

RL78 スマート・コンフィグレータ V1.2.0

R20UT4874JJ0103

Rev.1.03

2022.01.20

リリースノート

要旨

RL78 スマート・コンフィグレータをご使用いただきまして、誠にありがとうございます。

この添付資料では、本製品をお使いいただく上でのサポート機能および注意事項等を記載しております。

ご使用の前に、必ずお読みくださいますようお願い申し上げます。

目次

1. はじめに.....	3
1.1 システム要件.....	3
1.1.1 PC.....	3
1.1.2 開発ツール.....	3
2. サポート一覧.....	4
2.1 デバイス一覧.....	4
2.2 コンポーネント一覧.....	5
2.3 新規サポート.....	6
2.3.1 コンポーネント構成のインポート／エクスポート機能をサポート.....	6
2.3.2 コード生成設定「作成日付」のデフォルト値を変更.....	6
2.3.3 PDF ファイルの生成を無効にするオプションを追加.....	7
2.3.4 プロジェクトのビルド時にコード自動生成機能.....	7
2.3.5 BSP リビジョンアップデートとバージョンチェックのサポート.....	8
2.3.6 [概要] ページの改善.....	8
3. 変更内容.....	9
3.1 制限の修正.....	9
3.1.1 1バイト送信の UARTA コールバック関数の修正.....	9
3.1.2 ウォッチドッグ・タイマ・コンポーネントの割り込み優先度の日本語表示名の修正.....	9
3.2 仕様の変更.....	10
3.2.1 オンチップ・デバッグ設定の E20 エミュレータ選択を削除.....	10
3.2.2 R_Config_UARTn_Stop() 関数に出力停止コードを追加.....	11
3.2.3 r_cg_macrodriver.h に R_Systeminit() 関数宣言を生成するように改善.....	11
3.2.4 e ² studio プロジェクトを CS+ で開いた場合に、スマート・コンフィグレータの構成設定を開けるように改善.....	11
3.2.5 PORT の「入力バッファ」機能の改善.....	11
3.2.6 ELCL 向けに新しい API を追加.....	12
3.2.7 IIC 通信コードの改善.....	12
3.2.8 ELCL 初期化関数呼び出しの改善.....	13
3.2.9 端子割り当てのデフォルト選択として「PIORn = 0」を使用するよう改善.....	13
3.2.10 レジスタバンク警告メッセージの改善.....	13
3.2.11 オンチップ・デバッグのトレース機能を使用した場合の RAM 領域予約の改善.....	14

3.2.12 LLVM C++プロジェクトのサポート.....	16
3.2.13 UARTA 転送レート範囲の改善	16
3.2.14 オンチップ・デバッグ・トレースの DTC 予約済み RAM 領域にエントリを作成しないように改善	16
4. RENESAS TOOL NEWS の改修履歴.....	17
5. 制限事項.....	18
5.1 制限事項一覧.....	18
5.2 制限事項詳細.....	18
5.2.1 ヘルプの表示内容の制限.....	18
5.2.2 UART 送受信機能のクロック設定に関する制限.....	18
5.2.3 通信コンポーネントを同時に使用する場合の制限.....	19
5.2.4 CS+でスマート・コンフィグレータを開く際の制限.....	19
5.2.5 e ² studio [設定] ダイアログの [コンポーネント] メニューに関する制限.....	20
5.2.6 ELCL モジュールのダウンロード機能に関する制限.....	21
6. 注意事項.....	22
6.1 注意事項一覧.....	22
6.2 注意事項詳細.....	22
6.2.1 ビルドエラー・メッセージ「セクション.bss 仮想アドレス範囲が.dtc_vectortable と重複しています」 について.....	22
6.2.2 スマート・コンフィグレータのインストールにつて.....	23
改訂記録.....	24

1. はじめに

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。ルネサスデバイス用の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。

1.1 システム要件

RL78 スマート・コンフィグレータ V1.2.0 の動作環境は次の通りです。

1.1.1 PC

- IBM PC/AT 互換機 (Windows® 10 x64, Windows® 8.1 x64)
- プロセッサ: 1GHz 以上 (ハイパースレッディング, マルチコア CPU に対応)
- メモリ容量: 推奨 4GB 以上。最低 2GB 以上 (64 ビット版 Windows では 4G バイト以上)
- ハードディスク容量: 空き容量 300MB 以上
- ディスプレイ: 1024×768 以上の解像度, 65536 色以上

1.1.2 開発ツール

- ルネサスエレクトロニクス製 RL78 用コンパイラ CC-RL V1.11 以上
- LLVM for Renesas RL78 V10.0.0.202110 以上
- IAR Embedded Workbench for Renesas RL78 V4.21.1 以上
- SMS アセンブラ V1.00.00 以上

e² studio に SMS Assembler を追加する場合、e² studio 2021-04 以降の統合インストーラからインストールしてください ([統合開発環境 e² studio | Renesas](#))。

他のコンパイラ同様、e² studio セットアップウィザードの [追加ソフトウェア] - [Renesas Toolchains & Utilities] タブで選択しインストールします。

2. サポート一覧

RL78 スマート・コンフィグレータ V1.2.0 のサポートデバイス、コンポーネントについて説明します。

2.1 デバイス一覧

RL78 スマート・コンフィグレータ V1.2.0 のサポートデバイス一覧です。

表 2-1 サポートデバイス

グループ (HW マニュアル番号)	ピン数	デバイス名
RL78/G23 グループ (R01UH0896JJ0100)	30pin	R7F100GAFxSP, R7F100GAGxSP, R7F100GAHxSP, R7F100GAJxSP
	32pin	R7F100GBFxNP, R7F100GBGxNP, R7F100GBHxNP, R7F100GBJxNP, R7F100GBFxFP, R7F100GBGxFP, R7F100GBHxFP, R7F100GBJxFP
	36pin	R7F100GCFxLA, R7F100GCGxLA, R7F100GCHxLA, R7F100GCJxLA
	40pin	R7F100GEFxNP, R7F100GEGxNP, R7F100GEHxNP, R7F100GEJxNP
	44pin	R7F100GFFxFP, R7F100GFGxFP, R7F100GFHxFP, R7F100GFJxFP, R7F100GFKxFP, R7F100GFLxFP, R7F100GFNxFP
	48pin	R7F100GGFxFB, R7F100GGGxFB, R7F100GGHxFB, R7F100GGJxFB, R7F100GGKxFB, R7F100GGLxFB, R7F100GGNxFB, R7F100GGFxNP, R7F100GGGxNP, R7F100GGHxNP, R7F100GGJxNP, R7F100GGKxNP, R7F100GGLxNP, R7F100GGNxNP
	52pin	R7F100GJFxFA, R7F100GJGxFA, R7F100GJHxFA, R7F100GJJxFA, R7F100GJKxFA, R7F100GJLxFA, R7F100GJNxFA
	64pin	R7F100GLFxFA, R7F100GLGxFA, R7F100GLHxFA, R7F100GLJxFA, R7F100GLKxFA, R7F100GLLxFA, R7F100GLNxFA, R7F100GLFxFB, R7F100GLGxFB, R7F100GLHxFB, R7F100GLJxFB, R7F100GLKxFB, R7F100GLLxFB, R7F100GLNxFB, R7F100GLFxLA, R7F100GLGxLA, R7F100GLHxLA, R7F100GLJxLA, R7F100GLKxLA, R7F100GLLxLA, R7F100GLNxLA
	80pin	R7F100GMGxFA, R7F100GMHxFA, R7F100GMJxFA, R7F100GMKxFA, R7F100GMLxFA, R7F100GMNxFA, R7F100GMGxFB, R7F100GMHxFB, R7F100GMJxFB, R7F100GMKxFB, R7F100GMLxFB, R7F100GMNxFB
	100pin	R7F100GPGxFB, R7F100GPHxFB, R7F100GPJxFB, R7F100GPKxFB, R7F100GPLxFB, R7F100GPNxFB, R7F100GPGxFA, R7F100GPHxFA, R7F100GPJxFA, R7F100GPKxFA, R7F100GPLxFA, R7F100GPNxFA
128pin	R7F100GSJxFB, R7F100GSKxFB, R7F100GSLxFB, R7F100GSNxFB	

2.2 コンポーネント一覧

RL78 スマート・コンフィグレータ V1.2.0 のサポートコンポーネント一覧です。

表 2-2 サポートコンポーネント

✓ : サポート, - : 非サポート

No	コンポーネント	モード	RL78/G23	備考
1	A/D コンバータ		✓	
2	クロック出力/ブザー出力制御回路		✓	
3	コンパレータ		✓	
4	CSI 通信	送信	✓	
		受信	✓	
		送信/受信	✓	
5	D/A コンバータ		✓	
6	データ・トランスファ・コントローラ		✓	
7	ディレイ・カウンタ		✓	
8	分周器機能		✓	
9	外部イベント・カウンタ		✓	
10	IIC 通信(マスタモード)		✓	
11	IIC 通信(スレーブモード)		✓	
12	入力パルス間隔測定		✓	
13	入力信号のハイ/ロウ・レベル測定		✓	
14	割り込みコントローラ		✓	
15	インターバル・タイマ	8-bit カウントモード	✓	
		16-bit カウントモード	✓	
		16-bit キャプチャモード	✓	
		32-bit カウントモード	✓	
16	キー割り込み		✓	
17	ワンショット・パルス出力		✓	
18	ポート		✓	
19	PWM 出力		✓	
20	リアルタイム・クロック		✓	
21	リモコン信号受信機能		✓	
22	SNOOZE モード・シーケンサ		✓	
23	方形波出力		✓	
24	UART 通信	送信	✓	
		受信	✓	
		送信/受信	✓	
25	電圧検出回路		✓	
26	ウォッチドッグ・タイマ		✓	
27	ロジック&イベント・リンク・コントローラ		✓	ELCL ダウンロード機能を使用して追加してください。

2.3 新規サポート

2.3.1 コンポーネント構成のインポート／エクスポート機能をサポート

コンポーネント構成ファイルのインポート／エクスポート機能をサポートしました。コンポーネントの構成を xml ファイルにエクスポートしたり、エクスポートした xml ファイルから別のスマート・コンフィグレータ・プロジェクトにインポートしたりできます。

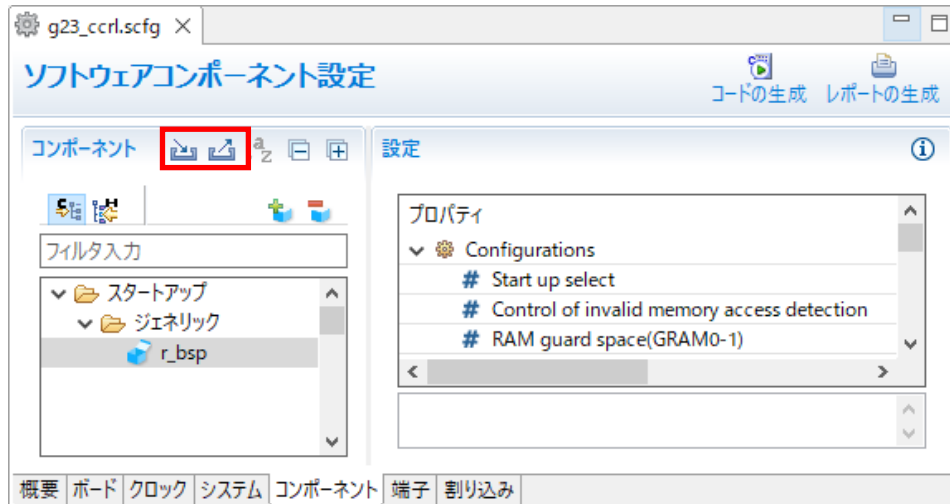


図 2-1 「コンポーネント」 ページの構成のインポート／エクスポート

2.3.2 コード生成設定「作成日付」のデフォルト値を変更

コード生成設定「作成日付」のデフォルト値を、「出力する」から「出力しない」に変更しました。

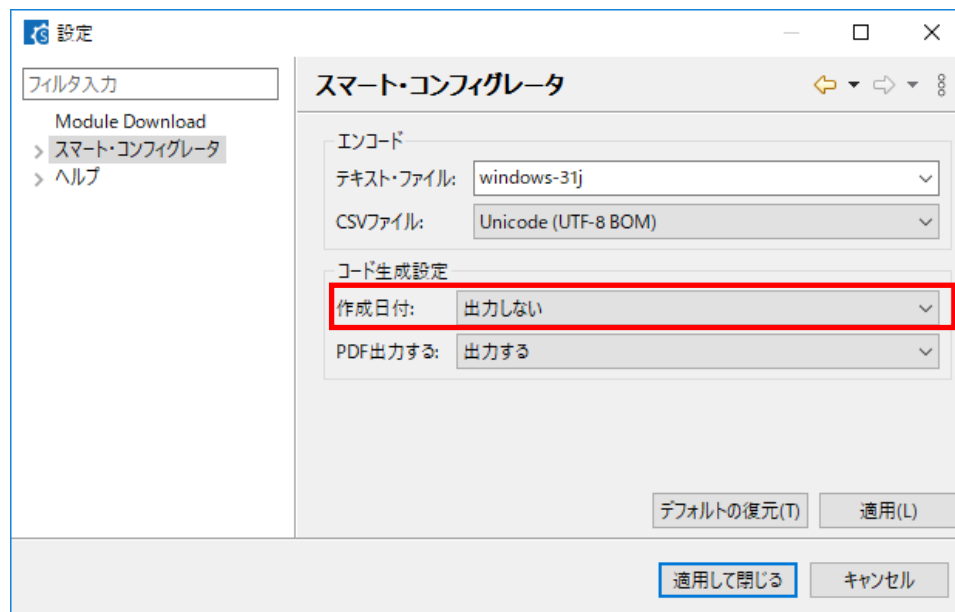


図 2-2 コード生成時の警告ダイアログ

2.3.3 PDF ファイルの生成を無効にするオプションを追加

RL78 Software Integration System コンポーネンが追加されたときに、プロジェクトへの PDF ファイルの生成を無効にするオプションを追加しました。

このオプションを使用するには、[ウィンドウ] - [設定] - [スマート・コンフィグレータ] よりスマート・コンフィグレータ設定ページに移動し、[PDF 出力する] 項目を「出力しない」に設定してください。

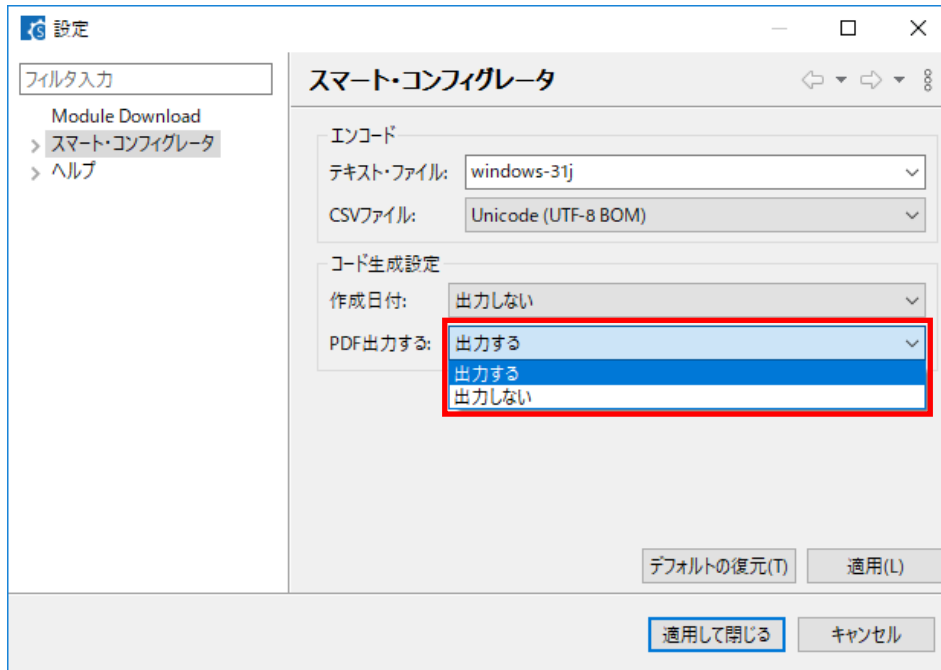


図 2-3 PDF 出力設定のオプション

2.3.4 プロジェクトのビルド時にコード自動生成機能

e² studio 2022-01 から GUI の設定とビルドされたコードの不一致を防ぐために、プロジェクトのビルド時にコード生成を最初に行うようになりました。

この機能を使用したくない場合は、プロジェクトのプロパティを開き、[SC Code Generation Builder] の選択を解除してください。

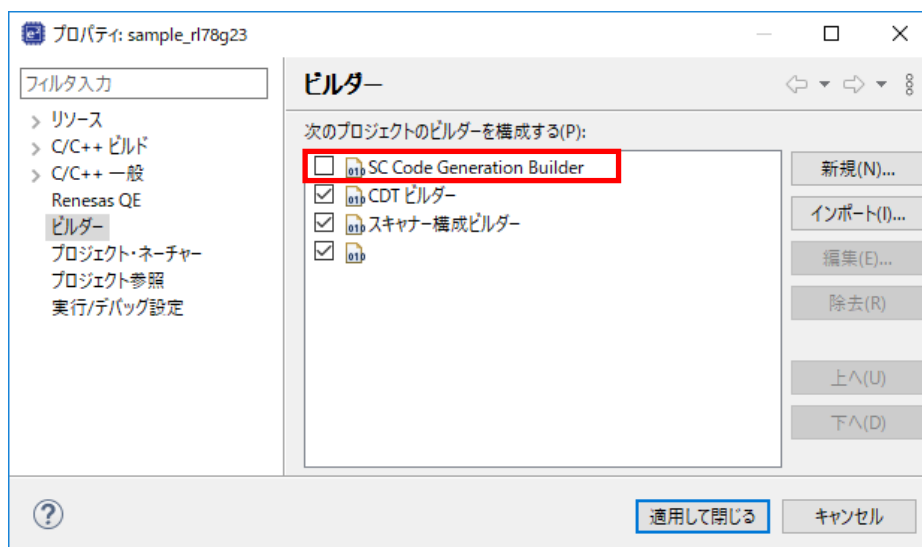


図 2-4 ビルダー設定のコード生成オフ

2.3.5 BSP リビジョンアップデートとバージョンチェックのサポート

- 1) スマート・コンフィグレータのプロジェクト作成時に追加されるデフォルトの BSP を rev. 1.13 に更新しました。
- 2) スマート・コンフィグレータと BSP を正しい組み合わせで使用していただくために、バージョンチェック用のマクロ (BSP_CFG_CONFIGURATOR_VERSION) を追加しました。
 - a) BSP_CFG_CONFIGURATOR_VERSION は、r_bsp_config.h に生成されます。

```

/* Version number of Smart Configurator.
   This macro definitions is updated by Smart Configurator.
   If you are using e2studio, set the following values.
   2021-04 : 1001
   2021-07 : 1010
   2021-10 : 1010
   If you are using Smart the standalone version of Smart Configurator,
   set the following values.
   v1.0.1 : 1001
   v1.1.0 : 1010
*/
#define BSP_CFG_CONFIGURATOR_VERSION /* Generated value. Do not edit this manually */

```

- b) 古いバージョンのスマート・コンフィグレータを使用すると、以下のビルドエラーが発生します。

```

#error "Make sure that the value of BSP_CFG_CONFIGURATOR_VERSION defined in
r_config/r_bsp_config.h matches the version of Smart Configurator you are using. If
they do not match, change the settings. If they match, you need to upgrade your
Smart Configurator. Please upgrade Smart Configurator."

```

2.3.6 [概要] ページの改善

[概要] ページを更新し、「ビデオ」や「最新情報」などへのリンクを追加しました。



図 2-5 [概要] ページ

3. 変更内容

RL78 スマート・コンフィグレータ V1.2.0 の変更内容について説明します。

3.1 制限の修正

表 3-1 制限の修正一覧

✓: 対象デバイス, -: 対象外デバイス

No	内容	RL78/G23	備考
1	1 バイト送信の UARTA コールバック関数の修正	✓	
2	ウォッチドッグ・タイマ・コンポーネントの割り込み優先度の日本語表示名の修正	✓	

3.1.1 1 バイト送信の UARTA コールバック関数の修正

UARTA 送信機能を使用して [ポーリングによる連続送信] モードでデータを送信する場合、1 バイトまたは複数のデータに関係なく、生成コードの仕様を以下のように変更しました。

- GUI で [送信完了] コールバック機能を選択できません。
- {Config_UARTAn}_user.c ファイルに以下の変更を行いました。
 - r_{Config_UARTAn}_PollingEnd_UserCode() を追加しました。
 - r_{Config_UARTAn}_send_1byte() と R_{Config_UARTAn}_Send_Polling() を削除しました。

3.1.2 ウォッチドッグ・タイマ・コンポーネントの割り込み優先度の日本語表示名の修正

ウォッチドッグ・タイマ・コンポーネントの割り込み優先度の日本語表示名を、以下のように変更しました。

- 「レベル 3 (低優先順位)」を「レベル 0 (高優先順位)」に変更
- 「レベル 0 (高優先順位)」を「レベル 3 (低優先順位)」に変更

3.2 仕様の変更

表 3-2 仕様の変更一覧

✓: 対象デバイス, -: 対象外デバイス

No	内容	RL78/G23	備考
1	オンチップ・デバッグ設定の E20 エミュレータ選択を削除	✓	
2	R_Config_UARTn_Stop() 関数に出力停止コードを追加	✓	
3	r_cg_macrodriver.h に R_Systeminit() 関数宣言を生成するように改善	✓	
4	e ² studio プロジェクトを CS+ で開いた場合に、スマート・コンフィグレータの構成設定を開けるように改善	✓	
5	PORT の「入力バッファ」機能の改善	✓	
6	ELCL 向けに新しい API を追加	✓	
7	IIC 通信コードの改善	✓	
8	ELCL 初期化関数呼び出しの改善	✓	
9	端子割り当てのデフォルト選択として「PIORn = 0」を使用するよう改善	✓	
10	レジスタバンク警告メッセージの改善	✓	
11	オンチップ・デバッグのトレース機能を使用した場合の RAM 領域予約の改善	✓	
12	LLVM C++ プロジェクトのサポート	✓	
13	UARTA 転送レート範囲の改善	✓	
14	オンチップ・デバッグ・トレースの DTC 予約済み RAM 領域にエントリを作成しないように改善	✓	

3.2.1 オンチップ・デバッグ設定の E20 エミュレータ選択を削除

E20 エミュレータは、RL78/G23 をサポートしていません。そのため、[システム] ページから E20 エミュレータの選択を削除しました。E20 エミュレータが選択されている古いプロジェクトを開くと、エミュレータ設定は自動的に E2Lite に変更されます。



図 3-1 E20 エミュレータ選択を削除

3.2.2 R_Config_UARTn_Stop() 関数に出力停止コードを追加

SAUmのUARTnを送信モードとして使用する場合、UARTn出力停止設定 (SOEmn = 0) が R_Config_UARTn_Stop() 関数に生成されます。

```

/*****
* Function Name: R_Config_UART0_Stop
* Description  : This function stops UART0 module operation.
* Arguments   : None
* Return Value : None
*****/
void R_Config_UART0_Stop(void)
{
    STMK0 = 1U; /* disable INTST0 interrupt */
    STO |= 0001 SAU CH0 STOP TRG ON; /* disable UART0 transmit */
    SOE0 &= (uint16_t)~ 0001 SAU CH0 OUIPUT ENABLE; /* disable UART0 output */
    STIFO = 0U; /* clear INTST0 interrupt flag */
}

```

図 3-2 R_Config_UARTn_Stop() 関数の SOEmn = 0

3.2.3 r_cg_macrodriver.h に R_Systeminit() 関数宣言を生成するように改善

スマート・コンフィグレータの設定 [API 出力関数] を「初期化 API 関数のみ出力する」に設定した場合に、r_cg_macrodriver.h に R_Systeminit() 関数宣言を生成するように改善しました。

3.2.4 e² studio プロジェクトを CS+で開いた場合に、スマート・コンフィグレータの構成設定を開けるように改善

e² studio のルネサス共通プロジェクトファイルでエクスポートしたスマート・コンフィグレータ・プロジェクトを CS+で開いたときに、スマート・コンフィグレータ起動後に構成設定を開けるように改善しました。

3.2.5 PORTの「入力バッファ」機能の改善

「入力バッファ」を「入力バッファオフ」に変更し、「入力バッファオフ」は、ポートを出力機能として使用する場合のみ選択可能です。



図 3-3 PORT の入力バッファオフ機能

R_<Config_PORT>_Create() :

- 「入力バッファオフ」が無効またはチェックを外すと、PDIDIS レジスタに `_00_PDIDISnx_INPUT_BUFFER_ON` が設定されます。
- 「入力バッファオフ」が有効でチェックされている場合、`_xx_PDIDISnx_INPUT_BUFFER_OFF` が PDIDIS レジスタに設定されます。

```

void R_Config_PORT_Create(void)
{
    /* Set PORT0 registers */
    ...
    PDIDIS0 = _00_PDIDISn4_INPUT_BUFFER_ON | _00_PDIDISn3_INPUT_BUFFER_ON |
              _00_PDIDISn2_INPUT_BUFFER_ON | _00_PDIDISn0_INPUT_BUFFER_ON;
    ...
    R_Config_PORT_Create_UserInit();
}

```

3.2.6 ELCL 向けに新しい API を追加

出力 ELCL 信号が INTELCL として選択されると、新しい API 関数が生成されます。

- R_Config_ELCL_xxxFunction_Start
- R_Config_ELCL_xxxFunction_Stop
- r_Config_ELCL_xxxFunction_interrupt (割り込みハンドラ関数)

3.2.7 IIC 通信コードの改善

IIC 通信 (マスタモード) コンポーネントを使用し、IIC0n としてリソースを選択すると、生成コードが改善され、理解し易くなりました。

- 1) R_Config_IICmn_StartCondition() および R_Config_IICmn_StopCondition() 関数での SDA, SCL 端子の待機操作に関する詳細コメントを追加しました。

```
void R_Config_IIC00_StartCondition(void)
{
    volatile uint8_t w_count;

    S00 &= (uint16_t)~_0001_SAU_CH0_DATA_OUTPUT_1; /* clear IIC00 SDA */

    /* Set delay to secure a hold time after SDA output low. The delay time depend on slave device.
    Here set 5us as default base on current clock */
    for (w_count = 0U; w_count <= IIC00_WAITTIME; w_count++)
    {
        NOP();
    }

    S00 &= (uint16_t)~_0100_SAU_CH0_CLOCK_OUTPUT_1; /* clear IIC00 SCL */
    SOE0 |= _0001_SAU_CH0_OUTPUT_ENABLE; /* enable IIC00 output */
    SS0 |= _0001_SAU_CH0_START_TRG_ON; /* enable IIC00 */

    /* Set delay to secure a hold time after SCL output low. The delay time depend on slave device.
    Here set 5us as default base on current clock */
    for (w_count = 0U; w_count <= IIC00_WAITTIME; w_count++)
    {
        NOP();
    }
}
```

図 3-4 SDA, SCL 端子の待機操作に関する詳細コメントの追加

- 2) エラー検出フラグをクリアし、IICm 割り込みハンドラ関数の R_Config_IICmn_StopCondition() 関数呼び出して、次の送受信操作が正しく機能するようにするコードを追加しました。

```
static void __near r_Config_IIC00_interrupt(void)
{
    volatile uint16_t w_count;

    /* Set delay to secure a hold time after SDA, SDL output. The delay time depend on slave device.
    Here set 20us as default base on current clock */
    for (w_count = 0U; w_count <= IIC00_WAITTIME_2; w_count++)
    {
        NOP();
    }

    if ((0x0002U == (SSR00 & _0002_SAU_PARITY_ERROR)) && (0U != g_iic00_tx_count))
    {
        SIR00 |= _0002_SAU_SIRMN_PECTMN; /* clear ACK error detection flag */
        R_Config_IIC00_StopCondition();
        r_Config_IIC00_callback_master_error(MD_NACK);
    }
    else if ((0x0001U == (SSR00 & _0001_SAU_OVERRUN_ERROR)) && (0U != g_iic00_tx_count))
    {
        SIR00 |= _0001_SAU_SIRMN_OVCTMN; /* clear overrun error detection flag */
        R_Config_IIC00_StopCondition();
        r_Config_IIC00_callback_master_error(MD_OVERRUN);
    }
    else
    {

```

図 3-5 エラー検出フラグのクリアと R_Config_IICmn_StopCondition() 関数の呼び出し

3.2.8 ELCL 初期化関数呼び出しの改善

ELCL 初期化関数の R_Config_ELCL_xxxFunction_Create() は、R_Systeminit で呼び出されるよう改善しました。

3.2.9 端子割り当てのデフォルト選択として「PIORn = 0」を使用するよう改善

デフォルト選択として「PIORn = 0」を使用するよう端子割り当てを改善しました。

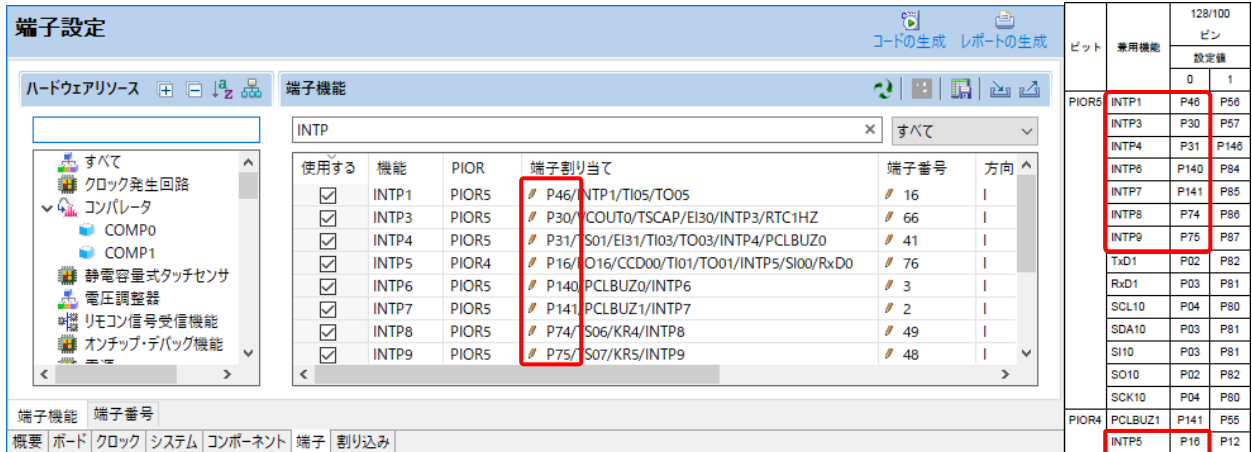


図 3-6 PIOR のデフォルト設定

3.2.10 レジスタバンク警告メッセージの改善

割り込みの優先度が同じで、同じレジスタバンクを指定した場合に、警告メッセージを表示しないように改善しました。



図 3-7 割り込みレジスタバンク

3.2.11 オンチップ・デバッグのトレース機能を使用した場合の RAM 領域予約の改善

スマート・コンフィグレータは、CS+および e² studio のトレース RAM 領域設定を自動的に設定します。



図 3-8 スマート・コンフィグレータのトレース機能設定

【CS+の場合】

[トレース RAM 領域への配置を制御する] は、スマート・コンフィグレータのトレース機能の設定により変更されます。

- [トレース機能設定] を「使用する」に選択した場合、[トレース RAM 領域への配置を制御する]は「はい(エラーを表示) (-OCDTR)」に設定されます。
- [トレース機能設定] を「使用しない」に選択した場合、[トレース RAM 領域への配置を制御する]は「いいえ」に設定されます。

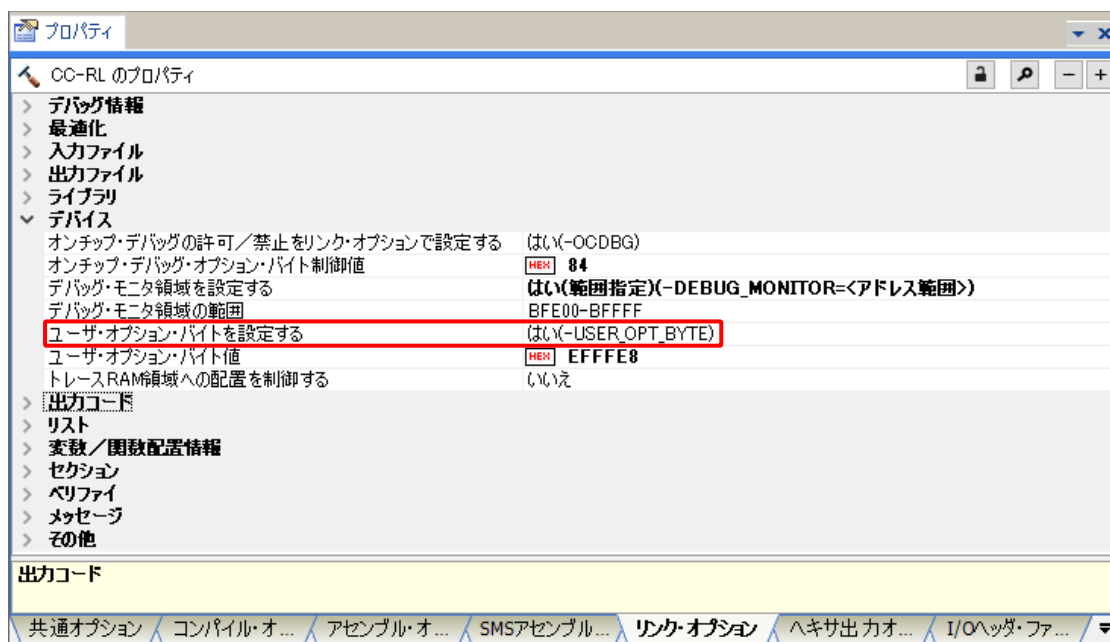


図 3-9 CS+の [トレース RAM 領域への配置を制御する]

【e² studio の場合】

1) LLVM for Renesas RL78 C/C++ Executable Project

スマート・コンフィグレータのトレース機能設定を使用する場合:

linker_script.ld file:

```
.....
.o cd_traceram 0xfc300(注1) (NOLOAD) : AT(0xfc300)
{
    KEEP(*(.o cd_traceram))
} >RAM
.....
```

注 1: 製品に応じた開始 RAM アドレス。

r_cg_vector_table.c:

```
.....
/* Secure trace RAM area */
#define OCDTRACERAM_SECT __attribute__((section(".o cd_traceram")))
uint8_t Ocd TraceRam[1024(注1)] OCDTRACERAM_SECT;
.....
```

注 1: 製品に応じた RAM サイズ。

2) Renesas CC-RL C/C++ Executable Project

「セクションのない RAM 領域」オプションは、スマート・コンフィグレータのトレース機能の設定により変更されます。

- [トレース機能設定] を「使用する」に選択した場合、[セクションを配置しない RAM 領域] は「セルフ RAM とトレース RAM」に設定されます。
- [トレース機能設定] を「使用しない」に選択した場合、[セクションを配置しない RAM 領域] は「なし」に設定されます。

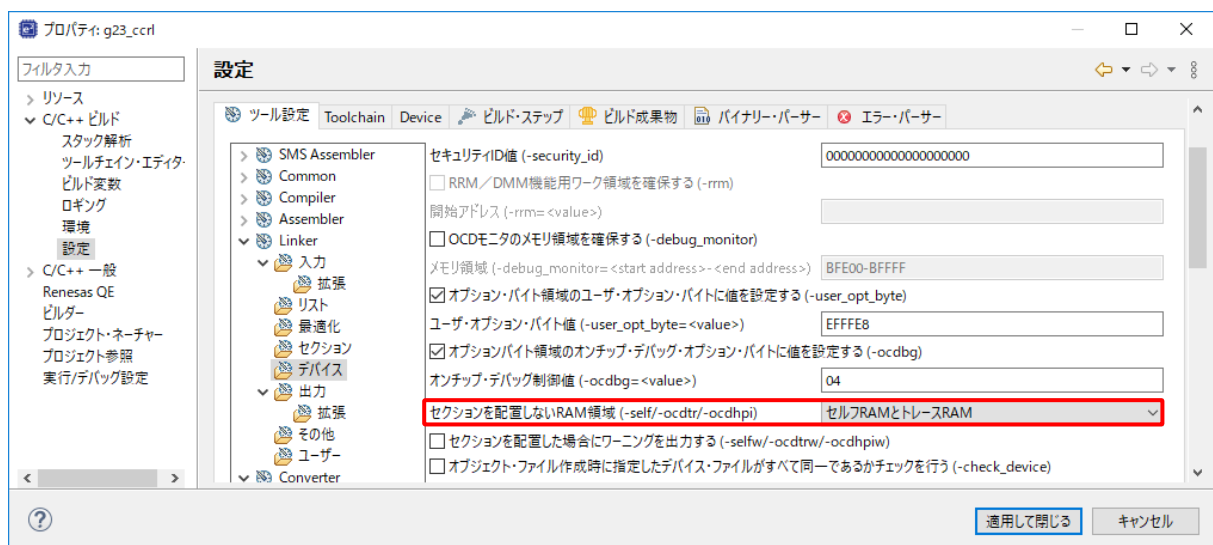


図 3-10 e² studio の [セクションを配置しない RAM 領域]

3.2.12 LLVM C++プロジェクトのサポート

スマート・コンフィグレータ V1.2.0 は、extern "C"コードを生成できるようになりました。

これにより e² studio の LLVM C++プロジェクトで使用できます。

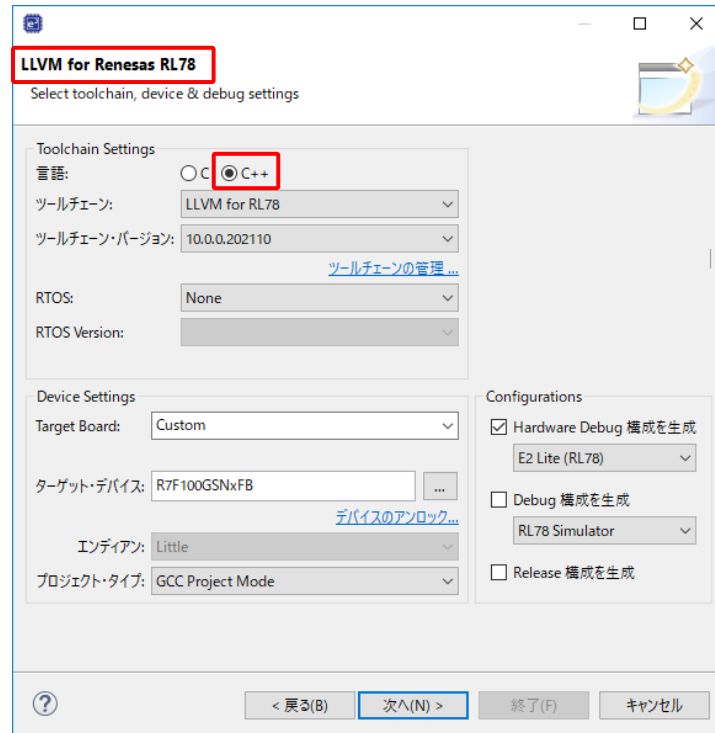


図 3-11 LLVM C++プロジェクト

3.2.13 UARTA 転送レート範囲の改善

UARTA の最大転送速度を 19200bps から 153600bps に変更しました。

3.2.14 オンチップ・デバッグ・トレースの DTC 予約済み RAM 領域にエントリを作成しないように改善

オンチップ・デバッグ・トレースの予約済み RAM 領域を使用するときにエントリが作成されないように、リンカ・スクリプト・ファイルに「NOLOAD」処理を追加しました。

linker_script.ld ファイル :

```
.....
.dtc_vectortable 0xffd00 (NOLOAD) : AT(0xffd00)
{
    KEEP*(.dtc_vectortable)
} >RAM
.dtc_controldata_0 0xffd40 (NOLOAD) : AT(0xffd40)
{
    KEEP*(.dtc_controldata_0)
} >RAM
.....
```


4. RENESAS TOOL NEWS の改修履歴

RENESAS TOOL NEWS 注意事項の改修状況について記載します。

発行日	資料番号	概要	対象デバイス	改修バージョン
2021/10/01	R20TS0757	1. LLVM for Renesas RL78 C/C++ Executable Project 作成時の注意事項 2. ポート入力バッファ機能に関する注意事項 https://www.renesas.com/document/tnn/notes-e-studio-smart-configurator-plug-smart-configurator-rl78	RL78/G23	V1.2.0

5. 制限事項

RL78 スマート・コンフィグレータ V1.2.0 の制限事項について説明します。

5.1 制限事項一覧

表 5-1 制限事項一覧

✓: 対象デバイス, -: 対象外デバイス

No	内容	RL78/G23	備考
1	ヘルプの表示内容の制限	✓	
2	UART 送受信機能のクロック設定に関する制限	✓	
3	通信コンポーネントを同時に使用する場合の制限	✓	
4	CS+でスマート・コンフィグレータを開く際の制限	✓	
5	e ² studio [設定] ダイアログの [コンポーネント] メニューに関する制限	✓	
6	ELCL モジュールのダウンロード機能に関する制限	✓	

5.2 制限事項詳細

5.2.1 ヘルプの表示内容の制限

スマート・コンフィグレータのヘルプメニュー [Help Contents] を開くと、下図のように「Smart Browser」が表示されます。

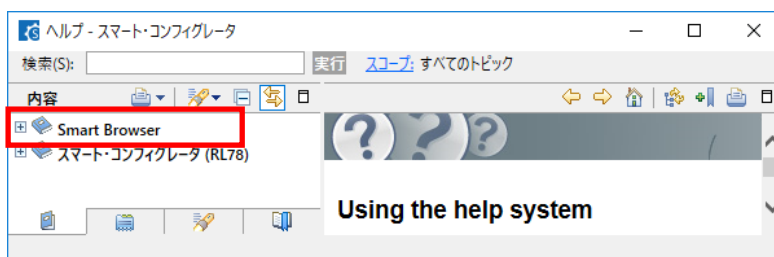


図 5-1 ヘルプメニューの [Help Contents]

5.2.2 UART 送受信機能のクロック設定に関する制限

UART 送受信機能を使用する場合は、「送信」と「受信」のクロック設定を同じ値に設定する必要があります。そうしないと、受信側が正しくデータを受信できません。

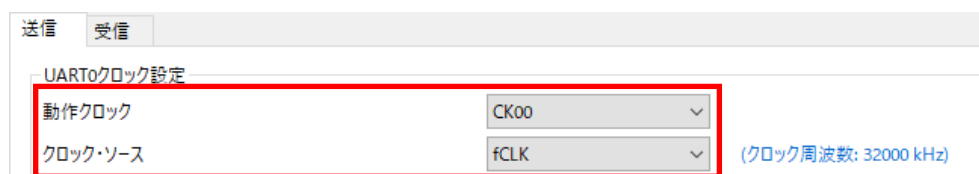


図 5-2 UART の送受信のクロック計設定

5.2.3 通信コンポーネントを同時に使用する場合の制限

UART n 受信機能を使用する場合は、奇数チャンネルの CSI n 1 機能と IIC n 1 機能を同時に使用しないように注意してください。

<80, 100, 128 ピン製品 >

ユニット	チャンネル	簡易SPI (CSI) として使用	UARTとして使用	簡易I2Cとして使用
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (LIN-bus対応)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	CSI31		IIC31

図 5-3 複数の通信コンポーネントを使用する場合

5.2.4 CS+でスマート・コンフィグレータを開く際の制限

CS+でスマート・コンフィグレータを正常に開くには、*.scfg ファイル（CS+プロジェクトフォルダの下にある）の名前がプロジェクト名と同じである必要があります。それ以外の場合、スマート・コンフィグレータは空のウィンドウを開きます。

この制限により、以下の2つの操作にご注意ください。

- e² studio プロジェクトを (e² studio の「ルネサス共通プロジェクトファイル」機能により) CS+にエクスポートする場合は、*.scfg ファイル名がプロジェクト名と同じであることを確認してください。
- 新しいCS+プロジェクトを作成するときは、ターゲットプロジェクトフォルダの下に複数の.scfg ファイルがないことを確認してください。このフォルダの下に複数の.scfg ファイルがある場合、スマート・コンフィグレータを開くと [File selection] ダイアログが表示されます。何もせずに [OK] をクリックしてください。

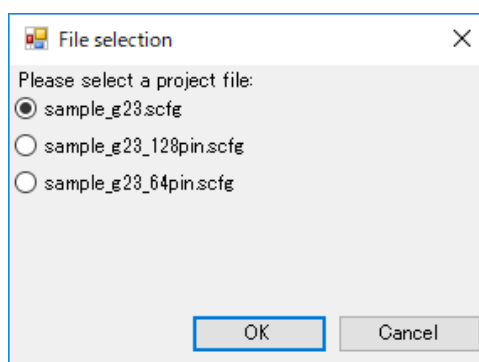


図 5-4 [File selection] ダイアログ

5.2.5 e² studio [設定] ダイアログの [コンポーネント] メニューに関する制限

e² studio 2022-01 インストール時に RX ファミリーを選択しない場合、e² studio [設定] ダイアログの [スマート・コンフィグレータ] - [コンポーネント] メニューは表示されません。

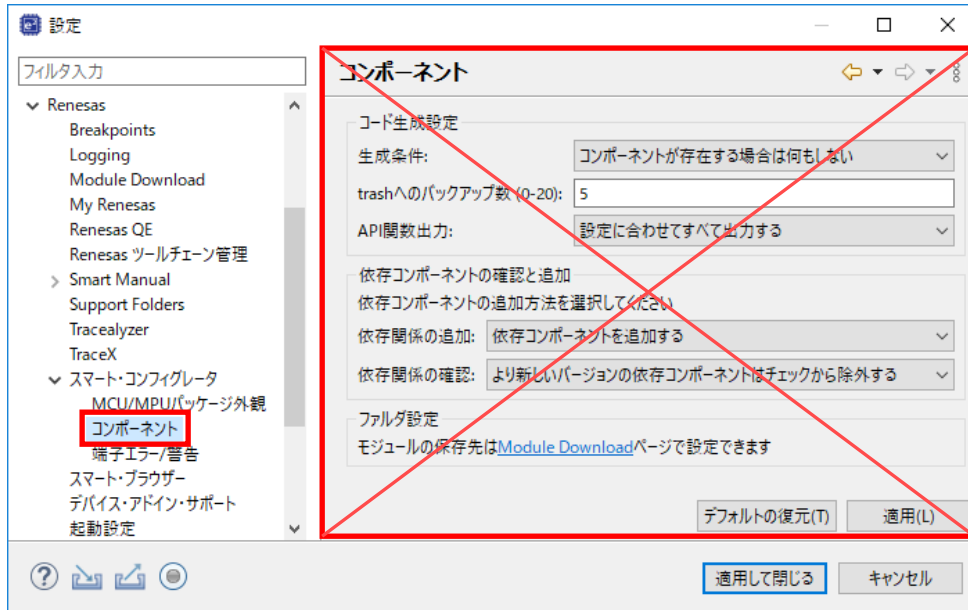


図 5-5 e² studio の [コンポーネント] メニュー

e² studio 2022-01 セットアップウィザードで RX ファミリーを選択してください。RX ファミリーをインストールすると、スマート・コンフィグレータの [コンポーネント] メニューが [設定] ダイアログに表示されます。

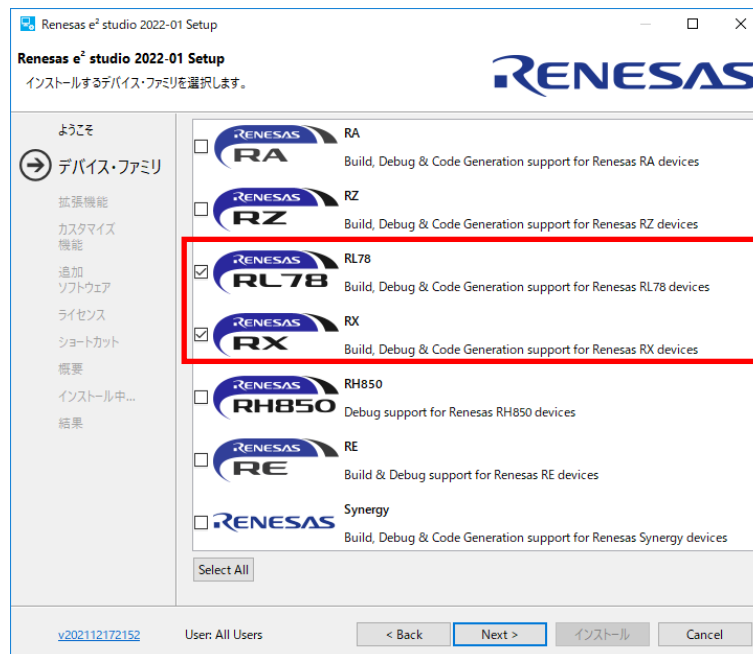


図 5-6 e² studio セットアップウィザードの RX ファミリー選択

5.2.6 ELCL モジュールのダウンロード機能に関する制限

RL78 スマート・コンフィグレータ V1.2.0 は、ELCL ダウンロード機能を使用できません。[コンポーネントの追加] ダイアログの [ELCL モジュールをダウンロードする] から ELCL モジュールを選択しても、ELCL モジュールはダウンロードできません。そのため、Web で公開している ELCL モジュールを RL78 スマート・コンフィグレータ V1.2.0 に同梱しましたので、ダウンロードする必要なく使用できます。

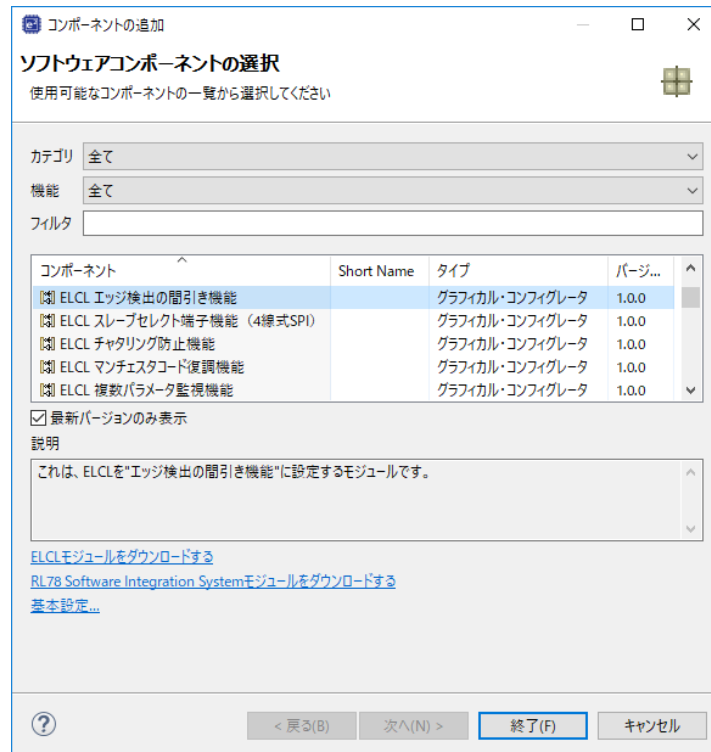


図 5-7 ELCL モジュール・ダウンロード・リンク

ELCL サンプルコードでコンポーネントリストに表示されていない ELCL モジュールを使用している場合、サンプルコードに同梱されている ELCL モジュールを、スマート・コンフィグレータがインストールされているフォルダの \RL78_Modules\ELCL_Modules フォルダにコピーしてください。コピー後は、スマート・コンフィグレータを再起動してください。

- `${Smart Configurator installation folder}\RL78_Modules\ELCL_Modules`

6. 注意事項

RL78 スマート・コンフィグレータ V1.2.0 の制限事項について説明します。

6.1 注意事項一覧

表 6-1 注意事項一覧

✓: 対象デバイス, -: 対象外デバイス

No	内容	RL78/G23	備考
1	ビルドエラー・メッセージ「セクション.bss 仮想アドレス範囲が.dtc_vectortable と重複しています」について	✓	
2	スマート・コンフィグレータのインストールにつて	✓	

6.2 注意事項詳細

6.2.1 ビルドエラー・メッセージ「セクション.bss 仮想アドレス範囲が.dtc_vectortable と重複しています」について

多くのコンポーネントと DTC コンポーネントを一緒に使用すると、一部のセクション・アドレスが重複するため、ビルドエラーが発生する場合があります。

```

CDT Build Console [LLVM_R7F100GCJxLA_case1]
ld.lld: error: section .bss virtual address range overlaps with .dtc_vectortable
>>> .bss range is [0xF9F00, 0xF9F31]
>>> .dtc_vectortable range is [0xF9F00, 0xF9F27]

ld.lld: error: section .bssf virtual address range overlaps with .dtc_controldata_0
>>> .bssf range is [0xF9F32, 0xF9F7F]
>>> .dtc_controldata_0 range is [0xF9F40, 0xF9F47]

ld.lld: error: section .bss load address range overlaps with .dtc_vectortable
>>> .bss range is [0xF9F00, 0xF9F31]
>>> .dtc_vectortable range is [0xF9F00, 0xF9F27]

ld.lld: error: section .bssf load address range overlaps with .dtc_controldata_0
>>> .bssf range is [0xF9F32, 0xF9F7F]
>>> .dtc_controldata_0 range is [0xF9F40, 0xF9F47]
clang: error: ld.lld command failed with exit code 1 (use -v to see invocation)
makefile:110: recipe for target 'LLVM_R7F100GCJxLA_case1.elf' failed
make: *** [LLVM_R7F100GCJxLA_case1.elf] Error 1
"make -j8 all" terminated with exit code 2. Build might be incomplete.

18:09:07 Build Failed. 2 errors, 0 warnings. (took 1s.846ms)

```

図 6-1 ビルドエラー・メッセージ

【回避策】 スマート・コンフィグレータは、「.bss」および「.bssf」セクション・アドレスを設定できません。したがって、ユーザーは、「.bss」および「.bssf」セクション・アドレスを手動で変更するか、DTC ベース・アドレスを変更して、このようなセクション・オーバーラップ・エラーを回避することを検討する必要があります。

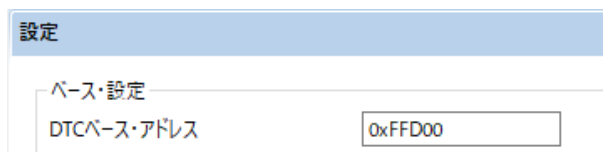


図 6-2 DTC ベース・アドレス設定

6.2.2 スマート・コンフィグレータのインストールにつて

インストールディレクトリに 64 文字以上を指定しないでください。

「指定されたパスが長すぎます」というエラーが発生し、スマート・コンフィグレータをインストールすることができません。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.01	2021.04.13	-	新規作成
1.02	2021.07.20	2	1.1.2 開発ツール 更新
		5	2.3 新規サポート 更新
		7	3 変更内容 更新
		12	4 制限事項 更新
1.03	2022.01.20	5	2.3 新規サポート 更新
		9	3 変更内容 更新
		18	4 RENESAS TOOL NEWS の改修履歴 追加
		19	5 制限事項 更新
		23	6 注意事項 追加

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策

CMOS製品の取り扱いの際は静電気防止を心がけてください。CMOS製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れしないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後、切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS製品の入力がノイズなどに起因して、 V_{IL} (Max.) から V_{IH} (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 V_{IL} (Max.) から V_{IH} (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違えば、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含まれます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、変更、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、改造、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等

高品質水準： 輸送機器（自動車、電車、船舶等）、交通管制（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。

7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア/ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限られません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア/ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものいたします。
13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレストシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。