
RL78 ファミリ

R01AN1232JJ0100

Rev.1.00

CubeSuite+ スタートアップ・ガイド編

2012.06.28

要旨

この資料は、CubeSuite+を用いた RL78 ファミリ サンプルコードの活用方法、および RL78 ファミリの開発ツールの基本的な操作を、ユーザに理解していただくことを目的としています。この資料を読みながら、実際にツールを操作することにより、サンプルコードの活用方法や開発ツールの基本的な操作に対する理解を、より深めていただけます。

この資料では、RL78/G13 のサンプルコードを用いて説明します。

対象デバイス

RL78/G13

目次

1. はじめに.....	2
2. 開発環境について.....	2
3. オブジェクト・モジュール・ファイルの生成.....	4
4. E1 エミュレータを使用したオンチップ・デバッグ.....	11
5. 付録 コード生成.....	16

1. はじめに

• 対象者

この資料は、サンプルコードを初めて活用されるお客様、および RL78 ファミリの開発ツールを初めて使用されるお客様を対象としています。なお、開発ツールを使用する際に、Windows®の操作方法に関する基礎知識を必要とします。

2. 開発環境について

RL78 ファミリを利用する場合の開発手順と開発ツールを説明します。

2.1 開発手順と開発ツールの関係

製品の開発手順と開発ツールについて、次の図に示します。

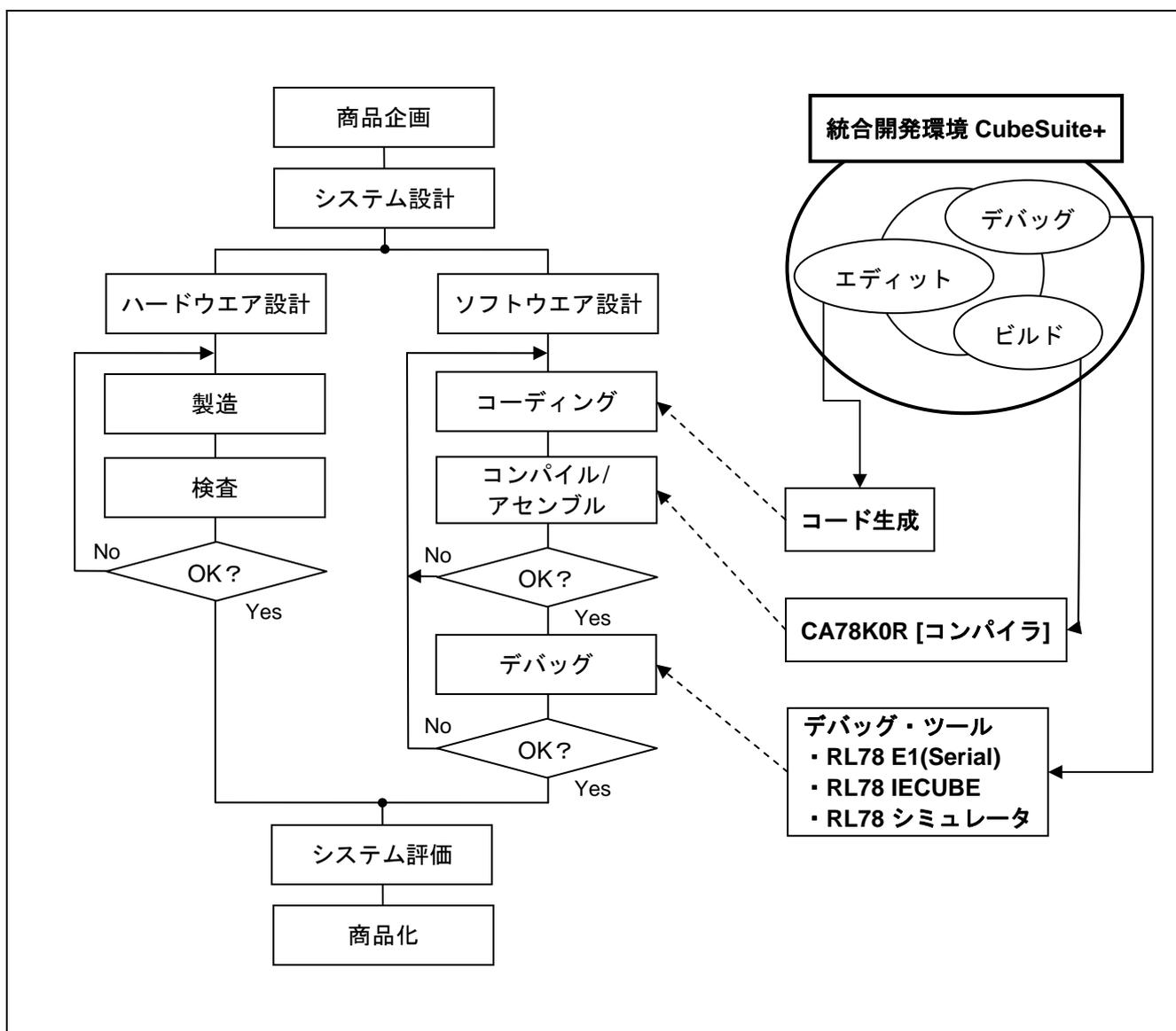


図 2-1 開発手順

2.2 使用する開発ツール

サンプルコードの動作確認をするためには、次の開発ツールを使用します。

2.2.1 CubeSuite+ (統合開発環境)

CubeSuite+は、Windows 上で使用する統合開発環境です。エディタ、コンパイラ、デバッガなどの開発ツールと連携し、効率的な開発ができます。おもに、次のものが用意されています。

a. デバイス依存情報ファイル

デバイス固有の情報が入ったファイルです。

b. CA78K0R (C コンパイラ)

RL78 ファミリ用の組み込み制御プログラムを C 言語で記述するために開発された汎用性、移植性の高い C コンパイラです。CA78K0R を Windows 上で使用するためには、CubeSuite+が必要です。

c. デバッグ・ツール

- RL78 E1(Serial)

E1 エミュレータを使用したオンチップ・デバッグを行います。

- RL78 IECUBE

フルスペック・エミュレータ IECUBE を使用したデバッグを行います。

- CubeSuite+ RL78 シミュレータ

RL78 ファミリ用の命令シミュレータです。

ホスト PC 上で、ソース・コードと TAU CH0~CH3 のインターバル・タイマ機能の動作をシミュレートできます。

d. 設計ツール

- 端子配置

端子配置図から GUI 操作で兼用端子の設定をする機能です。設計書への端子配置図の貼り付けや、端子配置の EXCEL 出力が可能です。

- コード生成

マイコン周辺機能 (タイマ, シリアル I/F, A/D など) を制御するデバイスドライバプログラムを, GUI 設定により自動生成できる機能です。

2.2.2 代表的な CPU 評価ボード

- CPU ボード (例 QB-R5F100LE-TB (RL78/G13_64 ピン/ROM64KB) など)
- Renesas Starter Kit (例 Renesas Starter Kit for RL78/G13 など)
- RL78G13 Stick スターターキット (RL78G13-STICK)

3. オブジェクト・モジュール・ファイルの生成

サンプルコードからオブジェクト・モジュール・ファイル (HEX, LMF ファイル) を生成するために、CubeSuite+を用いてサンプルコードをビルドします。

この章では、サンプルコードのプロジェクト作成からビルドまでの動作例を説明します。

CubeSuite+のプロジェクト作成方法の詳細については、「CubeSuite+統合開発環境 ユーザーズマニュアル 起動編 (R20UT0975)」を参照してください。

注意 RL78 ファミリのサンプルコードでは、プロジェクトで最初に指定したデバイス品名を変更することはできません。そのため、プロジェクトのデバイス品名を変更する場合は新規にプロジェクトを作成してください。

備考 1 ダウンロードしたサンプルコードのプロジェクトをそのまま使用する場合は、“ダウンロードしたファイル名+拡張子[* .mtpj]”のファイルをダブルクリックして CubeSuite+のプロジェクトを開いてください。その場合は、“3.2 オプション・バイトの設定”からの手順で実施してください。

2 LMF (ロード・モジュール) ファイルとは、シンボル等のデバッグ情報を含むオブジェクト・ファイルです。

3 HEX (ヘキサ) ファイルとは、LMF ファイルから生成したデバッグ情報を含まない HEX 形式のオブジェクト・ファイルです。

3.1 新規プロジェクトの作成とサンプルコードの読み出し

3.1.1 サンプルコードの準備

ダウンロードしたファイルを解凍し、任意の場所（フォルダ）に保存してください。

3.1.2 CubeSuite+の起動と新規プロジェクトの作成準備

CubeSuite+を起動し、ツールバーの **スタート(S)** をクリックすると、スタートパネルがオープンします。スタートパネルがオープンしたら、図 3-1 に示す“新しいプロジェクトを作成する”の **GO** ボタンをクリックし、新規にプロジェクトを作成します。

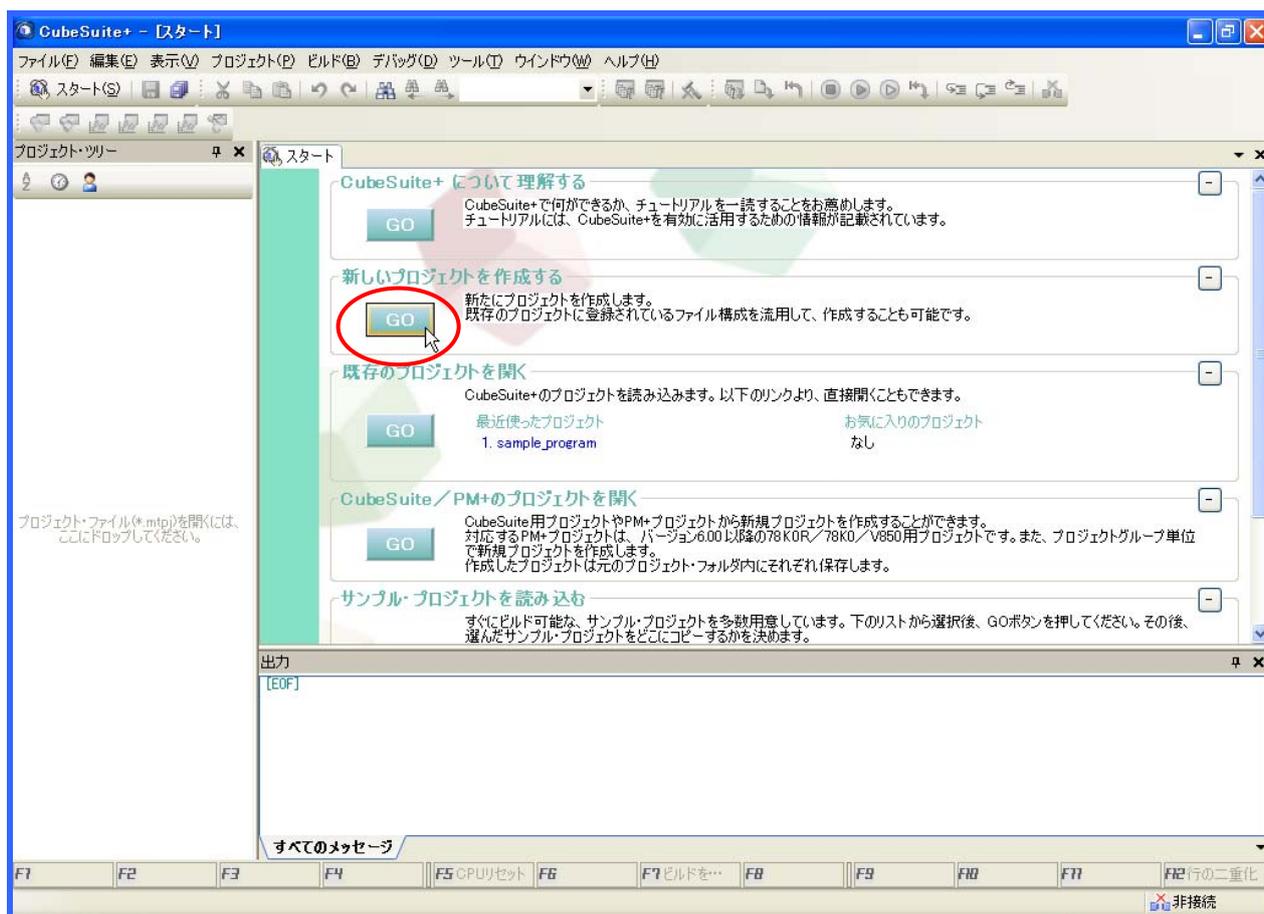


図 3-1 CubeSuite+ のスタートパネル

3.1.3 プロジェクトの指定

図 3-2 に示す“プロジェクト作成ダイアログ”画面が表示されたら、次の設定を行います。

プロジェクトで使用するデバイス品名を選択します。^注

プロジェクト名とプロジェクト・ファイルの作成場所を指定します。

(ただし、指定した作成場所の下にプロジェクト名のフォルダを作成しない場合は、[プロジェクト名のフォルダを作成する(A)]のチェックを外してください。)

[既存のプロジェクトのファイル構成を流用する(S)]をチェックし、[流用元のプロジェクト(P)]に流用元のサンプルコードのプロジェクト・ファイル名を指定します。ここで指定することで、流用もとのソース・ファイルが自動的に読み込まれます。

また、指定した流用元プロジェクトのファイル構成をコピーして使用する場合は、[プロジェクト・フォルダ以下の構成ファイルをコピーして流用する(Q)]にチェックをしてください。

[作成(C)]ボタンをクリックして、プロジェクトを作成します。

【注】 使用する品種名が選択欄にない場合、デバイス依存情報ファイルをインストールしてください。

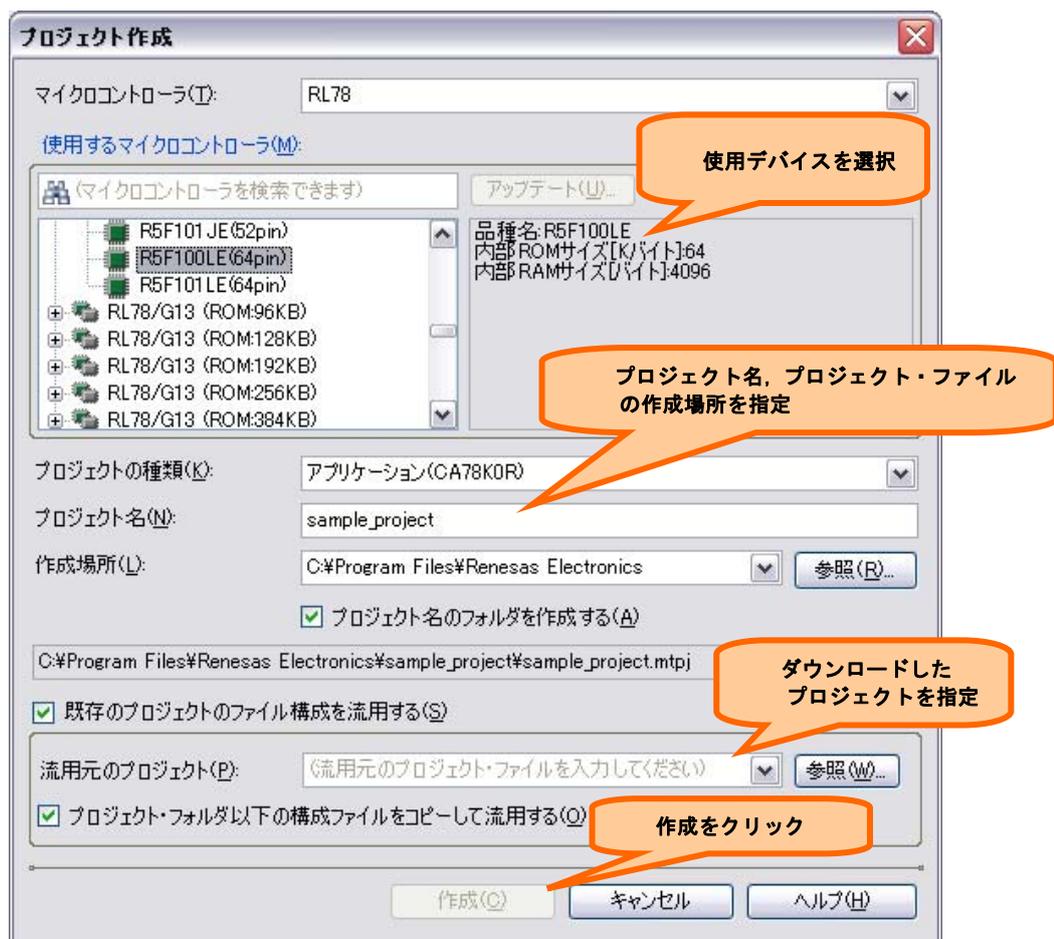


図 3-2 プロジェクト作成ダイアログ

3.2 オプション・バイトの設定

オプション・バイトは、ユーザ・オプション・バイトとオンチップ・デバッグ・オプション・バイトで構成されています。電源投入時またはリセット起動時に、マイコンは自動的にオプション・バイトの設定値を参照して、動作を開始します。

ここでは、RL78/G13の設定を例に挙げて、CubeSuite+の[CA78K0R (ビルド・ツール)]のプロパティから、オプション・バイトを設定する手順を説明します。

3.2.1 オンチップ・デバッグ・オプション・バイトの設定

オンチップ・デバッグ・オプション・バイトを設定することにより、マイコンの持つオンチップ・デバッグ機能を有効にします。

[CA78K0R のプロパティ]画面が表示されたら、画面下の[リンク・オプション]タグを選択します。

[デバイス]カテゴリを開き、[オンチップ・デバッグを設定する] プロパティで [はい(-go)] を選択すると、[オンチップ・デバッグ・オプション・バイト制御値] プロパティ、[デバッグ・モニタ領域開始アドレス] プロパティ、[デバッグ・モニタ領域サイズ[バイト]] プロパティが表示されます。

[オンチップ・デバッグ・オプション・バイト制御値] プロパティ：

オンチップ・デバッグ・オプション・バイトの制御値を 0x なしの 16 進数で指定します。

(RL78/G13 の場合、オンチップ・デバッグ動作禁止(0x04)、オンチップ・デバッグ動作許可(0x84/0x85)を設定します)

[デバッグ・モニタ領域開始アドレス] プロパティ：

デバッグ・モニタ領域の開始アドレスを 0x なしの 16 進数で指定します。

(デフォルト：内部 ROM のエンド・アドレス-1024 +1)。

[デバッグ・モニタ領域サイズ[バイト]] プロパティ：

デバッグ・モニタ領域のサイズを 10 進数で指定します。指定可能な値の範囲は、0 ~1024 です (デフォルト：512)。

3.2.2 ユーザ・オプション・バイトの設定

ユーザ・オプション・バイトを設定することにより、ウォッチドッグ・タイマ(WDT)の設定、電圧検出回路(LVD)の設定、システム予約領域の設定を行います。

ユーザ・オプション・バイトの設定も、[リンク・オプション] タブの [デバイス] カテゴリで行います。[ユーザ・オプション・バイトを設定する] プロパティで [はい(-gb)] を選択すると、[ユーザ・オプション・バイト値] プロパティが表示されます。

[ユーザ・オプション・バイト値] プロパティ：

ユーザ・オプション・バイト値を 0x なしの 16 進数で指定します。(RL78/G13 の場合、WDT 停止(0xEF), LVD リセットモード 2.81V (0x7F), 動作クロック $f_{\text{H}}=32\text{MHz}$ (HS モード)(0xE8)を設定します)

備考 オプション・バイトの設定方法についての詳細は、「CubeSuite+統合開発環境 ユーザーズマニュアル RL78,78K0R ビルド編 (R20UT0730)」を参照してください。

注意 オプション・バイトは、リンク・オプションで設定する方法以外に、ソース・コード上でプログラミングする方法があります。なお、同時にオプション・バイトを設定した場合は、リンク・オプションで設定した値が有効になります。

3.3 オブジェクト・モジュール・ファイルの生成（ビルド）

サンプルコードからオブジェクト・モジュール・ファイル（HEX, LMF ファイル）を生成するために、CubeSuite+を用いてサンプルコードをビルドします。

ビルドには、次の種類があります。

種類	説明
ビルド	ビルド対象ファイルのうち、更新されたファイルのみビルドを実行します。
リビルド	ビルド対象のすべてのファイルのビルドを実行します。
ラピッド・ビルド	ビルド設定の変更と平行してビルドを実行します。
パッチ・ビルド	プロジェクトが持つビルド・モードを一括してビルドを実行します。

備考 ビルド方法の詳細は、「CubeSuite+統合開発環境 ユーザーズマニュアル RL78,78K0Rビルド編 (R20UT0730)」を参照してください。

ここでは、“ビルド”を例に挙げて、HEXファイルの作成手順を説明します。

3.3.1 HEX ファイルの指定

プロジェクトを開いたら、次の手順で HEX ファイル名を設定します。

左側の“プロジェクト・ツリー”にある[CA78K0R (ビルド・ツール)] (図 3-3 中に赤い下線で示す) をダブルクリックして、[CA78K0R のプロパティ]画面を表示させます。

[CA78K0R のプロパティ]画面が表示されたら、画面下の[オブジェクト・コンバート・オプション]タグを選択します。

[ヘキサ・ファイル]カテゴリの[ヘキサ・ファイル名]プロパティに「任意のファイル名+拡張子 (.hex)」を入力します。（デフォルト設定は、プロジェクト・ファイル名と同じファイル名になっています）

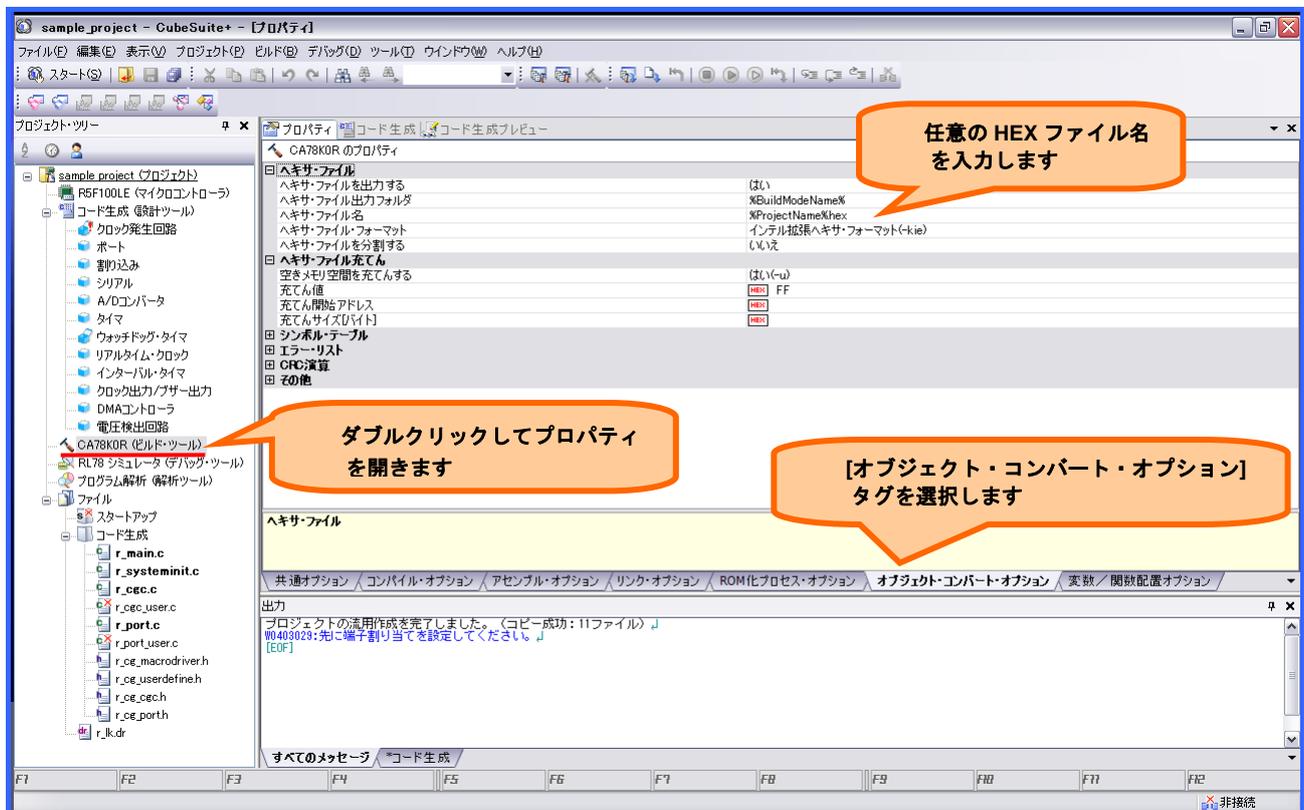


図 3-3 オブジェクト・コンバータ・オプション画面

3.3.2 ビルドの実行

図 3-4 中の枠で示す  ボタンをクリックすることで、ビルドを実行します。ビルドを実行すると、ソース・ファイルから HEX ファイル (*.hex) が生成され、プロジェクト・ファイルのフォルダ内に格納されます。ビルドが終了すると、CubeSuite+画面の出力画面に[ビルド終了 (エラー:0 個, 警告:0 個)]のメッセージが表示されます。

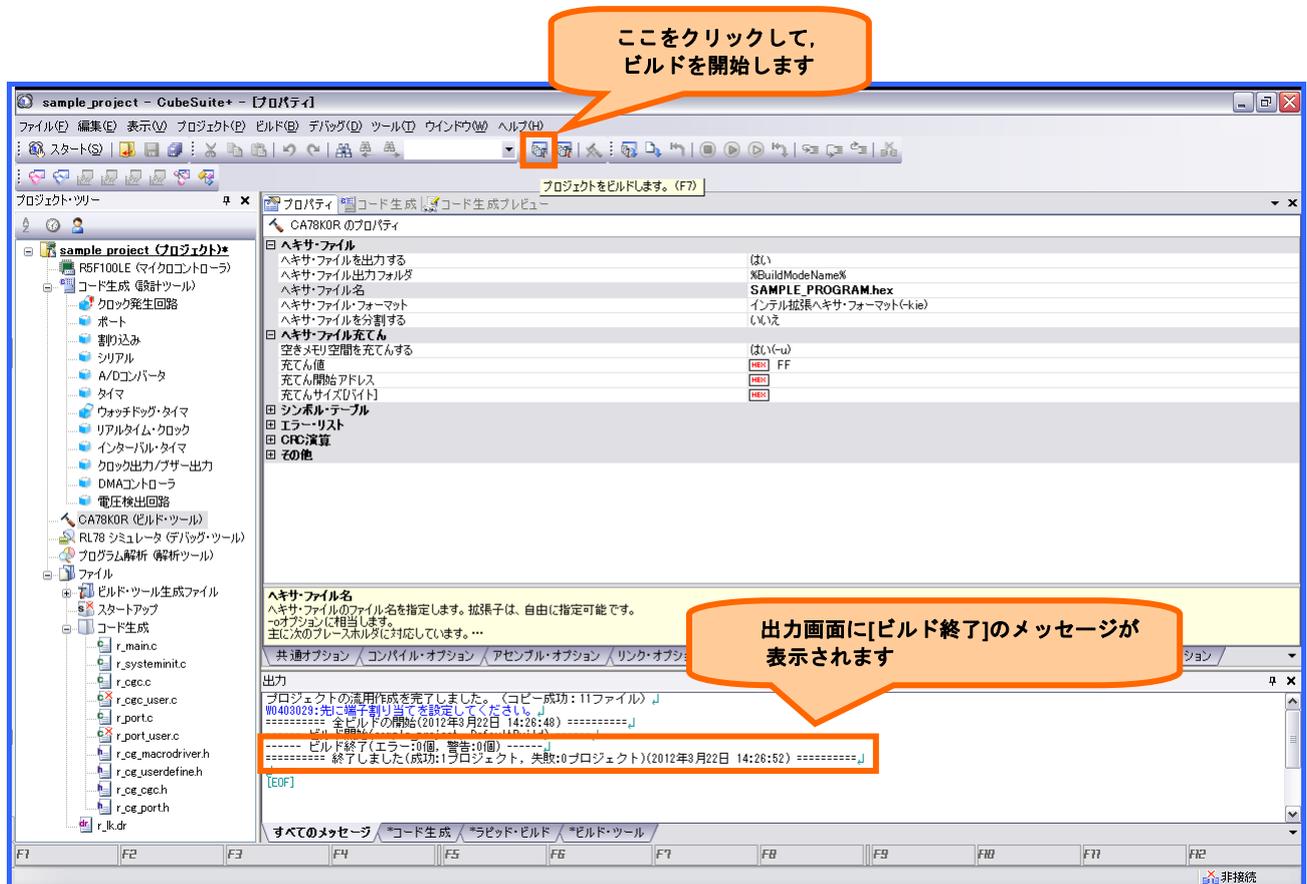


図 3-4 ビルドの実行画面

備考 HEX ファイル (*.hex) は、ルネサス製の E1 エミュレータとフラッシュ書き込みソフトウェア Renesas Flash Programmer (RFP)を使用することで、マイコンのフラッシュメモリへプログラムを書き込むことができます。フラッシュ書き込みの詳細は、「Renesas Flash Programmer フラッシュ書き込みソフトウェア ユーザーズマニュアル (R20UT0599)」を参照してください。

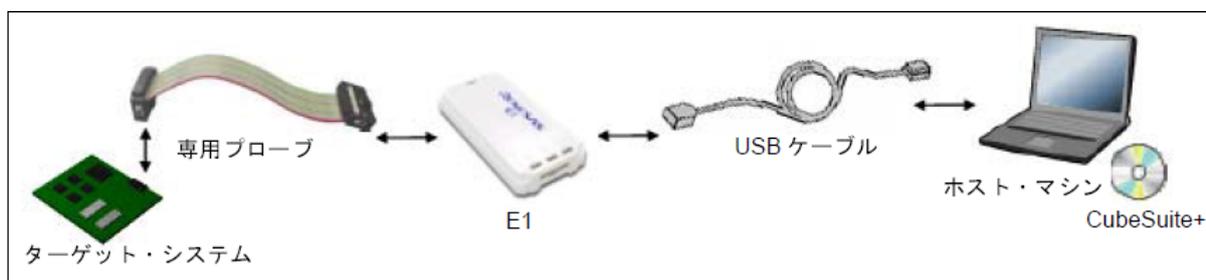
4. E1 エミュレータを使用したオンチップ・デバッグ

CubeSuite+ が提供するデバッグ機能を使用することにより、RL78 ファミリ用に開発されたプログラムを、効率良くデバッグすることができます。

ここでは、ルネサス製 E1 エミュレータ を使用したオンチップ・デバッグの設定例を説明します。CubeSuite+ が提供するデバッグ機能の詳細については、「CubeSuite+統合開発環境 ユーザーズマニュアル RL78 デバッグ編 (R20UT0978)」を参照してください。

4.1 デバッグを始める前の準備

図 4-1 に示すように、ホスト・マシン、E1 エミュレータ、および必要に応じてターゲット・ボードを接続します。接続方法についての詳細は、「E1/E20 エミュレータ ユーザーズマニュアル 別冊 RL78 接続時の注意事項 (R20UT1994)」を参照してください。



【注】 ターゲット・システムとの通信方式として、シリアル通信のみサポートしています (JTAG 通信は使用不可)。

図 4-1 ホスト・マシンとデバッグ・ツールとの接続例【E1】

4.2 オンチップ・デバッグの設定手順

4.2.1 デバッグ・ツールの選択/切り替え

プロジェクト内で使用するデバッグ・ツールを選択します。使用するデバッグ・ツールの選択/切り替えは、プロジェクト・ツリーパネル上の [RL78 デバッグ・ツール名(デバッグ・ツール)] ノードを右クリックすることで表示されるコンテキスト・メニューにより行います (図 4-2 参照)。

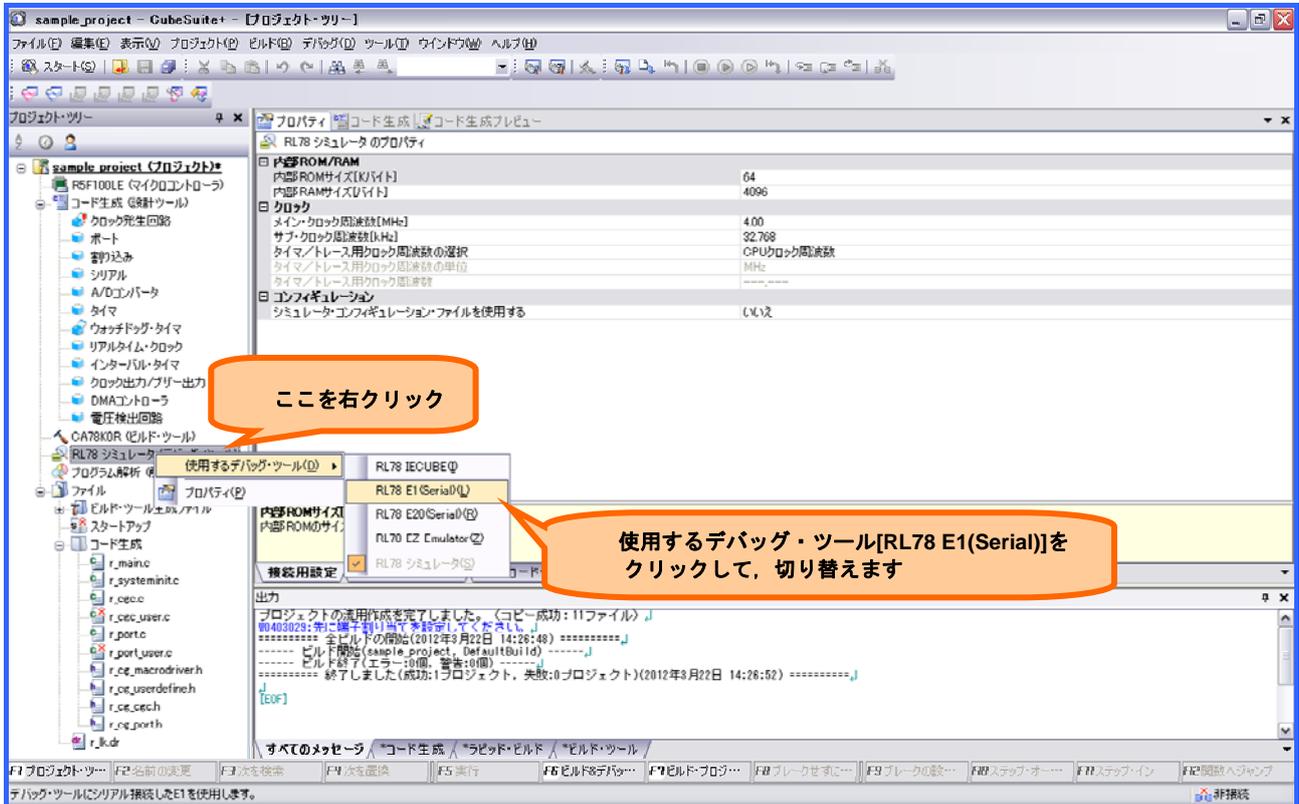


図 4-2 デバッグ・ツールの選択/切り替え【E1】

4.2.2 オンチップ・デバッグの動作環境の設定

E1 エミュレータを使用する場合の動作環境の設定を図 4-3 のプロパティパネルで行います。ここでは、「ターゲット・ボードと接続」の設定を行います。その他項目の設定については、「CubeSuite+統合開発環境 ユーザーズマニュアル RL78 デバッグ編 (R20UT0978)」を参照してください。

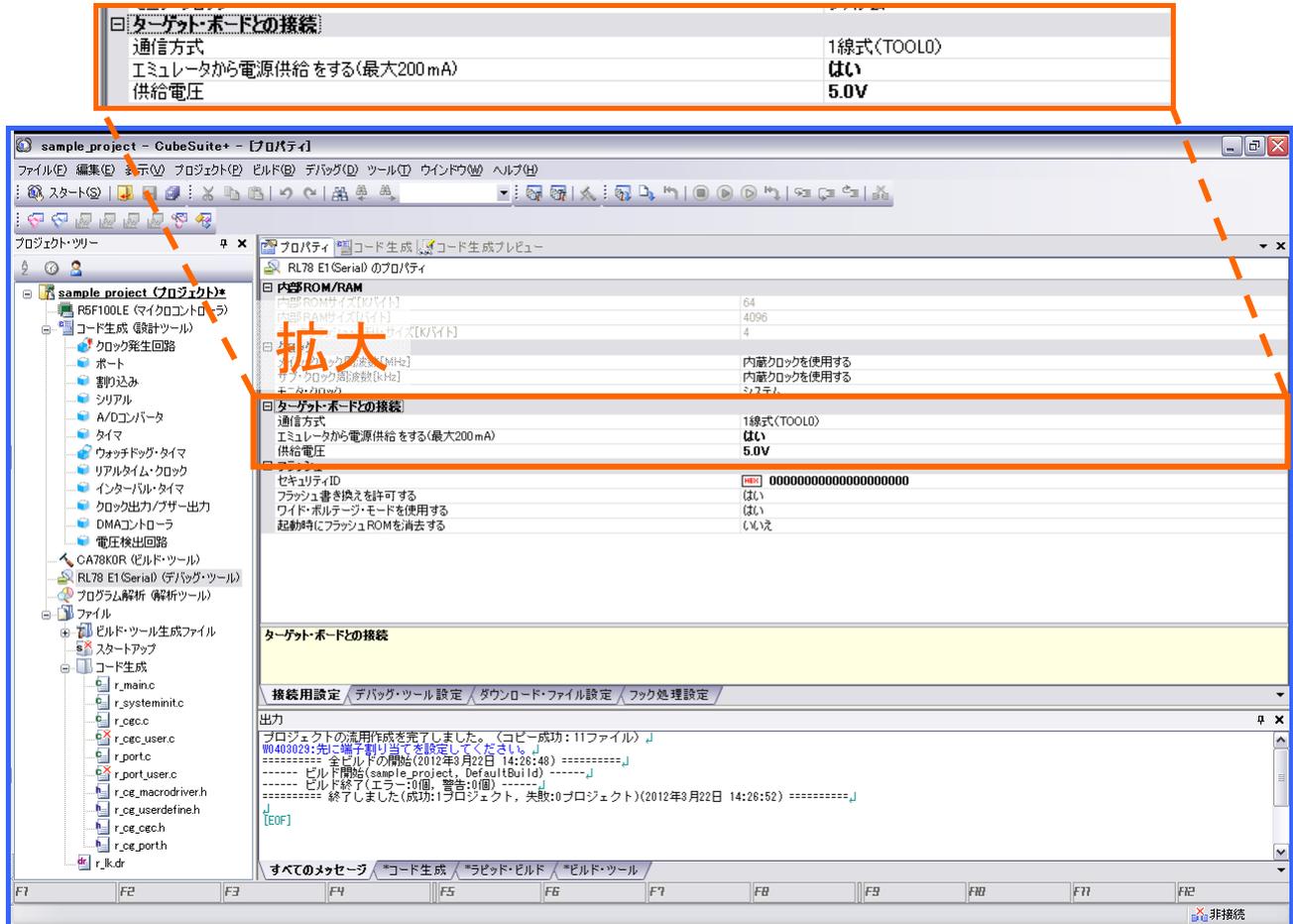


図 4-3 動作環境設定【E1】（プロパティパネル）

a. 通信方式

E1 エミュレータ がターゲット・システム上の RL78/G13 とシリアル通信を行う際の通信方式を指定します。1 線式(TOOL0) を選択してください。

b. エミュレータから電源供給をする(最大 200mA)

E1 エミュレータからターゲット・システムへ電源を供給するか否かを指定します。電源を供給する場合は [はい] を選択してください（デフォルトでは [いいえ] が指定されます）。なお、エミュレータから電源供給しない場合は、[いいえ] を選択して、外部から電源を供給してください。

c. 供給電源

このプロパティは、[エミュレータから電源共有をする(最大 200mA)] プロパティにおいて、[はい] を指定した場合のみ表示されます。ターゲット・システムへ供給する電圧を指定します。

例：5.0V …ターゲット・システムへ 5.0V の電圧を供給します。

【注】 E1 エミュレータと接続中にこのカテゴリ内のプロパティを変更することはできません。

4.2.3 CubeSuite+にデバッグ・ツールを接続する

デバッグ・ツールバーの  ボタンをクリックすることで、デバッグ・ツールと接続したのち、LMF ファイルのダウンロードを実行します。

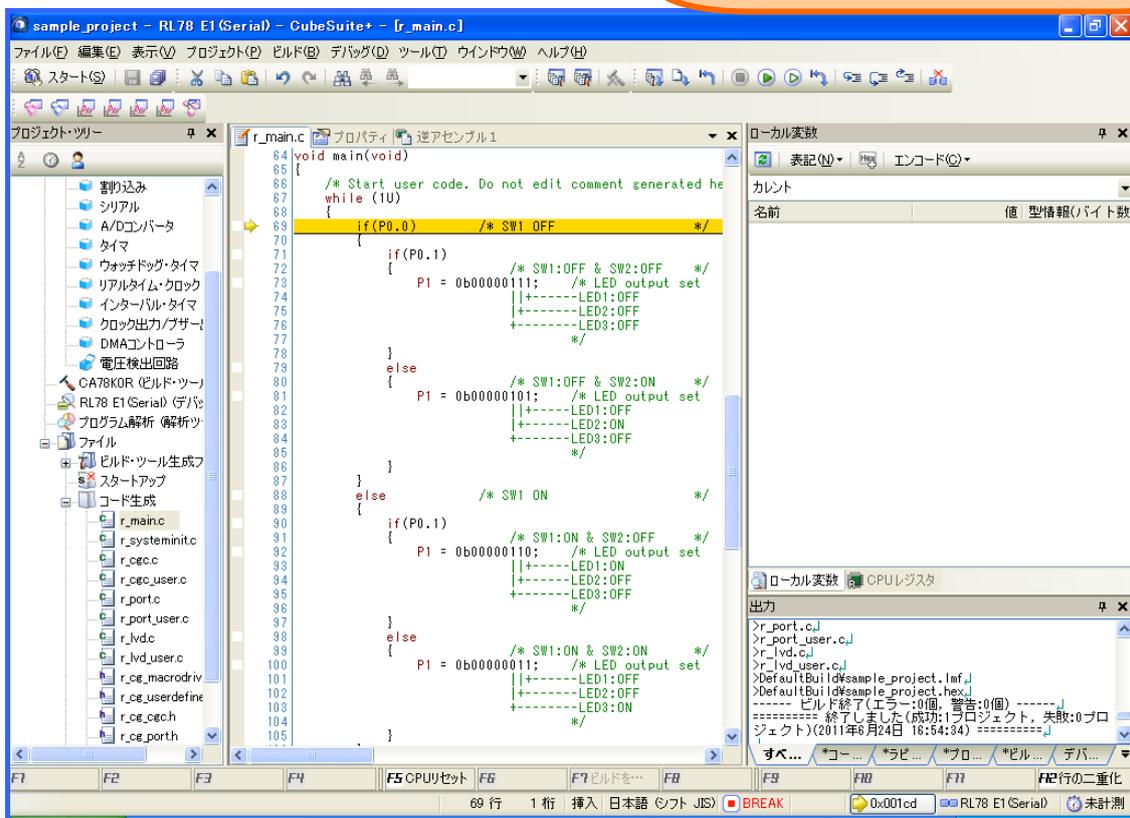
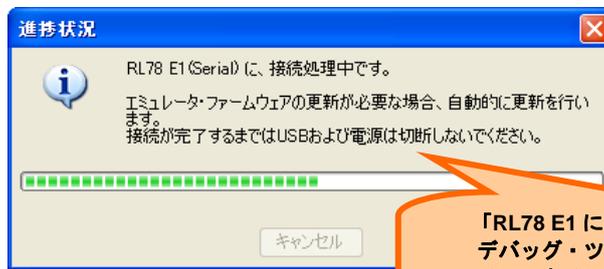
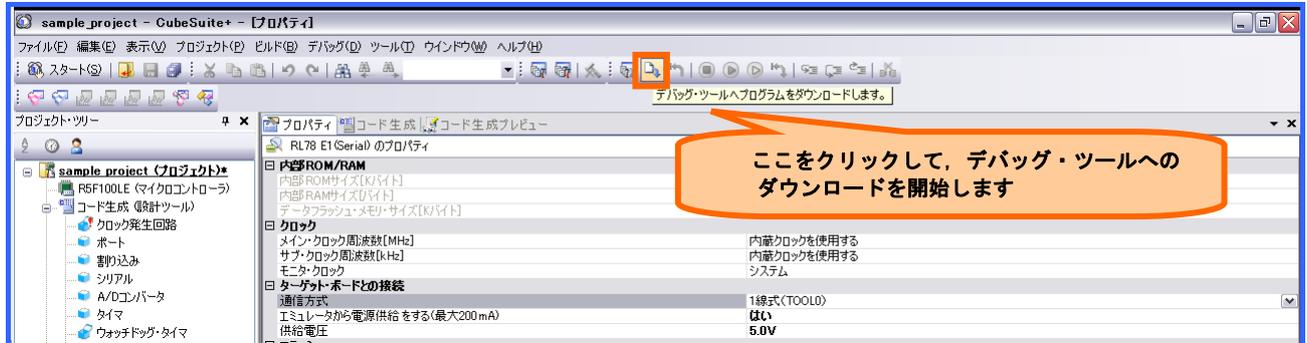


図 4-4 デバッグ・ツールへのダウンロード例

4.2.4 CubeSuite+からデバッグ・ツールを切断する

デバッグ・ツールバーの  ボタンをクリックすることにより、現在接続しているデバッグ・ツールとの通信を切断します。デバッグ・ツールと切断すると、メイン・ウインドウのステータスバーが、 のように変化します。



図 4-5 デバッグ・ツールから切断したステータスバー

5. 付録 コード生成

5.1 コード生成の使用方法

CubeSuite+のコード生成は、RL78 ファミリが提供している周辺機能（クロック発生回路の機能、ポートの機能など）を制御する際に必要な情報を CubeSuite+ のパネル上で設定することにより、対応するソース・コード（デバイス・ドライバ・プログラム）を出力します。

ここでは、コード生成の操作手順を説明します。コード生成の詳細は、「CubeSuite+統合開発環境 ユーザーズマニュアル RL78 設計編 (R20UT0976)」を参照してください。

5.1.1 コード生成パネルでの設定

使用する周辺機能（クロック発生回路の機能、ポートの機能など）のコード生成パネルをオープンして、必要な情報を設定します。

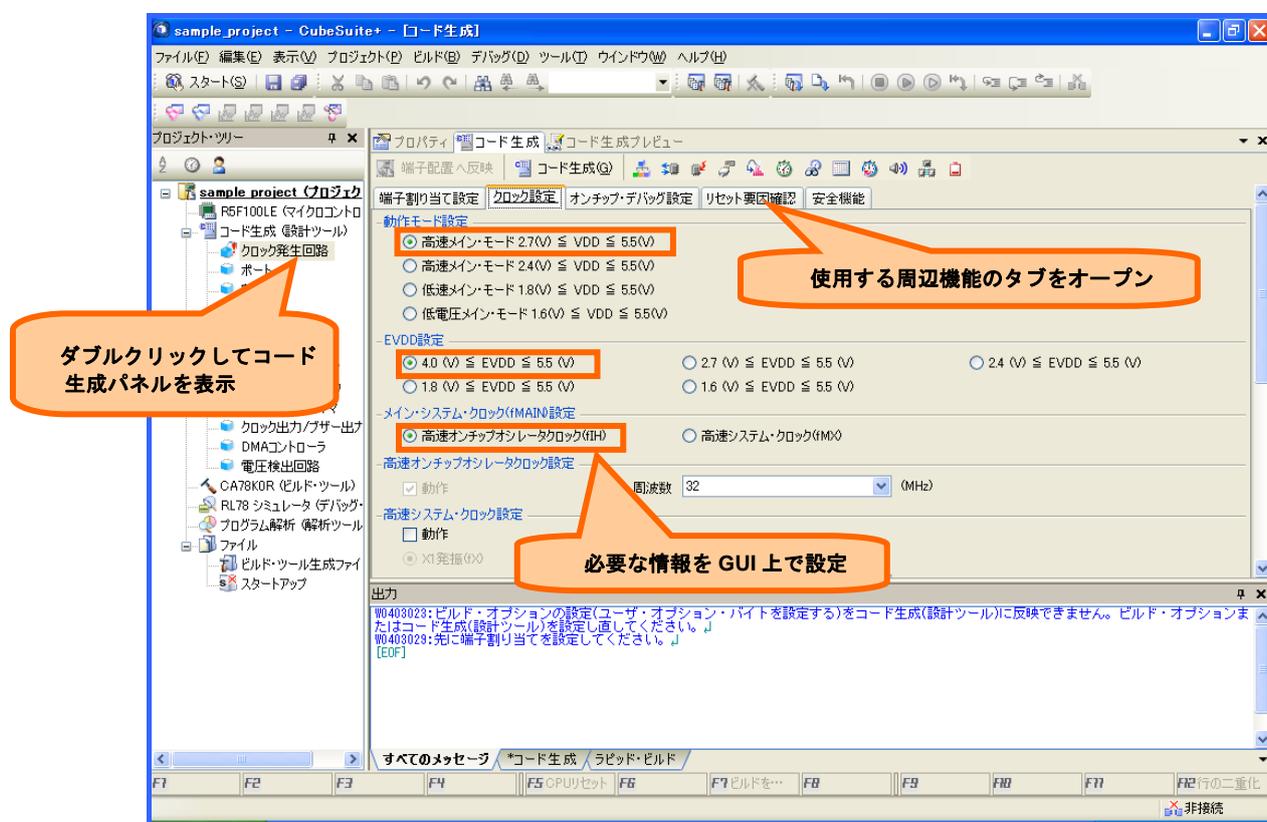


図 5-1 コード生成パネルの設定例

備考 コード生成パネルで入力間違いや必須項目の未入力、設定する端子が他の機能で既に使われているなどの場合に、情報として誤っていることを示す  アイコンを該当箇所に表示します。  アイコン上にマウス・カーソルを移動した際には、これらの警告に関する情報（解決するためのヒント）がポップアップ表示されます。

5.1.2 ソース・コードの確認

[コード生成プレビュー] を選択してオープンしたコード生成プレビューパネルで、コード生成パネル上で設定した情報に応じたソース・コードを確認できます。

5.1.3 ソース・コードの出力

コード生成パネルの[コード生成(G)] ボタンをクリックし、ソース・コードを出力します。

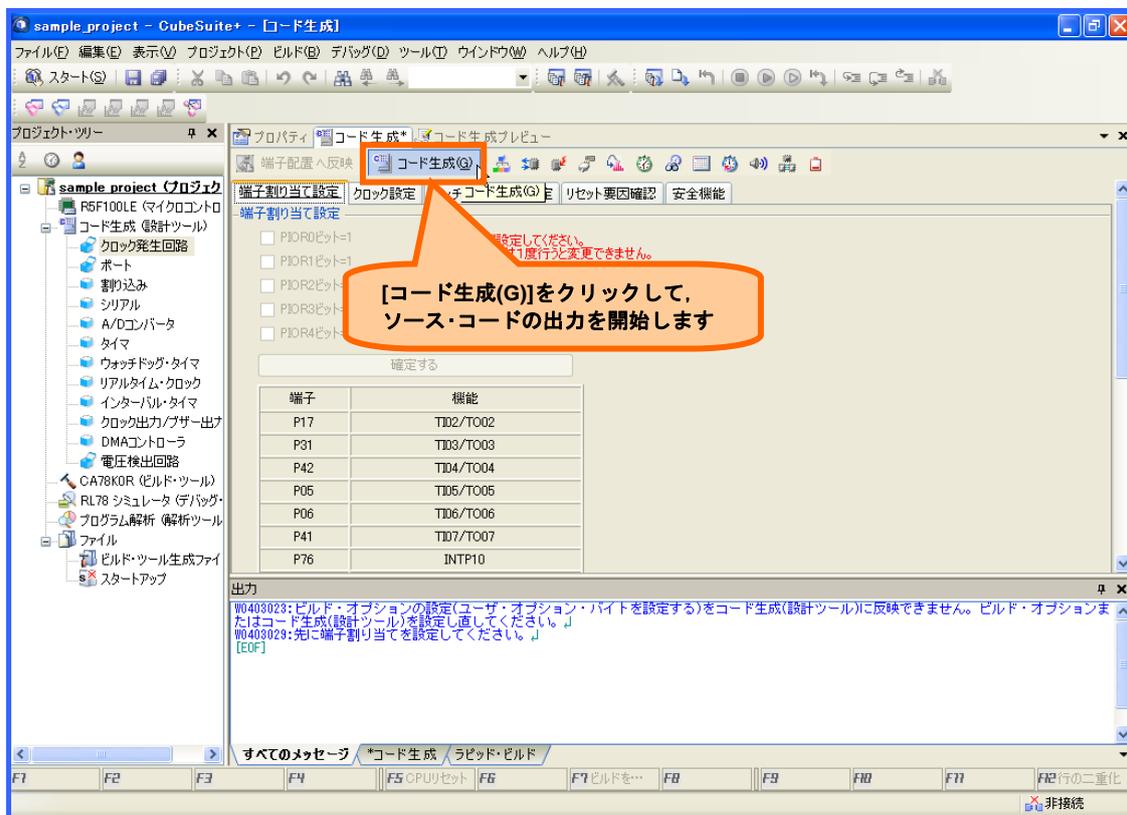


図 5-2 ソース・コードの出力例

5.2 コード生成が出力するソース・コード規則

5.2.1 ソース・コードの構成

以下に、コード生成が出力するファイルの一覧を示します。

表 5.1 コード生成の出力ファイル一覧

出力単位	ファイル名	出力内容
各周辺機能	r_cg_周辺機能名.c	初期化関数, API 関数 ^注
	r_cg_周辺機能名_user.c	割り込み関数, コールバック関数
	r_cg_周辺機能名.h	レジスタへの代入値マクロを定義
プロジェクト	r_main.c	main 関数
	r_systeminit.c	各周辺機能の初期化関数コール R_CGC_Get_ResetSource のコール
	r_cg_macrodriver.h	全ソース・ファイルで共通使用する マクロを定義
	r_cg_userdefine.h	空ファイル (ユーザ定義用)
	r_lk.dr	リンク・ディレクティブ

【注】 API 関数の詳細は、「CubeSuite+統合開発環境 ユーザーズマニュアル RL78 設計編 (R20UT0976)」を参照してください。

5.2.2 ソース・コード規則

ここでは、“サンプルコード A/D コンバータ編”のソース・コードを例に挙げて命名規則を説明します。

main.c

```

<リスト省略>
/*****
Exported global variables and functions (to be accessed by other files)
*****/
/* Start user code for global. Do not edit comment generated here */
uint16_t    result_buffer;          /* AD converter result buffer */

/* End user code. Do not edit comment generated here */
/*****
* Function Name: main
* Description : This function implements main function.
* Arguments : none
* Return Value : none
*****/
void main(void)
{
/* Start user code. Do not edit comment generated here */
uint8_t count;                    /* Loop counter */

result_buffer = 0U;                /* Initialize result buffer */

DI();                              /* Disable interrupt */
R_ADC_Set_OperationOn();          /* Enable comparator operation */

/* ---- stabilization wait time(about 1us) ---- */
for (count=0U; count<3U; count++)
{
NOP();
}
R_ADC_Start();                    /* Start AD converter */

<リスト省略>
}
/* End user code. Do not edit comment generated here */
}
    
```

備考 これらコメント文の文字列は変更しないでください。

r_adc.c

```

<リスト省略>
void R_ADC_Create(void)
{
    ADCEN = 1U; /* supply AD clock */

<リスト省略>
    /* Set ANIO pin as analog input */
    PM2 |= 0x01U;
    ADM0 = 28_AD_CONVERSION_CLOCK_5 | 00_AD_TIME_MODE_NORMAL_1 | 00_AD_OPERMODE_SELECT;
    ADM1 = 00_AD_TRIGGER_SOFTWARE | 00_AD_CONVMODE_CONSELECT;
<リスト省略>
/*****
* Function Name: R_ADC_Start
* Description : This function starts the AD converter.
* Arguments : none
* Return Value : none
*****/
void R_ADC_Start(void)
{
    ADIF = 0U; /* clear INTAD interrupt flag */
    ADMK = 0U; /* enable INTAD interrupt */
    ADCS = 1U; /* enable AD conversion */
}
/*****
End of function R_ADC_Start
*****/

```

2進定数

レジスタ(ADM0 レジスタ) に代入する値は、
下線で示した Byte の総和(OR)が入ります

備考 これらコメント文の文字列は変更しないでください。

a. グローバル変数の定義

グローバル変数の定義は，“ /* Start user code for global. Do not edit comment generated here */ ”以降に記述してください。なお，“ /* End user code. Do not edit comment generated here */ ”までの範囲は、再度コード生成を行っても書き換わることはありません。

注意 この領域以外でグローバル変数定義した場合、再度コード生成を行った際に、CubeSuite+によって上書き消去されます。

b. ユーザ・プログラムの記述

ユーザ・プログラムは，“ /* Start user code. Do not edit comment generated here */ ”以降に記述してください。なお，“ /* End user code. Do not edit comment generated here */ ”までの範囲は、再度コード生成を行っても書き換わることはありません。

注意 この領域以外でユーザ・プログラムを記述した場合、再度コード生成を行った際に、CubeSuite+によって上書き消去されます。

c. API 関数

API 関数は、ユーザ・プログラムでコールして使用してください。それにより、コード生成で出力した各周辺機能のレジスタ設定や動作処理を行います。

API 関数の詳細は、「CubeSuite+統合開発環境 ユーザーズマニュアル RL78 設計編 (R20UT0976)」を参照してください。

d. 2進定数

ビット・レジスタに直接記述する場合、コード生成のソース・コードは2進定数で記述します。

表 5.1 2進定数の型 一覧

添字なし2進数	int, unsigned int, long int, unsigned long int
u, またはUの添字付き	unsigned int, unsigned long int
l, またはLの添字付き	long int, unsigned long int
u, またはUの添字, およびl, またはLの添字付き	unsigned long int

e. レジスタ設定

コード生成で出力したレジスタは、“r_cg_ 周辺機能名.h”に自動的に定義した変数の総和(OR)を代入して設定しています。

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<http://japan.renesas.com/>

お問い合わせ先

<http://japan.renesas.com/contact/>

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。

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、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

1. 未使用端子の処理

【注意】未使用端子は、本文の「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。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】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. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレスがあります。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。

4. クロックについて

【注意】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。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違っていると、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。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。

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連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の侵害等に関し、当社は、一切その責任を負いません。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。
3. 当社製品を改造、改変、複製等しないでください。
4. 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。お客様の機器の設計において、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には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、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に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
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
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障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おりますが、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、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害など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、機器またはシステムとしての出荷保証をお願いいたします。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、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、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。お客様がかかる法令を遵守しないこと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。
11. 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固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

- 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とその総株主の議決権の過半数を直接または間接に保有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。



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<http://japan.renesas.com/contact/>