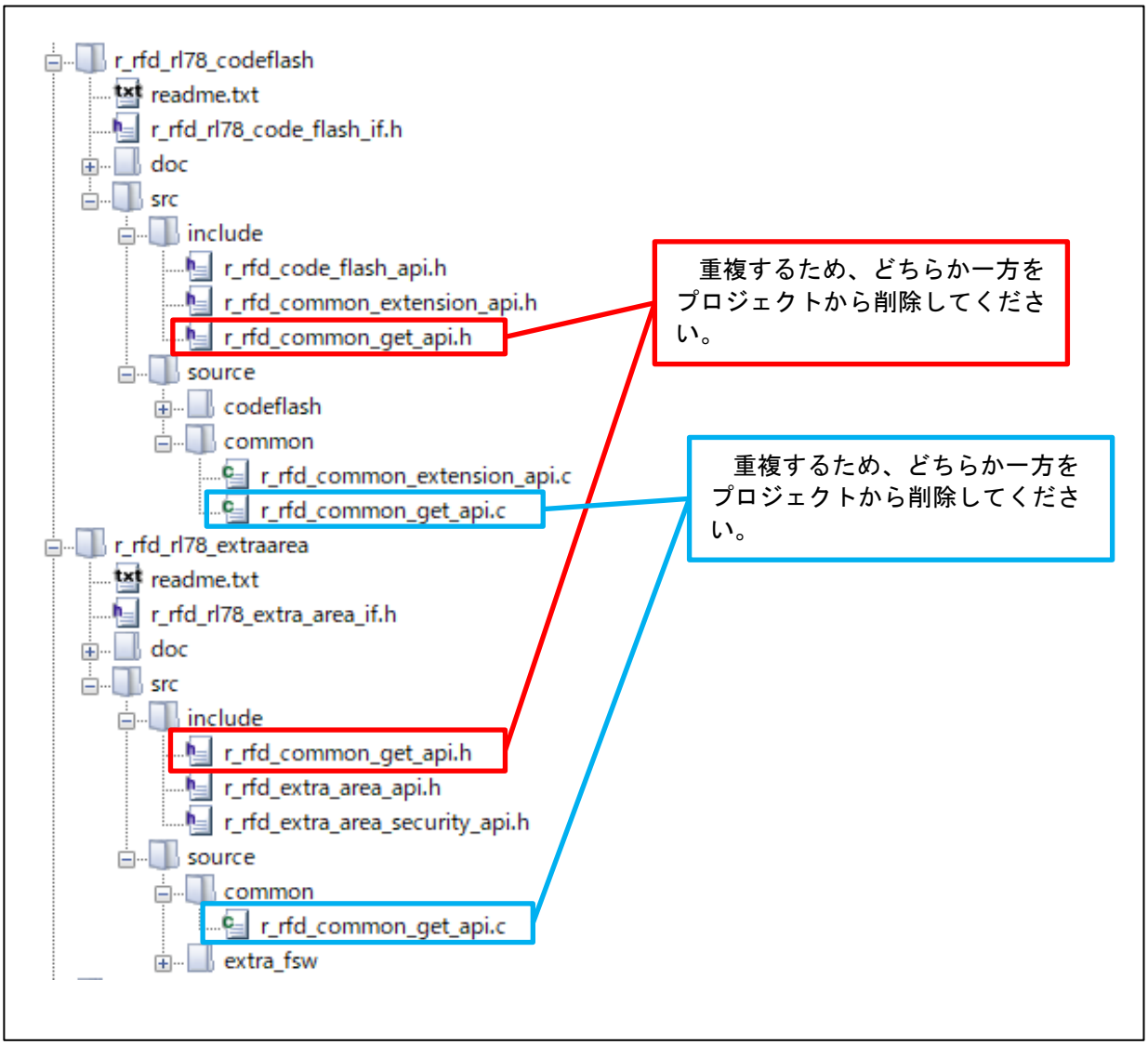


展開時の r_rfd_rl78_common フォルダ

2.2.1.1　r_rfd_rl78_codeflash と r_rfd_rl78_extraarea を同時に使用する場合の注意事項

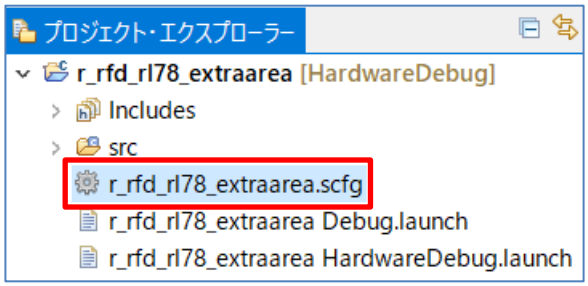
r_rfd_rl78_codeflash と r_rfd_rl78_extraarea を同時に使用する場合、下図のように"r_rfd_common_get_api.c"、"r_rfd_common_get_api.h"が重複します。

各ファイルとも、どちらか一方をプロジェクトから削除してください。

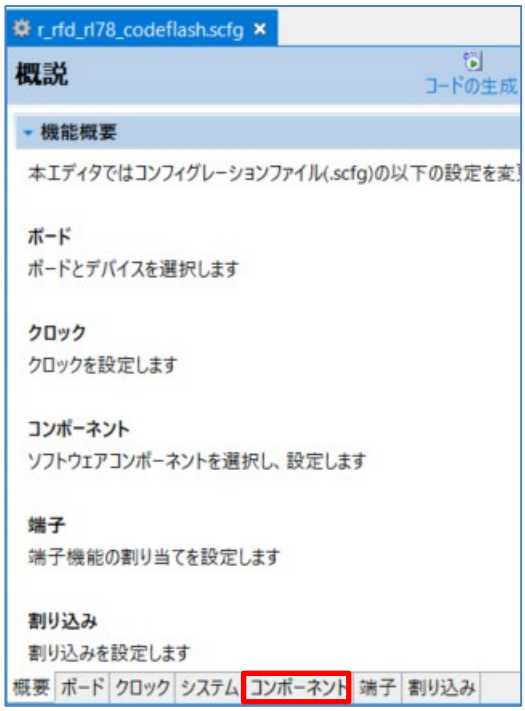


2.2.2　e² studio(CC-RL)の場合

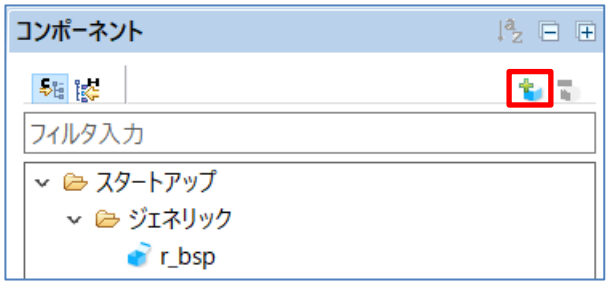
① e² studio を起動後スマート・コンフィグレータのプロジェクトファイルを開きます。



② "コンポーネント"タブを選択します。



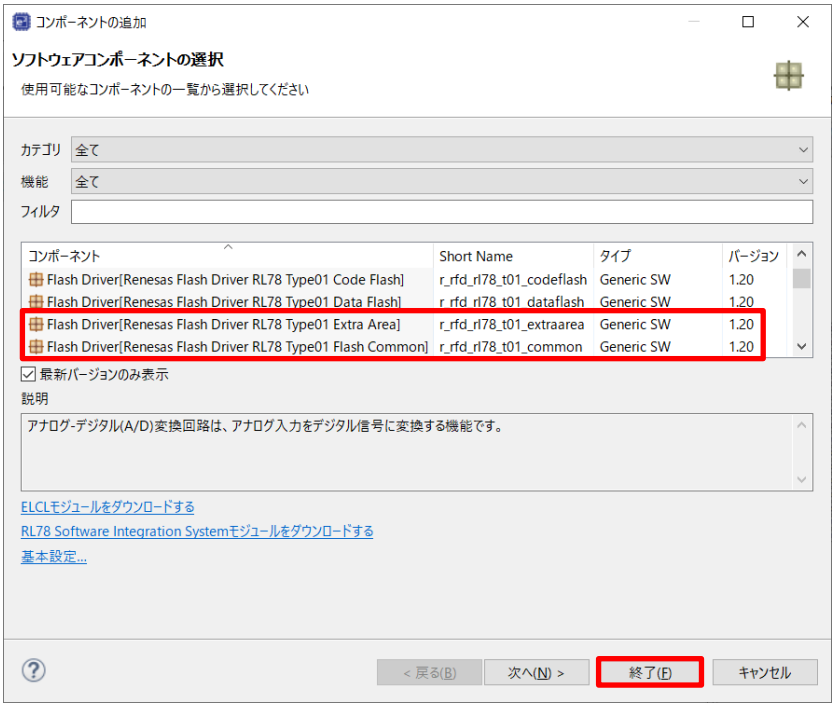
③ コンポーネントの追加ボタンを押し、"コンポーネントの追加"ダイアログを開きます



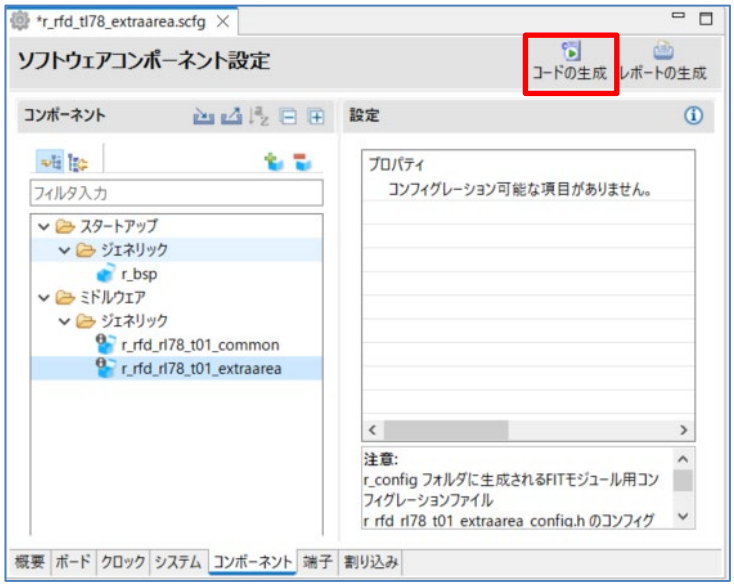
④ 以下のコンポーネントを選択し、"終了"を押します

・Flash Driver[Renesas Flash Driver RL78 Type01 Extra Area](r_rfd_rl78_t01_extraarea)

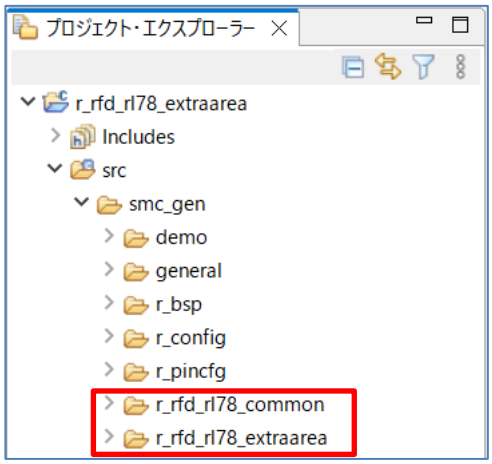
・Flash Driver[Renesas Flash Driver RL78 Type01 Flash Common](r_rfd_rl78_t01_common)



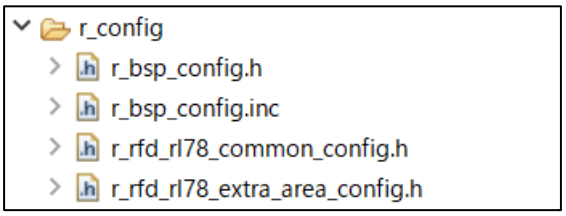
⑤ コード生成ボタンを押し、コードを生成します。



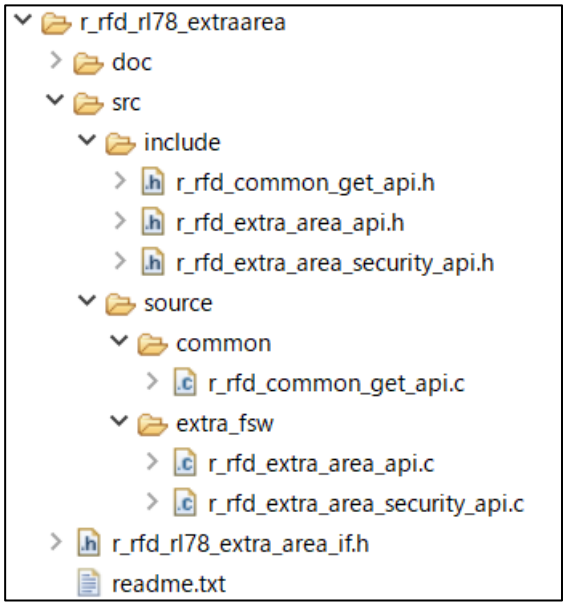
⑥ プロジェクトツリーに"r_rfd_rl78_common"フォルダ、"r_rfd_rl78_extrarea"フォルダが追加されます。



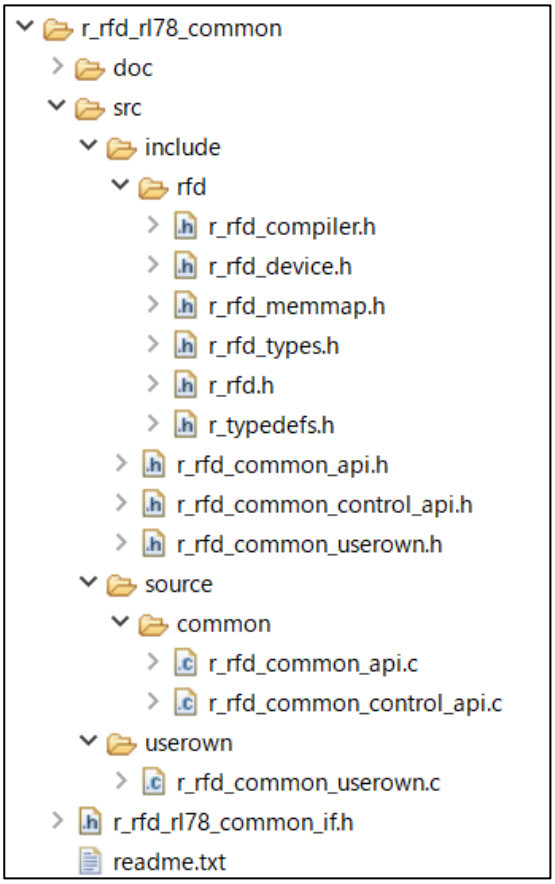
各フォルダはそれぞれ以下のように展開されます。



展開時の r_config フォルダ



展開時の r_rfd_rl78_extraarea フォルダ

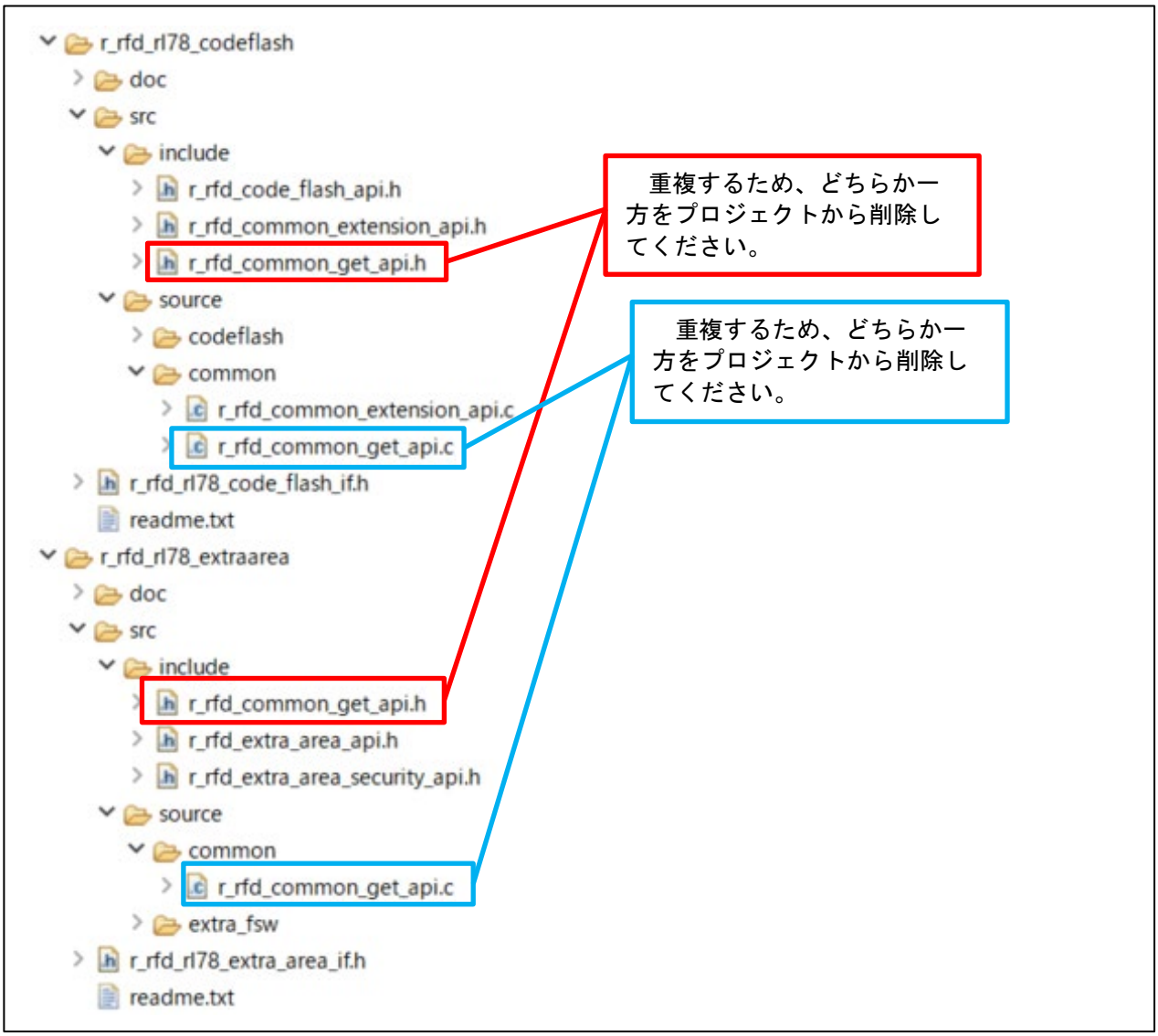


展開時の r_rfd_rl78_common フォルダ

2.2.2.1 r_rfd_rl78_codeflash と r_rfd_rl78_extraarea を同時に使用する場合の注意事項

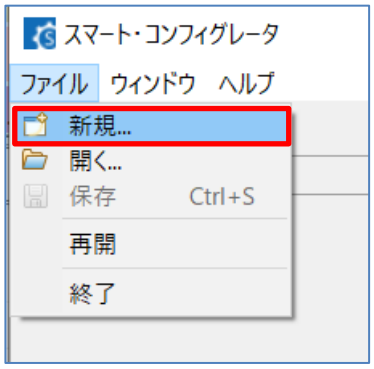
r_rfd_rl78_codeflash と r_rfd_rl78_extraarea を同時に使用する場合、下図のように "r_rfd_common_get_api.c"、"r_rfd_common_get_api.h"が重複します。

各ファイルとも、どちらか一方をプロジェクトから削除してください。



2.2.3　IAR EW for Renesas RL78 の場合

① Smart Configurator for RL78 を起動し、"ファイル" [新規]を選択します。



② プラットフォームとツールチェーンを選択します。

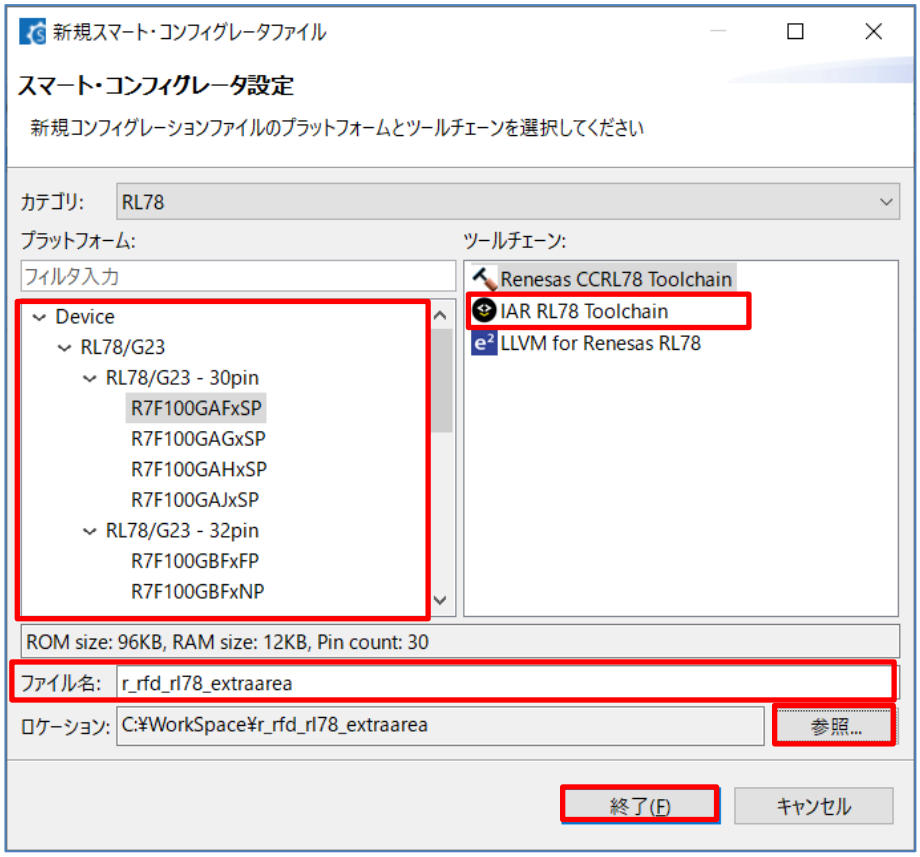
プラットフォームは IAR EW for Renesas RL78 のプロジェクトで選択したものと同じものを選択します。

ツールチェーンは"IAR RL78 Toolchain"を選択します。

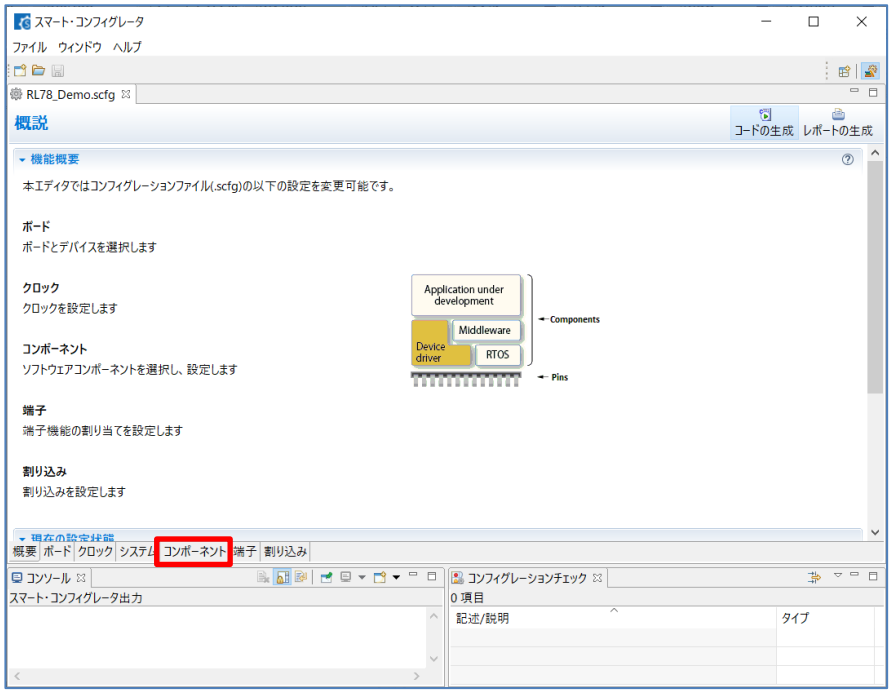
ファイル名は任意の名前を付けます。

"参照"ボタンを押し、IAR EW for Renesas RL78 のプロジェクトフォルダをロケーションに指定し、"終了"を押します。

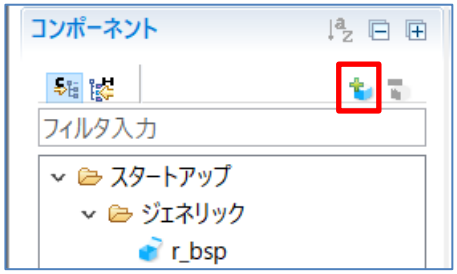
設定したロケーションに「.setting」フォルダと、「<ファイル名>.scfg」ファイルが作成されます。



③ コンポーネントタブを選択します。



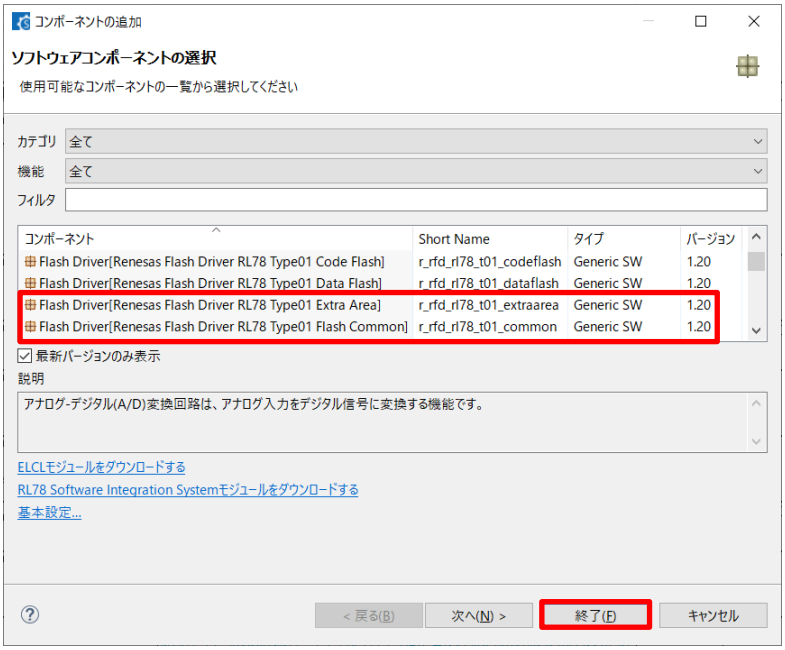
④ コンポーネントの追加ボタンを押し、"コンポーネントの追加"ダイアログを開きます。



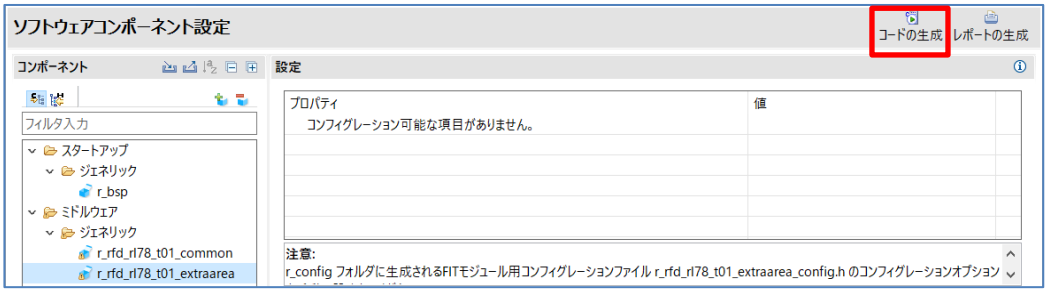
⑤ 以下のコンポーネントを選択し、"終了"を押します。

・Flash Driver[Renesas Flash Driver RL78 Type01 Extra Area](r_rfd_rl78_t01_extraarea)

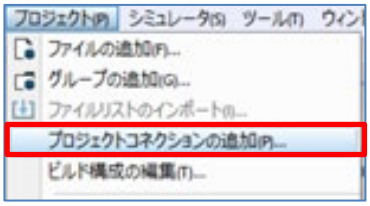
・Flash Driver[Renesas Flash Driver RL78 Type01 Flash Common](r_rfd_rl78_t01_common)



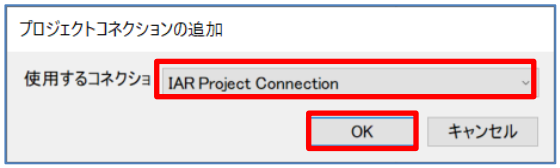
⑥ コード生成ボタンを押し、コードを生成完了後スマート・コンフィグレータを閉じます。



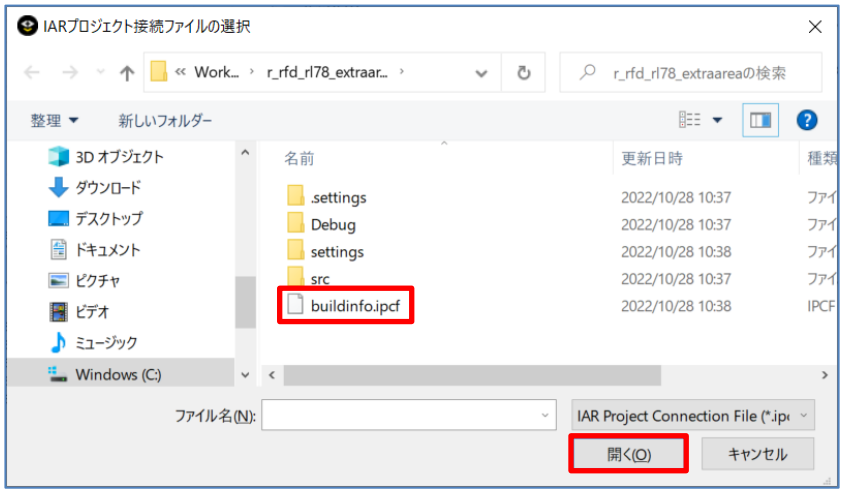
⑦ IAR EW for Renesas RL78 を起動後"プロジェクト" [プロジェクトコネクションの追加]を選択し、プロジェクトコネクションの追加ダイアログを開きます。



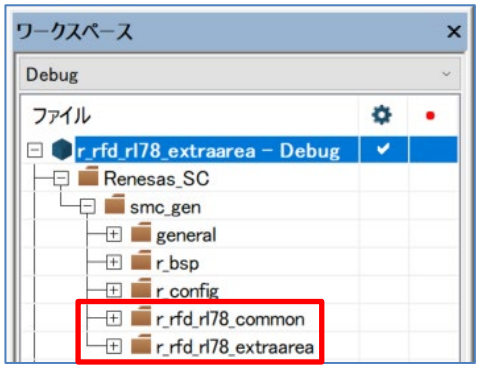
⑧ "IAR Project Connection"を選択し、"OK"を押します。



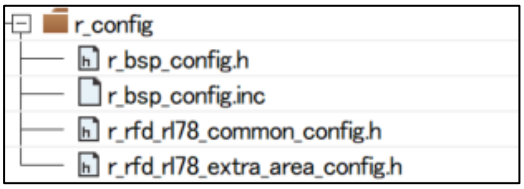
⑨ スマート・コンフィグレータで作成した icpf ファイルを選択し、"開く"を押します。



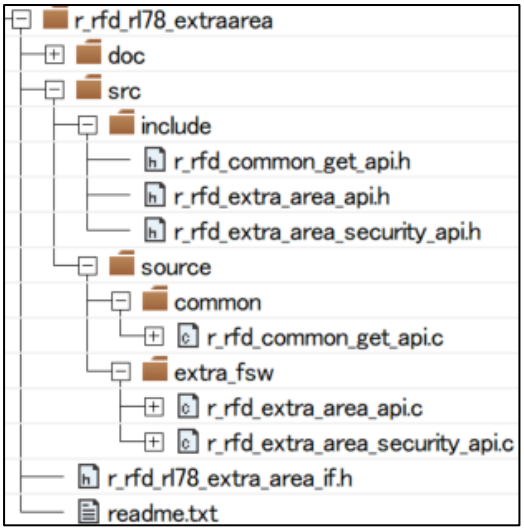
⑩ ワークスペースに"r_rfd_rl78_common"、" r_rfd_rl78_extraarea"が追加されます。



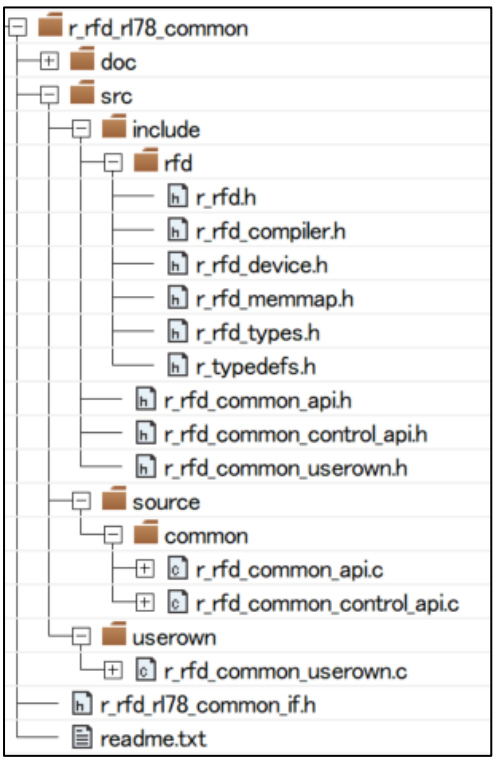
各フォルダはそれぞれ以下のように展開されます。



展開時の r_config フォルダ



展開時の r_rfd_rl78_extraarea フォルダ

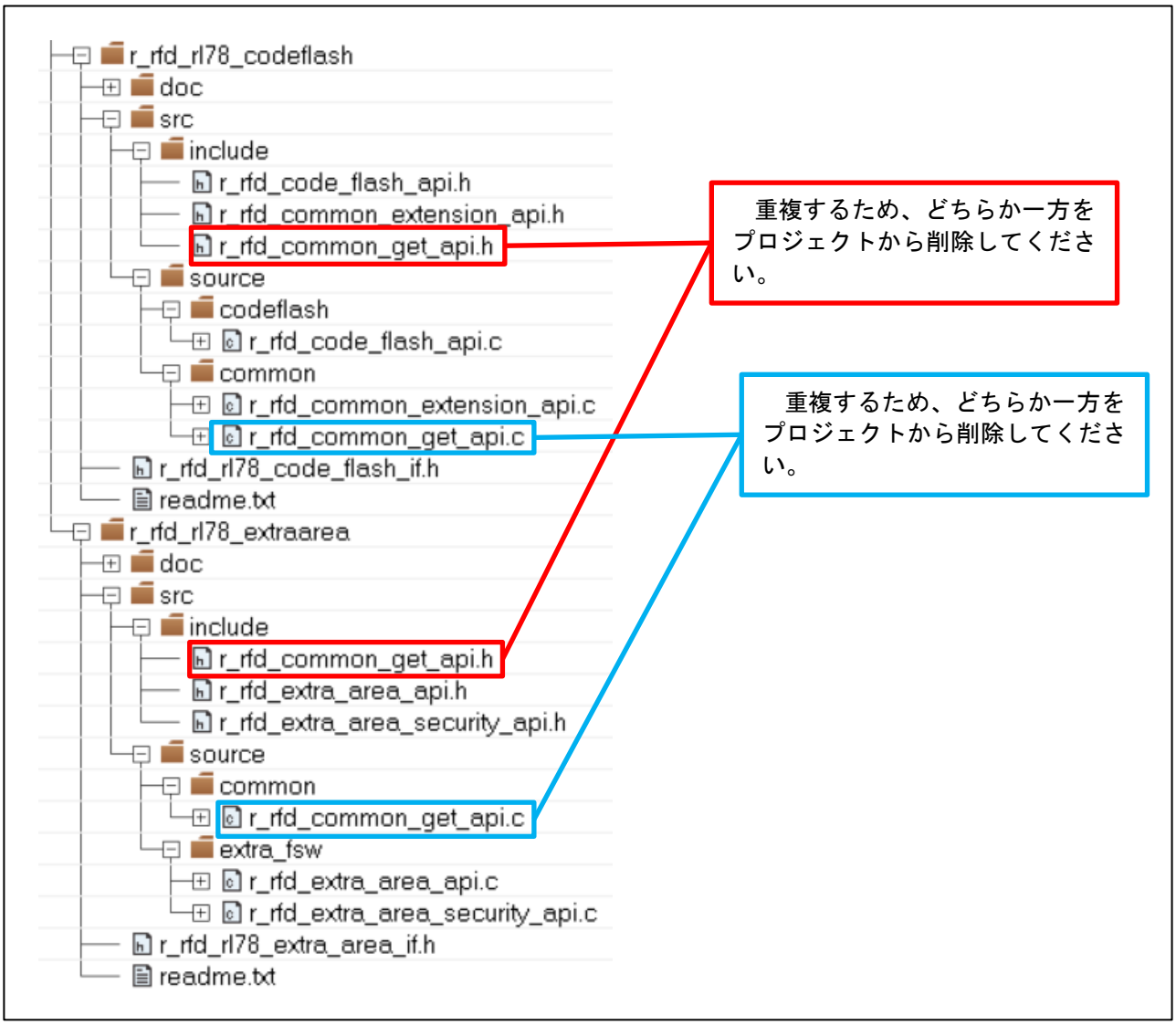


展開時の r_rfd_rl78_common フォルダ

2.2.3.1　r_rfd_rl78_codeflash と r_rfd_rl78_extraarea を同時に使用する場合の注意事項

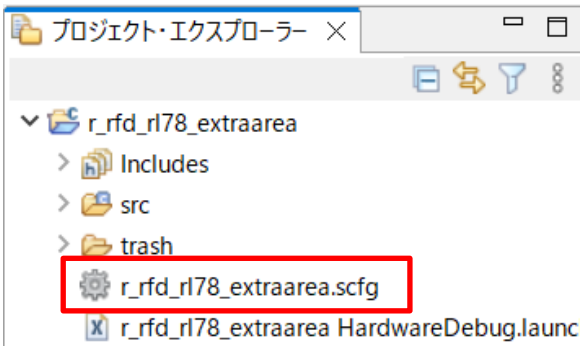
r_rfd_rl78_codeflash と r_rfd_rl78_extraarea を同時に使用する場合、下図のように"r_rfd_common_get_api.c"、"r_rfd_common_get_api.h"が重複します。

各ファイルとも、どちらか一方をプロジェクトから削除してください。



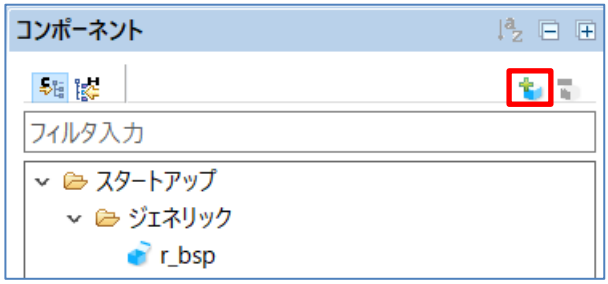
2.2.4　e² studio(LLVM)の場合

① e² studio を起動後スマート・コンフィグレータのプロジェクトファイルを開きます。



② "コンポーネント"タブを選択します。

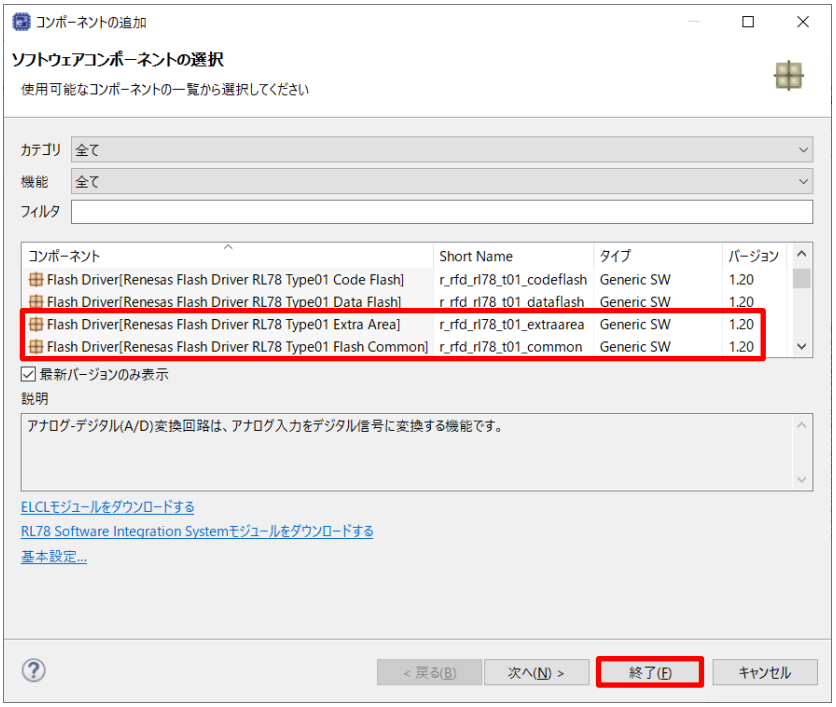
③ コンポーネントの追加ボタンを押し、"コンポーネントの追加"ダイアログを開きます



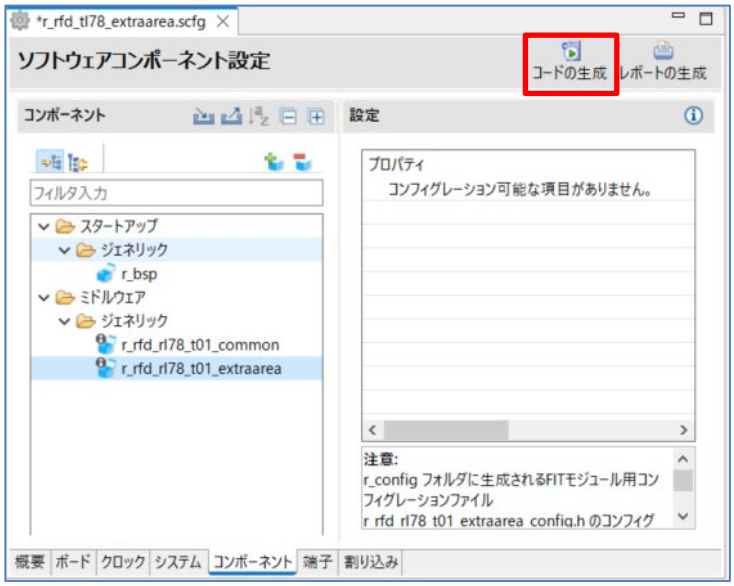
④ 以下のコンポーネントを選択し、"終了"を押します

・Flash Driver[Renesas Flash Driver RL78 Type01 Extra Area](r_rfd_rl78_t01_extraarea)

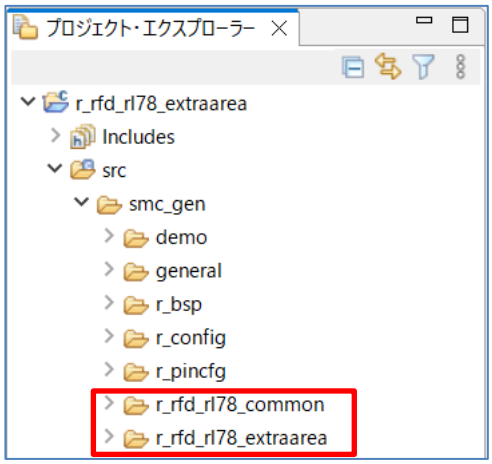
・Flash Driver[Renesas Flash Driver RL78 Type01 Flash Common](r_rfd_rl78_t01_common)



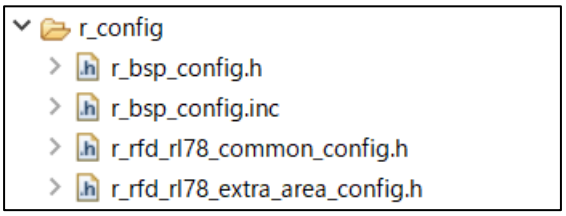
⑤ コード生成ボタンを押し、コードを生成します。



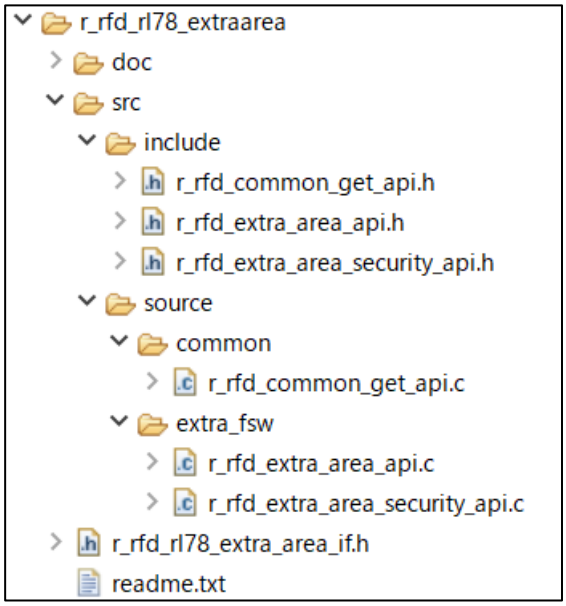
⑥ プロジェクトツリーに"r_rfd_rl78_common"フォルダ、"r_rfd_rl78_extraarea"フォルダが追加されます。



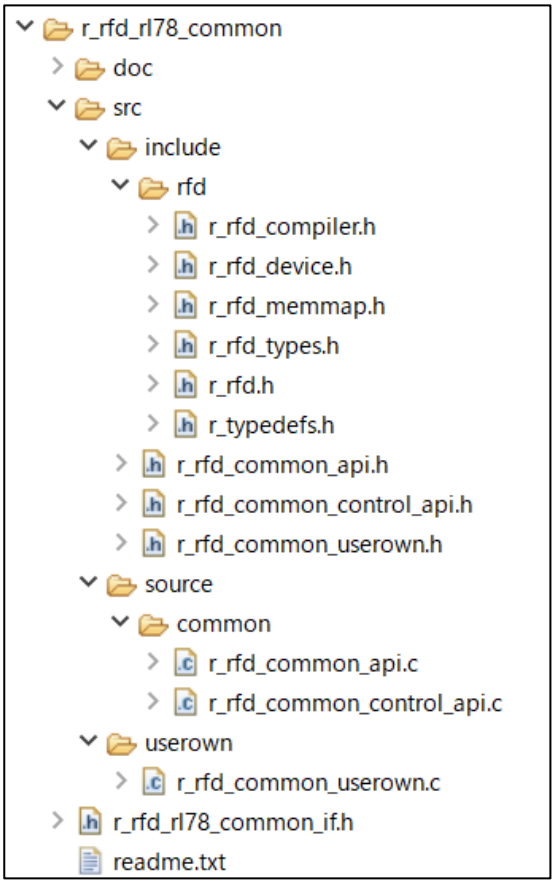
各フォルダはそれぞれ以下のように展開されます。



展開時の r_config フォルダ



展開時の r_rfd_rl78_extraarea フォルダ

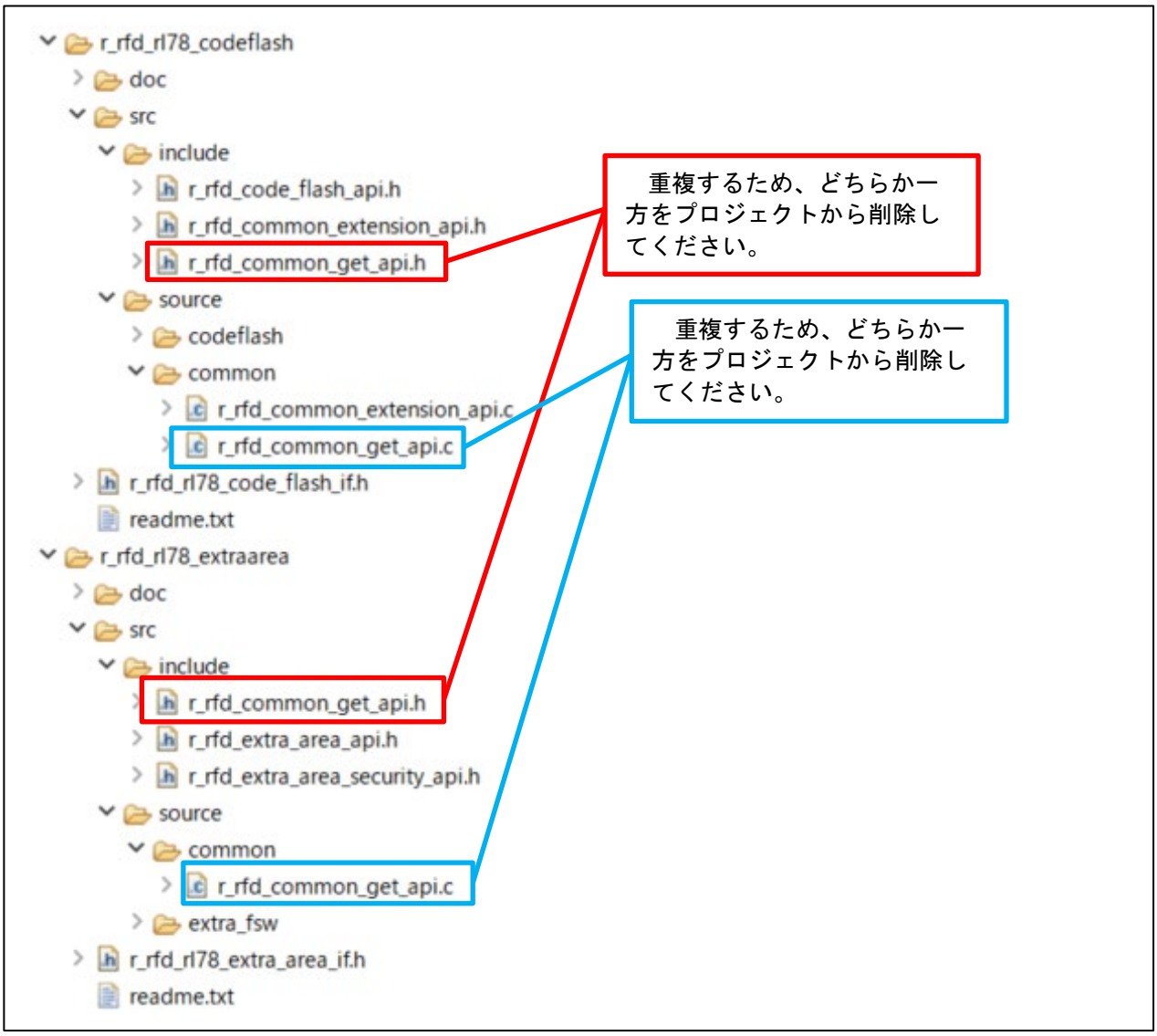


展開時の r_rfd_rl78_common フォルダ

2.2.4.1　r_rfd_rl78_codeflash と r_rfd_rl78_extraarea を同時に使用する場合の注意事項

r_rfd_rl78_codeflash と r_rfd_rl78_extraarea を同時に使用する場合、下図のように"r_rfd_common_get_api.c"、"r_rfd_common_get_api.h"が重複します。

各ファイルとも、どちらか一方をプロジェクトから削除してください。



2.3　サンプルプログラムをプロジェクトに登録

① "EX_sample.zip"を解凍します。

解凍後に生成されたデバイス名のフォルダは、対象デバイスのみを残し、その他は、削除してください。例えば、RL78/G23 を使用する場合、"RL78_G23"フォルダのみを残し、非対象の"RL78_G24"等は、フォルダごと削除します。

データ・フラッシュ、コード・フラッシュのサンプルプログラムと同時に使用する場合は、共通ファイルが重複することを防ぐため、解凍時に同一のフォルダ名として解凍してください。

② CS+、e^2 studio または IAR でサンプルプログラムのフォルダをプロジェクトに登録します。

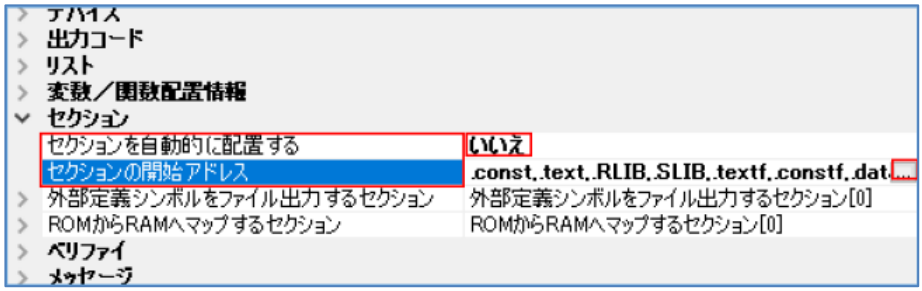
※サンプルプログラム内の、使用するコンパイラ・パッケージ以外のフォルダに含まれるファイルは登録不要です。

③ セクション設定
③-1：CS+の設定

CS+でのセクション項目の設定は、"リンク・オプション"タブで設定(各領域共通)

- [セクション]　項目を設定します。

[セクションを自動的に配置する]を一度"いいえ"に設定すると[セクションの開始アドレス]にセクションが表示されるようになり、表示された最も右の" ... "ボタンで、"セクション設定"画面を表示します。

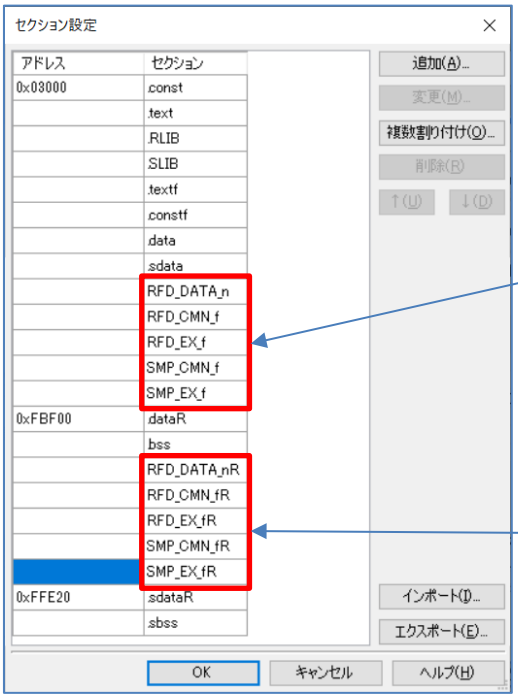


以降、RL78/G23 のサンプルのフォルダ名("RL78_G23")は、対象デバイスのフォルダ名に読み替えてください。RL78/G24 を使用する場合は、フォルダ名を"RL78_G24"に読み替えます。

エクストラ領域書き換えに必要なセクションを"セクション設定"画面で追加します。

プログラム領域へ追加：RFD_DATA_n, RFD_CMN_f, RFD_EX_f, SMP_CMN_f, SMP_EX_f

RAM へ追加：RFD_DATA_nR, RFD_CMN_fR, RFD_EX_fR, SMP_CMN_fR, SMP_EX_fR

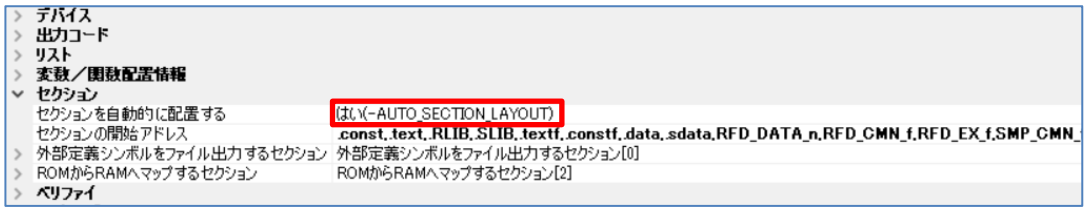


追加セクション
RFD_DATA_n
RFD_CMN_f
RFD_EX_f
SMP_CMN_f
SMP_EX_f

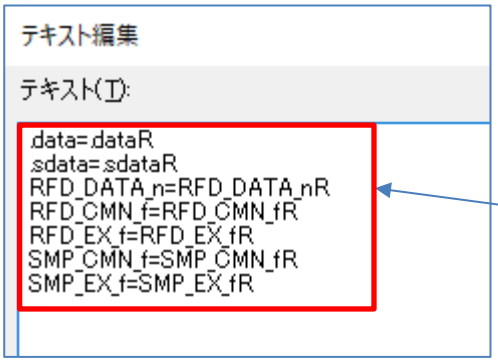
RFD_DATA_nR
RFD_CMN_fR
RFD_EX_fR
SMP_CMN_fR
SMP_EX_fR

注) 各デバイスを使用する場合のサンプルプロジェクトの設定については、「RL78　ファミリ Renesas Flash Driver RL78 Type01　ユーザーズマニュアル」(R20UT4830) Rev.1.20 以降の「RFD RL78 Type01 サンプル・プロジェクトの作成」章の「デバイス変更に伴う設定」項を参照してください。

"OK"ボタン押下後、[セクションを自動的に配置する]を"はい"に戻してください。



[ROM から RAM へマップするセクション]の最も右の" ... "ボタンで、"テキスト編集"画面を表示して、ROM から RAM へマップするセクションを追加します。

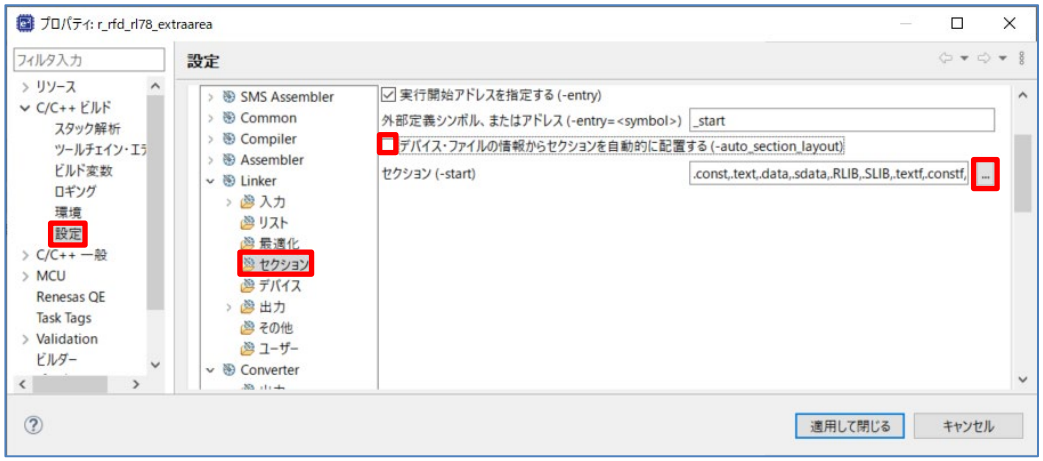


ROM から RAM へ マップするセクション
.data=.dataR
.sdata=.sdataR
RFD_DATA_n=RFD_DATA_nR
RFD_CMN_f=RFD_CMN_fR
RFD_EX_f=RFD_EX_fR
SMP_CMN_f=SMP_CMN_fR
SMP_EX_f=SMP_EX_fR

③-2：e² studio(CC-RL)の設定

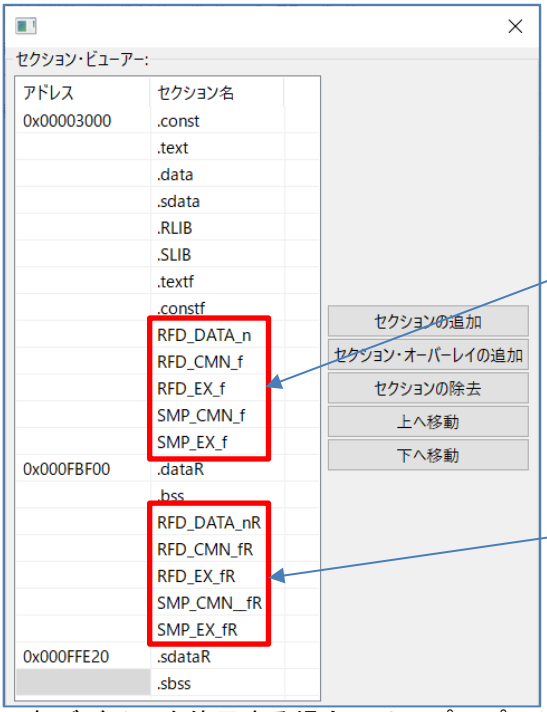
e² studio(CC-RL)でのセクション設定は"セクション・ビューアー"画面で行います。

"セクション・ビューアー"画面は[プロジェクト] - [プロパティ]でプロパティウインドウを開き、"C/C++ビルド" [設定] – "Linker" [セクション]で表示される[デバイス・ファイルの情報からセクションを自動的に配置する(-auto_section_layout)]のチェックを外し、[セクション(-start)]の最も右の" ... "ボタンで表示します。



プログラム領域へ追加：RFD_DATA_n, RFD_CMN_f, RFD_EX_f, SMP_CMN_f, SMP_EX_f

RAM へ追加：RFD_DATA_nR, RFD_CMN_fR, RFD_EX_fR, SMP_CMN_fR, SMP_EX_fR

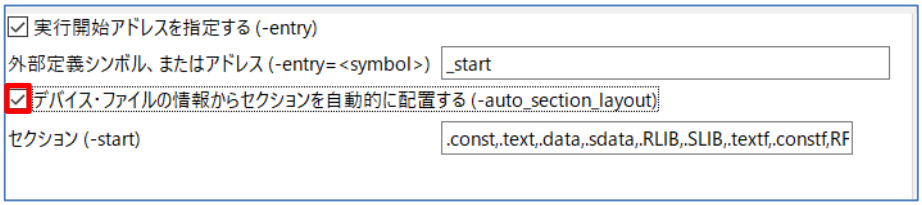


追加セクション
RFD_DATA_n
RFD_CMN_f
RFD_EX_f
SMP_CMN_f
SMP_EX_f

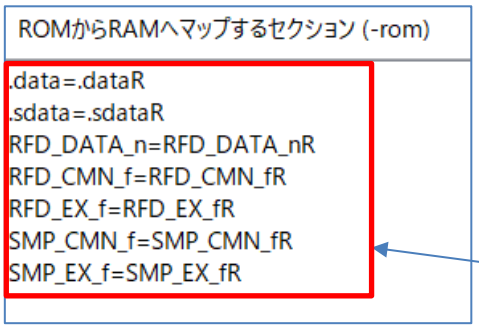
RFD_DATA_nR
RFD_CMN_fR
RFD_EX_fR
SMP_CMN_fR
SMP_EX_fR

注) 各デバイスを使用する場合のサンプルプロジェクトの設定については、「RL78 ファミリ Renesas Flash Driver RL78 Type01 ユーザーズマニュアル」(R20UT4830) Rev.1.20 以降の「RFD RL78 Type01 サンプル・プロジェクトの作成」章の「デバイス変更に伴う設定」項を参照してください。

"OK"ボタン押下後、[デバイス・ファイルの情報からセクションを自動的に配置する (-auto_section_layout)]をチェックして下さい。



"C/C++ビルド" [設定] － "Linker" [出力]　で表示した画面で[ROM から RAM へマップするセクション(-rom)]を設定します。



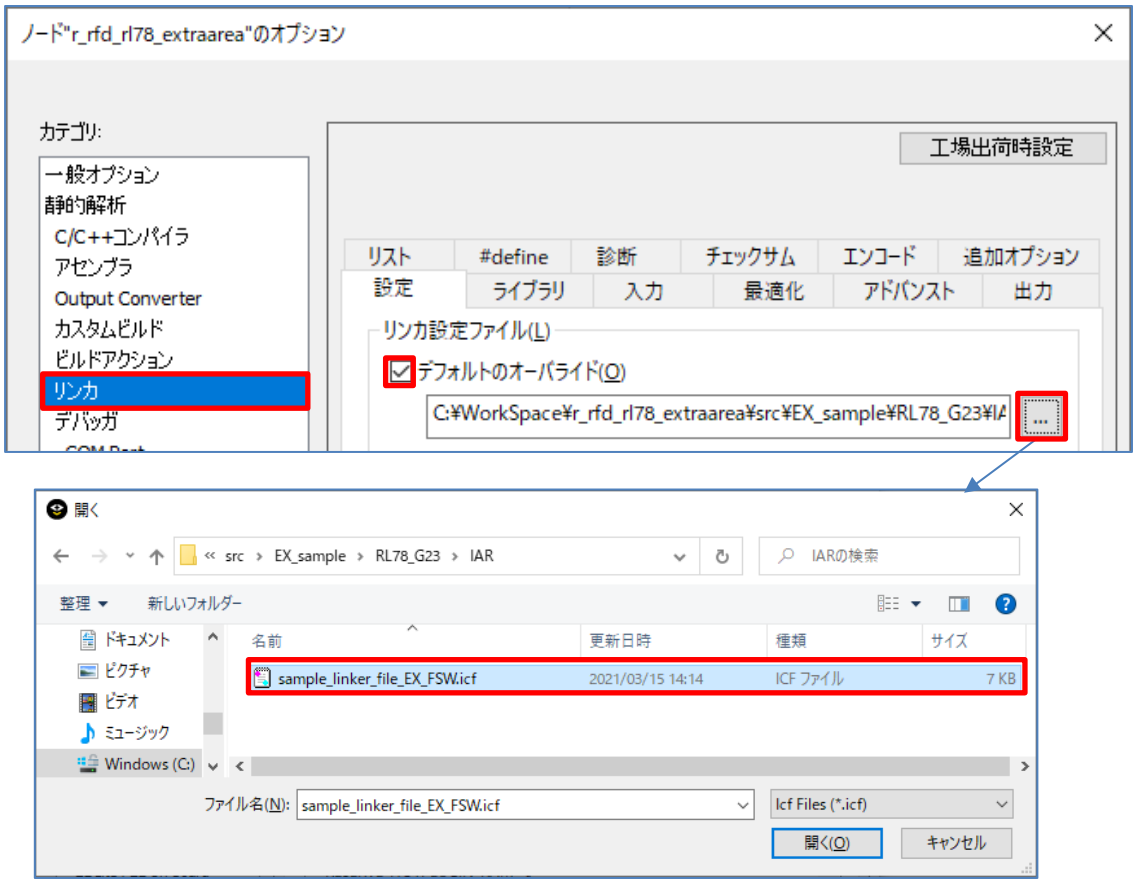
ROM から RAM へ マップするセクション
.data=.dataR
.sdata=.sdataR
RFD_DATA_n=RFD_DATA_nR
RFD_CMN_f=RFD_CMN_fR
RFD_EX_f=RFD_EX_fR
SMP_CMN_f=SMP_CMN_fR
SMP_EX_f=SMP_EX_fR

③-3：IAR EW for Renesas RL78 の設定

IAR Embedded Workbench では、ビルドで実行するリンク設定をリンカ設定ファイル(*.icf)に記述します。ツリーで[プロジェクト]のマウス右クリックで"オプション"を選択、表示された画面内の[リンカ]で、[設定] - [デフォルトのオーバライド(O)]にチェックを入れ、" ... "ボタンの"開く"画面でリンカ設定ファイル(*.icf)を選択します。ここでは、RFD RL78 Type01 用に準備されている" sample_linker_file_(領域名).icf"ファイルを選択します。書き換え領域ごとのリンカ設定用ファイル(*.icf)は以下の通りです。

- エクストラ領域書き換え用：sample_linker_file_EX_FSW.icf

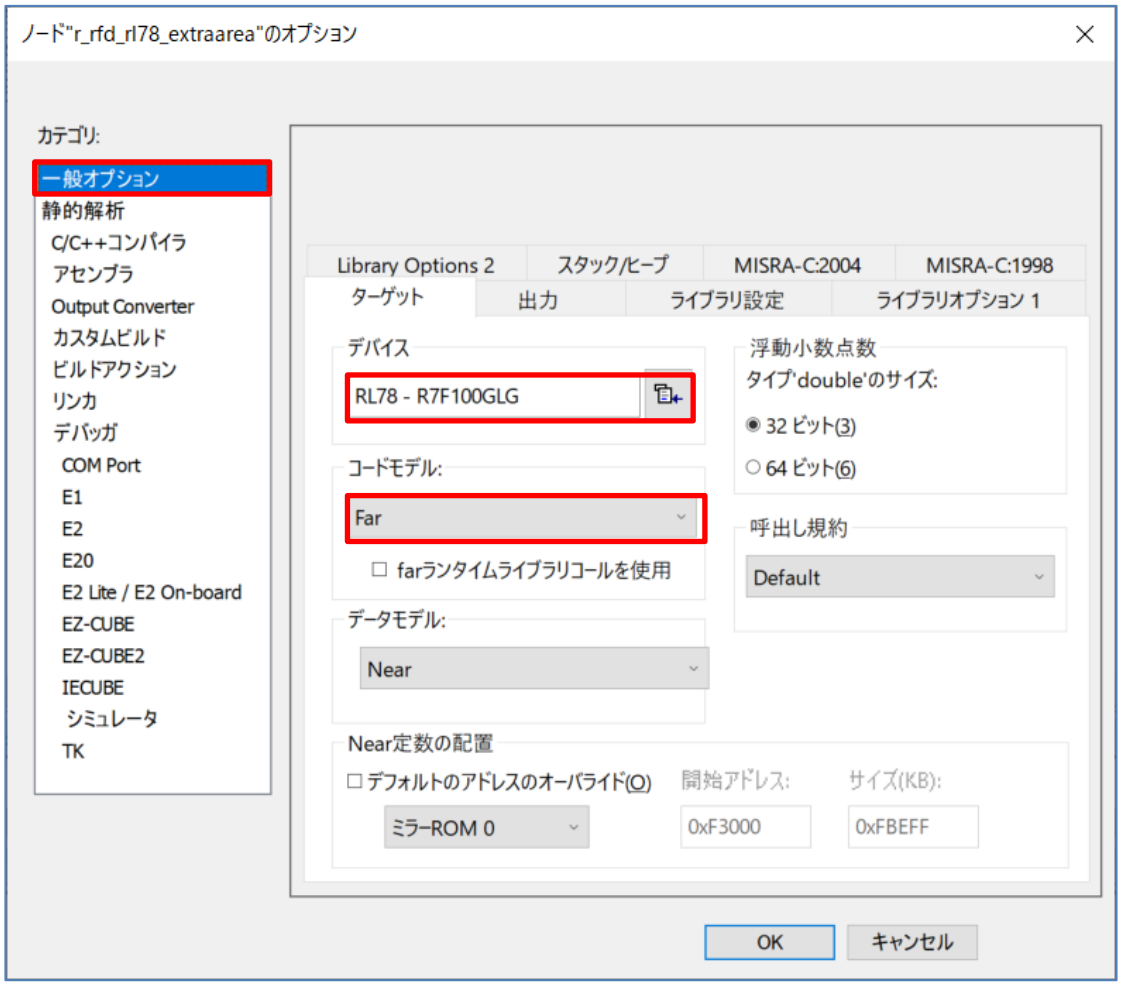
  RL78/G23：(\EX_sample\RL78_G23\IAR\)



※コード・フラッシュやデータ・フラッシュと同時に使用する場合、同時に使用する領域のサンプルプログラムに対応するよう icf ファイルを変更してください。

注) 各デバイスを使用する場合のサンプルプロジェクトの設定については、「RL78 ファミリ Renesas Flash Driver RL78 Type01 ユーザーズマニュアル」(R20UT4830) Rev.1.20 以降の「RFD RL78 Type01 サンプル・プロジェクトの作成」章の「デバイス変更に伴う設定」項を参照してください。

"オプション"の画面内の[一般オプション] - [ターゲット]タブの項目を設定します。[デバイス]で対象デバイスを、[コードモデル:]で"Far"を選択します。



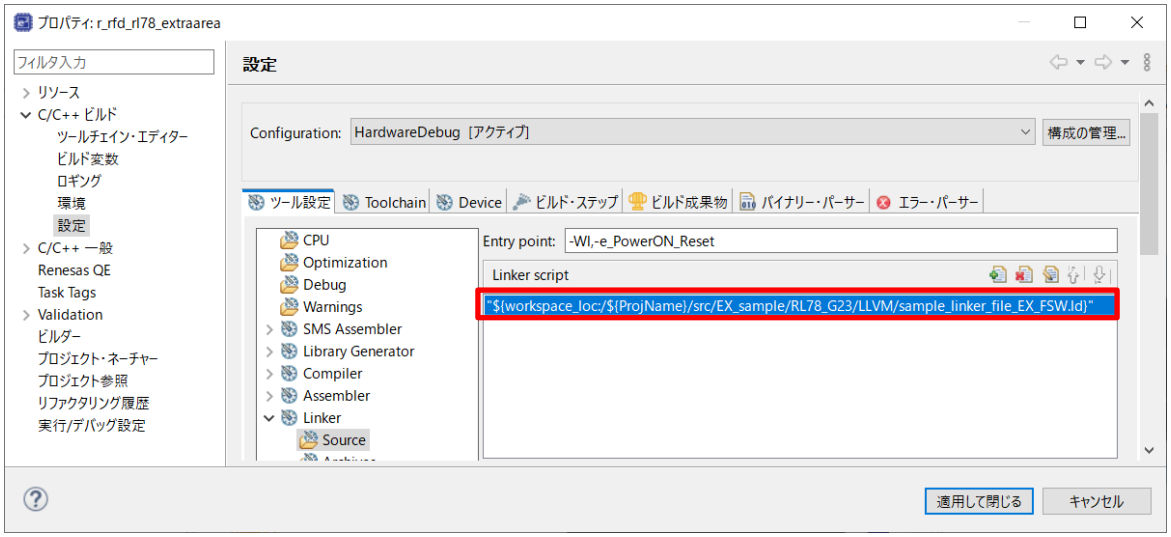
③-4：e² studio(LLVM)の設定

e² studio(LLVM) では、ビルドで実行するリンク設定をリンカ・スクリプト(*.ld)に記述します。

[プロジェクト]-[プロパティ]でプロパティウインドウを開き、"C/C++ビルド"[設定]-"Linker" [Source] で表示した画面でリンカ・スクリプトのパスを設定します。

追加するリンカ・スクリプトのパスを以下に示します。

${workspace_loc:/${ProjName}/src/EX_sample/RL78_G23/LLVM/sample_linker_file_EX_FSW.ld}



※src フォルダ直下に EX_sample.zip を解凍した場合のリンカ・スクリプトのパスとなります。

注) 各デバイスを使用する場合のサンプルプロジェクトの設定については、「RL78　ファミリ Renesas Flash Driver RL78 Type01　ユーザーズマニュアル」(R20UT4830) Rev.1.20 以降の「RFD RL78 Type01 サンプル・プロジェクトの作成」章の「デバイス変更に伴う設定」項を参照してください。

④ インクルード・パスの設定

図は RL78/G23 の場合を示しています。ここでも、RL78/G23 のサンプルのフォルダ名("RL78_G23")は、対象デバイスのフォルダ名に読み替えてください。RL78/G24 を使用する場合は、フォルダ名を"RL78_G24"に読み替えます。

④-1：CS+の設定

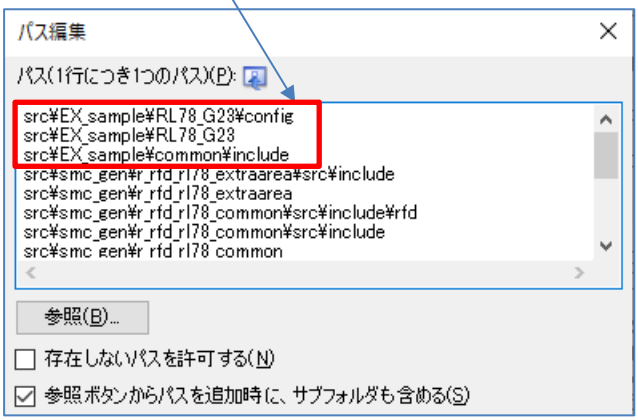
CS+でのインクルード・パスの設定は、"共通オプション"タブで設定します。

- [よく使うオプション(コンパイル)] － [追加のインクルード・パス]で"パス編集"ウインドウを表示して、インクルード・ファイルのパスを追加します。

"2.2.1 CS+の場合"の⑤でコード生成を実施した段階で、サンプルプログラム以外のインクルード・パスは登録されているため、サンプルプログラムのインクルード・パスを登録します。

追加するインクルード・パスを以下に示します。

src\EX_sample\RL78_G23\config
src\EX_sample\RL78_G23
src\EX_sample\common\include



※src フォルダ直下に EX_sample.zip を解凍した場合のインクルード・パスとなります。

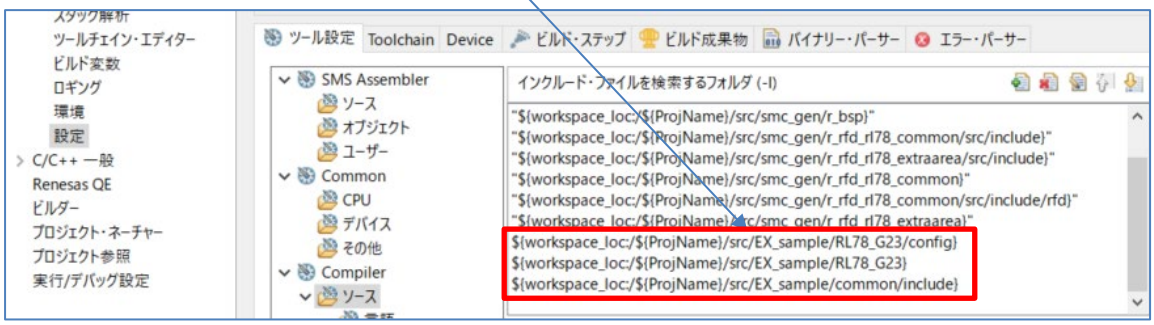
④-2：e² studio(CC-RL)の設定

e² studio(CC-RL) でのインクルード・パスの設定は、"プロパティ"ウインドウで設定します。

- "C/C++ビルド" [設定] ‐ "Compiler" [ソース]　で表示した画面でインクルード・ファイルのパスを設定します。

追加するインクルード・パスを以下に示します。

```
${workspace_loc:/${ProjName}/src/EX_sample/RL78_G23/config}
${workspace_loc:/${ProjName}/src/EX_sample/RL78_G23}
${workspace_loc:/${ProjName}/src/EX_sample/common/include}
```



※src フォルダ直下に EX_sample.zip を解凍した場合のインクルード・パスとなります。

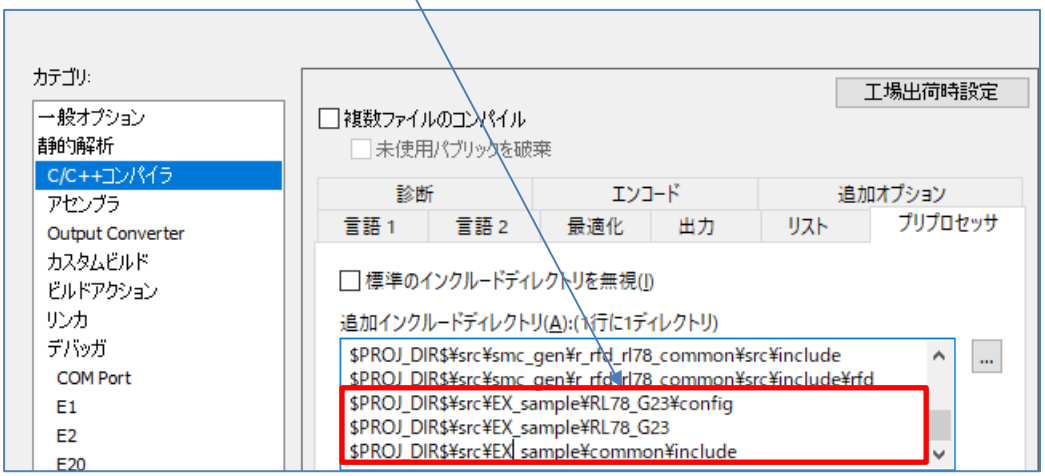
④-3：IAR EW for Renesas RL78 の設定

IAR Embedded Workbench でのインクルード・パスの設定は、カテゴリの"C/C++コンパイラ"を選択し、"プリプロセッサ"タブで設定します。

- [追加インクルード・ディレクトリ(A) : (1 行に 1 ディレクトリ)]で"パス編集"ウインドウを表示して、インクルード・ディレクトリのパスを設定します。

追加するインクルード・パスを以下に示します。

```
$PROJ_DIR$\src\EX_sample\RL78_G23\config
$PROJ_DIR$\src\EX_sample\RL78_G23
$PROJ_DIR$\src\EX_sample\common\include
```



※src フォルダ直下に EX_sample.zip を解凍した場合のインクルード・パスとなります。

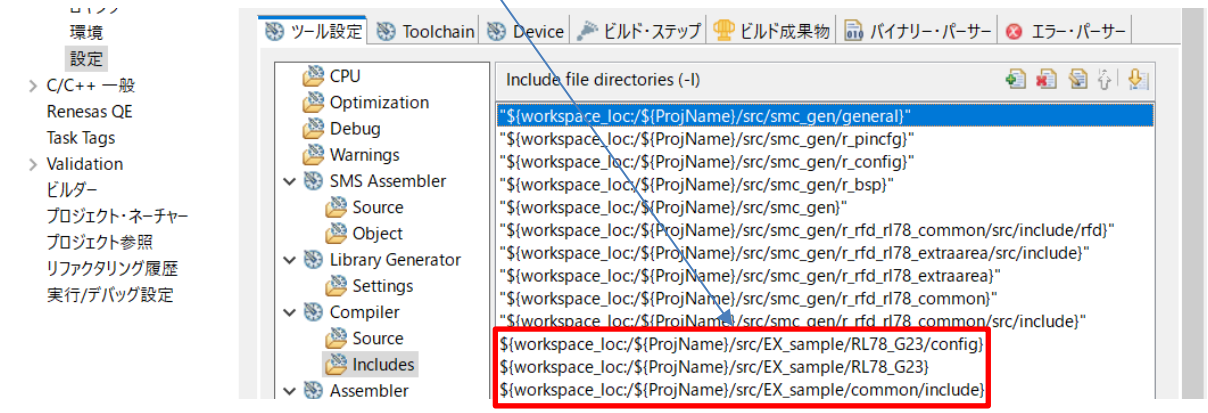
④-4：e² studio(LLVM)の設定

e² studio (LLVM)でのインクルード・パスの設定は、"プロパティ"ウインドウで設定します。

- "C/C++ビルド" [設定] － "Compiler" [Includes]　で表示した画面でインクルード・ファイルのパスを設定します。

追加するインクルード・パスを以下に示します。

```
${workspace_loc:/${ProjName}/src/EX_sample/RL78_G23/config}
${workspace_loc:/${ProjName}/src/EX_sample/RL78_G23}
${workspace_loc:/${ProjName}/src/EX_sample/common/include}
```



※src フォルダ直下に EX_sample.zip を解凍した場合のインクルード・パスとなります。

⑤ ユーザ定義マクロの設定

⑤-1　CS+の設定

Renesas Flash Driver RL78 Type01 ユーザーズマニュアル(R20UT4830) 「ユーザ定義マクロの設定」を参照ください。

⑤-2　e² studio(CC-RL)の設定

Renesas Flash Driver RL78 Type01 ユーザーズマニュアル(R20UT4830) 「ユーザ定義マクロの設定」を参照ください。

⑤-3　IAR EW for Renesas RL78 の設定

Renesas Flash Driver RL78 Type01 ユーザーズマニュアル(R20UT4830) 「ユーザ定義マクロの設定」を参照ください。

⑤-4　e² studio(LLVM)の設定

Renesas Flash Driver RL78 Type01 ユーザーズマニュアル(R20UT4830) 「ユーザ定義マクロの設定」を参照ください。

⑥ デバイス項目の設定

⑥-1　CS+の設定

Renesas Flash Driver RL78 Type01 ユーザーズマニュアル(R20UT4830) 「デバイス項目の設定」を参照ください。

⑥-2　e² studio(CC-RL)の設定

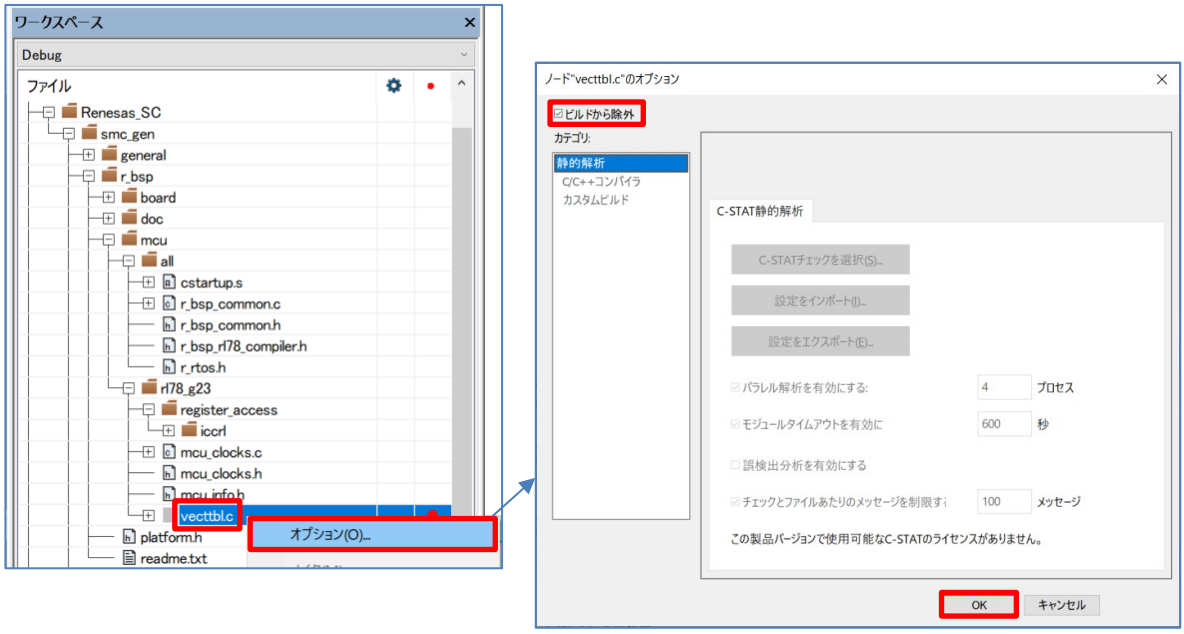
Renesas Flash Driver RL78 Type01 ユーザーズマニュアル(R20UT4830) 「デバイス項目の設定」を参照ください。

⑥-3　IAR EW for Renesas RL78 の設定

サンプルプログラムとして用意した vecttbl.c を用いてビルドすることで、ユーザ・オプション・バイトを 0x6EFFE8 に、オンチップ・デバッグ・オプション・バイトを 0x85 に設定します。

スマート・コンフィグレータでコード生成時、"smc_gen\r_bsp\mcu\rl78_g23\ vecttbl.c"が生成され、vecttbl.c が重複するため、こちらを無効にする必要があります。

ツリーで[プロジェクト]内の" Renesas_SC\smc_gen\r_bsp\mcu\rl78_g23\ vecttbl.c"をマウス右クリックで"オプション"を選択、表示された画面内の[ビルドから除外]にチェックを入れ、ビルド対象から除外してください。



⑥-4　e² studio(LLVM)の設定

サンプルプログラムとして用意した vects.c を用いてビルドすることで、ユーザ・オプション・バイトを 0x6EFFE8 に、オンチップ・デバッグ・オプション・バイトを 0x85 に設定します。

スマート・コンフィグレータでソースコード生成時、"smc_gen/general/r_cg_vect_table.c"が生成され、vects.c と設定対象が重複するため、こちらを無効にする必要があります。

プロジェクト・エクスプローラーから[プロジェクト]内の"smc_gen/general/r_cg_vect_table.c"をマウス右クリックし、[リソース構成]-[ビルドから除外...]機能により、ビルド対象から除外してください。

⑦ メインプログラムからサンプル用プログラムの実行

r_flash_sample_extraarea_rl78g2x.c に記載されている sample_extraarea_main 関数を、作成したプロジェクトの main 関数で呼び出すよう記述し、ビルド、ダウンロード、実行してください。

※sample_extraarea_main 関数のプロトタイプ宣言を記載したヘッダファイル

" r_flash_sample_extraarea_rl78g2x.h"を用意していますので、インクルードしてください。

2.4　サンプルプログラムの動作確認

フラッシュ・シールド・ウインドウ(FSW)を制御する 4byte(32bit)の領域を書き込む動作を以下の方法で確認します。

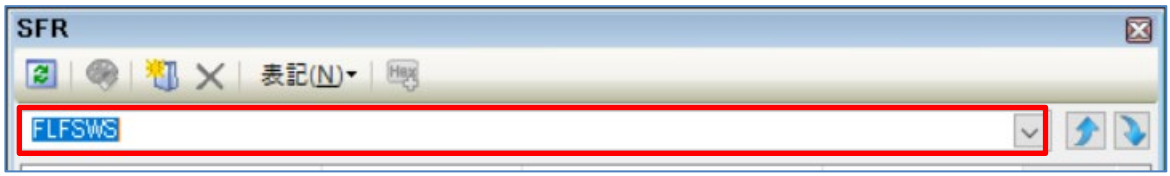
2.4.1　CS+の場合

① [デバッグ] – [デバッグ・ツールへダウンロード]を選択し、デバッグを開始します。

② [表示] – [SFR]を選択し SFR ウインドウを表示します。

③ SFR ウインドウに"FLFSWS"、"FLFSWE"を表示します。

検索のテキストボックスに"FLFSWS"を記入し Enter キーを押すと表示されます。



④ [デバッグ] – [実行]を選択しプログラム実行後、SFR ウインドウの値が変化していることを確認します。

FLFSWS：0x8000

FLFSWE：0x8040

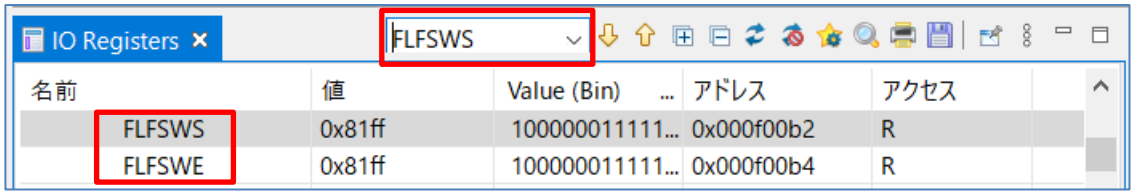
2.4.2　e² studio(CC-RL)の場合

① [実行] – [デバッグ]を選択し、デバッグを開始します。

② [ウインドウ] – [ビューの表示] – [その他] – [IO Registers]を選択し、IO Registers ビューを表示します。

③ IO Registers ビューに"FLFSWS"、"FLFSWE"レジスタを表示します。

検索用のテキストボックスに"FLFSWS"を記入して Enter キーを押すと表示されます



④ [実行] – [再開]を選択しプログラム実行後、式ビューの"FLFSWS"、"FLFSWE"が以下の値に変化していることを確認します。

FLFSWS：0x8000

FLFSWE：0x8040

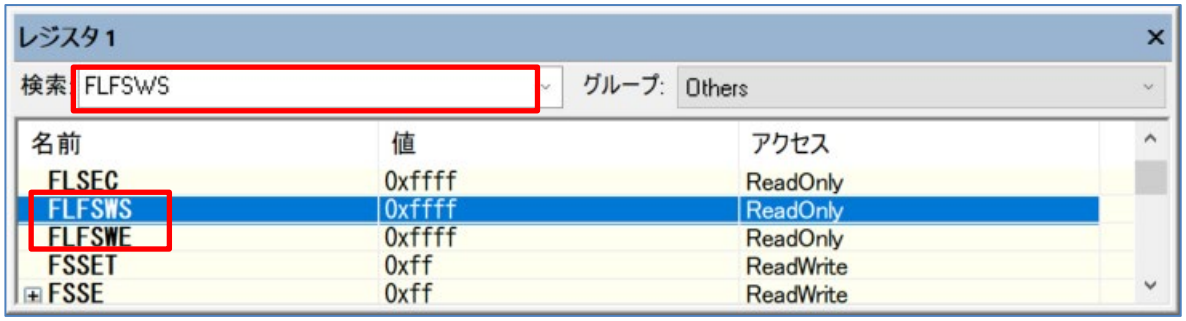
2.4.3　IAR EW for Renesas RL78 の場合

① [プロジェクト] – [ダウンロードしてデバッグ]を選択し、デバッグを開始します。

② [表示] – [レジスタ]を選択後、レジスタ(1)～レジスタ(4)のいずれかを選択しレジスタウインドウを表示します。

③ レジスタウインドウの"FLFSWS"、"FLFSWE"を表示します。

検索のテキストボックスに"FLFSWS"を記入し、Enter キーを押すと表示されます。



④ [デバッグ] – [実行]を選択しプログラム実行後、レジスタウインドウの"FLFSWS"、"FLFSWE"が以下の値に変化していることを確認します。

FLFSWS：0x8000

FLFSWE：0x8040

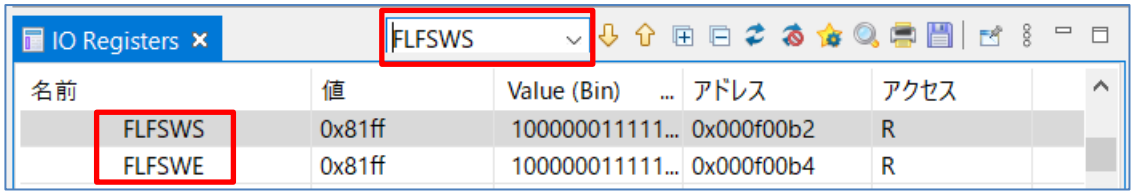
2.4.4　e² studio(LLVM)の場合

① [実行] – [デバッグ]を選択し、デバッグを開始します。

② [ウインドウ] – [ビューの表示] – [その他] – [IO Registers]を選択し、IO Registers ビューを表示します。

③ IO Registers ビューに"FLFSWS"、"FLFSWE"レジスタを表示します。

検索用のテキストボックスに"FLFSWS"を記入して Enter キーを押すと表示されます



④ [実行] – [再開]を選択しプログラム実行後、式ビューの"FLFSWS"、"FLFSWE"が以下の値に変化していることを確認します。

FLFSWS：0x8000

FLFSWE：0x8040

3. 注意事項

(1) コード・フラッシュ/エクストラ領域の書き換え操作

コード・フラッシュ/エクストラ領域を書き換え操作する場合は、実行するプログラムを RAM に配置してください。

(2) データ・フラッシュ領域を操作する場合の前提条件

データ・フラッシュ領域を操作する前に、必ず、データ・フラッシュ・コントロール・レジスタ(DFLCTL レジスタ)の DFLEN ビット[bit0] = 1(データ・フラッシュのアクセス許可)に設定しておく必要があります。

(3) フラッシュ・メモリ書き換え操作中のプログラム実行

RL78/G23 のセルフ・プログラミングは、フラッシュ・メモリ・シーケンサを使用し、フラッシュ・メモリの書き換えを制御します。フラッシュ・メモリの書き換えが可能な"フラッシュ・メモリ制御モード"では、操作対象のフラッシュ・メモリは参照できなくなります。

- コード・フラッシュ・プログラミング・モードでは、コード・フラッシュ・メモリを参照することができません。コード・フラッシュ・プログラミング・モード中に実行する ROM(コード・フラッシュ・メモリ)上の RFD RL78 Type01 の関数、ユーザ・プログラム、およびそれぞれの参照データは、事前に RAM へコピーして、RAM 上で実行、参照する必要があります。
- データ・フラッシュ・プログラミング・モードでは、データ・フラッシュ・メモリを参照することができません。データ・フラッシュ・プログラミング・モード中に参照するデータは、事前に RAM へコピーして、RAM 上で参照する必要があります。

(4) セルフ・プログラミング中のクロック条件

セルフ・プログラミング中は、高速オンチップ・オシレータを動作させておく必要があります。

中速オンチップ・オシレータは停止(MIOEN = 0)させ、メイン・オンチップ・オシレータ・クロック(fOCO)は高速オンチップ・オシレータを選択(MCM1 = 0)してください。

また、セルフ・プログラミングを実行する場合は、メイン・システム・クロックを使用し、X1 発振回路、または、高速オンチップ・オシレータを選択してください。中速オンチップ・オシレータやサブシステム・クロックでセルフ・プログラミングを実行することはできません。

※セルフ・プログラミング中のクロック設定については、対象となる RL78 マイクロコントローラのユーザーズマニュアルを参照してください。

(5) オンチップ・デバッガでセルフ・プログラミングのデバッグをする場合の注意事項

オンチップ・デバッガでセルフ・プログラミングのデバッグをする場合、デバッグ実行時に RAM の先頭アドレスから 128byte の領域を使用するため、この領域を空けてください。それと同時に、ご使用の開発環境が CS+, e^2 studio の場合は、デバッガでフラッシュのセルフ・プログラミングを行う設定をしておく必要があります。

- CS+の設定例：

  プロジェクトの"RL78 E2 [Lite](デバッグ・ツール)"から"接続用設定"タブを選択、"フラッシュ"の「Flash のセルフ・プログラミングを行う」を「はい」に設定します。
- e^2 studio の設定例：

  プロジェクトの"プロパティ"から"実行/デバッグ設定"を選択し、対象の"HardwareDebug"設定を編集します。"Debugger"タブを選択後、"Connection Settings"タブを選択し、表示された「フラッシュのセルフ・プログラミングを行う」を「はい」に設定します。

4. 関連文書

各ドキュメントの最新版をルネサス エレクトロニクスホームページから入手してください。
(https://www.renesas.com).

No	Document Title	Document Number
1	RL78/G22 ユーザーズマニュアル　ハードウェア編	R01UH0978
2	RL78/G23 ユーザーズマニュアル　ハードウェア編	R01UH0896
3	RL78/G24 ユーザーズマニュアル　ハードウェア編	R01UH0961
4	RL78 ファミリ ボードサポートパッケージモジュール	R01AN5522
5	RL78 ファミリ Renesas Flash Driver RL78 Type01 ユーザーズマニュアル	R20UT4830
6	E1/E20/E2 エミュレータ, E2 エミュレータ Lite ユーザーズマニュアル別冊 (RL78 接続時の注意事項)	R20UT1994

5. 改訂記録

Rev.	発行日	改定内容	
		Page	概要
1.00	2021.12.20	-	新規作成
1.10	2023.5.31	-	RL78/G22, RL78/G24 を追加
1.20	2024.3.22	-	LLVM コンパイラをサポート

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うと、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、ノイズ幅射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
   標準水準：　コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等
   高品質水準：輸送機器（自動車、電車、船舶等）、交通制御（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またはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
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。

© 2024 Renesas Electronics Corporation. All rights reserved.