

RL78 スマート・コンフィグレータ

R20AN0581JJ0104 Rev.1.04 2025.04.21

ユーザーガイド:IAR 編

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(以下、スマート・コンフィグレー タと略す)の基本的な使用方法とスマート・コンフィグレータの生成ファイルを IAR Embedded Workbench のプロジェクトに追加するまでの手順について説明します。

スマート・コンフィグレータおよび統合開発環境の対象バージョンは以下の通りです。

- IAR Embedded Workbench for Renesas RL78 V5.10.3 以降
- RL78 スマート・コンフィグレータ V1.13.0 以降

対象デバイス

サポートしているデバイスは、以下の URL をご参照ください。

https://www.renesas.com/rl78-smart-configurator



目次

1. 概要	5
1.1 目的	5
1.2 特長	5
1.3 ソフトウェア・コンポーネント	5
2. 使用前の準備	6
2.1 スマート・コンフィグレータのインストール	6
2.2 スマート・コンフィグレータのアンインストール	6
3. スマート・コンフィグレータの操作方法	7
3.1 操作手順	7
3.2 起動	8
3.3 コンフィグレーションファイルの作成と読み込み	9
3.3.1 新規作成	9
3.3.2 既存のコンフィグレーションファイルを開く	
3.4 ウィンドウ	
3.4.1 メインメニュー	
3.4.2 ツールバー	
3.4.3 スマート・コンフィグレータビュー	
3.4.4 MCU/MPU パッケージビュー	14
3.4.5 コンソールビュー	
3.4.6 コンフィグレーションチェックビュー	15
4. 周辺機能の設定	
 周辺機能の設定 4.1 ボード設定 	16
 周辺機能の設定 4.1 ボード設定 4.1.1 デバイス選択 	16
 4. 周辺機能の設定 4.1 ボード設定 4.1.1 デバイス選択 4.1.2 ボード選択 	
 4. 周辺機能の設定 4.1 ボード設定 4.1.1 デバイス選択 4.1.2 ボード選択 4.1.3 ボード設定のエクスポート 	
 4. 周辺機能の設定 4.1 ボード設定 4.1.1 デバイス選択 4.1.2 ボード選択 4.1.3 ボード設定のエクスポート	
 4. 周辺機能の設定 4.1 ボード設定 4.1.1 デバイス選択 4.1.2 ボード選択 4.1.3 ボード設定のエクスポート 4.1.4 ボード設定のインポート 4.2 クロック設定 	
 4. 周辺機能の設定 4.1 ボード設定 4.1.1 デバイス選択 4.1.2 ボード選択 4.1.3 ボード設定のエクスポート	
 4. 周辺機能の設定 4.1 ボード設定	
 4. 周辺機能の設定 4.1 ボード設定 4.1.1 デバイス選択 4.1.2 ボード選択 4.1.3 ボード設定のエクスポート	
 4. 周辺機能の設定 4.1 ボード設定	
 4. 周辺機能の設定 4.1 ボード設定 4.1.1 デバイス選択 4.1.2 ボード選択 4.1.3 ボード設定のエクスポート	
 4. 周辺機能の設定 4.1 ボード設定	
 4. 周辺機能の設定	
 4. 周辺機能の設定 4.1 ボード設定 4.1 デバイス選択 4.1.2 ボード選択 4.1.3 ボード設定のエクスポート 4.1.4 ボード設定のインポート 4.2 クロック設定 4.3 コンシステム設定 4.4 ポーネント設定 4.4 ポーネント設定 4.4.1 コンポーネント・ビューとハードウェア・ビューの切り替え 4.4.2 ソフトウェア・コンポーネントの追加 4.4.3 ソフトウェア・コンポーネントの追加 4.4.4 CG ドライバの設定 4.4.5 CG コンフィグレーションのリソース変更 4.4.6 SNOOZE モード・シーケンサの設定 	
 4. 周辺機能の設定 4.1 ボード設定 4.1 デバイス選択 4.1.2 ボード選択 4.1.3 ボード設定のエクスポート 4.1.3 ボード設定のインポート 4.4 ボード設定のインポート 4.2 クロック設定 4.3 コンシステム設定 4.4 ポーネント設定 4.4 ポーネント設定 4.4.1 コンポーネント・ビューとハードウェア・ビューの切り替え 4.4.2 ソフトウェア・コンポーネントの追加 4.4.3 ソフトウェア・コンポーネントの追加 4.4.3 ソフトウェア・コンポーネントの削除 4.4.4 CG ドライバの設定 4.4.5 CG コンフィグレーションのリソース変更 4.4.6 SNOOZE モード・シーケンサの設定 4.4.7 SMS データファイルの更新 	
 4. 周辺機能の設定 4.1 ボード設定 4.1 デバイス選択 4.1.2 ボード選択 4.1.2 ボード設定のエクスポート 4.1.3 ボード設定のインポート 4.1.4 ボード設定のインポート 4.2 クロック設定 4.3 コンシステム設定 4.4 ポーネント設定 4.4 ポーネント設定 4.4 ポーネント設定 4.4.1 コンポーネント・ビューとハードウェア・ビューの切り替え 4.4.2 ソフトウェア・コンポーネントの追加 4.4.3 ソフトウェア・コンポーネントの削除 4.4.4 CG ドライバの設定 4.4.5 CG コンフィグレーションのリソース変更 4.4.6 SNOOZE モード・シーケンサの設定 4.4.7 SMS データファイルの更新 4.4.8 ELCL 固定機能モジュールのダウンロード 	16 16 16 17 18 19 20 21 21 21 22 24 25 26 28 31 32
 4. 周辺機能の設定 4.1 ボード設定 4.1.1 デバイス選択 4.1.2 ボード選択 4.1.3 ボード設定のエクスポート 4.1.4 ボード設定のインポート 4.4 ボード設定のインポート 4.2 クロック設定 4.3 コンシステム設定 4.4 ポーネント設定 4.4 ポーネント設定 4.4.1 コンポーネント・ビューとハードウェア・ビューの切り替え 4.4.2 ソフトウェア・コンポーネントの追加 4.4.3 ソフトウェア・コンポーネントの追加 4.4.3 ソフトウェア・コンポーネントの削除 4.4.4 CG ドライバの設定 4.4.5 CG コンフィグレーションのリソース変更 4.4.6 SNOOZE モード・シーケンサの設定 4.4.7 SMS データファイルの更新 4.4.8 ELCL 固定機能モジュールのダウンロード 4.4.9 固定機能 ELCL コンポーネントの設定 	
 4. 周辺機能の設定 4.1 ボード設定 4.1.1 デバイス選択 4.1.2 ボード選択 4.1.3 ボード設定のエクスポート 4.1.3 ボード設定のインポート 4.4 ボード設定のインポート 4.2 クロック設定 4.3 コンシステム設定 4.4 ポーネント設定 4.4 ポーネント設定 4.4.1 コンポーネント・ビューとハードウェア・ビューの切り替え 4.4.2 ソフトウェア・コンポーネントの追加 4.4.3 ソフトウェア・コンポーネントの削除 4.4.4 CG ドライバの設定 4.4.5 CG コンフィグレーションのリソース変更 4.4.6 SNOOZE モード・シーケンサの設定 4.4.7 SMS データファイルの更新 4.4.8 ELCL 固定機能モジュールのダウンロード 4.4.9 固定機能 ELCL コンポーネントの設定 4.4.10 ELCL Flexible Circuit の作成と編集 	16 16 16 17 18 19 20 21 21 21 21 21 22 24 25 26 28 31 32 33 34
 4. 周辺機能の設定 4.1 ボード設定 4.1.1 デバイス選択 4.1.2 ボード選択 4.1.3 ボード設定のエクスポート 4.1.4 ボード設定のインポート 4.2 クロック設定 4.3 コンシステム設定 4.4 ポーネント設定 4.4 ポーネント設定 4.4.1 コンポーネント・ビューとハードウェア・ビューの切り替え 4.4.2 ソフトウェア・コンポーネントの追加 4.4.3 ソフトウェア・コンポーネントの削除 4.4.4 CG ドライバの設定 4.4.5 CG コンフィグレーションのリソース変更 4.4.6 SNOOZE モード・シーケンサの設定 4.4.7 SMS データファイルの更新 4.4.8 ELCL 固定機能モジュールのダウンロード 4.4.9 固定機能 ELCL コンポーネントの設定 4.4.10 ELCL Flexible Circuit の作成と編集 4.4.11 RL78 Software Integration System モジュールのダウンロード 	16 16 16 17 18 19 20 21 21 21 21 22 24 25 26 28 31 32 33 34 39
 4. 周辺機能の設定	16 16 16 17 18 19 20 21 20 21 21 22 24 25 26 28 31 32 33 34 39 41



4.4.14	BSP コンフィグレーションのバージョン変更	43
4.4.15	コンポーネント構成のエクスポート	45
4.4.16	コンポーネント構成のインポート	45
4.4.17	コンポーネントの基本設定	46
4.5 端-	子設定	49
4.5.1	PIOR 機能による端子割り当ての変更	50
4.5.2	ソフトウェア・コンポーネントの端子割り当て変更	51
4.5.3	MCU パッケージビューを使用した端子の設定	52
4.5.4	端子機能から端子番号の表示	53
4.5.5	端子設定のエクスポート	54
4.5.6	端子設定のインポート	54
4.5.7	ボード端子情報を使用した端子設定	55
4.5.8	端子のフィルタ機能	55
4.5.9	端子エラー/警告の設定	56
4.6 割	し り 込み設定	57
4.6.1	ッニッシン 割り込み優先レベルの設定	57
462	割り込みバンクの設定	58
		00
5. 競合	の管理	59
51 U	ソースの競合	59
52 端-	/ · · · · / / / / / / / / / / / / / / /	59
0		
6. ソー	スの生成	61
6.1 ソ-	ースファイルの生成	61
6.2 生)	或ファイルの構成とファイル名	62
6.3 ク	ス・シート・	65
64 端-		66
65 割	, (X)	67
0.0 81		01
7. 統合	開発環境への読み込み	68
7.1 IAF	R Embedded Workbench への読み込み	68
7.2 IAF	マロジェクトのビルド	69
8. 그—·	ザープログラムの作成	70
8.1 ⊐-	-ド生成タイプのカスタムコード追加方法	70
8.2 그-	ーザーアプリケーションコードの使用方法	72
9. 生成	ソースのバックアップ	73
10. レポ	ートの生成	74
10.1 全語	設定内容レポート(PDF または txt 形式)	74
10.2 端-	子機能リスト、端子番号リストの設定内容(csv 形式)	75
10.3 MC	ン パッケージ図(png 形式)	75
		-
11. 그— [.]	ザーコード保護機能	76
11.1 그-	ーザーコード保護機能の指定タグ	76
11.2 그·	ーザーコード保護機能の使用例	76
11.3 競1	合発生時の対応方法	77



11.3.1 5 11.3.2 5	競合発生条件 競合の解決方法	77 78
12. ヘルブ	າ	80
13. 参考ド	ギキュメント	81



1. 概要

1.1 目的

本アプリケーションノートは、IAR 社製統合開発環境 IAR Embedded Workbench およびスマート・コン フィグレータを使用したプロジェクトの作成、基本的な使用方法と IAR Embedded Workbench のプロジェ クトに追加するまでの手順について説明しています。

IAR Embedded Workbench の使い方は、IAR Embedded Workbench のユーザーズマニュアルを参照して ください。

1.2 特長

スマート・コンフィグレータは、「ソフトウェアを自由に組み合わせられる」をコンセプトとしたユー ティリティです。SW 統合機能によるミドルウェアのインポート、ドライバコード生成、端子設定の3つの 機能でお客様のシステムへのルネサス製ドライバの組み込みを容易にします。

1.3 ソフトウェア・コンポーネント

スマート・コンフィグレータは、3 種類のソフトウェア・コンポーネント(コード生成(CG)、グラ フィカル・コンフィグレータ、RL78 Software Integration System)に対応します。それぞれのソフトウェ アが対応するドライバは、以下の通りです。

- <u>コード生成(DTC、A/D コンバータ、割り込みコントローラなど)</u> CG ドライバは、DTC、AD コンバータ、割り込みコントローラなどのマイコン周辺機能の制御プログ ラムです。コード生成機能を使用したソフトウェア・コンポーネントの組み込みが便利です。
- グラフィカル・コンフィグレータ(SMS、ELCL) グラフィカル・コンフィグレータ・モジュールは、他のドライバ設定に比べてグラフィカルな GUI を 提供することで、複雑な構成の設定を容易にします。SNOOZE モード・シーケンサ(SMS)とロジッ ク&イベント・リンク・コントローラ(ELCL)のソフトウェア・コンポーネントを提供しています。
- <u>RL78 Software Integration System (静電容量センサユニット (CTSU2L) など)</u>

RL78 Software Integration System モジュールは、ドライバ、ミドルウェア、アプリケーション SW のソフトウェア・コンポーネントで、コードを生成するための簡単な GUI を提供します。



2. 使用前の準備

2.1 スマート・コンフィグレータのインストール 下記 URL から「RL78 スマート・コンフィグレータ」をダウンロードしてください。 https://www.renesas.com/rl78-smart-configurator

インストーラ起動後、インストーラの手順に従ってインストールしてください。インストールには、管理 者権限が必要です。

2.2 スマート・コンフィグレータのアンインストール

スマート・コンフィグレータをアンインストールするには、コントロールパネルの [アプリと機能] から 「Smart Configurator for RL78」を選択してください。



3. スマート・コンフィグレータの操作方法

3.1 操作手順

スマート・コンフィグレータで IAR 関連のファイルを生成し、IAR Embedded Workbench に読み込むまでの手順を図 3-1 に示します。IAR Embedded Workbench の操作については、IAR 社の関連ドキュメントを参照してください。





3.2 起動

Windows $\lambda \beta - \beta \lambda = - m \beta$ [Renesas Electronics Smart Configurator] \rightarrow [Smart Configurator for RL78 Vx.x.x]を選択します。選択後、スマート・コンフィグレータのメインウィンドウが起動します。

【注】Vx.x.x はご使用のバージョンに読み替えてください。



3.3 コンフィグレーションファイルの作成と読み込み

スマート・コンフィグレータは、使用するマイクロコントローラ、ビルド・ツール、周辺機能、端子機能 などの設定情報をコンフィグレーションファイル (*.scfg) に保存し、参照します。

3.3.1 新規作成

メインツールバーの 📫 [新規コンフィグレーションファイル] ボタンをクリックするとダイアログが表 示されます。

- (1) [プラットフォーム:] で、デバイスを選択します。
- (2) [ツールチェーン:] で、 [IAR RL78 Toolchain] を選択します。
- (3) [ファイル名:] に、ファイル名を入力します。
- (4) [ロケーション:]を確認します。変更したい場合は、[参照]をクリックして保存先を選択してくだ さい。
 - 【注】 [コード生成] ボタンをクリックすると、*.eww、*.ewp、*.ewd、main.c、および buildinfo.ipcf ファイルがこの場所に生成されます。
- (5) [終了] をクリックして、コンフィグレーションファイルを作成します。

	【⑤ 新規スマート・コン:	フィグレータファイル	,	_		×			
	スマート・コンフィグレータ設定								
	新規コンフィグレーションファイルのプラットフォームとツールチェ ーンを選択してください								
	カテゴリ: RL78					~			
	プラットフォーム:		ツールチェーン:						
(1)	フィルタ入力		A Renesas CCRL78	3 Toolchai	n ()				
	> RL78/G23 -	64pin	A IAR RL78 Toolch	nain	(2)			
	> RL78/G23 -	80pin	e ² LLVM for Renes	as RL78					
	> RL78/G23 -	100pin 128ain							
	* RE78/023 -	iSJxFB							
	R7F1000	SKxFB							
	R7F1000	iSLxFB							
	R7F1000	iSNxFB							
	ROM size: 96KB, RAM	size: 12KB, Pin cou	nt: 30						
(3)	ファイル名: Smart_C	onfigurator_Exampl	e						
(4)	ロケーション: D:\RL78	IAR			参照	ę			
			(5) 終了(F)	キャンヤ	214			

図 3-3 コンフィグレーションファイルの新規作成

- (6) 任意のコンポーネントを追加し設定したあと、コードを生成し、プロジェクトを保存します。
 - 【注】 *.eww、*.ewp、*.ewd、および main.c ファイルは初回のコード生成でのみ生成されますが、 buildinfo.ipcf ファイルはコード生成のたびに生成されます。



3.3.2 既存のコンフィグレーションファイルを開く

メインツールバーの □ [既存コンフィグレーションファイルを開く] アイコンをクリックすると、 [ファイルを開く] ダイアログが表示されます。*.scg ファイルを選択して、 [開く] をクリックしてください。

【3 開<						×
← → ~ ↑ 📙 > PC > V	/indov	vs (C:) > smartconfigurator > workspace	~ Ō	workspaceの検索		P
整理 ▼ 新しいフォルダー				833	•	?
🎝 ミュージック	^	名前 ^	更新日時	種類	サイズ	
🎬 Windows (C:)		Smart_Configurator_Example.scfg	2018/10/03 17:03	SCFG ファイル		1 KB
🚔 Local disk (D:)						
🤿 ネットワーク						
	~					
ファイル名(<u>N</u>	: Sm	art_Configurator_Example.scfg	~	Smart Configuration	files	\sim
				開<(<u>Q</u>)	キャンセ	IL

図 3-4 既存のコンフィグレーションファイルを開く



3.4 ウィンドウ

スマート・コンフィグレータを起動すると、メインウィンドウが表示されます。メインウィンドウの構成 を 「図 3-5 メインウィンドウ」に示します。



図 3-5 メインウィンドウ

- (1) メニューバー
- (2) メインツールバー
- (3) スマート・コンフィグレータビュー
- (4) MCU/MPU パッケージビュー
- (5) コンソールビュー
- (6) コンフィグレーションチェックビュー

RL78 スマート・コンフィグレータ

3.4.1 メインメニュー

ファイル	ウインドウ	ヘルフ

メインメニューの一覧を「表 3-1 メニュー一覧」に示します。

表	3-1	メニュ	
~	•		500

メニュー		内容				
ファイル	新規	プロジェクトを新規に作成するための [新規スマート・コンフィグ				
		レータファイル]ダイアログを表示します。				
	開く	既存のプロジェクトを開くための[開く]ダイアログを表示します。				
	保存	プロジェクトを同名で保存します。				
	再開	スマート・コンフィグレータを再起動します。				
	終了	スマート・コンフィグレータを終了します。				
ウィンドウ	設定	プロジェクトのプロパティを設定するための [設定] ダイアログを表				
示します。						
ビューの表示 ウィンドウの表示を設定するための [ビューの表示] ダイアロ						
		示します。				
ヘルプ	ヘルプ目次	ヘルプを表示します。				
	ホームページ	ルネサス Web サイトのスマート・コンフィグレータページを開きま				
		す。				
	リリースノート	ルネサス Web サイトを開き、リリースノートを表示します。				
	ツールニュース	ルネサス Web サイトを開き、ツールニュースを表示します。				
	APIマニュアル	ルネサス Web サイトを開き、API マニュアルを表示します。				
	説明	バージョン情報を表示します。				

3.4.2 ツールバー

🗂 🗁 🔡

メインメニューの一部の機能は、ツールバーのボタンに割り当てられています。各ツールバーボタンに対応するメインメニューを「表 3-2 ツールバーボタンとメインメニューの対応」に示します。

ツールバーボタン	対応するメインメニュー
	[ファイル] → [新規]
6	[ファイル] → [開く]
	[ファイル] → [保存]

表 3-2	ツールバーオ	ドタンとメイ	インメニュー	-の対応
	2 / V · · · ·			· · · · · · · · · · · · · · · · · · ·



3.4.3 スマート・コンフィグレータビュー

[概要]、[ボード]、[クロック]、[システム]、[コンポーネント]、[端子]、[割り込み]の7つのページから構成されます。タブをクリックして、ページを選択すると選択したタブに応じて内容が切り替わります。

德 *Smart_Cor	ifigurator_Example.scfg ×			
概説				🐻 📄 コードの生成 レポートの生成
▼ 機能概要				?
	概要 <u>振要</u> をクリックすると、スマート・コンフィグレータの機能 ことができます。	もを確認する		
rum,	動画		Application Code	
	<u>スマート・コンフィグレータの紹介</u> 関連動画		Software Components	imart 🕧
	最新情報 <u>最新情報</u> をクリックすると、最新リリースの情報を確 できます。 すべての <u>リリースノート</u> を表示	認することが	RTOS Device Drivers	Configurat
	現在のパージョン: V1.13.0 製品ドキュメントとよくあるご質問(FAC <u>ユーザーガイド</u> <u>APIマニュアル</u> <u>FAQ</u> : <u>Smart Configurator</u>	U)	MCU Hardware	9
▼ 現在の設定	2状態			
注:コード生/	成の動作は、 <u>設定</u> ページでカスタマイズできます。 ①		D' (400)	
使用し(いる)		KAM size: 32KB,	Pin count: 100)	
生 成 元 ロ ク =	ンゴノ (PROJECI_LOC+): src+smc_gen		福未…	
マンポーナン		15-25-51	边中	
	ト パータ	170	設定 Copfig ADC(ADC: 使用中)	
S Board S	upport Packages, - v1.90 (r. bsp)	1.90	r bsp(使用中)	
⊘ ポート		1.7.0	Config_PORT(PORT: 使用中)	v
概要 ボード ク	ロック システム コンポーネント 端子 割り込み			

図 3-6 スマート・コンフィグレータビュー

3.4.4 MCU/MPU パッケージビュー

MCU/MPU パッケージ図上に端子状態を表示します。端子設定を変更することもできます。

MCU/MPU パッケージビューは、[割り当て機能]、[ボード機能]、[シンボル名] の3種類を切り替えることができます。

- [割り当てられた機能]は、端子設定の割り当て状態を表示します。
- [ボード機能]は、ボードの初期設定情報を表示します。
 ボードの初期端子設定情報は、[ボード]ページの[ボード:]で選択したボードの端子情報です。
 (4.1 ボード設定、4.5.7 ボード端子情報を使用した端子設定を参照ください。)
- [シンボリック名] は、ユーザーが端子に定義したシンボル名を表示します。シンボリック名のマクロ 定義は、Pin.h ファイルにポート読み出しまたは書き込みの関数とともに生成します。



図 3-7 MCU パッケージビュー

3.4.5 コンソールビュー

スマート・コンフィグレータビューまたは MCU/MPU パッケージビューでの設定変更内容が表示されます。

ע-עעב 😫	B,	R	1	Ē	•	3 -	. 🗆	
スマート・コンフィグレータ出力								
M05000001: 端子 3 (こ PCLBUZ0 の機能が割り当てられています M05000001: 端子 117 (こ ANIO の機能が割り当てられています								~
								\sim
<							3	>

図 3-8 コンソールビュー

3.4.6 コンフィグレーションチェックビュー

ドライバが使用する割り込み、周辺機能、端子、その他の設定に競合が発生した場合に、その内容を表示 します。

🗟 Configuration Problems 🛛		≱ ▽ □ □
8 errors, 0 warnings, 0 others		
	タイプ	
> 🔞 割り込み (2 項目)		
> 😣 周辺機能 (2 項目)		
> 😣 端子 (2 項目)		
> 😣 設定 (2 項目)		

図 3-9 コンフィグレーションチェックビュー



4. 周辺機能の設定

周辺機能は、スマート・コンフィグレータビューから選択します。

4.1 ボード設定

[ボード] ページでは、ボードおよびデバイスの変更が可能です。

4.1.1 デバイス選択

[…] ボタンをクリックすると、デバイスが選択できます。

🔅 *Smart_Co	onfigurator_Example.scfg $ imes$	
デバイス選	選択	🐻 💼 コードの生成 レポートの生成
デバイス選択	R	2 4
ボード: デバイス:	カスタムユーザボード ~ R7F100GLGxFB ボード情報をダウンロードする	
概要ポード	クロック システム コンポーネント 端子 割り込み	

図 4-1 デバイス選択

【注】デバイスの変更は IAR プロジェクトのデバイス (マイクロコントローラ) には反映されません。

デバイスを変更すると、次のメッセージが表示されます。

各ボタンの操作については、「表 4-1 デバイス変更の確認操作」を参照してください。

ि नें।	イス変更の確認 ×	{	▶ デバイス変更の確認	×
?	デバイスの変更により全ての設定が更新されます。新しいデバイスがサポートしない設定は破棄され ます。	(デバイスの変更により全ての設定が更新されます。新しいデバイスがサポートしたます。	い設定は破棄され
	続けますか?		続けますか?	
	(はい(Y) いいえ(N)		保存して続ける 続ける	キャンセル

図 4-2 デバイス変更の確認

表 4-1 デバイス変更の確認操作

ダイアログ・ボタン	操作説明
はい(Y)	選択したデバイスに変更します。
いいえ(N)	選択したデバイスへの変更をキャンセルします。
保存して続ける	現在の設定内容を構成ファイルに保存し、選択したデバイスに変更しま す。
続ける	現在の設定内容を構成ファイルに保存せずに、選択したデバイスに変更し ます。
キャンセル	デバイス変更をキャンセルします。



4.1.2 ボード選択

[] をクリックすると、リストからボードが選択できます。ボード選択により、ボードの接続に合わせて端子、クロック及びシステムが自動で設定されます。

🔯 *Smart_Configurator_Example.scfg ×		
テバイス選択	じ コードの生成	 レポートの生成
デバイス選択		è
ボード: カスタムユーザボード デバイス: R7F100GLGxFB … <u>ボード情報をダウンロードする…</u>		
概要 ボード クロック システム コンボーネント 端子 割り込み		

図 4-3 ボード選択

選択したボードの構成に応じて、以下の項目が変更されます。

- 端子割り当て(初期端子設定)
- メインクロック周波数
- サブクロック周波数
- デバイス
- オンチップ・デバッグ動作設定とエミュレータ設定

ボードを変更すると、図 4-2 または下記のメッセージが表示されます。 各ボタンの操作については、「表 **4-2** ボード変更の確認操作」を参照してください。



図 4-4 ボード変更の確認

表 4-2 ボード変更の確認操作

ダイアログ・ボタン	操作説明
保存して続ける	現在の設定内容を構成ファイルに保存し、選択したボードに変更します。
続ける	現在の設定内容を構成ファイルに保存せずに、選択したボードに変更しま す。
キャンセル	ボード変更をキャンセルします。

【注】選択したボードに応じて、デバイスが変更されます。 デバイスの変更は、IAR プロジェクトのデバイス(マイクロコントローラー)には反映されません。



4.1.3 ボード設定のエクスポート

ボード設定後に今後の参考としてボード設定のエクスポートができます。ボード設定のエクスポートは、 以下の手順で行います。

- (1) [ボード] ページで、[ボードの設定をエクスポート] 🚮 ボタンをクリックします。
- (2) 出力場所を選択し、エクスポートするファイル名(表示名)を入力します。

∰ *Smart_Configurator_Example.scfg ×	- 8
デバイス選択	🐻 😑 コードの生成 レポートの生成
デバイス選択	(1) 관련
ボード: カスタムユーザボード ~ デバイス: R7F100GLGxFB … <u>ボード情報をダウンロードする…</u>	
概要 ボード クロック システム コンポーネント 端子 割り込み	

図 4-5 ボード設定のエクスポート(bdf 形式)

4.1.4 ボード設定のインポート

ボード設定のインポートは、以下の手順で行います。

- (1) [ボードの設定をインポート] i ボタンをクリックし、bdf ファイルを選択してください。
- (2) インポートしたボード設定がボードの選択肢に追加されます。

*Smart_Configurator_Example.scfg ×	- 8
デバイス選択	🛐 🔒 コードの生成 レポートの生成
デバイス選択	(1) ലെഫ്
ポード: カスタムユーザボード (2) 〜	
デパイス: R7F100GLGxFB	
<u>ボード情報をダウンロードする</u>	
概要 ボード クロック システム コンポーネント 端子 割り込み	

図 4-6 ボード設定のインポート(bdf 形式)

ー度インポートしたボード設定は、同じデバイスグループの他のプロジェクトでもボードの選択肢に表示 されます。



4.2 クロック設定

[クロック] ページでは、システム・クロックを設定することができます。[クロック] ページで作成した設定は、各コンポーネントのクロックソースとして使用されます。

クロック設定を変更するには、以下の手順で行います。

- (1) 動作モードと EVDD 設定を指定します。
- (2) デバイス操作に必要なクロックを選択します(デフォルトは、高速オンチップ・オシレータが選択されています)。
- (3) ボードの仕様に従って各クロックの周波数を指定します(一部の内部クロックでは周波数が固定され ていることに注意してください)
- (4) マルチプレクサ・シンボルで、出力クロックのクロックソースを選択します。



図 4-7 [クロック] ページ



4.3 コンシステム設定

[システム] ページでは、オンチップ・デバッグを設定できます。この設定は r_bsp ファイルに反映され ます。

例えば、以下のよう (1) ~ (3)を設定し、4) の [コードを生成] _- 📴 ボタンをクリックします。

	(4)	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
 ① Iミュレータを使う 	⊖ сомポ−ト	
2) E2 Lite		
● 使用する		
3) (● 使用する		
○ 使用する		
◉ 使用する		
0x000000000000000000000000000000000000		
(.	 (1) ・ エミュレータを使う (2) ・ 臣 E 2 Lite ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	 (1) ● Iミュレータを使う ○ COMポート (2) ● E2 Lite ● 使用する (3) ● 使用する (9) 使用する ● 使用する

図 4-8 [システム] ページ

オンチップ・デバッグ設定は、以下の r_bsp ファイルに反映されます。

¥<ProjectDir>¥src¥smc_gen¥r_config¥r_bsp_config.h

オンチップ・デバッグ設定コードの値を変更する場合は、[システム] ページのオンチップ・デバッグ設定 を変更してコードを生成してください。

#if defi	<pre>#if defined(ICCRL78)</pre>										
/* Option byte setting(When using IAR)		*/									
#define	BSP_CFG_OPTBYTE0_VALUE	(OxEFU)	/*	Generated	value.	Do	not	edit	this	manually	*/
#define	BSP_CFG_OPTBYTE1_VALUE	(Ox3AU)	/*	Generated	value.	Do	not	edit	this	manually	*/
#define	BSP_CFG_OPTBYTE2_VALUE	(0xE8U)	/*	Generated	value.	Do	not	edit	this	manually	*/
#define	BSP_CFG_OPTBYTE3_VALUE	(0x84U)	/*	Generated	value.	Do	not	edit	this	manually	*/

```
図 4-9 r_bsp_config.hのオプションバイト設定
```

【注】 これらの設定値は、MCUタイプまたは型番によって異なります。 詳細は、デバイスのハードウェア・マニュアル を参照してください。



4.4 ポーネント設定

[コンポーネント] ページは、ドライバとミドルウェアをソフトウェア・コンポーネントとして組み合わせ ます。追加したコンポーネントは、左側のコンポーネント・ツリーに表示されます。

🔅 Smart_Configurator_Example.scfg $ imes$			
ソフトウェアコンポーネント設定	-c	─── ドの生成	
コンポーネント 🚵 🖾 🖧 🕒 🕀	設定		(i)
 ● 「「」 ● スタートアップ ◆ ご スタートアップ ◆ ジェネリック ● ジェネリック ● r_bsp 一 概要 「ボード」 クロック システム コンボーネント 	コンポーネント・ツリー 		

図 4-10 [コンポーネント] ページ

4.4.1 コンポーネント・ビューとハードウェア・ビューの切り替え

コンポーネント・ツリーでは、コンポーネント・ビューとハードウェア・ビューの2つのツリー表示を提供しています。 以下のアイコンをクリックすることで、表示を切り替えることができます。

- (1) [コンポーネント・ビュー] 5 アイコン:
 コンポーネント・ツリーに、コンポーネントのカテゴリごとにコンポーネントを表示します。
- (2) [ハードウェア・ビュー] パマン:
 コンポーネント・ツリーに、ハードウェア・リソース階層でコンポーネントを表示します。



図 4-11 コンポーネント・ビューとハードウェア・ビューの切り替え

RENESAS

4.4.2 ソフトウェア・コンポーネントの追加 コンポーネントを追加するには、以下の2つの方法があります。

- (a) [コンポーネントの追加] 🍗 アイコンからのコンポーネント追加
- (b) ハードウェア・リソース・ノードからのコンポーネント追加

[コンポーネントの追加] 눟 アイコンからのコンポーネント追加 (a) について説明します。

a-1. [コンポーネントの追加] 🍾 アイコンをクリックします。

🔅 Smart_Configurator_Example.scfg 🗙	□ □
ソフトウェアコンポーネント設定	じ コードの生成 レポートの生成
コンポーネント 🚵 🛃 📮 🕀 設定	0
概要 ボード クロック システム コンポーネント 端子 割り減	<u>」</u> み

図 4-12 コンポーネントの追加

- a-2. [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] ページの リストからコンポーネントを選択します(例: A/D コンバータ)。
- a-3. [タイプ] は [コード生成] であることを確認してください。
- a-4. [次へ] をクリックします。

く コンポ	ーネントの追加		_		×
ソフトウ 使用可能	ェアコンポーネントの選択 ^{Eなコンポーネントの一覧から選択してく}	(ださい		ł	
カテゴリ	全て				~
機能	全て				\sim
フィルタ					
コンポー		Short Name	917 a-3.	バージー	^
⊞ A/0	コンバータ		コード生成	1.4.0	
Boa	ard Support Packages v1.60	r_bsp	RL78 Software Integration System	1.60	
🖶 D/4	Aコンバータ		コード生成	1.3.0	
ta ELC	L AND		グラフィカル・コンフィグレータ	1.1.0	
ta ELC	L D flip flop		グラフィカル・コンフィグレータ	1.2.0	
ta ELC	L EXOR		グラフィカル・コンフィグレータ	1.1.0	
ta ELC	L OR		グラフィカル・コンフィグレータ	1.1.0	~
 ✓ 最新. 説明 アナログ 	ハージョンのみ表示 -デジタル(A/D)変換回路は、アナログ	入力をデジタル信	号に変換する機能です。		^
		-			v
RL78 So	ftware integration Systemモジュール	<u>/をタワンロードする</u>			
# + =0.5	= = =				
空中設入	<u>L</u>				
			a-4.		
?		< 戻る(B)	次へ(N) > 終了(F)	キャンセ	μ

図 4-13 コード生成コンポーネントの追加



- a-5. [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(例: Config_ADC)。
- a-6. 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(例: ADC)。
- a-7. [終了] をクリックします。

【 コンポーネントの追加	_	
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		
A/Dコンパータ a-5. コンフィグレーション名: Config_ADC リソース: ADC a-6.		~
a-7. ? <戻٥(B) 次へ(N) > 終7(F)		キャンセル

図 4-14 コンポーネントの追加

- ハードウェア・リソース・ノードからのコンポーネント追加(b)について説明します。
- b-1. [👺 (ハードウェア・ビュー)] アイコンをクリックし、ツリー・ビューをハードウェア・リソース 階層表示にします。
- b-2. ハードウェア・リソース・ノードをダブルクリックします(例: A/D コンバータ)。
- b-3. [コンポーネントの追加] ダイアログのリストに選択したハードウェア・リソース・ノードのコンポー ネントが表示されます。
- b-4. これ以降は、[コンポーネントの追加] 🍾 アイコンからのコンポーネント追加(a-3)と同じ手順です。

Smart_Configurator_Example.scfg ×	【 コンポーネントの追加		
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の選択 使用可能カフリポーネントの選択		-
コンポーネント 油 ぱ ぱ₂ □ 田 b.1) を で た い の た い の た ト ト ト ト ト ト ト ト ト ト ト ト ト ト ト ト ト ト	(史用可能なコンホーネントの一覧から選択してください) カテゴリ 全て 機能 全て フィルタ		~ ~
 リアルタイムクロック コ2 ビット・インターパル・タイマ クロック出力/ブザー出力制御回路 	コンポーネント <u>b^3</u>) Short Name # A/Dコンパータ	タイプ コード生成	/(` 1.4.0
○ キャットックダイズ ○ オットトックダイズ ○ クイロンパータ ○ コンパピータ ○ シリアル・アレイ・ユニット ○ シリアル・インタフェースIICA ○ シリアル・インタフェースUARTA ○ リリアル・インタフェースUARTA ○ リーム	 ✓ 最新パージョンのみ表示 説明 アナログ-デジタル(A/D)変換回路は、アナログ入力をデジタル信号に変換 PL78 Software Integration Systemモジュールをグロンロードする 	ぬする機能です。	· ^
 リモノノ信号受信機能 データ・トランスファ・コントローラ ロジック&イベントリンクコントローラ 割り込み機能 すー割り込み ご 電圧検出回路 SNOOZET=ド・シーケンサ 	ILCLモジュールをダウンロードする 基本設定…		
	? < 戻る(B) 次へ(N) > 終了	7 (F)	キャンセル

図 4-15 ハードウェア・リソース・ノードからの追加



4.4.3 ソフトウェア・コンポーネントの削除

プロジェクトからソフトウェア・コンポーネントを削除するには、以下の手順で行います。

- コンポーネント・ツリーからソフトウェア・コンポーネントを選択します。
 Shift または Ctrl キーを押下しながらクリックすると複数のソフトウェア・コンポーネントを選択できます。
- (2) [コンポーネントの削除] 🖥 アイコンをクリックします。



図 4-16 ソフトウェア・コンポーネントの削除

コンポーネント・ツリーから、選択したソフトウェア・コンポーネントが削除されます。

削除したソフトウェア・コンポーネントのソースファイルをコード生成するため、[コード生成] ユードの生成 ボタンを押してください。



4.4.4 CG ドライバの設定

CG コンフィグレーションを設定するには、以下の手順で行います。

- コンポーネント・ツリーにある CG コンフィグレーションをクリックし、選択します(例: Config_ADC)。
- (2) 右側の設定パネルでドライバを設定します。以下に手順と画面の例を示します
 - a. [分解能設定] で [10 ビット] を選択します。
 - b. [トリガ・モード設定]で[ソフトウエア・トリガ・ノーウエイト・モード]を選択します。
 - c. [A/D チャネルの選択] で [ANI0] を選択します。
 - d. [変換時間] で [2112/fCLK] を選択します。

*Smart_Configurator_Example.scfg ×					° 🗆
ソフトウェアコンポーネント設定				こードの生成 レポートの	生成
コンポーネント 🚵 🛃 🎝 🗈 🕀	設定			i) ^
転換 1 フィルタ入力	コンパレータ動作設定 ④ 停止	○許可			
マ	- 分解能設定 ● 10ビット	O 8Ľ"≻ト	O 12Ľ″yト		
	VREF(+) 設定	○ AVREFP	○ 内部基準電圧		
(1) Config_ADC	VREF(-) 設定 ● VSS	OAVREFM			
(2) t	 トリガ・モード設定 シフトウエア・トリガ・ノーウエイト・モー シフトウエア・トリガ・ウエイト・モード ハードウェア・トリガ・ノーウエイト・モー ハードウェア・トリガ・ウエイト・モード INTTM01 	ب ۲			
	動作モード設定 ・連続セレクト・モード ワンショット・セレクト・モード A/Dチャネルの選択	 連続スキャン・モード ワンショット・スキャン・モード ANIO 	~		
	変換時間設定 変換時間モード 変換時間 (2	標準1) d. 2112/fCLK	∨ ∨ (66 µs)		ł
	変換結果上限/下限値設定 ● ADLL ≦ ADCRn ≦ ADULで割り込 ○ ADUL < ADCRnまたはADLL > AD	み要求信号(INTAD)を発生 CRnで割り込み要求信号(INTAD)を発生			
概要 ポード クロック システム コンポーネント 端	上限値(ADUL) 子 割り込み	255			~

図 4-17 CG ドライバの設定

CG コンフィグレーションのコード生成は、デフォルトで生成する設定になっています。

CG コンフィグレーションを右クリックし、 [✓ □-F生成] をクリックすると、 [□-F生成] に変わり コードを生成しません。

[コード±成]をクリックすると、 [✓ コード±成]に変わりコードを生成します。

4.4.5 CG コンフィグレーションのリソース変更

スマート・コンフィグレータでは、ユーザーは CG コンフィグレーションのリソースを変更することがで きます(例: TAU0_1 から TAU0_3 に変更)。互換性のある設定は、現在のリソースから新しく選択したリ ソースへ移行することができます。

現在のソフトウェア・コンポーネント用にリソースを変更するには、以下の手順で行います。

- (1) CG コンフィグレーションを右クリックします(例: Config_TAU0_1)。
- (2) コンテキスト・メニューから [リソースの変更] を選択します。



図 4-18 リソースの変更

- (3) [リソースの選択] ダイアログにある新しいリソースを選択します(例: TAU0_3)。
- (4) [次へ] ボタンが有効になるので、クリックします。

1 3 リソースの				×
リソースの	選択			
使用可能な	リソースの一覧から選択してください			
動作モード:	8ビット・カウンタ・モード			~
リソース:	TAU0_3 (3)			~
	TAU0_1			
	TAU1_1 TAU1_3 ITL000 ITL001 ITL012 ITL013			
	(4) <戻る(B) 次へ(N)> 終了(F)	キャンセ	JV

図 4-19 [コンポーネント] ページー新しいリソースの選択



- (5) コンフィグレーション設定は、[コンフィグレーション設定の選択] ダイアログに表示されます。
- (6) 設定が変更可能であるかを確認します。
- (7) テーブル内の設定を使用するか、デフォルト設定を使用するか選択します。
- (8) [終了] をクリックします。

🚺 リソースの選択				×
コンフィグレーション設定の選択				
コンフィグレーション設定一覧 (7)				
使用する設定の選択? 🔘 以下の設定を	吏用する ○ デフォルト設定を修	使用する		
設定項目 (6)	値		変更可能	^
動作クロック	CK02		はい	
クロック・ソース	fCLK/2		はい	
動作モード設定	上位8ビット		はい	
インターバル時間(上位8ビット)	10		はい	
間隔単位	μs		はい	
インターバル時間(下位8ビット)	10		はい	
間隔単位	μs		はい	V
<			3	>
	(8)			
< 戻る(B)	次へ(N) > 終了	(F)	キャンセル	/

図 4-20 新しいリソース設定の確認

リソースは、自動的に更新されます(例: INTTM01 から INTTM03 へ)。

∰ *Smart_Configurator_Example.scfg ×				- 8
ソフトウェアコンポーネント設定				🕲 🛅 コードの生成 レポートの生成
コンポーネント 🚵 🛃 🖧 🖻 🖽	設定			í
 ■ ■<th>クロック設定 動作クロック クロック・ソース</th><th>CK02 fCLK/2</th><th>~</th><th>(クロック周波数:16000 kHz)</th>	クロック設定 動作クロック クロック・ソース	CK02 fCLK/2	~	(クロック周波数:16000 kHz)
💣 r_bsp マ 🗁 ドライバ	- 動作モート設定 ● 上位8ビット	○下位8ピット	C)上位と下位8ビット
✓	インターバル・タイマ設定			
	インターバル時間(上位8ビット)	10	μs ~ ((実際の値:10)
	インターバル時間(下位8ビット)	10	$\mu s \qquad \lor$	
	□ カウント開始時にINTTM03割り込	みを発生する		
	- 割り込み設定 ✓ タイマ・チャネル3のカウント完了でき	削り込み発生(INTTM03)		
	優先順位	レベル3(低優先順位)	\sim	
	✓タイマ・チャネル3のカウント完了で書	削り込み発生(INTTM03H)		
	優先順位	レベル3(低優先順位)	~	
概要 ボード クロック システム コンポーネント	端子 割り込み			

図 4-21 自動的に更新されるリソース



コンフィグレーション名を変更する場合は、以下の手順で行います。

- (9) CG コンフィグレーションを右クリックします。
- (10) [リネーム] を選択して、コンフィグレーションに再度名前をつけます(例: Config_TAU0_1 から Config_TAU0_3 へ)。

コンポーネント	눱 🗳 🔁 🕀
	10 T
フィルタ入力	
🗸 🗁 ଅଧି–୮୮୬	f
🗸 🗁 ジェネリ	10
💣 r_bs	۰ ۱
(9) 💽 Con	ig_TAU0_1
	✓ コード生成
	初期化APIのみ出力
	リソースの変更
1	削除
	複製
(10)	ሀネ-ム
	リセット時のデフォルト
	▶ コンフィグレーションの追加 >

図 4-22 コンフィグレーション・リネーム

4.4.6 SNOOZE モード・シーケンサの設定

SNOOZE モード・シーケンサ(SMS)は、グラフィカル・コンフィグレータ・タイプのコンポーネントで、[ソフトウェア・コンポーネントの選択]から追加できます。

フトウェアコンポーネントの選	択			
史用可能なコンポーネントの一覧から選打	尺してください		1	+
カテゴリ 全て				~
幾能全て				~
ารมหร				
^ コンポーネント	Short Name	タイプ	パージ	^
■ IIC通信 (マスタ・モード)		コード生成	1.8.0	
■ PWM出力		コード生成	1.9.0	
🖾 SNOOZEモード・シーケンサ		グラフィカル・コンフィグレータ	1.4.0	1
		コード生成	1.7.0	
	rm_touch	RL78 Software Integratio	2.00	•
 Touch middleware. 日 日	rm_touch イ状態でも他の周辺機能のf (イ中に処理を実行することで	RL78 Software Integratio 言号をトリガとして起動できるため、、 *低消費電力を実現できます。	2.00 CPU から独	
田ouch middleware. コ 最新パージョンのみ表示 説明 SNOOZE モード・シーケンサは、スタンパ 立した動作が可能です。CPU がスタンパ 178 Software Integration System To Software Integration System To Software Integration System To Software Integration System To Software Integration System System	m_touch イ状態でも他の周辺機能の パイ中に処理を実行することで	RL78 Software Integratio 言号をトリガとして起動できるため、 化消費電力を実現できます。	2.00 CPU から独	~ ~ ~
田ouch middleware. コ 最新パージョンのみ表示 説明 SNOOZE モード・シーケンサは、スタンパ 立した動作が可能です。CPU がスタンパ RZ78 Software Integration Systemモ注 it(CLモジュールをダウンロードする	m_touch イ状態でも他の周辺機能の パイーに処理を実行することで ジュールをダウンロードする	RL78 Software Integratio 言号をトリガとして起動できるため、 化消費電力を実現できます。	2.00 CPU から独	
田ouch middleware. コ 最新パージョンのみ表示 説明 SNOOZE モード・シーケンサは、スタンパ 立した動作が可能です。CPU がスタンパ 3L78 Software Integration Systemモン ににモジュールをグウンロードする 基本設定	m_touch イ状態でも他の周辺機能の パイーに処理を実行することで ジュールをダウンロードする	RL78 Software Integratio 言号をトリガとして起動できるため、 化消費電力を実現できます。	2.00 CPU から独	
田ouch middleware. コ 最新パージョンのみ表示 説明 SNOOZE モード・シーケンサは、スタンパ 立した動作が可能です。CPU がスタンパ はた動作が可能です。CPU がスタンパ はていたシュールをグウンロードする 基本設定…	m_touch イ状態でも他の周辺機能の パイーに処理を実行することで ジュールをダウンロードする	RL78 Software Integratio 言号をトリガとして起動できるため、 化消費電力を実現できます。	2.00 CPU から独	
■ Touch middleware.] 最新パージョンのみ表示 説明 SNOOZE モード・シーケンサは、スタンパ 立した動作が可能です。CPU がスタンパ 31.78 Software Integration Systemモン (CLモジュールをグウンロードする 基本設定	m_touch イ状態でも他の周辺機能のf (イード処理を実行することで ジュールをグウンロードする	RL78 Software Integratio 言号をトリガとして起動できるため、 低消費電力を実現できます。	2.00 CPUから独	
■ Touch middleware. □ 最新パージョンのみ表示 説明 SNOOZEモード・シーケンサは、スタンパ 立した動作が可能です。CPU がスタンパ にした動作が可能です。CPU がスタンパ はCQモジュールをダウンロードする 基本設定	m_touch イ状態でも他の周辺機能の (イ中に処理を実行することで ジュールをダウンロードする	RL78 Software Integratio 言号をトリガとして起動できるため、 (低消費電力を実現できます。	2.00 CPU から独	

図 4-23 SNOOZE モード・シーケンサ (SMS) の追加



SNOOZE モード・シーケンサ(SMS)の GUI は、以下の「図 4-24 SNOOZE モード・シーケンサ(SMS)GUI」表示となり、コード生成と比較してよりグラフィカルに表示されます。 ブロックをドラッグ・アンド・ドロップすることで構成できます。



図 4-24 SNOOZE モード・シーケンサ(SMS)GUI

表 4-3	SMS GUI	エリ	アの	説明
-------	---------	----	----	----

エリア		説明
(1) SMS ブロック		SMS で使用できるブロックを表示します。
		ブロックはシーケンス(機能)を形成するためのパーツで、A/D 電圧取
		得、比較&分岐、1byte 転送などがあります。
(2) ツールバー	€	キャンパスを拡大します。
	P	キャンパスを縮小します。
	E.	SMS データ管理ダイアログを表示し、使用する変数などを管理します。
	è	SMS シーケンスをインポートします。このアイコンをクリックすると幾
		つかのサンプルシーケンスをご使用いただけます。
	4	SMS シーケンスをエクスポートします。
	~	SMS データファイルを更新します。
	i	SMS データファイルの情報を表示します。
(3) 起動トリガ選択		起動トリガを選択します。
(4) 使用リソース		キャンパスで使用しているレジスタ、命令数を表示します。
(5) キャンパス		SMS ブロックを配置して、シーケンスを作成します。
(6) コンソール		SMS で使用できないコンポーネント設定時にメッセージを表示します。

以下のように SMS ブロックを設定します。

- (1) ブロック・リストからブロックを選択します(例: CSI マスタ受信)
- (2) [CSI マスタ受信] ブロックを、キャンバスの Start ブロックと Finish ブロック間(〇のインジケーターが表示されない位置)にドラッグ&ドロップします。
- (3) [CSI マスタ受信] ブロックをダブルクリックして、CSI マスタ受信設定ダイアログを開きます。
- (4) CSI マスタ受信設定ダイアログのプロパティを設定します。
- (5) [データ管理] 設定を開くと、受信データ用の変数/アドレスを編集できます。
- (6) 設定が必要なブロックは、右下が赤く表示されます。正しく設定されると緑に変わります。
- (7) 同じようにいくつかのブロックを追加して、シーケンスを作成します。



図 4-25 SMS ブロック設定

4.4.7 SMS データファイルの更新

以下の手順で、SMS データファイル (ブロック、シーケンス)の更新が行えます。更新することで新しい ブロック、シーケンスをご使用いただけます。

- (1) [SMS データファイルを更新] 🔩 ボタンをクリックして、SMS データファイルの更新を行います。
- (2) SMS データファイルの更新を確認します。
- (3) 新しいバージョンが存在した場合、自動的にダウンロードして更新します。



図 4-26 SMS データファイルのダウンロード

4.4.8 ELCL 固定機能モジュールのダウンロード

ELCL(ロジック&イベント・リンク・コントローラ)のソフトウェアコンポーネントタイプは、グラ フィカル・コンフィグレータです。ELCL コンポーネントには2種類あり、1種類は「スレーブセレクトピ ン機能」、「チャタリング防止機能」などの固定機能 ELCL コンポーネントで、もう1種類は ELCL Flexible Circuit で、フレキシブルに ELCL 回路を作成できます。ELCL 固定機能モジュールは、[コンポーネ ントの追加] ダイアログから追加できます。コンポーネントリストに含まれていない ELCL 固定機能モ ジュールを使用したい場合は、[ELCL モジュールをダウンロードする] のリンクより、ダウンロードできま す。

00/10/10			ß			- D X
カテゴリ	全て		RL78	ELCLモジュールのダウンロード		- 14-
機能	全て		<u>ダウ</u>	ンロードするRL78 ELCLモジュールを選択してください	(2)	
フィルタ				P/LII		オペア認知
านส-	-\$\\	Short Name		21 M	1.1.0	9 A CLEW
# Δ/Γ	אירט ערם	Shore Hame		EXOR	1.1.0	選択をすべて解除
Boa	ard Support Packages v1.90	r_bsp		エッジ検出の間引き機能	2.0.0	
🖶 Cap	pacitive Sensing Unit driver.	r_ctsu		スレーブセレクト端子機能(4線式SPI)	2.0.0	
🖶 D/#	Aコンパータ			複数パラメータ監視機能	2.0.0	
Call ELC	CL AND			Dフリップフロップ	1.1.0	
	CL D flip flop			AND	1.1.0	
	CL EXOR			OR or NOR	1.1.0	
	パージョンのみまテ			Through	1.1.0	
説明				チャタリング防止機能	2.0.0	
7+07	ブーデジタル(Δ/D)変換回路は アナログス	力をデジタル信号に変換する機能	7	RL78/G23 Common ELCL Module	2.0.3	
////	-) シリル(R/U) 友快回回は、 /) ロノハ	ノコモノンノル 昌号 に交通失う シ成形		マンチェスタコード復調機能	2.0.0	
					(3)	_
RL78 So	oftware Integration Systemモジュールを	ダウンロードする	1		ダウンロード	* キャンセル
ELCLES	ジュールをダウンロードする					
基本設定	<u>Æ</u>					
			40.7	7 (T) de la de ll		
(2)						

ダウンロードした ELCL 固定機能モジュールは、コンポーネン選択トリストに自動的に追加させます。

カテゴリ 全て				`
機能 全て フィルタ				`
	Short Name	タイプ	バージ	,
ELCL Flexible Circuit		グラフィカル・コンフ	1.0.0	
3 ELCL OR		グラフィカル・コンフ	1.1.0	
[3] ELCL selector		グラフィカル・コンフ	1.1.0	
🕄 ELCL Through		グラフィカル・コンフ	1.1.0	
🕼 ELCL エッジ検出の間引き機能		グラフィカル・コンフ	2.0.0	
[3] ELCL スレーブセレクト端子機能 (4線式SPI)		グラフィカル・コンフ	2.0.0	
🕼 ELCL チャタリング防止機能		グラフィカル・コンフ	2.0.0	
🕼 ELCL マンチェスタコード復調機能		グラフィカル・コンフ	2.0.0	
党明 アナログ-デジタル(A/D)変換回路は、アナログ入力?	ビデジタル信号に変換す	る機能です。		^
	ウンロードする			

R20AN0581JJ0104 Rev.1.04 2025.04.21



4.4.9 固定機能 ELCL コンポーネントの設定

以下の手順で、固定機能 ELCL コンポーネントを設定します。

- (1) [ソフトウェア・コンポーネントの選択] から固定機能 ELCL コンポーネントを選択します
 (例: ELCL スレーブセレクト端子機能(4線式 SPI))
- (2) [構成] パネルでドライバを構成します。
 - a. [Input signal selector] :入力信号を選択します。
 - b. [Event control (link processor)] : 論理セルブロックを選択します。
 - c. [Output signal selector] : 出力信号を選択します。
- (3) リンクをクリックすることで、アプリケーションノートを開くことができます。



図 4-29 固定機能 ELCL コンポーネントの設定



4.4.10 ELCL Flexible Circuit の作成と編集

ELCL (ロジック&イベント・リンク・コントローラ) Flexible Circuit コンポーネントは、グラフィカル・ コンフィギュレータの新しいコンポーネント・タイプで、コンポーネントリストから選択して使用すること ができます。

געב 🛐	Image: System 2017 - - - ×					
ソフトウェアコンボーネントの選択 使用可能なコンポーネントの一覧から選択してください						
カテゴリ	全て				~	
加強合物	47 V					
	Ξ.(-	
ノイルタ						
コンポー	^ ネント	Short Name	タイプ	バージョン	^	
₩ A/0	コンバータ		コード生成	1.7.0		
H Boa	rd Support Packages v1.90	r_bsp	RL78 Software Integratio	1.90		
🕀 Cap	acitive Sensing Unit driver.	r_ctsu	RL78 Software Integratio	2.00		
- D/A	コンバータ		コード生成	1.5.0		
🛤 ELC	L AND		グラフィカル・コンフィグレータ	1.1.0		
🕅 ELC	L D flip flop		グラフィカル・コンフィグレータ	1.1.0		
🕼 ELC	L EXOR		グラフィカル・コンフィグレータ	1.1.0	_	
e∰ ELC	L Flexible Circuit		グラフィカル・コンフィグレータ	1.0.0		
🕼 ELCL OR			グラフィカル・コンフィグレータ	1.1.0	~	
☑ 最新パージョンのみ表示 説明 The logic and event link controller (ELCL) links signals output by peripheral functions to other specified peripheral functions through internal logic cell blocks, allowing direct communications between peripheral functions without CPU intervention. ▲ RL78 Software Integration Systemモジュールをグウンロードする ELCLモジュールをグウンロードする 基本設定 ▲						
?		< 戻る(B)	欠へ(N) > 終了(F)	+72	1セル	

図 4-30 ELCL Flexible Circuit コンポーネントの追加



ELCL Flexible Circuit コンポーネントは、ELCL 回路の作成と編集のためのドラッグアンドドロップ操作 をサポートする直感的な GUI を提供し、回路設計後に ELCL レジスタ設定を自動的に生成できます。



表 4-4	ELCL Flexible Circuit GUI エリ	ア説明

エリア		説明
(1)ELCL 要素		ELCL で使用できる要素を表示します。
(2)プロパティ		選択した ELCL 要素の設定を表示します。
(3) ツールバー	A	キャンパスを拡大します。
	P	キャンパスを縮小します。
	(?)	ヘルプを表示します。
(4) キャンバス	入力	ELCL 入力要素を配置する領域です。
	論理セルブロック	ELCL 論理セルブロック要素を配置する領域です。
	出力	ELCL 出力要素を配置する領域です。



以下の手順に従って、ELCL Flexible Circuit を作成します。

- (1) パネルから入力/論理セル/出力の ELCL 要素をキャンバスにドラッグ&ドロップします。
- (2) キャンバスで入力/論理セル/出力の ELCL 要素を選択し、プロパティを設定します。
- (3) 始点を終点にドラッグ&ドロップして接続します。
- (4) ELCL 回路作成後、[コードの生成] ボタンをクリックすると、ELCL レジスタ設定が生成されます。
- 【注】 手順 (1)、(2)、(3) は決まった操作ではなく、各ステップを自由に操作し ELCL 回路を作成または 編集できます。



図 4-32 ELCL Flexible Circuit 作成手順


以下に、ELCL 回路を簡単に作成し、正しい設計に導くのに役立つ GUI 操作の詳細を示します。

(1) 始点を終点にドラッグ&ドロップして接続します。



【注】 ポイントをドラッグした時、接続可能な終点だけを薄緑色でハイライト表示します。

(2) ハードウェアの制限を超えた接続や設定を行った場合、各部にエラーが表示されます。 ユーザーは、示されたメッセージに基づいてエラーを解決できます。



図 4-35 ELC	L Flexible	Circuit	エラ	一表示
------------	------------	---------	----	-----

表 4-5	ELCL Flexible Circuit エラーメッセージー覧
-------	----------------------------------

No.	エラーメッセージ
1	現在選択されている入力イベント信号は、pass-through/AND/OR/EX-OR 回路の入力 0 に接続でき
1	ません。
2	信号選択レジスタ xxx はすべて使用されており、割り当てられません。
3	ELL1SEL4 と ELL1SEL5 は両方とも使用されています。同じ論理セルブロック内の別のフリッ
5	プフロップで使用している信号と同じ信号を使用してください。
1	このクロック設定を使用する場合は、別のフリップフロップで使用している信号と同じ信号を設
4	定してください。
5	フリップフロップのセットとリセットは、異なる信号を選択する必要があります。
6	ELCL 回路には割り当てられていないリソースが存在するか、ライン接続が完了していないた
0	め、一部のコードが生成されないか、正しくありません。
7	入力信号に割り込み要求信号を接続した場合、出力信号に周辺機能のハードウェアトリガ以外は
	接続できません。
8	xxx はすでに使用されています。利用可能な他のリソースを選択してください。



4.4.11 RL78 Software Integration System モジュールのダウンロード

RL78 Software Integration System モジュールは、ドライバ、ミドルウェア、アプリケーション SW のソ フトウェア・コンポーネントで、コードを生成するための簡単な GUI を提供し、[コンポーネントの追加] ダイアログからダウンロードできます。

- (1) [図 4-12 コンポーネントの追加] 🍾 アイコンをクリックして [コンポーネントの追加] ダイアログを 開きます。
- (2) [RL78 Software Integration System モジュールをダウンロードする] リンクをクリックして、使用する RL78 Software Integration System モジュールをダウンロードします。

געב 🛐	ーネントの追加			
ソフトウ 使用可能	エアコンポーネントの選択 ^{能なコンポーネントの一覧から選択して}	ください		
カテゴリ	全て			~
機能	全て			\sim
วามจ				
コンポー	-ネント	Short Name	タイプ	バージ ^
⊞ A/0	ロコンバータ		コード生成	1.4.0
Boa Boa D/A	ard Support Packages v1.60 Aコンバータ CLAND	r_bsp	RL78 Software Integration System コード生成 グラフィカル・コンフィグレータ	1.60 1.3.0 1.1.0 x
<				>
 ✓ 最新. 説明 アナログ 	バージョンのみ表示 	ブ入力をデジタル信号に	変換する機能です。	^
				~
RL78 So ELCL王シ 基本設定	ftware Integration Systemモジュー. ジュールをダウンロードする 差			
?		< 戻る(B) 次	へ(N) > 終了(F) キ	ヤンセル

図 4-36 RL78 Software Integration System モジュールのダウンロード

【注】 ダウンロードには、My Renesas へのログインが必要です。ログインしていない場合は、次のダイア ログボックスからログインを求められます。

6	×
My Renesas ログイン	My Renesas
My Renesasアカウントの情報を入力してください。	
メール・アドレス:	
パスワード:	
	パスワードをお忘れの方?
e [*] studioは使用状況を記録し、ルネサスに送信することができます。送信されたデータは、製品・サ 用されます。	ナービスの改善や、より良いサポート・メンテナンスのために使
○ 下記リンクの「プライバシーポリシー」と「収集する情報について」を理解し、ルネサスに使用状況の	のデータが送信されることに同意する。
○ ルネサスに使用状況のデータが送信されることに同意しない。	
<u>ブライバシーボリシー 収集する情報について</u>	
この設定は、後で(設定)ワイントワで変更することができます。	
My Renesas へご登録いただくと、ツール製品などのダウンロードサービスやメールニュースなどの各種	サービスをご利用いただけるようになります。
	ログイン キャンセル

図 4-37 My Renesas ログインダイアログ



- (3) [RL78 Software Integration System モジュールのダウンロード] ダイアログで必要なモジュールを選択します。
- (4) [参照] をクリックして、ダウンロードしたモジュールを保存する場所を選択します。
- (5) [ダウンロード]をクリックすると、モジュールのダウンロードが開始されます。

90	ンロードするRL78	Software Integration System	モジュールを選択して	ください。		(3) 🖆
	タイトル		ドキュメントNo.	リビジョン	発行百	^	すべて選
	RL78 Family F	S2012 Sensor Control Mo	R01AN6196EJ	Rev.1.12	2023-04-2	-	根をすべ
\square	RL78 Family H	HS400X Sensor Control M	R01AN6446EJ	Rev.1.02	2023-04-2		12/12/3
	RL78 Family H	HS300x Sensor Control M	R01AN6194EJ	Rev. 1.22	2023-03-0		
\square	RL78 Family 2	MOD4410, ZMOD4450 a	R01AN6197EJ	Rev. 1.21	2023-03-0		
\square	RL78 Family S	Sensor I2C Communicatio	R01AN6193EJ	Rev.1.11	2023-03-0		
\square	RL78 Family (DB1203 Sensor Control M	R01AN6379EJ	Rev. 1.02	2023-03-0		
\square	RL78 Family F	S1015 Sensor Control Mo	R01AN6198EJ	Rev. 1.01	2023-03-0.		
\square	RL78 Family F	S3000 Sensor Control Mo	R01AN6195EJ	Rev.1.01	2023-03-0		
\square	RL78ファミリ CT	「SUモジュール Software Inte	R11AN0484JJ	Rev.1.30	2023-02-1	~	
<					>		6
ŦŸ.	ュール・フォルダー・	パス:					Ò
	C:¥Users¥	¥.eclipse¥com.renesas.g	platform_download	d¥RL78_Mo	dules¥Generi	Modu	les #

図 4-38 RL78 Software Integration System モジュールのダウンロード



4.4.12 RL78 Software Integration System モジュールの追加

RL78 Software Integration System モジュールを追加する手順は、以下の通りです。

- (1) [図 4-12 コンポーネントの追加] なイアログを 開きます。
- (2) コンポーネントリストから RL78 Software Integration System タイプのコンポーネントを選択します。複数のモジュールを選択したい場合は、Ctrl キーを押下しながらクリックします。
- (3) [終了] をクリックします。

	パーネントの追加				2
ソフトウ 使用可能	フェアコンポーネントの選択 能なコンポーネントの一覧から選択して	てください		ŧ	
カテゴリ	全(
機能	全て				
フィルタ					
	^				_
コンボ-	ーネント	Short Name	917	バージ.	1
# A/	Dコンハータ and Support Dackagos y1.60	r ben	コート生成 PL 78 Software Integration System	1.4.0	1
t Ca	pacitive Sensing Unit driver.	r ctsu	RL78 Software Integration System	1.30	1
	1 75.00 5	-	- 110 0		
🛛 🖶 D//	AJ2/1-2		コード生成	1.3.0	
< D/.	AJ2/\-9			1.3.0	
■ D/. く 量 最新 説明 依存モ The Cl config applica	A Jンバーダ iパージョンのみ表示 :ジュール: r_bsp パージョン 1.13 TSU2L module is a CTSU2L driver gured assuming access via the Tou ation.	for the Capacitive uch middleware la	□-+生成 Sensing Unit. The CTSU2L module is yer, but can also be accessed from the	1.3.0	
■ D/. く 量 最新 説明 依存モ The Cl config applic. RL78 Sc	A Jンバーダ デジュール: r_bsp パージョン 1.13 TSU2L module is a CTSU2L driver 1 gured assuming access via the Tou ation. <u>oftware Integration Systemモジュー</u>	for the Capacitive uch middleware la ルをダウンロードする	□-+生成 Sensing Unit. The CTSU2L module is syer, but can also be accessed from the	1.3.0	
■ D/. < 最新 説明 依存モ The CI config applic RL78 Sc ELCLモジ 基本語	A コンバーダ デジュール: r_bsp パージョン 1.13 TSU2L module is a CTSU2L driver 1 gured assuming access via the Tou ation. Software Integration Systemモジュー ジュールをグウンロードする 零	for the Capacitive uch middleware la ルをダウンロードする	□-卜生成 Sensing Unit. The CTSU2L module is over, but can also be accessed from the	1.3.0	
■ D/. く 量 最新 説明 依存モ The Cl config applic <u>ELCLE</u> 基本設	A Jンバーダ パージョンのみ表示 ジュール: r_bsp パージョン 1.13 TSU2L module is a CTSU2L driver 1 yured assuming access via the Tou ation. Systemモジュー ジュールをグウンロードする 定	for the Capacitive uch middleware la ルをグウンロードする	□-卜生成 Sensing Unit. The CTSU2L module is syer, but can also be accessed from the	e user	
■ D/ く 量 最新 説明 依存モT config applic <u>RL78 Sc</u> <u>ELCLモ</u> 基本設置	A J ンバーダ パージョンのみ表示 ジュール: r_bsp パージョン 1.13 TSU2L module is a CTSU2L driver 1 Jured assuming access via the Tou ation. Systemモジュー ジュールをダウンロードする 定	for the Capacitive uch middleware la ルをダウンロードする	□-+生成 Sensing Unit. The CTSU2L module is iyer, but can also be accessed from the (3)	1.3.0	

図 4-39 RL78 Software Integration System モジュールの追加



4.4.13 RL78 Software Integration System モジュールの設定

RL78 Software Integration System を使用するには、構成オプションを設定し、設定方法はコンポーネントによって異なります。

構成パネルで構成オプションを設定しコード生成を行うと、RL78 Software Integration System の構成ファイルに自動的に設定が反映されます。

【注】RL78 Software Integration System モジュールの構成ファイルは、r_config フォルダに生成されます。

ソフトウェアコンポーネント設	定		🕲 📄 コードの生成 レポートの生成
コンポーネ 🚵 🛃 🎝 🗋 🕀	設定		(1)
 ● ● ○ ○ スタートアップ ◇ ◇ ジェネリック ◇ ご たい ○ ○	プロパティ ◆ 肇 Configurations # Parameter check # Data transfer of INTCTSUWR and INTCTSURD # Data strange address setting for CTSURD # Data storage address setting for CTSURD # Data storage address setting for CTSUWR # Interrupt level for INTCTSUWR # Interrupt level for INTCTSURD # Interrupt level for INTCTSURD * Interrupt level for INTCTSURD * Interrupt level for INTCTSUFN * I SCAPING * TSON端子 * TSO1端子	値 Use system default Interrupt handler Setting in r_ctsu Disable 0xFF800 0xFF800 Level 2 Level 2 Level 2 Level 2 ー 使用する 一 使用する	
概要 ボード クロック システム コンポーネ	シト 端子 割り込み		

図 4-40 RL78 Software Integration System モジュールの設定



4.4.14 BSP コンフィグレーションのバージョン変更

BSP コンフィグレーションのバージョン変更は、以下の手順で行います。

(1) コンポーネント・ツリーから、バージョンを変更する r_bsp コンポーネントを右クリックします。

鬱 *Smart_Configurator_Example.scfg	×			- 0
ソフトウェアコンポーネント設定	È		^で ⊐−ドの生成 レポ	 −トの生成
コンポーネ 🚵 🛃 🖓 🕒 🕀 🛔	設定			í
- 11 日本 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12	プロパティ マ 谷 Configur	ations	値	^
	# Start u	up select	Enable (use BSP startup)	
	# Contr	ol of invalid memory access detection	Disable	
r bsp	# RAM	guard space(GRAM0-1)	Disabled	
(「」	恋 面	of control registers of port function(GPORT)	Disabled	
11 9 39 013	2.2	of registers of interrupt function(GINT)	Disabled	
😫 削除		of control registers of clock control function, voltage detector, an	Disabled	
リセット時の	デフォルト	lash access control(DFLEN)	Disables	
		zation of peripheral functions by Code Generator/Smart Configura	Enable	
	# API fu	Inctions disable(R_BSP_StartClock, R_BSP_StopClock)	Enable	~
	<			>
				< >
概要 ボード クロック システム コンポーネン	小端子 割り込み			

図 4-41 BSP コンフィグレーションのバージョン変更

- (2) コンテキスト・メニューから [バージョンの変更...] を右クリックします。
- (3) [バージョンの変更] ダイアログボックスで変更したいバージョンを選択します。デバイスが対応していないバージョンを選択した場合、[選択されたバージョンはターゲット・デバイスまたはツールチェーンをサポートしていません。] と表示されますので、対応しているバージョンを選択してください。
- (4) [次へ] をクリックします。

🚺 バージョンの変更	[×
バージョンの選	択					
変更後のバージョン	を選択してください。					
⊐ນ ສ ⇒ນ⊾⊘.	- her					
コノルーネノト治: 現ちのげ ごうい	r_bsp				 	
現在のハーション	1.00					
変更後のハーション:	1.01					~
	< 戻る(B)	次へ(N) >	終了(F)	キャンセル	v
	< 戻る(B)	次^(N) >	終了(F)	キャンセル	,

図 4-42 BSP コンポーネントのバージョン選択



(5) バージョン変更により、変更する設定項目の一覧が表示されますので、問題ないことを確認し、[終 了] をクリックします。

【○ パージョンの変更 設定の変更内容 以下の設定が追加または削除されます。	_		×
設定項目 変更はありません。		状態	
<			>
< 戻る(B) 次へ(N) > 終了(F)		キャンセ	IL

図 4-43 設定変更項目の確認

(6) [バージョンを変更し、コードを生成しますか。] と表示されますので、問題なければ [はい] をクリックします。

🕼 バージョンの変更		×
パージョンを変更し、コードを生成しますか。	(はい(Y) いいえ(N)	

図 4-44 バージョンの変更確認

(7) BSP コンポーネントのバージョンが変更され、自動的にコード生成が実行されます。



4.4.15 コンポーネント構成のエクスポート

[コンポーネント] ページの [崎 (コンフィグレーションのエクスポート)] ボタンをクリックすると、現 在の設定を*.xml ファイルとしてエクスポートできます。

コンポーネント 📐	
-5t≣ is	😧 🗟
フィルタ入力	
🗸 🗁 አል-ዞኤን	Ĵ
 ◇ ジェネリッ ☆ r_bsj ◇ ⇒ ドライバ ◇ ⇒ タイマ ♂ Conf 	יל b fig_TAU0_1

図 4-45 コンフィグレーションのエクスポート

4.4.16 コンポーネント構成のインポート

[🏜 (コンフィグレーションのインポート)] ボタンをクリックし、エクスポートされた*.xml ファイルを 選択すると、コンポーネントの構成がインポートできます。

コンポーネント 🔃	⊿ ļa ⊨ 🕀
	😜 🗟
フィルタ入力	
🗸 🗁 スタートアップ	
🗸 🗁 ジェネリック	1
💣 r_bsp	
🗸 🍋 ドライバ	
🗸 🦢 ବିଧ୍ୟ	
💣 Config	_TAU0_1

図 4-46 コンフィグレーションのインポート

4.4.17 コンポーネントの基本設定

モジュールの保存先、依存関係などのコンポーネントの基本設定を変更できます。変更するには、[コン ポーネントの追加] ダイアログ(図 4-13 コード生成コンポーネントの追加)に表示される [ソフトウェ ア・コンポーネントの選択] ページの [基本設定] リンクをクリックし、[設定] ダイアログを表示させま す。

10000000000000000000000000000000000000		- 🗆 ×
フィルタ入力	コンポーネント	
Logging Module Download My Renesas > Scripting マスマート・コンフィグレータ MCU/MPUパッケージ外観 コンポーネント 端子エラー/警告 > ヘルプ	 バックアップの設定 ✓ バックアップ設定を有効にする trashへのバックアップ数 (1-20): 5 コード生成コンポーネントの設定 API関数出力: 設定に合わせてすべて出力する APIコードスタイル: マクロで値を表記 	~
	FIT(RX) / SIS(RL78 / RISC-V MCU)コンポーネントの設定 コード生成時の動作: すべてのコンポーネントファイルを再生成する 依存コンポーネントの確認と追加 依存コンポーネントの追加方法を選択してください 依存関係の追加: 依存コンポーネントを追加する 依存関係の確認: より新しいパージョンの依存コンポーネントはチェックから除外す	~ ~
	ファルダ設定 モジュールの保存先は <u>Module Download</u> ページで設定できます デフォルトの復分	元(T) 適用(L)
	適用して閉じ	る キャンセル

図 4-47 コンポーネントの基本設定

【注】 1. ユーザーは、[trash へのバックアップ数(1~20)] オプション(図 4-48 に示す)を設定することで、バックアップのため、トラッシュフォルダに生成したフォルダの数を制限できます。 制限を 超えると、新しいフォルダが古いフォルダを置き換えます。

バックアップの設定	
✔ バックアップ設定を有効にする	
trashへのバックアップ数 (1-20):	5

図 4-48 バックアップ数の設定



【注】 2. コード生成時の動作には、「設定ファイルを更新する」と「すべてのコンポーネントファイルを 再生成する」の2つのオプションがあります。デフォルトの設定は、「設定ファイルを更新す る」です。「設定ファイルを更新する」を選択し、コード生成する場合、スマート・コンフィグ レータはプロジェクト内にファイルが存在するかどうかをチェックします。ファイルが存在する 場合、そのファイルは書きされません。ただし、設定ファイル(例:xxx_config.h)はコード生成 するたびに更新されます。「すべてのコンポーネントファイルを再生成する」を選択しコード生 成する場合、ファイルが常に上書きされます。

- FIT(RX) / SIS(RL78 /	RISC-V MCU)コンポーネントの設定	
コード生成時の動作:	すべてのコンポーネントファイルを再生成する	~
−依存コンポーネントの確	設定ファイルを更新する すべてのコンポーネントファイルを再生成する	

図 4-49 [コード生成時の動作:]の変更

【注】 3. 初期化 API 関数のみを生成したい場合は、[API 関数出力:] リストボックスで「初期化関数のみ出 力する」に変更してください。".h"、".c"ファイルの voidR_ {ConfigurationName} _Create (void) 、void R_ {ConfigurationName} _Create_UserInit (void) のみが生成されます。 デフォル トのオプション設定「設定に合わせてすべて出力する」に変更するとすべての API 関数が再度生 成されます。

コード生成コンポー	ネントの設定
API関数出力:	設定に合わせてすべて出力する ~
APIコードスタイル:	設定に合わせてすべて出力する
	初期化関数のみ出力する

図 4-50 [API 関数出力:] の変更

スマート・コンフィグレータ V1.4.0 から、初期化 API のみ出力する機能を個別の構成 (コード 生成コンポーネント) に適用できるようになりました。 選択したコンポーネントを右クリック し、コンテキスト・メニューから [初期化 API のみを出力] を選択してください。

👻 🗁 ドライバ			
🗙 🗁 割り込み			
💣 Config_INT(-		_
	\checkmark	コード生成	
		初期化APIのみ出力	
		リソースの変更	
	*	削除 複製 リネーム	
		リセット時のデフォルト	
	+	コンフィグレーションの追加	>

図 4-51 個別構成の [初期化 API のみ出力] 設定

【注】 4. HEX 値でコードを生成するには、下図の [マクロを使用せず即値(16 進数)で表記] オプションに変 更してください。[マクロで値を表記] に戻すと、マクロ記述ですべての API が生成されます。

コード生成コンポー	ネントの設定
API関数出力:	設定に合わせてすべて出力する ~
APIコードスタイル:	マクロで値を表記 ~
「FIT(RX) / SIS(RL7 コード生成時の動1	マクロで値を表記 マクロを使用せず即値(16進数)で表記 年: 該定ノアイルを更新する

図 4-52 [API コードスタイル:] の変更



【注】 5. モジュールのバージョンとその依存関係が不一致の場合に、警告メッセージ W04020011 を表示 します。モジュールとその依存関係の改訂履歴を確認し、使用しているモジュールに変更が不要 な場合は、この警告を無視してかまいません。この警告を消すには、コンポーネント基本設定の [依存関係の確認:] リストボックスで「依存コポーネントのバージョンをチェックしない」を選択 し、[OK] をクリックします。

コード生成コンポー	ネントの設定
API関数出力:	設定に合わせてすべて出力する ~
APIコードスタイル:	マクロで値を表記 ~
	マクロで値を表記
FII(NA) / SIS(NL/	マクロを使用せず即値(16進数)で表記
コード生成時の動	乍: すべてのコンボーネントファイルを再生成する ~

図 4-53 [依存関係の確認:]の変更



4.5 端子設定

端子ページは、端子機能の割り当てに使用します。周辺機能別に端子機能を表示する[端子機能]リスト と、端子番号順に全ての端子を表示する[端子番号]リストの2つの表示があり、タブをクリックして切り 替えることができます。[端子機能]リストには各周辺機能の端子機能が表示され、[端子番号]リストにはす べての端子が端子番号順に表示されます。

f設定							コードの生成	レポートの
ドウェアリソース 🗉 🖃 🔩 💑	端子機能						∂ ∎	R 24
ィルタ文字列を入力	フィルタスス	り (* = any str	ing, ? = any ch	aracter)			すべて	
🚣 इर्गर 🔺	使用する	機能	PIOR	端子割り当て	端子番号	方向	備考	コメント
\$# 1/Oポート		ANIO		P20/ANI0/AVREFP/EI20	/ 117	1	この端子を初期化するコンポーネントが	
2019/2 発生回路		ANI1		✓ 設定されていません	✓ 設定されていません	なし		
ダイマ・アレイ・ユニット		ANI2			✓ 設定されていません	なし		
V TAU0		ANI3		✓ 設定されていません	✓ 設定されていません	なし		
VIAU00		ANI4		✓ 設定されていません	✓ 設定されていません	なし		
TAU01		ANI5			/ 設定されていません	なし		
TAU02		ANI6			/ 設定されていません	なし		
IAU03		ANI7		✓ 設定されていません	/ 設定されていません	なし		
IAU04		ANI8		✓ 設定されていません	✓ 設定されていません	なし		
TAU05		ANI9		✓ 設定されていません	✓ 設定されていません	なし		
TAU06		ANI10		✓ 設定されていません	/ 設定されていません	なし		
IAU07		ANI11			/ 設定されていません	なし		
		ANI12			✓ 設定されていません	なし		
TAU10		ANI13			✓ 設定されていません	なし		
		ANI14			✓ 設定されていません	なし		
		ANI16		🥒 設定されていません	✓ 設定されていません	なし		
		ANI17			/ 設定されていません	なし		
	<)

図 4-54 端子ページ (端子機能)

[ボード]ページでボードを選択すると、[ボード機能] にボードの初期端子機能情報が表示されます。 また、[機能] 選択リストに表示される []] アイコンは、ボードの初期端子機能を示します。

子設定							じ コードの生成	 レポ−トの
子番号							III I	1 22
ィルタ入力	(* = any string, ? = any character)						すべて	
岩子番号	端子名	ボード機能	機能	方向	備考	シンボリック名	אנאב	
1	P142/SCK30/SCL30	P142	設定されていません	なし			Pmod1-4[RTS]	
2	P141/PCLBUZ1/INTP7	P141	設定されていません	なし			Pmod1-1[CTS]	
3	P140/PCLBUZ0/INTP6	INTP6	設定されていません	なし			Pmod1-7[INT]	
4	P120/ANI19/IVCMP1/EI120		設定されていません	なし			J4-12	
5	P37/ANI21		設定されていません	なし			J4-11	
6	P36/ANI22		設定されていません	なし			J4-10	
7	P35/ANI23		設定されていません	なし			J4-9	
8	P34/TxDA1	TxDA1	設定されていません	なし			J9-2[IO1/TX]	
9	P33/RxDA1	RxDA1	定されていません 🗸	なし			J9-1[IO0/RX]	
10	P32/CLKA1		設定されていま	th,			J4-8	
11	P106/TI17/TO17		P33				J9-4[IO~3]	
12	P105/TI16/TO16		RxDA1				J9-5[IO~4]	
13	P104/TI15/TO15		BZAECHUCUKEN	146			J9-6[IO~5]	
14	P103/TI14/TO14		設定されていません	なし			J9-7[IO~6]	
15	P47/INTP2	P47	設定されていません	なし			Pmod2-7[IO1]	
16	P46/INTP1/TI05/TO05		設定されていません	なし			J9-8[IO~7]	
17	P45/SO01		設定されていません	なし			J4-7	
18	P44/SI01/SDA01	SDA01	設定されていません	なし			Grove3-2[SDA]	
10	D42/SCK01/SCL01	SCL01	設定されていません	tel			Group2 1[SCI1	

図 4-55 端子ページ (端子番号)



4.5.1 PIOR 機能による端子割り当ての変更

PIOR「フィルタ機能」は、端子機能設定の管理、端子機能設定の再構成、端子機能の競合のチェックに 便利な機能です。PIOR機能の割り当てを変更するには、以下の手順で行います。

- (1) ツールテキスト入力ボックスに「pior1」と入力し、PIOR1 に関連するすべての端子機能を表示させ ます。
- (2) 端子割り当ての1つを変更すると、PIOR1に関連するすべての端子機能割り当てが自動的に再割り 当てされます。
- (3) 端子エラーメッセージは、[備考] 欄と [構成問題ビュー] に表示されます。
- (4) 端子エラーメッセージが表示されたら、端子割り当てを再構成する必要があります。

ハードウェアリソース 🕀 🖻 ↓ªZ	🔒 端子機	能 (1)							• • • • • • •	. 24 24
フィルタ文字列を入力	pior1								× すべて	\sim
	◆ 使用 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	する 機能 RxD2 SCK00 SCK20 SCL00 SCL00 SCL00 SDA20 SDA20 S100 S120 S000 S200 S000 S200 TxD0	PIOR PIOR1 PIOR1 PIOR1 PIOR1 PIOR1 PIOR1 PIOR1 PIOR1 PIOR1	 端子割り当て 沙定されていません PSS/PCLBUZ/SCK00 砂定されていません 砂定されていません 砂定されていません 砂定されていません 砂定されていません 砂定されていません ジ定されていません ションパンロン(NTP5/SI00/RxD0 ションパンロン(NTD5/SI00/RxD0 ションパンロン(NTD5/SI00/RxD0 ジェンスの(NTD5/SI00/RxD0 ジェンスの(NTD5/SI00/RxD0<	 端子番号 設定されていません ア2 設定されていません 設定されていません 設定されていません 設定されていません ア5 設定されていません ア5 設定されていません ジェントでいません 	方 な <mark>IO な</mark> なななな I な O <mark>な</mark> な	備考 複数の端: ソフトウェア ソフトウェア	(3) テ機能が同一端子にも コンポーネントにより使用 コンポーネントにより使用	刊り当てられています 見されていますが、 月されていますが、	× ۲ ۲
8子機能 端子番号 要 ポード クロック システム コンポーネント 端子 , コンフィグレーションチェック × mors, 0 warnings, 0 others	割り込み								c	78 -
 (3) (3) (34) (5項目) (54010003: PCLBUZ1 (Config_PCLBUZ 64010003: SCK00 (Config_CS00で設定 65000010: 第子72 を推載の機能で使用 6505000011: SCA20を「端子"ページで割り当 6505000011: SCA20を「端子"ページで割り当 	1で設定) が使用 2) が使用する端 月できません。端 当ててください。 iててください。	用する端子と次の端 子と次の端子が競; i子 72 (こ PCLBUZ1,		:す : SCK00 (Config_CSI00で設定), SCK00 (Pin Allocatorで) LBUZ1 (Config_PCLBUZ1で設定), PCLBUZ1 (Pin Allocato /割り当てられています。	型 設定), 端子 rで設定), 端子 端子 端子 端子 端子 端子					

図 4-56 PIOR フィルタ機能

PIOR 設定コードは bsp ファイル: ¥<ProjectDir>¥src¥smc_gen¥r_bsp¥r_config¥ r_bsp_config.h に生成されます。 PIOR 設定コード値を変更したい場合は、関連するピンの割り当てを変更して、再度コードを生成してください。

💮 *Smart_Configurator_Example.scfg 🗙							- 0
端子設定						じ コードの生成	 レポートの生成
ソフトウェアコンポーネント 🗉 🗈 🔩 🔜	端子機能					- 2 🗉 🗉	1 è 4
	フィルタ入力) (* = any strin	ig, ? = any char	acter)		すべて	\sim
v 🍒 r_bsp	使用する	機能	PIOR	端子割り当て	端子番号	方向	備考
✓ ▲ クロック出力 / ブザー出力制御回路		PCLBUZO	PIOR3	P140/PCLBUZ0/INTP6	/ 3	0	
Contig_PCLBUZ0	<						>
端子機能 端子番号							
概要 ボ−ド クロック システム コンボーネント 端子 割	り込み						
⊨ r_bsp_config h ⊠							
560 128/100-pin							^
561 0 : PCLB020 - P140 562 1 : PCLB020 - P31							
563 -*/	_						
564 #define BSP CFG PIOR3	(0) /*	Generate	d value.	Do not edit this ma	nually */		~

図 4-57 PIOR のコード生成



4.5.2 ソフトウェア・コンポーネントの端子割り当て変更

スマート・コンフィグレータは、プロジェクトに追加されるソフトウェア・コンポーネントに端子を割り 当てます。端子の割り当ては端子ページで変更可能です。

このページでは、端子機能と端子番号のリストを表示します。

端子機能リストにあるソフトウェア・コンポーネントの端子割り当てを変更するには、以下の手順で行います。

- (1) [ハードウェア・リソース表示とソフトウェア・コンポーネント表示の切り替え] 🚠 をクリックして、ソフトウェア・コンポーネントによって表示するように変更します。
- (2) ソフトウェア・コンポーネントを選択します(例: Config_INTC)。
- (3) [使用する] タブをクリックし、使用した端子でソートします。
- (4) 端子機能リストの端子割り当て欄、または端子番号欄で、端子割り当てを変更します(例: P46 から P56)。
- (5) または、 [選択されたリソースの次の端子割り当て先] 🔮 ボタンをクリックし、端子割り当てを変 更します。クリックするごとに、機能を割り当てられる端子が表示されます。



図 4-58 端子設定- [端子機能] リストの端子配置設定

スマート・コンフィグレータでは、ユーザーは他のソフトウェア・コンポーネントにリンクすることな く、[端子] ページで端子機能を有効にすることができます。それらの端子を別のソフトウェア・コンポーネ ントが使用する他の端子と区別するため、図 4-58 の備考のように"この端子を初期化するコンポーネント がありません。"という注意書きがつけられます。この場合、初期化コードは生成されませんので、コン ポーネントを追加してください。



4.5.3 MCU パッケージビューを使用した端子の設定
 MCU パッケージビューでは、端子設定をパッケージ図で表示します。ユーザーは MCU パッケージビューをイメージファイルにキャプチャーでき、回転や拡大、縮小ができます。
 MCU パッケージビューで端子を設定するには、以下の手順で行います。

- (1) [拡大] 🎤 ボタンをクリックするか、マウスホイールをスクロールして、ビュー内を拡大します。
- (2) 端子の上で右クリックします。
- (3) 端子に割り当てる信号を選択します。
- (4) [設定の変更...] で、端子の色をカスタマイズすることができます。



図 4-59 MCU パッケージビューを使用した端子設定

4.5.4 端子機能から端子番号の表示

端子機能に関連付けられている端子番号に移動できます。端子機能か端子番号に移動するには、以下の手順で移動します。

- (1) [端子機能] タブで、使用する端子機能を右クリックしポップアップメニューを開きます。
- (2) [端子番号タブにジャンプ]を選択します。
- (3) [端子番号] タブに移動し、(1) で選択した端子機能の端子番号を表示します。

JAXAE								コードの生成	. DW-P03
フトウェアコン	. ⊕ ⊝ ↓ª₂ 歳	端子機能						3 🔳	1
イルタ文字列	「を入力	フィルタ入	力 (* = any string,	? = any charad	:ter)			すべて	
> 📥 r_bsp		使用する	機能	PIOR	端子割り当て	ť	第七報 문	方向	備者
🗸 🚠 割り込	みコントローラ		INTRO		D127/EI127/INTD0		20	1	
🚽 Co	onfig_INTC		INTPO	DIODE	/ P137/E1137/INTPU / 読売されていませ/	,	29	t tst	
~	-		INTPO	PIORS	/ 設定されていません	0	設定されていません	/なし かり	
			INTP2	DIOD:	/ 設定されていません	0	設定されていません	なし	
			INTP3	PIORS	/ 設定されていません	/	設定されていません	なし	
			INTP4	PIORS	認定されていません 見また(EO1c/CCD00/TI01/ しまた) しまた(EO1c/CCD00/TI01/ しまた) しまた(EO1c/CCD00/TI01/ しまた) しょうしょう しょうしょ しょうしょう しょうしょ しょ しょ しょうしょ しょうしょ しょうしょ しょ しょ しょうしょ しょうしょ しょ しょ しょうしょ しょ しょうしょ しょ しょ	TOO1/INT /	設定されていません	40	
			INTEG	PIOR4	P 16/EU 16/CCD00/1101/ // D140/DCLDUZ0/INTDc	1001/101	/0 #*フェエロ カゴル・シン・	۱ ۴	_
			INTD7	DIORE	# F140/PCLB0Z0/INTP6 # 記座されていませ/		「ホナ番号ツノにジヤン」		
			INTP7	PIORS	/ 設定されていません		コメントを端子番号タフ	のコメントに追	皇加
			INTP8	PIORS	/ 設定されていません		コメント削除		
			INTP9	PIORS	/ 設定されていません		端子割り当て		
			INTP10	PIOR1	/ 設定されていません				
			INTP11	PIOR1	/ 設定されていません		煽す割り目(解除		
		-							
mart Confic	・垂ち ック システム コンポーネ gurator Example.scfg	ント 端子 割 ×	り込み		↓				
mart_Config	審 す ック システム コンポーネ gurator_Example.scfg	ント 端子 割 ×	1952み		•			7- Kott	
^{₩₩} [₩] [™] -ド クロ: mart_Confi <u>c</u> そ設定 番号	● マ ック システム コンポーネ gurator_Example.scfg	ント 端子 割 ×	1り込み	· · · · · · · · · · · · · · · · · · ·	•			で コードの生成	ー に レポートの当 日 - 「「」」 とい
ポード クロ: mart_Confic *設定 番号	● マ ック システム コンポーネ gurator_Example.scfg	ント 端子 割 ×	り込み		•			コードの生成 重 すべて	 と レポートの3 ■ ≧」
warte 1993 J ボード クロ: mart_Config ご設定 番号 子番号	● す ック システム コンポーネ gurator_Example.scfg 端子名	ント 端子 割 ×	1り込み	ボード機	↓ 能 機能	方向	備考	で コードの生成 重 すべて	ー レポートのき 114 224 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 334 234 234 234 334 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234 234
mart_Config mart_Config で設定 番号 子番号 71	● す ック システム コンポーネ gurator_Example.scfg 端子名 P54/SCK31/SCL31	ント 端子 割 ×	1り込み	ポード機	能機能 設定されていません	方向 , なし	備考	で コードの生成 副 すべて	ー レポートの5 11 22 12 23 24 25 シンボリック名
warte 1973 3 ボード クロ: mart_Config 子番号 71 72	ック システム gurator_Example.scfg 端子名 PS4/SCK31/SCL31 PS5/PCLBUZ1/SCK0	ント 端子 割 ×	IIり込み	ボード機	能 機能 設定されていません 設定されていません	方向 , なし , なし	備考	で コードの生成 建 すべて	ー レポートのき 読 ご シンボリックキ
mart_Config mart_Config 子番号 子番号 71 72 73	 (当 マ) システム コンポーネ gurator_Example.scfg 端子名 P54/SCK31/SCL31 P55/PCLBUZ1/SCK0 P56/INTP1 	ント 端子 割 ×	Iリ込み	ボード機	 総機能 設定されていません 設定されていません 設定されていません 設定されていません 	方向 , なし , なし	信考	で コードの生成 建 すべて	 レポートの当 レポートの当 シンポリックキ
☆me 173 J ボード クロ: *設定 番号 71 72 73 74	 (当 7) ジステム コンボーネ gurator_Example.scfg 端子名 P54/SCK31/SCL31 P55/PCLBUZ1/SCK0 P56/INTP1 P57/INTP3 	ント 端子 割 × 0	り込み 	ボード機	 ・ ・	方向 , なし , なし , なし	備考	で コードの生成 建 すべて	 レポートの当 レポートの当 シンボリックネ
mart_Config *設定 番号 子番号 71 72 73 74 75	 (当 マ) ジステム コンボーネ gurator_Example.scfg 端子名 P54/SCK31/SCL31 P55/PCLBUZ1/SCK0 P56/INTP1 P57/INTP3 P17/E017/CCD01/TI 	ント 端子 割 × 0 0 0 2/TO02/SO	り; <u>入</u> み 	ボード機	能 機能 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません	方向 方 なし , なし , なし , なし	備考	で コードの生成 建 すべて	ー レポートのき 騙 ご シンポリック名
xme / 70 xmart_Config 2 ひつ: 2 2 2 2 3 74 75 76	 (当 マ) ジク システム コンボーネ gurator_Example.scfg 端子名 P54/SCK31/SCL31 P55/PCL8UZ1/SCK0 P56/INTP1 P57/INTP3 P17/E017/CCD01/TI P16/E016/CCD00/TI 	ント 端子 割 × 0 0 0 0 0 0 0 0 0 0 0 0 0	り込み 	ボード機 INTP5	能 機能 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません	方向 , なし , なし , なし , なし , なし , なし , , , , , , , , , , , , , , , , , , ,	備考	で この 上の 生成 上の 生成 上の 生成 して 、 、 、 、 、 、 、 、 、 、 、 、 、	こ レポートのき ■] № シンポリックネ
☆元 / クロ: mart_Config ・設定 番号 子番号 71 72 73 74 75 76 77	 (当 マ) ジステム コンボーネ gurator_Example.scfg 端子名 P54/SCK31/SCL31 P55/PCLBUZ1/SCK0 P56/INTP1 P57/INTP3 P17/EO17/CCD01/TI P16/EO16/CCD00/TI P15/EO15/SCK20/SC 	ント 端子 割 × 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Iリシムみ 00/TxD0 (P5/Sl00/RxD0 002	ボード機 INTP5	能 機能 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません 認定されていません	方向 , 方 , なし , なし , なし , なし , なし , なし , なし , なし	備考	で コードの生成 運 すべて	□ レポートの当 ■ № シンポリック名
xme / ym 3 ボード / クロ: mart_Config 2 設定 番号 71 72 73 74 75 76 77 78	 (当 マ) ジク システム コンボーネ gurator_Example.scfg 端子名 P54/SCK31/SCL31 P55/PCL8UZ1/SCK0 P56/INTP1 P57/INTP3 P17/EO17/CCD01/TI P15/EO15/SCK20/SC P14/VCOUT1/EO14/ 	ント 端子 割 × 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	り込み り込み 	ボード機 INTP5 IO03 RxD2	能 機能 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません の下P5 設定されていません RxD2	方向 , 方向 , なし , なし , なし , なし , なし , なし , なし , なし	備考 この端子を初期(で コードの生成 副 すべて 化するコン	レポートの当 レポートの当 レポートの当 シンボリックネ シンボリックネ
mart_Config mart_Config つい で設定 番号 71 72 73 74 75 76 77 78 79	 (当 マ) ジステム コンボーネ gurator_Example.scfg 端子名 P54/SCK31/SCL31 P55/PCLBUZ1/SCK0 P56/INTP1 P57/INTP3 P17/EO17/CCD01/TI P15/EO15/SCK20/SC P14/VCOUT1/EO14/ P13/IVREF1/E013/D 	ント 端子 割 × 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	り込み) り込み 00/TxD0 (P5/Sl00/RxD0 102 DA20/SCLA0/Tl03/ DA20/SCLA0/Tl03/	ボード機 INTP5 TO03 RxD2 TxD2	能 機能 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません 取定されていません 取定されていません 取定されていません 取定されていません 取定されていません 取定されていません	方向 , なし , なし , なし , なし , なし , なし , なし , し , ひ	備考	 ごう-ドの生成 すべて すべて たするコン 化するコン 	 レポートの当 レポートの当 シンポリック名
xme / 70 mart_Config mart_Config 一般定 番号 子番号 71 72 73 74 75 76 77 78 79 80	 (当 マ) ジステム コンボーネ gurator_Example.scfg 端子名 P54/SCK31/SCL31 P55/PCLBUZ1/SCK0 P56/INTP1 P57/INTP3 P17/EO17/CCD01/TI P15/EO15/SCK20/SC P14/VCOUT1/EO14/ P13/IVREF1/EO13/Ts P12/E12/E012/SOC 	ント 端子 割 × 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	り込み) 00/TxD0 00/TxD0 1P5/Sl00/RxD0 102 DA20/SCLA0/Tl03/ DA0/Tl04/T004 LTxD/INTP5/Tl05/T	ボード機 INTP5 TO03 RxD2 TxD2 O05	能 機能 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません RxD2 TxD2 TOQLTxD	方向 , なし , なし , なし , なし , なし , なし , なし , なし	備考 信考 この端子を初期 この端子を初期	 ごードの生成 すべて すべて はするコン 化するコン 	 レポートの当 レポートの当 シンポリック名
	(単 マ) ック システム コンポーネ gurator_Example.scfg 端子名 P54/SCK31/SCL31 P55/PCLBUZ1/SCK0 P56/INTP1 P57/INTP3 P17/EO17/CCD01/TI P16/EO16/CCD00/TI P15/EO15/SCK20/SC P14/VCOUT1/EO14/ P13/IVREF1/EO13/T5 P12/E112/EO12/SO0 P11/E112/EO12/SO0 P11/E112/EO11/SIO0	ント 端子 割 × 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	り込みみ 00/TxD0 [P5/Sl00/RxD0 102 DA20/SCLA0/Tl03/7 DA40/Tl04/T004 LTxD/INTP5/Tl05/T Isc/)(SD400/Tl06/T	ボード機 INTP5 IO03 RxD2 TxD2 O05 O06	能 機能 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません アレ2 TxD2 TOOLTxD TOOLTxD TOOLTxD	方向 方なし , ななし , なし , なし , なし , なし , なし , なし ,	備考 この端子を初期 この端子を初期	 ご一ドの生成 すべて すべて 	 レポートの当 レポートの当 シンポリックネ
xmme 1 mm J ボード クロ: mart_Config 2 設定 番号 子番号 71 72 73 74 75 76 77 78 79 80 81 82	(当 マ ック システム コンボーネ gurator_Example.scfg 端子名 P54/SCK31/SCL31 P55/PCLBUZ1/SCK0 P56/INTP1 P57/INTP3 P17/E017/CCD01/TI P16/E016/CCD00/TI P15/E015/SCK20/SC P14/VCOUT1/E014/ P13/IVREF1/E013/Tb P12/E112/E011/SI00 P11/E111/E011/SI00 P10/E110/F010/SCK2	ント 端子 割 × 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	19込みみ 00/TxD0 175/SI00/RxD0 102 DA20/SCLA0/TI03/ DA20/SCLA0/TI03/ DA20/TI04/TO04 LTxD/INTP5/TI05/TI RxD/SDA00/TI06/TI 7/TO07	ボード機 INTP5 TO03 RxD2 TxD2 O05 O06	能 機能 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません RxD2 TxD2 ToOLTxD TOOLTxD TOOLTxD TOOLTxD TOOLTxD	方向 方なし がなし がなし がなし がなし に の の の し	備考 この端子を初期 この端子を初期	コードの生成 すべて まべて 化するコン 化するコン	 レポートの当 レポートの当 シンボリックネ
America (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1	● マ ック システム コンボーネ gurator_Example.scfg 端子名 P54/SCK31/SCL31 P55/PCL8UZ1/SCK0 P56/INTP1 P17/E017/CCD01/TI P16/E016/CCD00/TI P15/E015/SCK20/SC P14/VCOUT1/E014/ P13/IVREF1/E013/TS P12/E112/E012/SO0 P11/E111/E011/SI00 P10/E110/E010/SCKI P90	ント 端子 割 × 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	00/TxD0 00/TxD0 IP5/SI00/RxD0 02 DA20/SCLA0/TI03/ DA20/SCLA0/TI03/ DA20/SCLA0/TI03/ TA0/TI04/TO04 LTxD/INTP5/TI05/T RxD/SDA00/TI06/T 17/TO07	ボード機 INTP5 TOO3 RxD2 TxD2 OO5 OO6	能 機能 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません RxD2 TxD2 ToOLTxD TOOLTxD TOOLRxD 設定されていません 設定されていません	方向 方なし かななし がなし り たし し り なし り っ な し り っ な の し り っ な し り っ な な し り っ な る し り っ な る し り っ な し し う っ な し し う っ な し し う っ な し し う っ な し し う っ な し う っ な し	備考 この端子を初期 この端子を初期	コードの生成 ほう すべて 化するコン… 化するコン…	 レポートの当 レポートの当 シンボリックキ
mart_Config mart_Config つロ: で、 一、 一、 一、 一、 一、 一、 一、 一、 一、 一	第マ ック システム コンボーネ gurator_Example.scfg 端子名 P54/SCK31/SCL31 P55/PCLBUZ1/SCK0 P56/INTP1 P15/E015/SCK20/SC P16/E015/SCK20/SC P14/VCOUT1/E014/ P13/IVREF1/E013/T5 P12/EI12/E012/SO0 P11/EI11/E011/SI00 P10/EI10/E010/SCK P90 P90 P91	ント 端子 割 × ・ の の の の の の の の の の の の の の の の の の	リシムみ 00/TxD0 [P5/Sl00/RxD0 002 DA20/SCLA0/TI03/ DA40/TI04/TO04 LTxD/INTP5/TI05/T RxD/SDA00/TI06/T 17/TO07	ボード機 INTP5 TOO3 RxD2 TxD2 O05 O06	能 機能 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません RxD2 TxD2 TOOLTxD TOOLTxD TOOLTxD TOOLTxD 設定されていません 設定されていません 設定されていません	方向 方なし かなし かなし かなし なしし なしし なしし なしし な	備考 信考 この端子を初期1 この端子を初期1	 ごうードの生成 運動 すべて 	ー レポートの当 ■ ≧ シンポリックネ

図 4-60 端子番号タブにジャンプ

4.5.5 端子設定のエクスポート

端子設定をエクスポートして、参照することができます。端子設定のエクスポートは、以下の手順で行います。

- (1) 端子ページの [ボードの設定をエクスポート] 📥 ボタンをクリックします。
- (2) 出力場所を選択し、エクスポートするファイル名を入力します。

XML フォーマットでエクスポートしたファイルは、同じデバイスの型名がある他のプロジェクトにイン ポートすることができます。

子設定							词 🖆 コードの生成 レポートの
フトウェアコンポ 🕀 📄 🖧	端子機能						2 🖬 🖬 🏜
フィルタ文字列を入力	フィルタ入力) (* = any sti	ring, ? = any cha	racter)			すべて
🗸 💑 r_bsp	使用する	機能	PIOR	端子割り当て	端子番号	方向	備考
🔋 r_bsp		INTPO		P137/EI137/INTP0	/ 29	1	この端子を初期化す
◇ 益割り込みコントローフ		INTP1	PIOR5	P46/INTP1/TI05/TO05	/ 16	1	
Config_INIC		INTP2		🥒 設定されていません	✓ 設定されていません	なし	
		INTP3	PIOR5	∥ 設定されていません	✓ 設定されていません	なし	
> 🚋 199-ЛЛ•91 х		INTP4	PIOR5	∥ 設定されていません	✓ 設定されていません	なし	
		INTP5	PIOR4	🥒 設定されていません	∥ 設定されていません	なし	
	<						>

図 4-61 端子設定を XML ファイルヘエクスポートする

端子ページの [.csv ファイルにリストを保存] 🌇 ボタンをクリックすることで、端子設定を CSV 形式で エクスポートします。

4.5.6 端子設定のインポート

現在のプロジェクトに端子設定をインポートするには、[ボードの設定をインポート] 🚵 ボタンをクリック し、端子設定を含む XML ファイルを選択してください。設定がプロジェクトにインポートされると、この ファイルに指定された設定は、端子設定ページに反映されます。

子設定							🔽 🖆 コードの生成 レポートの
トウェアコンポ 🕀 🖻 📲	端子機能						- 🕂 🖪 🖪 🔂
ィルタ文字列を入力	フィルタ入さ	ኃ (* = any str	ing, ? = any cha	racter)			すべて
🗸 💑 r_bsp	使用する	機能	PIOR	端子割り当て	端子番号	方向	備考
🔋 r_bsp		INTPO		P137/EI137/INTP0	/ 29	1	この端子を初期化す
・ 🏧 割り込みコントローラ		INTP1	PIOR5	P46/INTP1/TI05/TO05	/ 16	1	
Config_INTC		INTP2		✓ 設定されていません	✓ 設定されていません	なし	
A/Dコンバータ		INTP3	PIOR5		∥ 設定されていません	なし	
🍒 インターバル・タイマ		INTP4	PIOR5		∥ 設定されていません	なし	
		INTP5	PIOR4		✓ 設定されていません	なし	
	<						>

図 4-62 端子設定を XML ファイルからインポートする

【注】 端子設定は反映されますが、コンポーネント設定には反映されません。



4.5.7 ボード端子情報を使用した端子設定

ボードの初期端子設定を一括で行えます。端子を一括で設定するには、以下の手順で行います。

- (1) [ボード] ページで、 [カスタムユーザーボード] 以外のボード設定情報を選択します。
 (4.1.2 ボード選択参照)
- (2) MCU/MPU パッケージで [ボード機能] を選択します。 (ボードの初期端子設定が参照できます)
- (3) [端子設定] ページを開き、[ボードの初期端子割り当ての設定] 🛃 ボタンをクリックします。
- (4) [ボードの初期端子割り当ての設定] ダイアログが開いたら [すべて選択] をクリックしてください。
- (5) [OK] ボタンをクリックします。



図 4-63 ボードの初期端子設定

端子設定を一度に設定しない場合は、手順(4)で個別に設定してください。

4.5.8 端子のフィルタ機能

端子ページの [端子機能] タブ、[端子番号] タブでフィルタ範囲を指定し、より簡単に端子設定を参照する ことができます。

ソフトウェアコンポーネント 🕀 📄 📫	z 🖧	端子機能						🥺 🔢 📑 🗠 🖂
フィルタ文字列を入力		フィルタ入力	(* = any string	, ? = any charac	ter)			すべて 〜
✓ 売 r_bsp ● r_bsp ↓ 売 割扱みコントローラ	^	使用する	機能	PIOR	端子割り当て	端子番号	方向	備 機能 PIOR
Config_INTC	~	<						端子割り当(端子番号 方向
端子機能 端子番号 概要 ボード クロック システム コンポーネント 🗴	岩子 割り	り込み						備考 コメント

図 4-64 端子機能のフィルタ

端子	番号							📕 📑 🔛	4
71)	ブルタ入力 (* = any string, ? = any character)								
端	子番号	端子名	ボード機能	機能	方向	備考	シンボリック名	すべて 端子番号	
	113	P24/ANI4/TS22	ANI4	ANI4	1	この端子を初期化するコン		端子名	1
	114	P23/ANI3/ANO1/IVREF0/TS21/EI23		設定されていません	なし			ボード機能	1
<								() () () () () () () () () () () () () (1
端子橋	吊子機能 端子番号								
概要:	ボード クロ	ック システム コンポーネント 端子 割り込み						ー・シンホリック名 コメント	

図 4-65 端子番号のフィルタ



4.5.9 端子エラー/警告の設定

[端子エラー/警告]設定を使用して、コンフィグレーションチェックビューの端子設定問題の表示方法を 制御できます。制御を変更したい場合は、[新規コンポーネント]ダイアログで [基本設定] リンクをクリッ クし、[設定]ダイアログを表示します。次に、[スマート・コンフィグレータ]の[端子エラー/警告]を選択 し、設定を変更します。



図 4-66 端子エラー/警告の設定

例:[ソフトウェアコンポーネント]の設定を「情報」から「エラー」に変更。

	▼ ソフトウェア 期以来です。	コンポーネント りた#FZ機能を	市田ナスソフトウィ	ירק	ポーネントがたい	Fron	
		- -	EW3 0 / / / / 1	., .,			
💮 *Smart_Config	gurator_Example.scfg ×				- 0	「 」 MCU/MPUパッケージ ×	- 0
端子設定				™ ⊐−ドの5		🔚 🔺 🔎 🏓 🖻 Райузтала 🖂 айузтала.	•
端子番号				I	8 1 2 2		-
I				すり	۲ζ γ		
第子書号 110 111 112 113 114 115 116 € 117 118 119 120 121 122	 端子名 P27/ANI7/TS25 P26/ANI6/TS24 P25/ANI5/TS23 P24/ANI4/TS22 P23/ANI5/TS23 P22/ANI2/ANO0/TS20/EI22 P22/ANI2/ANO0/TS20/EI22 P21/ANI1/AVREFP/EI20 P130 P102/TI06/TO06 P07 P04/SCK10/SCL10 P03/ANI16/TS29/SI10/RxD1/SDA10 	ボード機能 ANI7 ANI6 ANI5 ANI4 P07 P04 RxD1	機能 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません いません いません いません 設定されていません 設定されていません 設定されていません 設定されていません 設定されていません	方 なななななななな 向 し ななななななな し ななななな な し	備考 へ この端子	RENESAS RI78G23 R7F100GSNxFB	
端子機能 端子 概要 ボード クロッ	番号 ック システム コンボーネント <u> 端子 </u> 割り込み ョンチェック ×] ँः ▶ 凡// ▽ 8 °	- 0
1 error, 0 warning	s, 0 others	~					
記述/説明 ✔ ⑧ 端子 (1 項 ◎ 1050000	目) 012: ANIOを使用するコンポーネントのコンフィグレ	ーションがありません。生き	丸コ−ド以外で端子の初期	月化を行う	場合は、このメッセージを	型 無視してください。 端子	

図 4-67 ソフトウェアコンポーネントのエラー



4.6 割り込み設定

[割り込み] ページには、各ベクタ番号によりすべての割り込みが表示され、 [コンポーネント] ページで 選択した周辺モジュールの割り込みを確認・設定できます。 コード生成タイプのコンポーネントで割り込 みを使用すると、割り込みの状態が「使用中」に変わります。

- (1) 使用中の割り込みのみを表示するには、[図(設定した割り込みの表示)]ボタンをクリックしてください。
- (2) グループ割り込みは、割り込みテーブルでは折りたたまれます。グループ割り込みリストの割り込みを見るには、[展開] ≫ ボタンをクリックしてください。

別り込み設定							で コードの生成 レ	<u>し</u> ポートの生
設定済み割り込る	バクタ						((1)
フィルタ文字	列を入力				ベクタ番号			~
ベクタ番号	ベクトルテーブルアドレス	割り込み	割り込み要求元	周辺機能	優先レベル	状態	バンク指定	備考
3	0000AH	INTP1	Pin input edge detection	INTC	レベル3(低優先順位)	使用中	なし	
2) > 18	00028H	INTSRE1/INTTM03H			レベル3(低優先順位)	使用中	なし	
24	00034H	INTAD	End of A/D conversion	ADC	レベル3(低優先順位)	使用中	なし	
4								>

図 4-68 割り込みページ

4.6.1 割り込み優先レベルの設定

[割り込み] ページの割り込み優先レベルは、以下の手順で変更できます。

- (1) 優先レベルを変更したい割り込みを表示します。
- (2) 優先レベルセルをクリックし、ドロップダウンリストから割り込み優先レベルを設定します。

🔅 *Smart_Configurat	or_Example.scfg \times							
割り込み設定							いしています。 し コードの生成 し	 √ポートの生成
設定済み割り込みべ	79							8
フィルタ文字列を	入力			4	ベクタ番号			~
ベクタ番号	ベクトルテーブルアドレス	割り込み	割り込み要求元	周辺機能	優先レベル	状態	バンク指定	備考
3	0000AH	INTP1	Pin input edge detection	INTC	レベル3(低優先順位)	使用中	なし	
> 18	00028H	INTSRE1/INTTM03H		(2)	レベル3(低優先順位)	使用中	なし	
24	00034H	INTAD	End of A/D conversion	ADC	レベル3(低優先順位) ∨	使用中	なし	
					レベル0(高優先順位)			
					UA111			
<					レベル2 レベル3(低優先順位)			>
磯要 ホート クロック ジ	システム コンボーネント 端子 割	り込み						

図 4-69 割り込み優先レベル設定



4.6.2 割り込みバンクの設定

[割り込み] ページの割り込みバンクは、以下の手順で変更できます。

- (1) バンクを変更したい割り込みを表示します。
- (2) バンク指定セルをクリックし、ドロップダウンリストからバンク(None / 1 / 2 / 3)を設定します。
- (3) 優先度の異なる複数の割り込みに対して同じバンクを指定すると警告マークが表示され、[備考] と [Configuration Problems] に警告メッセージが表示されます。必要に応じて再設定してください。

2.112											
ィルタ文字列	を入力						(1)		ベクタ番号		
の番号	ベクトルテーブルアドレス	割り込み	割り込み要求元	周辺機能	優先レベル	状態	バンク指定	備考	(3)		
0	00004H	INTWDTI	Watchdog timer interval	WDT	レベル0(高優先順位)	使用中	なし,	1	(3)		
1	00006H	INTLVI	Voltage detection	LVD	レベル1	使用中	A 1	優先度の異なる複数	の割り込み機能に対し	て、1つのレジスタバンク	を指定することはできません
2	00008H	INTPO	Pin input edge detection	INTC	レベル2	使用中	14U				
3	0000AH	INTP1	Pin input edge detection	INTC	レベル3(低優先順位)	使用中	なし				
4	0000CH	INTP2	Pin input edge detection	INTC	レベル3(低優先順位)	使用中	A 1	優先度の異なる複数	の割り込み機能に対し	て、1つのレジスタバンク	を指定することはできません
5	0000EH	INTP3	Pin input edge detection	INTC	レベル3(低優先順位)	使用中	741.				
6	00010H	INTP4	Pin input edge detection	INTC	レベル3(低優先順位)	使用中	なし 🗸				
7	00012H	INTP5	Pin input edge detection	INTC	レベル3(低優先順位)	(2)	なし				
8	00014H	INTST2/INT			レベル3(低優先順位)	(2)	1				
9	00016H	INTSR2/INT			レベル3(低優先順位)		2				
10	00018H	INTSRE2/IN			レベル3(低優先順位)		1	/			

図 4-70 割り込みバンク設定

割り込みバンクの設定は、コンポーネントの{*ConfigurationName*}_user.c ファイルに生成されたコードに反映されます。



図 4-71 割り込みバンクコード (IAR プロジェクト)

実際に生成されるコード仕様は、コンパイラによって異なります。 詳細については、対応する IDE の ユーザーガイドを参照してください。

5. 競合の管理

コンポーネントの追加、端子や割り込みの設定をすると、リソースの競合に関する問題が起こる可能性が あります。この情報はコンフィグレーションチェックビューに表示されます。表示された情報を参照して、 競合問題を解決してください。なお、競合が発生していてもコードは生成できます。

5.1 リソースの競合

コンフィグレーションで、二つのソフトウェア・コンポーネントが同じリソース(例: ADC)を使うよう に設定した場合、コンポーネント・ツリーにエラーマーク 🛚 が表示されます。

コンフィグレーションチェックビューに周辺機能の競合に関するメッセージが表示され、周辺機能に競合が 見つかったソフトウェア設定をユーザーに知らせます。

🔅 *Smart_Configurator_Example.scfg 🗙				- 8
ソフトウェアコンポーネント設定) コードの	🕒 生成 レポートの生成
コンポーネント 🚵 🖾 🖓 🖂 🕀 設	定			^
 ・ ・	□ンパレ-タ動作設定 ● 停止 ○許可			
	分解能設定 ● 10ピット ● 8ピット VREF(+) 設定		() 12ピット	~
概要 ボード クロック システム コンボーネント 端子 割り	込み			>
6 errors, 0 warnings, 0 others				0 0 -
記述/説明	^	型		
 ◆ ③ 割り込み(2項目) ⑥ E04010005: INTAD (Config_ADC_Duplicate) ⑧ E04010005: INTAD (Config_ADCで設定)と次 ◆ ⑧ 周辺機能(2項目) 	で設定) と次の割り込みのベクタが競合しています : INTAD (Config_A この割り込みのベクタが競合しています : INTAD (Config_ADC_Duplica	DCで設定) . 割 ateで設定) . 割	り込み り込み	
 ◎ E04010001:周辺機能 A/Dコンパータ (Config. ◎ E04010001:周辺機能 A/Dコンパータ (Config. 	ADC_Duplicateで設定) は次の設定で使用されています : Config_AE ADCで設定) は次の設定で使用されています : Config_ADC_Duplica	DC. 周 te. 周	辺機能 辺機能	
 ※ ● 地子(2 項目) ⑥ E04010003: ANI0 (Config_ADC_Duplicateで ⑧ E04010003: ANI0 (Config_ADCで設定)が使 	設定) が使用する端子と次の端子が競合しています : ANIO (Config_A 用する端子と次の端子が競合しています : ANIO (Config_ADC_Duplic	ADCで設定). 端 rateで設定). 端	子 子	

図 5-1 リソースの競合

5.2 端子の競合

端子の競合がある場合、エラーマーク 🔕 がツリーと端子機能リストに表示されます。

*Smart_Configurator_Example.scfg	×								- 0
端子設定								で <mark>い</mark> コードの生成 レポ・	 トの生成
ארבעזבעזייי די בעזייייייייייייייייייייייייייייייייייי	端子機能							2 🔳 📑	èс
フィルタ文字列を入力	フィルタ入力) (* = any string	, ? = any chara	acter)				すべて	\sim
> 📩 r bsp	使用する	機能	PIOR	端子割り当て	端子番号	方向	備考		^
		🐼 ANIO		P20/ANI0/AVREFP/EI20	/ 117	1	複数の端子機能が同一端・	子に割り当てられていま	र जे
or Config_ADC		ANI1		🦉 設定されていません	🖉 設定されていません	なし			
Y 42 /1−1 - /1 - /1 - DODT		ANI2			/ 設定されていません	なし			
@ Config_PORT		ANI3			/ 設定されていません	なし			~
	<								>
·····································									
端子機能 师丁曲节									
概要 ボード クロック システム コンポーネン	小 端子 割り	込み							

図 5-2 端子の競合



競合情報の詳細は、コンフィグレーションチェックビューに表示されます。

🔝 Configuration Problems 🕴	* □
3 errors, 0 warnings, 0 others	
記述/説明	タイプ
→ 😢 端子 (3 項目)	
😵 E04010003: ANIO (Config_ADCで設定) が使用する端子と次の端子が競合しています : P20 (Config_PORTで設定)、 P20 (Pin Allocatorで設定).	端子
🧕 E04010003: P20 (Config_PORTで設定) が使用する端子と次の端子が競合しています : ANI0 (Pin Allocatorで設定)、ANI0 (Config_ADCで設定).	端子
🔞 E05000010: 端子 117を複数の機能で使用できません。端子 117 に P20, ANIO の機能が割り当てられています。	端子

図 5-3 端子競合のメッセージ

エラーマークのあるツリーノードを右クリックし、 [競合の解決]を選択して競合を解決してください。

יזרלידעריעיעיעיעיעיעיעיעיעיעיעיע פּאַזעיעיעיע עזעיעע	端子機能	
フィルタ文字列を入力	フィルタ入力) (* = any stri
> ▲ r_bsp マ ▲ A/Dコンパータ ② Config_ADC マ ▲ ポート ③ Confia_PORT すべて割り当て解	使用する 夕 し 除	機能 ② P20 CCD00 CCD01 CCD02 CCD03 CCD04
競合の解決		
端子機能 端子番号 概要 ボード クロック システム コンポーネン	小端子割	り込み

図 5-4 端子競合の解決

選択されたノードの機能は、他の端子に再度割り当てられます。



6. ソースの生成

ソースの生成は、コンフィグレーションチェックビューで競合が発生していても生成できます。

6.1 ソースファイルの生成

スマート・コンフィグレータビューの [コードの生成] - Forest ボタンをクリックすると、設定した内容に応じたソースファイルを出力します。

Image: Smart_Configurator_Example.scfg ×		- 0
ソフトウェアコンポーネント設定	じ コードの生成	■ レポートの生成

図 6-1 ソースファイルの生成

スマート・コンフィグレータは、¥<PeojectDir>¥src¥smc_gen にソースファイルを生成し、IAR 関連ファ イルをコンフィグファイルの保存場所(3.3.1 新規作成)に生成します。

既にファイルを生成している場合は、バックアップコピーを生成(9.生成ソースのバックアップ)します。



6.2 生成ファイルの構成とファイル名

スマート・コンフィグレータが出力するフォルダとファイルを図 6-2 生成ファイルの構成とファイル名 に示します。なお、main()関数は IAREW でプロジェクト作成時に生成する {*Project name*}.c に含まれます。

"ConfigName"はコンポーネント設定で設定したコンフィグレーション名、"Project name"は IAREW で設定したプロジェクト名を示します。



図 6-2 生成ファイルの構成とファイル名



フォルダ	ファイル	説明
{ProjectName}	{ <i>ProjectName</i> }.eww	このファイルは、最初のコード生成で1回だけ生成されま
		す。{ <i>ProjectName</i> }.ewp ファイルパスは、このファイルで指定
		されます。
	{ProjectName}.ewp	このファイルは、最初のコード生成で1回だけ生成されま
		す。このファイルの最後に buildinfo.ipcf ファイルと main.c
		ファイルを追加します。
	{ <i>ProjectName</i> }.ewd	このファイルは、最初のコード生成で1回だけ生成されま
		す。IAR Embedded Workbench によって生成されるデフォル
		トの*.ewd ファイルとまったく同じです。
	main.c	このファイルは、最初のコード生成で1回だけ生成されま
		す。main() 関数が含まれています。
	buildinfo.ipcf	このファイルは常に生成され、ソースファイルの登録情報が
		含まれています。RL78 スマート・コンフィグレータ V1.4.0
		から*.ipcf ファイル名前は、buildinfo.ipcf に変更されていま
		す。V1.4.0より前に作成されたプロジェクトを読み込むと、
		.ipcf ファイルは buildinfo.ipcf として再生成され、元の.ipcf
		ファイル ({ <i>ProjectName</i> }.ipcf) はフォルダに残ります。
general		このフォルダは常時生成されます。同じ周辺機能の CG ドラ
		イハで共通に使用される、ヘッタファイルとソースファイル
		を含みます。
	r_cg_ <i>xxx</i> .h ^(*1)	このファイルは SFR レジスタを設定するためのマクロ定義を
		含みます。
	r_smc_entry.h	このファイルは常時生成されます。このファイルには、ブロ
		ジェクトに追加される CG トライハのヘッタファイルが含ま
		れよす。ユーサーか追加するソースノアイルで UG トライハ の開始た住田士で担合。このコーズリのイン ちょうじがぶま
		の関数を使用する場合、このファイルのイングルートが必要
		じり。
	r_cg_macrodriver.n	このファイルは吊时生成されます。このヘッダファイルは、
	n en veerdefine b	トライバと使用される矢通のマラロ定我を含めより。
	r_cg_userdeline.n	ビのファイルは吊時生成されます。ユーリーは、専用のユー
	r og oveteminit e	「 「 」 「 限域に ()」 に 定報 と 追加 りることが (とより。
	r_cg_systemmit.c	Croste()関数を今みます。目辺機能の初期化に使用します
	1_cg_xxx_continion_user.c	を会みます
		このファイルけ対応する周辺機能を使用する場合に生成され
		このシアイルは対応する周辺限能を使用する場合に工成され ます。
	r og vyv common $h^{(*1)}$	このファイルけ対応する周辺機能を使用する場合に生成され
		このシットルは対応する周辺版化を使用する場合に工成され
r bsp		
1_000		フォルダ(board, doc, mcu)から構成されます。
		 main()実行前に MCU を記動する初期化コード(例・ス)
		タックのセットアップ、メモリの初期化)
		 iodefine.h (mcu フォルダ) にあるすべての SFR レジスタ
		の定義
		• r bsp のアプリケーションノート (doc フォルダ)
		プロジェクトで使用されるデバイスの r hsp h を含む
		platform.h もこのフォルダに生成されます。
general r_bsp	main.c buildinfo.ipcf r_cg_xxx.h ^(*1) r_smc_entry.h r_cg_macrodriver.h r_cg_userdefine.h r_cg_systeminit.c r_cg_xxx_common_user.c ^(*1) r_cg_xxx_common.c ^(*1)	 トの*.ewd ファイルとまったく同じです。 このファイルは、最初のコード生成で1回だけ生成されます。main()関数が含まれています。 このファイルは常に生成され、ソースファイルの登録情報が含まれています。RL78 スマート・コンフィグレータ V1.4.0 から*.ipcf ファイル名前は、buildinfo.ipcf に変更されています。V1.4.0 より前に作成されたプロジェクトを読み込むと、*.ipcf ファイルは buildinfo.ipcf として再生成され、元の*.ipcf ファイル (<i>lProjectName</i>).ipcf)はフォルダに残ります。 このファイルは常時生成されます。同じ周辺機能の CG ドライバで共通に使用される、ヘッダファイルとソースファイル を含みます。 このファイルは常時生成されます。同じ周辺機能の CG ドライバで共通に使用される、ヘッダファイルとソースファイル を含みます。 このファイルは常時生成されます。このファイルには、プロジェクトに追加される CG ドライバのヘッダファイルが含まれます。ユーザーが追加するソースファイルで CG ドライバの関数を使用する場合、このファイルのインクルードが必要です。 このファイルは常時生成されます。このヘッダファイルがるまれます。ユーザーが追加するソースファイルのインクルードが必要です。 このファイルは常時生成されます。このヘッダファイルにな、ドライバで使用される共通のマクロ定義を含みます。 このファイルは常時生成されます。全コンポーネントのCreate()関数を含みます。周辺機能の初期化に使用します。 このファイルは対応する周辺機能を使用する場合に生成されます。 このファイルは対応する周辺機能を使用する場合に生成されます。 このファイルは対応する周辺機能を使用する場合に生成されます。 このファイルは対応する周辺機能を使用する場合に生成されます。 このファイルに対応する周辺機能を使用する場合に生成されます。 このファイルに対応する周辺機能を使用する場合に生成されます。 このファイルに対応する周辺機能を使用する場合に生成されます。 このファイルに対応する周辺機能を使用する場合に生成されます。 このファイルに対応する周辺機能を使用する場合に生成されます。 ・ main()実行前に MCU を起動する初期化コード (例:スタックのセットアップ、メモリの初期化) ・ ioddfine.h (mcu フォルダ)にあるすべての SFR レジスの定義 ・ r_bsp のアプリケーションノート (<i>doc</i> フォルダ) プロジェクトで使用されるデバイスの r_bsp.h を含む、platform.h もこのフォルダに生成されます。

フォルダ	ファイル	説明
r_xxx/		このフォルダは、プロジェクトに追加された RL78 Software
rm_ <i>xxx</i> ^(*1)		Integration System モジュール用に生成されます。
		• doc フォルダ:アプリケーションノート
		・ r_xxx.c/rm_xxx.c ^{(*1) :} ソースファイル
		・ r_xxx.c/rm_ <i>xxx</i> .h ^{(*1) :} ヘッダファイル
		r_xxx_api.h/rm_xxx_api.h ^{(*1) :} すべての API とインターフェー
		ス定義のリスト
r_config		このフォルダは常時生成されます。MCU パッケージ、クロッ
-		ク、割り込み等のコンフィグレーションヘッダファイルを含
		みます。
	r_bsp_config.h	このファイルは常時生成されます。クロック初期化と他の
		MCU に関連する r_bsp の設定を含みます。いくつかの MCU
		関連の設定はスマート・コンフィグレータが生成し(例:
		パッケージタイプ)、他の設定(例:スタックサイズ)は
		ユーザーが手動で設定します。
	r_bsp_config.inc	このファイルは常時生成されます。
		構成ヘッダファイルを生成します。
	r_xxx_config.h/rm_xxx_config.h	このファイルは、プロジェクトに追加されるすべての RL78
	(*1)	Software Integration System ドライバ/ミドルウェアの構成
		ヘッダファイルです。
r_pincfg	Pin.h	このファイルは常時生成されます。
		端子のシンボルをサポートするために生成され、smc_entry.h
		に含まれています。
	Pin.c	このファイルは常時生成されます。
		[端子] ページで有効になっている端子設定が生成されます。
		PIOR を設定する必要のない端子設定を生成するだけです。
	r_ <i>xxx</i> _pinset.c	このファイルは、RL78 Software Integration System の端子設
		定ソースファイルです。
	r_ <i>xxx</i> _pinset.h	このファイルは、RL78 Software Integration System の端子設
		定ヘッダファイルです。
{ConfigName}		このフォルダは、プロジェクトに追加される CG ドライバ用
		に生成されます。このフォルダ内の API 関数には、
		ConfigName(設定名)を含んた名称かつけられています。
	{ConfigName}.c	このファイルは、トライハを初期化する関数
		(R_ConfigName_Create)、トフイハに行有な探作、1例えば
		スタート(R_ConfigName_Start)やストッフ (P ConfigName Start)を実行する関数を会社ます
	(ConfigName) waara	(N_OOMINGNAME_SUP) で大口りの肉奴で占がより。 ドライバの初期化(P_ConfigName_Crosts)の後に泊加すて
	{Coniigivariie}_user.c	IFフィハの初期に(R_COMIGNAME_Cleate)の後に追加する
		ここができる討り込みり一レスルーテノと関数を召みまり。 コーザーけ、東田のコーザーコード領域にコードと開始を迫
		ユーッーは、守用のユーッーコート限域にコートと) (男など) 加することができます
	(ConfigMama) h	パーファマーにん、くてみり。
	{Configivame}.n	{Coningivanie}.c こ {Coningivanie}_user.c のヘッタファイル ぐ
		7 0

【注】 *1. xxx は周辺機能名を示します。

6.3 クロック設定

[クロック] ページにあるクロックソースの設定は、¥src¥smc_gen¥r_config フォルダにある r_bsp_config.h ファイルのマクロに生成されます。main()を実行する前に r_bsp によって、クロック初期化 コードは処理されます。r_bsp_config.h ファイルには、他の MCU 関連の設定(例:パッケージ、スタック サイズ)も含みます。

Image: 1000000000000000000000000000000000000	JYTEXAL			し コードの生 #	しポートの生成
Bit P - P:		R		21020	PART OF A
BHE-P: ##47-0-4-44.00/7-50/0 EXOURAT: 4007 E-V0006 2.55 V Statushey BHP BABE: 22 (MAR) (M42) (STOPE-W6006)-2.88 ±050002:E-W0.08 (FMLX.8 ±0.47) BHP (Stope-W600)-2.88 ±050002:E-W0.08 (FMLX.8 ±0.47) BHP (Stope-W600)-2.88 ±050002:E-W0.08 (FMLX.8 ±0.47) BHP (BLE) (BLE) (BLE) (BLE) (BLE) (BL					
EVODB 2: 4 0 V c EVODD c 55 V I Rait / 57/*5 / -5 Rait / 5 Rait / 57/*5 / -5 Rait / 50/000000000000000000000000000000000	動作モード:	高速メイン・モード4.0(V)~5.5(V)	•		
Image:	EVDD設定:	4.0 V ≤ EVDD0 ≤ 5.5 V	•		
Ridk: 2 (M42) HUCO MRNBYT: WX STOPE-MPS-000-XMBJTS/SOUCEE-MA.0B (THECK # 27.57) Ridk Stafet 3 2.60.08 ZMA045 x Y H #27.57.752-7 Ridk Y H #27.57.752-7 Strip # Ridk Y H #27.57.752-7 Y H #27.57.752-7 <t< td=""><td>✓ 高速オンチゥブ・オシ</td><td>/~h</td><td></td><td></td><td></td></t<>	✓ 高速オンチゥブ・オシ	/~h			
HOCO MH&X: Int (STOPE- MP4 AUF SNO2ZE- K- K- 0F FM C.R. if 9/-7 NR # 528 / 50-00 & 21 / 50 / 50 / 10 / 10 / 10 / 10 / 10 / 1	周波数:	32 - (MHz)		8115	
(5)TOPE-EMPARTUR-TRABATO SNOOZEE-PA-08 FS9(Z-F57) REAL (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842) (842)<	HOCO 開始設定:	通常 👻	•	32	(MHz)
XHR EXERT CONSIGNER () Y + EXT X-50* 242 M/ALTE - HMA MILLITE -	(STOPモードからのリリー	ス時およびSNOOZEモードへの移行時に高速オンチップ		fMAIN	·
C.LK W Pikt/>1/50/15/L Bits: 4 MP A MP MP A MP	死振器を起動するための	設定があります。)		32	(MHz)
Image: 0,000 (042) BMP 0,000 0,000 BMP 0,000 0,000 Image: 1 0 BASE: 5 0,000 BASE: 5 0,000 BASE: 2,008 0,000 StripsEgas 0,000 0,000 Mitest=F: MIRESE 0,000 Mitest=F: Size 0,000				fclk	
W 中進7547/15/U-15 周波音: 4 (MHz) 5 (MHz) 5 (MHz) 8 5 (MHz) 9 5 (Mz) <t< td=""><td></td><td></td><td></td><td>32000 EMP</td><td>(kHz)</td></t<>				32000 EMP	(kHz)
萬法: 4 ● (MHz) 9月前 9月前 第1000 9月前 第1000 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	✔ 中連オンチゥブ・オシ	レータ		4	(MHz)
>>RBB \$>RBB \$\$\P\$RB \$\$\$\P\$RB \$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	周波数:	4 ~ (MHz)			
■ XIR#828 ■ MO(P ФЛФ - (MHz) ■ 21 ● ■ 21 ● ■ 21 ● ■ 21 ● ■ 21 ● ■ 21 ● ■ 21 ● ■ 21 ● ■ 21 ● ■ 21 ● ■ 21 ● ■ 21 ● ■ 22 ● ■ 22 ● ■ 22 ● ■ 22 ● ■ 22 ● ■ 22 ● ■ 22 ● ■ 22 ● ■ 22 ● ■ 22 ● ■ 22 ● ■ 22 ● ■ 22 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●			分用語		
#hfter-P: INFA #dista: 5 #dista: 5 #dista: 218/2 #dista: 22768 (Hu) 1 #dista: 22768 (Hu) 1 #dista: 22768 (Hu) 5 #dista: 22768 (Hu) 5 #dista: 22768 (Hu) 5 #dista: 22768 #dista: 500 #dista: *	X1発振回路		x1 *	fMXP	
用进数: 5 (MHz) 和形型工作标: 22.768 (MHz) 和水型工作不可不可不可不可不可不可不可不可不可不可不可不可不可不可不可不可不可不可不可	動作モード:	X1発振 👻			(MHz)
R 居安 空時編: 218/k → 52428.8 (m)	周波数:	5 (MHz)			
	発振安定時間:	2^18/fx = 52428.8(µs)			
第連型ンチウブ・ヨシレータ 周点意: 22.768 (H4z) 周点意: 22.768 (H4z) 日、意用得口クロッチドウブ・ケイブが動作。または「SXFP) 「B温度見作」 「 XTI現居自発 動作モード: XTI提后 (Hz) 「 XTI現居モード: 所温度見解1 中間の供給計可 (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (Hz) (fiL	
周進第:: 32.768 (4Hz) 間の参加偏相ななデドッジ・イマが参引、参えはSS/Pがを達まプチラフ・オン → 5を選択 動作モード: XT1現后 動作モード: 数17度用日 一 22.768 (4Hz) XT1現用モード: 然:温泉風俗1 	低速オンチップ・オシレータ		•	32.768	(kHz)
Internet 2007 FOR VITABLE FOR CONSULT OF VITABLE FOR CONSUL FOR CONSULT OF VITABLE FOR CON	周波数:	32.768 (kHz)			
✓ XT1現局器路 FSXR Bh年モ-ド: XT1現局 電波数: 32.768 (bHz) 32.768	レータを選択	ドップ・ショイル 動作を、新宅は13人にから3度なノナップ・なン		fSXP 32 768	(kHz)
▼ XT1%用回路 5XR 動作モード: XT1%用「「 関志策: 32.768 (bHz) XT1%用モード: 他注意知用「 (Háb モード: STOP,HALTE-ド時の供給許可					
動作モード: XTI発展 ~ 型波数: 32.768 (kHz) XTI発展モード: 他I注度発展1 ~ 供給モード: STOP,HALTE-ド時の供給許可 ~	✓ XT1発振回路			fSXR	(44-)
展進新: 32.768 0(Hz) XTT現毎モード: 他活動発展指 供給モード: STOP,HALTモード時の供給許可	動作モード:	XT1発掘		32.766	(M12)
XTT探察モ~ド: 他指後現解1 ~ 供店モ~ド: STOP,HALTE-ド時の供給許可 ~	周波数:	32.768 (kHz)			
#48モード: STOP,HALTE~ド時の供給許可 ◆ ・	XT1発掘モード:	低消费冤赔1			
•	供給モード:	STOP,HALTモード時の供給許可	•		
					• •

図 6-3 クロック設定と r_bsp_config.h に生成されるコード

フォルダ	ファイル	マクロ/関数	説明
r_config	r_bsp_config.h	クロックに関連する	これらの設定は、クロックソースのクロックページにある
		マクロ	ユーザーの選択を基に、スマート・コンフィグレータに
			よって生成されます。 <i>main()</i> を実行する前に、r_bsp はク
			ロックの初期化を処理します。
		MCU 設定に関連する	MCU 関連の設定は、スマート・コンフィグレータによっ
		マクロ	てマクロが生成されます(例:パッケージタイプ)。マク
			ロの詳細は、r_bsp フォルダのアプリケーションノート
			(¥src¥smc_gen¥r_bsp¥doc)を参照してください。

【注】コードの生成実行前の r_bsp_config.h は trash フォルダにバックアップされます。 (9. 生成ソースのバックアップ)



6.4 端子設定

[端子] ページの設定は、コンポーネントにより下記に示すソースファイルに生成されます。

1) <u>{ConfigName}を使用したドライバの端子初期化</u>

端子機能は¥src¥smc_gen¥{*ConfigName*}¥{*ConfigName*}.cのR_*ConfigName*_Createで初期化されます。 端子初期化コードは、main()を実行する前に処理されます。

*Smart_Configurator_Example.sc	fg ×					ami	
端子設定						○ コードの生成	じました。 □ □ □
עראידעראיז 🕀 🖻 🛱 腸	端子機能					3	
フィルタ文字列を入力 フィルタ入力 (* = any string, ? = any character)							~
> 💑 r_bsp	使用する	機能	PIOR	端子割り当て	端子番号	方向	備考
✓ 売 万形波出刀		TO01		P16/EO16/CCD00/TI01/TO01/INTP5/SI00/RxD0	/ 76	0	
	<						2
子機能 端子番号							
要 ボード クロック システム コンポー	ネント 端子 書	り込み					

図 6-4 Config_TAU0_1 の端子設定

フォルダ	ファイル	関数	ドライバ	説明
{ConfigName}	{ConfigName}.c	R_ConfigName_Create	CG	このドライバが使用する端子を API 関数が初期化します。main() 関数を 実行する前に、r_cg_systeminit はこ の関数を呼び出します。

2) <u>RL78 Software Integration System コンポーネントの端子初期化</u>

端子機能は¥src¥smc_gen¥r_pincfg¥{*ConfigName*}_pinset.c の R_{*PeripheralName*}_PinSetInit で初期 化されます。ユーザーは、main()で端子初期化コードを呼び出します。

續 *Smart_Configurator_Example.scf	g ×							
端子設定						で コードの生成	し レポート) の生成
עראידערע 🗈 🖻 🛱	端子機能					3	I 2	14
フィルタ文字列を入力	フィルタ入さ] (* = any	string, ? = any	/ character)		すべて		\sim
> 🚋 r_bsp	使用する	機能	PIOR	端子割り当て	端子番号	方向	備考	^
→ <u> </u>		TS00		P50/TS00/EI50/EO50/CCD03	/ 67	IO		
💓 r_ctsu		TS01		P31/TS01/EI31/TI03/TO03/INTP4/PCLBUZ0	/ 41	IO		~
	<						3	
端子機能 端子番号 概要 ボード クロック システム コンボー:	ネント 端子 害	りり込み						

図 6-5 r_ctsu の端子設定

フォルダ	ファイル	関数	ドライバ	説明
r_pincfg	{ConfigName} _pinset.c	R_{ <i>PeripheralName</i> } _PinSetInit	RL78 Software Integration System	このドライバが使用する端子を API 関数が初期化します。ユーザーは、 main() 関数でこの関数を呼び出す必 要があります。



6.5 割り込み設定

[割り込み] ページの設定は、いくつかのソースファイルに生成されます。

割り込み関数は、¥src¥smc_gen¥{*ConfigName*}¥{*ConfigName*}.c ファイルの R_*ConfigName*_Create で初期 化されます。

の込み設定							い コードの生成 レ)ポートの生
り込みベクタ								2
					ベクタ番号			~
ベクタ番号	ベクトルテーブルアドレス	割り込み	割り込み要求元	周辺機能	優先レベル	状態	バンク指定	^
0	00004H	INTWDTI	Watchdog timer interval	WDT	レベル3(低優先順位)	使用中	なし	
1	00006H	INTLVI	Voltage detection	LVD	レベル3(低優先順位)	使用中	なし	
2	00008H	INTPO	Pin input edge detection	INTC	レベル3(低優先順位)	使用中	1	
3	0000AH	INTP1	Pin input edge detection	INTC	レベル3(低優先順位)	使用中	なし	~
c								>

図 6-1 割り込み設定

項目	フォルダ	ファイル	ドライバ	説明
優先レベル	{ConfigName}	{ConfigName}.c	CG	優先レベルは、このファイルの R_ <i>ConfigName</i> _Create で初期化されま す。main() 関数を実行する前に、 r_cg_systeminit はこの関数を呼び出しま す。
バンク指定	{ConfigName}	{ConfigName}_user.c	CG	割り込みを次のように宣言します。 #pragma interrupt "Interrupt API Name"(vect="Interrupt Name", bank=RBbankNumber) 図 4-71 割り込みバンクコード (IAR プロジェクト) を参照してください。



7. 統合開発環境への読み込み

スマート・コンフィグレータで出力したソースコードを統合開発環境プラットフォームに読み込みます。

7.1 IAR Embedded Workbench への読み込み

スマート・コンフィグレータは、使用するコンパイラに IAR 環境を選択したとき、ソースファイルと共に IAR Embedded Workbench 関連ファイル (*.eww, *.ewp. *.ewd, main.c) を出力します。IAR Embedded Workbench でプロジェクト ファイルを作成する必要はありません。

下記の手順で使用してください。

- (1) IAR Embedded Workbench の [ファイル] メニューから [ワークスペースを開く] を選択します。
- (2) [ワークスペースを開く] ダイアログボックスで、プロジェクトファイルが保存されているフォルダを 参照し、プロジェクトファイル (*.eww) を選択して [開く] ボタンをクリックします。

❷ ワークスペースを開く								×
\leftrightarrow \rightarrow \checkmark	→ PC → (C:) Wit	ndow	s > workspace > iar_ew >	~	Q	,○ iar_ewの検索		
整理 ▼ 新しいフ:	ォルダー							?
🖊 ダウンロード		^	名前 ^	更新日時		種類	サイズ	
📃 デスクトップ			.settings	2022/10/27 20:10	6	ファイル フォルダー		
🛗 ドキュメント			src	2022/10/27 20:10	6	ファイル フォルダー		
📰 ピクチャ			h trash	2022/10/27 20:10	6	ファイル フォルダー		
🔚 ビデオ			smart_configurator_example.eww	2022/10/27 20:10	6	IAR IDE Workspace		1 KB
🎝 ミュージック								
🖆 (C:) Windows								
		< <						>
	ファイル名(N): smart_configurator_example.eww ~			ワークスペースファイル(*.e	ww)	\sim		
						開く(O)	キャンセノ	ل ا

図 7-1 *.eww ファイルの読み込み

(3) スマート・コンフィグレータによって出力したソースファイルは、IAR ワークスペース/プロジェクトに追加されます。

ワークスペース	•	џ×
Debug		~
ファイル	¢	•
🗆 🌒 smart configurator example – Debug *	×	
- 🛱 📕 Renesas_SC		
└───		
- 🕀 🛑 Config_ADC		٠
🚽 🖬 r_cg_ad.h		
- 🕀 🗟 r_cg_ad_common.c		•
🛛 🚽 🗟 r_cg_ad_common.h		
🚽 🖬 r_cg_macrodriver.h		
🛛 🕂 🖸 r_cg_systeminit.c		•
🚽 🖬 r_cg_userdefine.h		
📃 🖵 🗟 r_smc_entry.h		
⊢Ģ ≡ r_bsp		
		٠
l —⊕ iii doc		
- 🛛 🖬 mcu		٠
📃 🔚 platform.h		
🖵 🗎 readme.txt		
└─────────────────────────────		
🚽 🖬 r_bsp_config.h		
L I'r_bsp_config.inc		
- buildinfo.ipcf		
H 🖸 main.c		٠
🖵 🔳 Output		
smart_configurator_example		

図 7-2 IAR ワークスペース/プロジェクトへのソースファイルの追加



- (4) IAR Embedded Workbench の [プロジェクト] メニューから [オプション] を選択します。
- (5) [ノード "ProjectName" のオプション] ダイアログボックスで、[ターゲット]タブのデバイスを対象デ バイスに変更します。

テゴリ:	1			
12オノンヨン 動り解析				
C/C++コンパイラ	Library Ontions 2	フタックループ	MISRA-C:200	4 MISRA_C:1008
アセンブラ Output Convertor	ターゲット	出力 ライ	(ブラリ設定	ライブラリオプション 1
カスタムビルド ビルドアクション リンカ デバッガ COM Port E1 E2 E20 E2 Lite / E2 On-board E2-CUBE E2-CUBE	デバイス RL78 core S3 - Un コードモデル: Near データモデル: Near	specified レート マン マン マン マン マン マン マン マン マン マン マン マン マン	 浮動小数点 タイプ doubl 32 ビット(: 64 ビット() 呼出し規約 Default 	数 e'のサイズ: 3) 5) ∽
シミュレータ TK	Near定数の配置 ロデフォルトのアドレ ミラーROM 0	スのオーバライド(O 🖟	引始アドレス: DxF3000	ナイズ(KB): 0xF9EFF

図 7-3 ターゲット・デバイスの変更

7.2 IAR プロジェクトのビルド

プロジェクト名を右クリックして [すべてを再ビルド] を選択すると、ビルドを開始します。



図 7-4 プロジェクトのビルド



8. ユーザープログラムの作成

ここでは、スマート・コンフィグレータが出力したソースファイルへのカスタムコード追加方法について 説明します。

8.1 コード生成タイプのカスタムコード追加方法

コンポーネント・タイプが [コード生成] の場合、ソースコード出力の際に、同一ファイルが存在する場合には、以下のコメントで囲まれた部分に限り、該当ファイルをマージします。

/* Start user code for xxxx. Do not edit comment generated here */

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

[コード生成] の場合、指定した周辺機能ごとに3つのファイルを生成します。デフォルトのファイル名 は、「Config_xxx.h」、「Config_xxx.c」、「Config_xxx_user.c」となり、xxx は周辺機能を表します。 (例えば、A/D コンバータ(リソース ADC)の場合、xxx は "ADC"と名付けられます。) カスタムコード を追加するためのコメントは、「*.c」ファイルの先頭と最後および「*.h」ファイルの最後に設けられる 他、「Config_xxx_user.c」にある周辺機能の割り込み関数内にも追加されます。以下に ADC の例 (Config_ADC_user.c)を示します。



/**************************************
Includes ************************************
#include "r_cg_macrodriver.h"
#include "r_cg_userdefine.h"
#include "Config_ADC.h"
/* Start user code for include. 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 */
Ŭ
/**************************************
* Function Name: R_Config_ADC_Create_UserInit
* Description: This function adds user code after initializing the AD converter.
* Arguments: None
* Return Value: None

<pre>void R_Config_ADC_Create_UserInit(void)</pre>
{
/* Start user code for user init. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */
}
/***********
* Function Name: r Config ADC interrupt
* Description: This function is INTAD interrupt service routine
* Arguments: None
* Paturn Value: None

#pragma vector – INTAD vect
interrupt static void r. Config. ADC interrupt(void)
/* Start user code for r. Config. ADC interrupt. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */
Line user code. Do not can comment generated here /
ſ
/* Start user code for adding. Do not edit comment generated here */
/* End user code. Do not adit comment generated here */
/ End user code. Do not cult comment generated here //



8.2 ユーザーアプリケーションコードの使用方法

RL78 Software Integration System Modules およびコード生成のコードを使用するには、以下の手順で行います。

(1) {*Project name*}.c ファイルを開き、使用するモジュールのヘッダファイルをインクルードコードに追加します。

RL78 Software Integration System Modules の場合は、r_xxx.h です。 コード生成の場合は、r_smc_entry.h です。

ワークスぺース ▼ # ×	main.c* x
Debug ~	fo
77-111 🌞 🔸	tinclude "r_smc_entry.h" tinclude "r smc_ctsu.h"
Smart_Configurator_Exampl ✓	int main(void)
H⊐ ■Renesas_SC	return 0;
Config_ADC	
H = general ●	
- I r_config	
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	
L⊞ ∎r_pincfg	
I → □ buildinto.ipct → ⊡ i i main.c	
Le Cutput	
Smart Configurator Example	
Smart_Connigurator_cxample	

図 8-1 ヘッダファイルの追加

(2) main() 関数で生成された関数を呼び出し、アプリケーションコードを追加します。

コード生成の場合、端子初期化を含むドライバ初期化関数 (R_ConfigName_Create) は、デフォルト で r_cg_systeminit.c の R_Systeminit() 関数で呼び出されます。ドライバ固有の処理を実行するに は、アプリケーションコードを追加する必要があります。

例えば、開始 (R_ConfigName_Start) と停止 (R_ConfigName_Stop) です。

ワークスペース	•	ч×	main.c* ×
Debug		~	
77-1 JJJ ■ Smart_Configurator_Exampl ■ Renessas_SC ■ Smc_gen = Config_ADC = Smc_gen = Smc	*	•	<pre>include "r_smc_entry.h" include "r_smc_ctsu.h" int main(void) { R_Config_ADC_Start(); return 0; }</pre>
Smart_Configurator_Example			٢

図 8-2 コード生成関数コール

RL78 Software Integration System Modules の場合は、対応するアプリケーションノートの「API 機能」の章に記載されている例を参照してください。

詳細については、「13. 参考ドキュメント」 のスマート・コンフィグレータのアプリケーション例を参照してください。
9. 生成ソースのバックアップ

スマート・コンフィグレータには、以下の場所にソースコードをバックアップする機能があります。

<ProjectDir>¥trash¥<Date-and-Time>

[コード生成] コード生成] ボタンをクリックしてコードの生成を行うと、スマート・コンフィグレータはコー ド生成前のソースのバックアップを作成します。<Date-and-Time>は、コード生成を実行しバックアップ フォルダを作成した日時です。



10. レポートの生成

スマート・コンフィグレータは、ユーザー設定のレポートを提供します。レポートを生成するには、以下 の手順で行います。

10.1 全設定内容レポート(PDF または txt 形式)

スマート・コンフィグレータビューで [レポートの生成] 🚅 ボタンをクリックし、レポートを出力し ます。

戀 *Smart_Con	figurator_Example.scfg ×					
概説		で コードの生成 レポートの生が	成			
▼ 機能概要		٢	^			
	概要 感要をクリックすると、スマート・コンフィグレータの機能を確認することができます。 動画	Application Code				
	<u>スマート・コンフィグレータの紹介</u> 関連動画	Software Components				
	最新情報 <u>最新情報</u> 支 <u>新情報</u> をクリックすると、最新リリースの情報を確認することができます。 すべての <u>リリースノート</u> を表示 現在のパージョン: V1.13.0	RTOS Device Drivers				
	製品ドキュメントとよくあるご質問(FAQ) ユーザーガイド APIマニュアル FAQ: <u>Smart Configurator</u>	MCU Hardware				
▼ 現在の設定	2状態					
注:コード生の 使用しているが	成の動作は、設定 ページでカスタマイズできます。 ① ボード/デバイス: R7F100GSNxFB (ROM size: 768KB, RAM size: 48KB, Pin count	: 128)				
生成先ロケー	ション (PROJECT_LOC¥): src¥smc_gen	編集	~			
概要 ボード ク	概要 ボード クロック システム コンボーネント 端子 割り込み					

図 10-1 全設定内容レポート出力(PDF または txt 形式)

	—		×
設定内容のレポートを出力します			
オプション			
● 全ての項目を出力する			
○ 選択した項目を出力する			
□ ボード			
□ クロック			
□ システム			
> ☑ コンポーネント			
> ■ 端子			
□割り込み			
✓PDFで出力		<u>フォントi</u>	<u> </u>
□ テキストで出力			
D:¥		参照	l
ОК		キャンセ	IL

図 10-2 レポート出力ダイアログ



10.2 端子機能リスト、端子番号リストの設定内容(csv 形式)

スマート・コンフィグレータビューの端子ページで [.csv ファイルにリストを保存] 🔜 ボタンをクリック し、表示中の端子リスト(端子機能リストまたは端子番号リスト)の設定内容を出力します。

子設定							▶ コードの生成	レポートの	」 D生
ードウェアリソース 🕀	🖻 🞝 🔓	端子機能					3 🗉		
フィルタ文字列を入力		フィルタ入さ	ኃ (* = any st	ring, ? = any cl	naracter)		すべて		`
🚣 すべて	^	使用する	機能	PIOR	端子割り当て	端子番号	方向	備考	1
\$∰ 1/0ポート			ANIO		P20/ANI0/AVREFP/EI20	/ 117	1		
■ クロック発生回路			ANI1		✓ 設定されていません	✓ 設定されていません	なし		
> 🥨 ALA. AD			ANI2		∥ 設定されていません	∥ 設定されていません	なし		
びん クリンパンタイムクロック のかかり クリーク のかり クリーク ののかり クリーク ののかり クリーク ののかり クリーク ののかり クリーク のののかり クリーク ののかり クリーク のののかり クリーク ののかり クリーク のののかり クリーク のののかり クリーク のののかり クリーク のののかり クリーク のののかり クリーク ののののののののののののののののののののののののののののののののの	tul stra Cal Bits		ANI3		∥ 設定されていません	∥ 設定されていません	なし		
♥ 400) クロック曲 / J / ノリー曲 / Ji つ DCL BU Zo	利御回路		ANI4		🥒 設定されていません	∥ 設定されていません	なし		
			ANI5			🥒 設定されていません	なし		
			ANI6		🥒 設定されていません	🥖 設定されていません	なし		
			ANI7		🥒 設定されていません	🥒 設定されていません	なし		
			ANI8		🥒 設定されていません	🥒 設定されていません	なし		
DACO			ANI9		🥒 設定されていません	🥒 設定されていません	なし		
	~	<						>	

図 10-3 端子機能リスト、端子番号リスト出力(csv 形式)

10.3 MCU パッケージ図 (png 形式)

MCU パッケージビューの [端子配置図を保存] 🏬 ボタンをクリックし、MCU パッケージ図を出力します。



図 10-4 MCU パッケージ図出力 (png 形式)



11. ユーザーコード保護機能

RL78 スマート・コンフィグレータ V1.6.0 以降のバージョンより、新たなユーザーコード保護機能をサポートしました。図 11-1 の指定タグを追加することで、任意の位置にユーザーコードを追加できるようになりました。追加されたユーザーコードはコード生成時に保護されます。

ユーザーコード保護機能は、「コード生成コンポーネント」が生成したファイルのみサポートします。

11.1 ユーザーコード保護機能の指定タグ

ユーザーコード保護機能を使用する場合、図 11-1 のように、/* Start user code */ と /* End user code */ を挿入し、このタグの間にユーザーコードを追加してください。指定タグが完全に一致しない場合は、保護されません。



11.2 ユーザーコード保護機能の使用例

図 11-2 に示すように、図 **11-1** の指定タグを使用し、A/D コンバータモジュールの Create() 関数の中に 新しいユーザーコードを挿入します。その後、A/D コンバータの GUI 設定を更新し、再びコード生成する と、挿入されたユーザーコードが新たに生成されたファイルに自動的にマージされます。



図 11-2 ユーザーコードの保護機能

11.3 競合発生時の対応方法

11.3.1 競合発生条件

GUIの設定変更やスマート・コンフィグレータのバージョンアップにより、挿入したユーザーコードの前後にある生成コードに変更がある場合、生成コードに競合が発生します。

競合が発生した場合、図 11-3 のようにコンソールに競合メッセージが表示されます。

🖳 או-עעב 🖳 🖬 🖬 🚽 🖻 ד 📬 🖬	
スマート・コンフィグレータ出力	
M04000001: ファイルを生成: <u>src\smc_gen\Config_RTC\Config_RTC.h</u>	~
M04000001: ファイルを生成: <u>src\smc_gen\Config_RTC\Config_RTC.c</u>	
M00000005:赤色でハイライトされている上記のファイルには、ユーザコードのマージが競合しています。ファイルを開き、手動で競合を解決してください	
M00000002: コード生成の終了:C:\Users\\workspace\Smart Configurator Example\src\smc gen	
	×
<	>

図 11-3 生成コードの競合メッセージ

ユーザーは、コンソール メッセージに競合ファイルをクリックし、[ファイル比較]ビューを開き、次の章 11.3.2 競合の解決方法ように競合を解決できます。



11.3.2 競合の解決方法

競合を解決するには、競合が発生したファイルを開いて、下記の手順に従って手動でコードを修正してく ださい。

- (1) コンソールメッセージの競合ファイルをクリックし、[File Compare] ビューを開きます。
- (2) 図 11-4 のように、矢印 🕟 をクリックし左側パネルのコードを右側パネルにコピーします。



図 11-4 生成コードの競合解決前

(3) 図 11-5 のように、適切な位置にコードを追加し、不要なコードを削除します。



図 11-5 生成コードの競合解決後



(4) コードのマージ後、保存 🔚 アイコンをクリックし保存します。

8 • % • 1 \$\$ • 9 • 10 \$\$ \$\$ \$\$ \$\$		Q. 🛛 🖻 🛛 Q. 🖓 🖓 ८/С++ 🧟 Х?ЭУЛ	719
Smart_Configurator_Example.scfg		•	
<mark> </mark>			
◆ 後◆ 翻訳単位 ●◆ R_Config_RTC_Create			
€ СНЕЕ2-7- §		M 📰 😤 🗑 🔛 💩 🕸	12
xisting code	• 1	New code	
<pre>virt = vu; /* Set FATCK: */ RTCC0 = 00 RTC CLK 32XHZ; /* Set L2-24-bour system and period of Constant-period interrupt (INTRTC) */ RTCC0 = (00 RTC RTCHZ DISABLE 00 RTC 12HOUR MODE 00 RTC INTRTC MOT GENERATE); /* Start user code */ por3 & bescription = This function enables the real-time clock. * Arguments = None * Arguments = None * Return Value : None *</pre>	Ja	<pre>7 Jet let/let/let/let/let/let/let/let/let/let/</pre>	
80 void R_Config_RTC_Start(void) 81 {		83 84 }	¥

図 11-6 コードの削除と保存

左側パネルのコードを右側パネルにコピーするか、右側パネルのコードを直接編集することで、競合を手動で解決することもできます。

【注】 競合が解決された後も、競合メッセージをクリックすると、[ファイル比較] ビューを開くことが できます。



12. ヘルプ

スマート・コンフィグレータの詳細情報は、ヘルプを参照ください。ヘルプをメニューから選択するとヘ ルプダイアログが開き、スマート・コンフィグレータに関するヘルプ目次が表示されます。



図 12-1 ヘルプ表示

ヘルプは、[概要] ページページの 🕐 アイコンからも参照できます。

	- 8	10 ヘルプ × 📄 🗘 🌣 🖇 🗖 🗖
概説	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	哈内容 ≫ 検索 № 関連トビック 即 ブックマーク 副 インデックス
 - 機能概要 様取 「飯豆をクリックすると、スマート・コンフィグレータの機能を確認 することができます。 動画 スマート・コンフィグレータの機能を確認 することができます。 動画 スマート・コンフィグレータの機能を確認 ることができます。 最新情報 量新情報 量新情報 量新情報 量新情報 量新情報 量新情報 量新情報 金方できます。	Smart Configurator	 スコーブ: デフォルト ◆ Smart Browser 図 スマート・コンフィグルータ (RL78) ◆ (図 スマート・コンフィグルータ (RL78) ユマート・コンフィグルータ (RL78) ユマート・コンフィグルータワロジェクトの作成 スマート・コンフィグレータクロジェクトの作成 スマート・コンフィグレータクロジェクトの作成 ソフトウェアコンボーネントの様成 ELCLモジュールのダウンロード グランカル・コンフィグルータの追加 SMSD/(-ジョン更新 第15込み 静定 コードの生成 Developer Assistance MUCパッケーシジェー
 ▼現在の設定状態 注: コード生成の動作は、設定 ページでカスタマイズできます。③ (使用しているボード/デバイス: R7F100GSNxFB (ROM size: 768KB, RAM size: 48KB, Pin count: 128) 生成先ロケーション (PROJECT_LOCV): src¥smc_gen 使用しているコンボーネント: 年更 ボード / クロック シンテム コンボーネント (強子) 割り込み。 	~	

図 12-2 クイックスタート

どちらの方法でも同じヘルプを参照できます。



13. 参考ドキュメント 【ユーザーズマニュアル : ハードウェア】

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/テクニカルニュース】 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。

【ユーザーズマニュアル:開発環境】

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 (R20UT4852)

最新の IAR Embedded Workbench for Renesas RL78 マニュアルを IAR ホームページから入手してください。

【アプリケーションノート】

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。



改訂記録

Rev.	セクション	改訂内容
1.00	-	-
		新規作成
1.01	第2章 使用前の準備	2.1 IAR C プロジェクト新規作成 削除
	第3章 スマート・コン	3.1 操作手順 更新
	フィグレータの操作方法	3.3 コンフィグレーションファイルの作成と読み込み 更新
	第4章 周辺機能の設定	4.4.13 コンポーネント構成のエクスポート 追加
		4.4.14 コンポーネント構成のインポート 追加
		4.4.15 コンポーネントの基本設定 追加
	第6章 ソースの生成	6.2 生成ファイルの構成とファイル名 更新
	第7章 統合開発環境への 読み込み	7. 統合開発環境への読み込み 更新
1.02	第4章 周辺機能の設定	4.4.12 BSP コンフィグレーションのバージョン変更の注を削除
		4.4.15 コンポーネントの基本設定:図13-1コンポーネントの基本
		設定の変更
		4.4.15 コンポーネントの基本設定:注1,2の変更
		4.4.15 コンポーネントの基本設定:注3の追加
		4.6.2 割り込みバンクの設定:手順(3)の変更、図 4-55 割り込み
		バンク設定の変更
1.03	第3章 スマート・コン	3.4.4 MCU/MPC パッケージビュー 更新
	フィグレータの操作	
	第4章 周辺機能の設定	4.1.2 ボード選択 更新
		4.4.3 ソフトウェア・コンポーネントの削除 複数コンポーネント の指定方法を追加
		4.4.10 RL78 Software Integration System モジュールのダウンロー ド 更新
		4.4.11 RL78 Software Integration System モジュールの追加 追加
		4.4.12 RL78 Software Integration System モジュールの設定 更新
		4.5 端子設定 図 4-50, 図 4-51 更新
		4.5.3 MCU/MPU パッケージビューを使用した端子の設定 更新
		4.5.4 端子機能から端子番号の表示 追加
		4.5.9 端子エラー/端子警告の設定 追加
	第6章 ソースの生成	6.2 生成ファイルの構成とファイル名 r_pincfg を追加
	第8章 ユーザープログラ ムの作成	8.2 ユーザーアプリケーションコードの使用方法 追加
	第 11 章 ユーザーコード保 護機能	11 ユーザーコード保護機能 追加
1.04	第4章 周辺機能の設定	4.4.8 ELCL 固定機能モジュールのダウンロード 更新
		4.4.9 固定機能 ELCL コンポーネントの設定 更新
		4.4.10 ELCL Flexible Circuit の作成と編集 追加
	第6章 ソースの生成	6.2 生成ファイルの構成とファイル名



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_I(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。

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

- 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。
- 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。
- 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

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。

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 ロ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/