

初心者のための RL78 入門コース（第 1 回：開発環境）

RL78 は Renesas では、下位に位置付けされた MCU（マイクロ・コントローラ・ユニット）です。ファミリーの中には、8 ビットの RL78/G10 と呼ばれるシリーズもありますが、基本的には 16 ビットの MCU です。

RL78 は、応用分野によって内蔵された周辺機能が異なり、汎用の G シリーズ、LCD 制御可能な L シリーズ、産業向けの I シリーズ、自動車向けの F シリーズ他があります。その分類以外に USB 対応の RL78/xxC やスマートアナログと呼ばれる RL78/xxE などもあります。

ここでは汎用の G シリーズの中で RL78 の最初の製品であり、メインストリームと思われる RL78/G13 を対象にしてプログラムを中心に進めていきます。

第 1 回目と言うことで、環境（CS+CC-RL）の準備とその簡単な使い方を中心に説明していきます。ページ数の関係で、詳しい説明ができませんが、簡単なデバッグまでの一通りのことはできるかなと思っています。

次回は、またシミュレータを用いて、ポートの入力を含めたポート制御を行います。その中で、タイマをインターバル・タイマとして使う方法などを説明する予定です。その後、実際のボードを使うための説明を行います。

今回の内容

1. はじめに	P2
2. 開発環境	P2
3. CS+CC-RL のインストール	P2
4. CS+CC-RL の起動	P9
5. コード生成を利用したプログラムの作成方法（共通的な内容）	P14

以下、次回（第 2 回）の内容

6. コード生成を利用した実際のプログラム作成	P19
7. BlueBoard-RL78/G13_64pin での動作確認	P34

1. はじめに

RL78 が登場して、最初の頃に使われていた人たちから新しい世代に切り替わっている時期ではないかと思います。RL78 のマニュアルはかなり親切に記載されているのですが、あまりに厚すぎて、慣れていないと関係する項目を探すのが大変です。また、RL78 に関しては入門書が 1 冊だけで、比較的好く書いてあるとは思いますが、殆どの人が使用するであろうコード生成に関してちょっと紹介しているだけで、取り扱いが不十分だと思っています。

そこで、これまでの経験をまとめて、初心者向けの実践的な内容のコンテンツをつくることにしました。

使用するデバイスは RL78 のメインストリームと考えられる RL78/G13 とします。

2. 開発環境

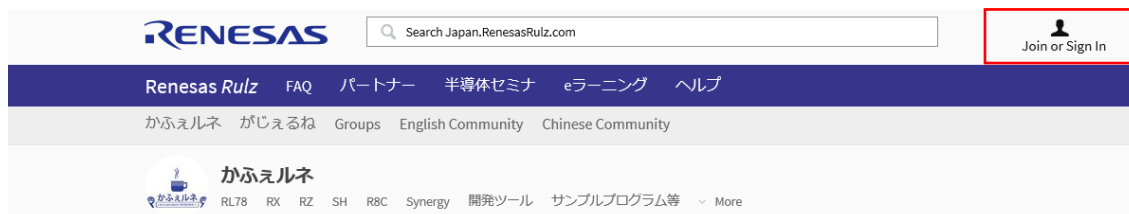
使用する開発環境としては、CS+CC-RL を使います（e2studio はバージョンアップでのトラブルが結構あるようなのでここでは使いません）。プログラムのデバッグも行うので、E1 または E2lite を使用します。すでに、E1 をもっているので、E1 を使いますが、これから入手するならば、安い E2lite を推奨します。また、必要に応じて CS+に含まれているシミュレータも使います。

また、ターゲットとなるハードウェアについては、できるだけ出来合いのものを使います。簡単に調べた範囲では、秋月電子で販売している「Blue Board-RL78/G13_64pin」が一番安そうだったので、これを最初のターゲットとします。LED（1 個）、スイッチ、可変抵抗が付いていて、LCD も使えますが、そのうちに、機能が不足するので、そのときはブレッド・ボードを使います。他に適当なボードがあれば、そちらへの移植も考えてみます。


3. CS+CC-RL のインストール

CS+CC-RL は無償評価版がルネサスの HP からダウンロードできます。ダウンロードには、My Renesas のアカウントが必要なので、最初にアカウントを申請しましょう。

下に示すかふえルネの TOP ページで、右上にある「Join or Sign in」をクリックします。



すると、My Renesas のログイン画面（<https://update.renesas.com/SSO/login>）が開きます。

 Enter your Email Address and password and click "Login" to access your account.

Email Address

Password

Login

- [Forgot your password?](#)
- [Don't have an account yet?Register here](#)

ここで、下側の「Don't have an account yet?Register here」をクリックするとアカウント作成ページが開くので、そこで指示に従って、必要な項目を入力していくことでアカウントを作成してください。

アカウントを作成したら、下記 URL のルネサスの「評価版ソフトウェアツール」のページを開きます。

https://www.renesas.com/ja-jp/products/software-tools/evaluation-software-tools.html?cid=tb_d370808try

開いたら、下記の部分までスクロールします。

統合開発環境 CS+ (旧CubeSuite+)

製品名	仕様・性能	試用期限
統合開発環境 CS+ for CC (RL78, RX, RH850用) 製品ページ 評価版ダウンロード	試用期限内は製品版(professional版)と同じ。 試用期限を過ぎると各MCUにより以下の制限があります。 [RH850ファミリ] リンクサイズを256Kバイト以内に制限しています。professional版の機能は使用できません。 [RXファミリ] リンクサイズを128Kバイト以内に制限しています。professional版の機能は使用できません。 [RL78ファミリ] リンクサイズを64Kバイト以内に制限しています。professional版の機能は使用できません。 コンパイラ※、デバuggを同梱。 ※ CC-RL, CC-RX, CC-RH	60日 初めて評価版ソフトウェアツールをインストールした後、最初にビルドを行った日から60日間の試用期間があります。 試用期間内は、機能に制限はありません。 61日目以降は、リンクサイズ、professional版の機能が制限されます。

ここで、「評価版ダウンロード」をクリックして、ツールダウンロードのページを開きます。


原稿作成時点では、4件が表示されるので、「【無償評価版】統合開発環境 CS+ for CC V6.00.00 (一括ダウンロード版)」をクリックしてダウンロードします。

ツールダウンロード (4)

フィルタ

製品/開発環境	<input checked="" type="checkbox"/> 統合開発環境 CS+ (旧 CubeSuite+) + 絞り込み
データの種類	<input checked="" type="checkbox"/> 無償評価版 + 絞り込み

 【無償評価版】 統合開発環境 CS+ for CA,CX V4.00.01 (分割ダウンロード版)
Jul.20.17

 【無償評価版】 統合開発環境 CS+ for CA,CX V4.00.01(一括ダウンロード版)
Jul.20.17

 【無償評価版】 統合開発環境 CS+ for CC V6.00.00 (分割ダウンロード版)
Jul.20.17

 【無償評価版】 統合開発環境 CS+ for CC V6.00.00 (一括ダウンロード版)
Jul.20.17

1ページの件数 20 1 / 1ページ

ダウンロードのページが開くので、スクロールして、一番下まで行くと、下記の表記があるので、「ダウンロード」をクリックします。


ダウンロード

製品サービス	ファイル名	ファイルサイズ	ダウンロード リンク
【無償評価版】 統合開発環境 CS+ for CC V6.00.00 (一括ダウンロード版)	CSPlus_CC_Package_V60000.exe	532.81 MByte	ダウンロード

すると、My Renesas のログイン画面が表示されます。

RENESAS

MyRenesas

 Enter your Email Address and password and click "Login" to access your account.

Email Address

Password

Login

Forgot your password?

Don't have an account yet?Register here

E-mail アドレスとパスワードを入力して、ログインしてください。すると、同意事項の画面が表示されるので、その下の方の「同意する」をクリックしてダウンロードに進みます。

同意する 同意しない

下記のダウンロード画面が開いたら、「実行(R)」をクリックするとダウンロードが始まります。500MB 以上あるので、通信環境によっては、かなり時間がかかります。

ダウンロード (D4000427)

公開日: Jul.20.17
カテゴリ: 無償評価版

【無償評価版】統合開発環境 CS+ for CC V6.00.00 (一括ダウンロード版)

概要

統合開発環境 CS+ の概要については、[データシート](#)を参照してください。
本製品は一括ダウンロードです。分割は[こちら](#)。

対応MCU

RH850ファミリ

RXファミリ

RL78ファミリ

対応マイコン型名は、[データシート](#)を参照してください。

主な同梱モジュール

カテゴリ	製品名
本体	CS+ for CC 共通部分 V6.00.00
コンパイラ	RH850コンパイラ CC-RH V1.06.00
	RXコンパイラ CC-RX V2.07.00
	RL78コンパイラ CC-RL V1.05.00
	RH850ファミリ デバイス依存情報 V6.00.00

renesas.com から CSPlus_CC_Package_V60000.exe (532 MB) を実行または保存しますか?

以下に示す画面が表示されるので、「CS+のセットアップを開始する」をクリックしてインストールします。

ルネサス エレクトロニクス マイコン開発ツール インストーラ はじめにお読みください

English


Integrated Development Environment for Embedded Systems

はじめに

このたびは、CS+をご使用いただきまして、ありがとうございます。
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。

[はじめにお読みください](#)

このファイルにはインストール時の注意事項の他、CS+対応マイコンの情報を掲載しています。
(閲覧には、Adobe® Acrobat® Reader が必要です。Adobe Acrobat Readerの情報は、[Adobe Systems Incorporated](#) のWebサイトにてご覧いただけます。)

CS+対応マイコンの最新情報は、弊社Webサイト「[統合開発環境 CS+](#)」をご覧ください。

インストール

下記のボタンをクリックして、CS+のセットアップを開始してください。

- CS+では、Microsoft 社が提供しているMicrosoft .NET Framework 4.5.2 と 言語バックおよび Microsoft Visual C++ 2010 SP1 のランタイムライブラリを使用します。
ご利用のPCにインストールされていない場合は、CS+のセットアップ時にインストールを行います。

※Windows Vista、Windows 7環境で、WEGから入手した無償評価版をご利用のお客様へ

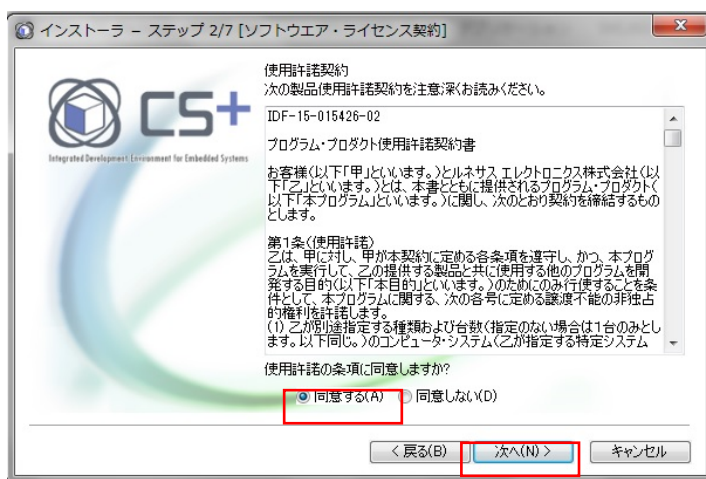
ご利用のPCにMicrosoft .NET Framework 4.5.2がインストールされていない場合、PCをネットワークに接続した状態でセットアップを行ってください。
ネットワークに接続していないPCでセットアップを行う場合は、[Microsoft ダウンロードセンター](#)を参照して、Microsoft .NET Framework 4.5.2 をインストールしてから、CS+のセットアップを開始してください。

- Microsoft .NET Framework 4.5.2 と Microsoft Visual C++ 2010 SP1 ランタイムライブラリは、最新の状態でご利用ください。
最新版に関する情報は、Microsoft社のWebサイトをご確認ください。

以下のようにインストール画面が表示されます。

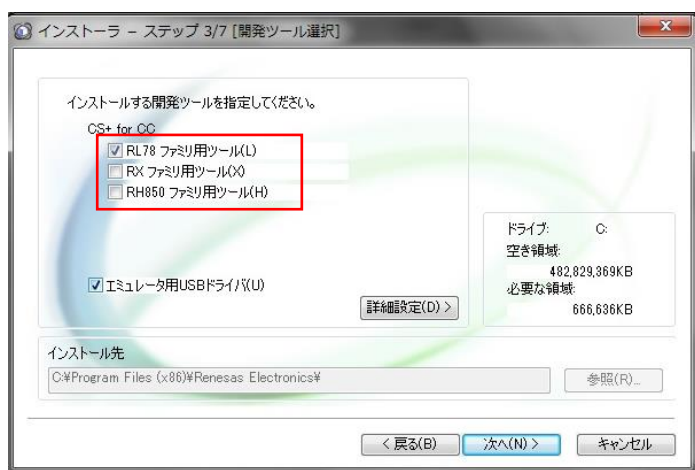


ここは、注意事項です。無用なトラブルを避けるために他のプログラムを終了させ、「次へ(N)」をクリックします。



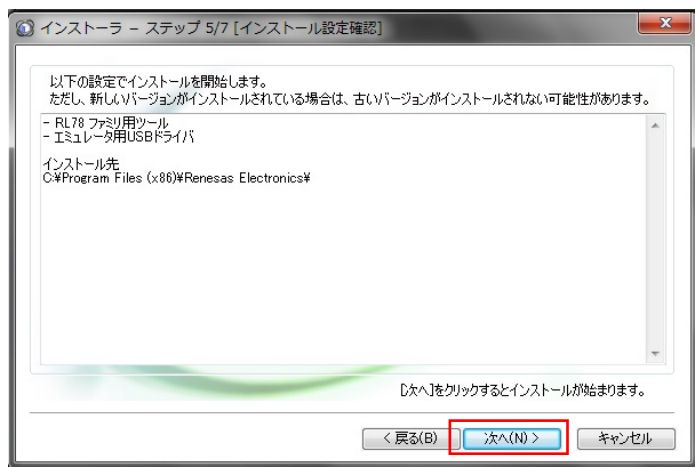
使用許諾契約を読み、「同意する(A)」を選択し、「次へ(N)」をクリックするとインストールが始まります。

最初に、対象のデバイスを指定します。

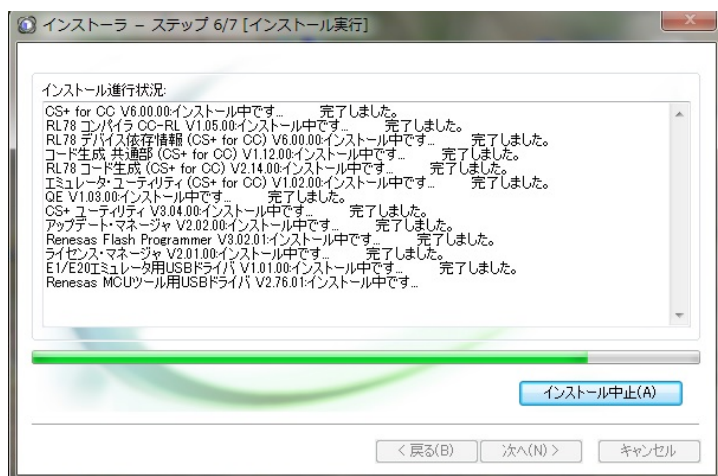


ここでは、RL78 だけにしておきます。RL78 以外のチェックを外し、「次へ(N)」をクリックします（ここで選択しなくても後で必要になったときに追加できます。）

インストール内容を確認してくるので、「次へ(N)」をクリックします。



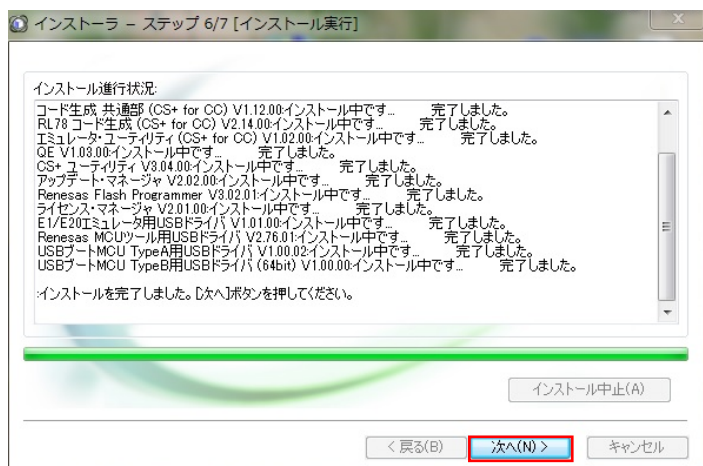
CS+関係のインストールが完了したら、USB ドライバのインストールを行います。



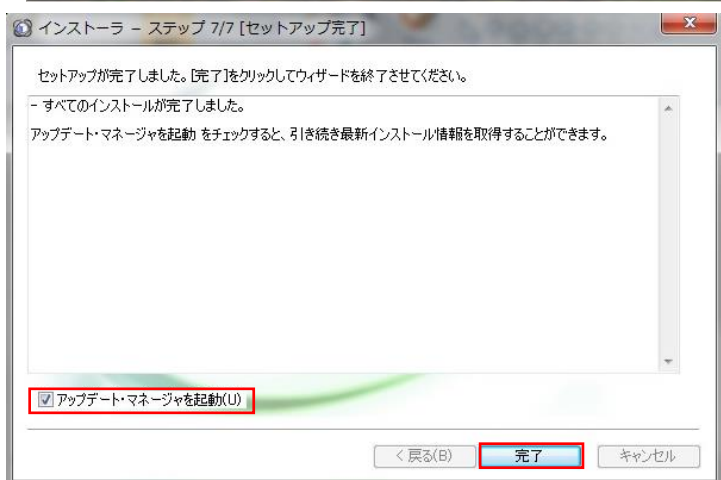
CS+関係のインストールに続く、USB 関係のデバイスドライバのインストールで Windows から警告が表示されるので、ここは、「インストール(I)」をクリックします。



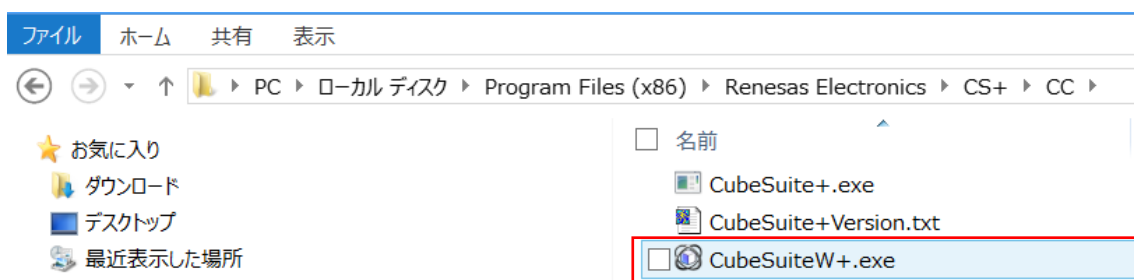
インストールが完了したら、以下の画面が表示されるので、「次へ(N)」をクリックします。



これで、インストールが完了して、完了画面が表示されるので、「次へ(N)」をクリックします。



アップデート・マネージャが起動しますが、さっと眺めて閉じて結構です。インストールが完了すると、「Program Files(x86)」の下に「Renesas Electronics」の下に「CS+」¥「CC」ディレクトリが作成され、「CubeSuiteW+.exe」ファイルができます。これがCS+です。



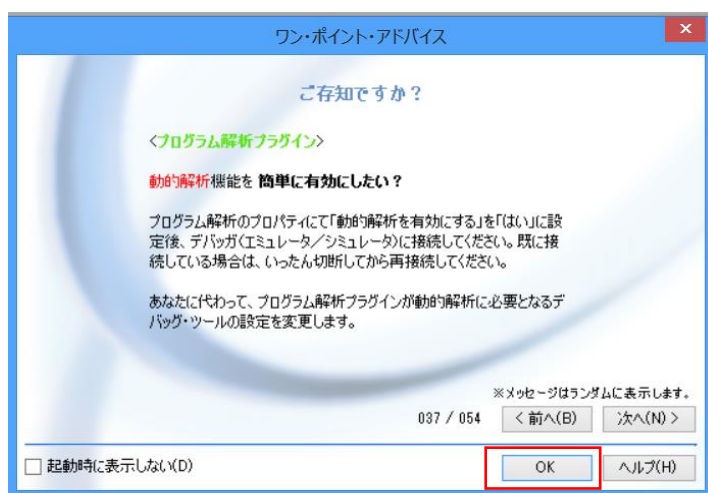
4. CS+CC-RL の起動

CubeSuiteW+.exe をダブル・クリックして、CS+を最初に起動すると、右の My Renesas のログイン画面が表示されます。

ここは、My Renesas の ID（メール・アドレス）とパスワードを入力してログインしておきましょう。



My Renesas ログイン画面のスクリーンショット。タイトルは「My Renesas ヘロログイン」。内容は「My Renesas へ登録いただく、ツール製品などのダウンロードサービスやメールアドレ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。」と説明されている。中央には「ログイン」という大きな文字があり、その下に「メール・アドレス(E)」と「パスワード(P)」の入力欄がある。パスワード欄には「資格情報を記憶する(R)」というチェックボックスがある。下部には緑色の「ログイン」ボタンがあり、右下には「登録は こちら から」というリンクがある。



通常、CS+を起動したときには、左に示すような、ワン・ポイント・アドバイスが表示されます。

表示しないようにもできますが、立ち上がり時間が気にならないなら、表示するようにしておくことをお勧めします。

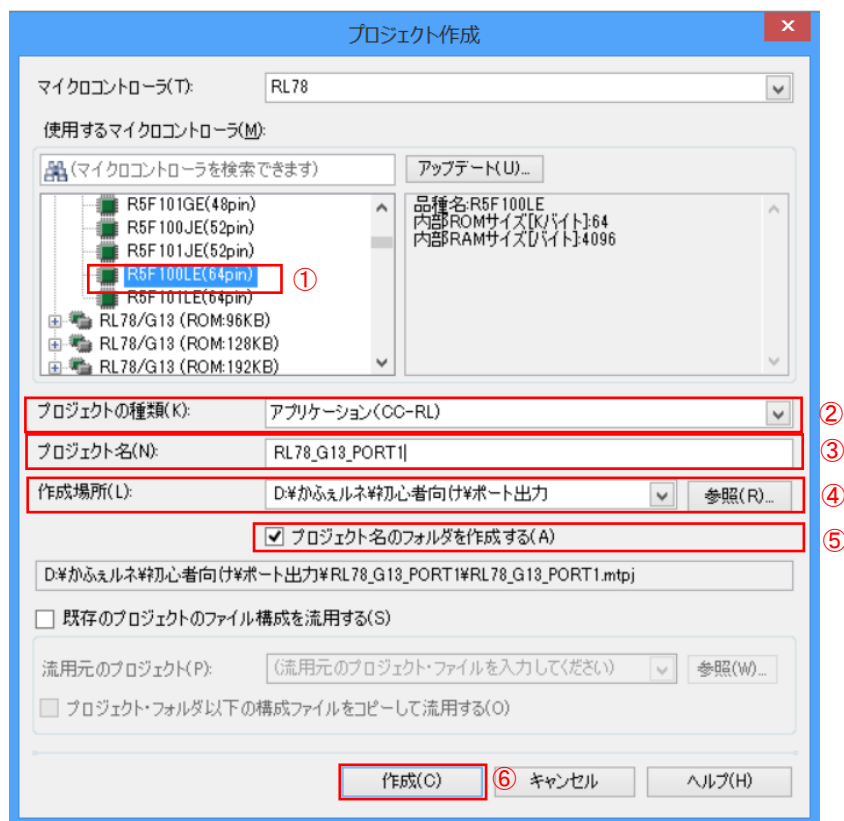
「OK」をクリックすると、下に示すスタート画面が表示されます。一番上にはチュートリアル

の起動ボタンがあるので、暇を見てチュートリアルを眺めてください。



通常は、「新しいプロジェクトを作成する」から始めます。

プロジェクトを作成すると下の「プロジェクト作成」画面が表示されるので、最初に「使用するマイクロコントローラ(M)」を選択します。ここでは、RL78/G13 で ROM64KB, 64pin の「R5F100LE」を選択します。(①)



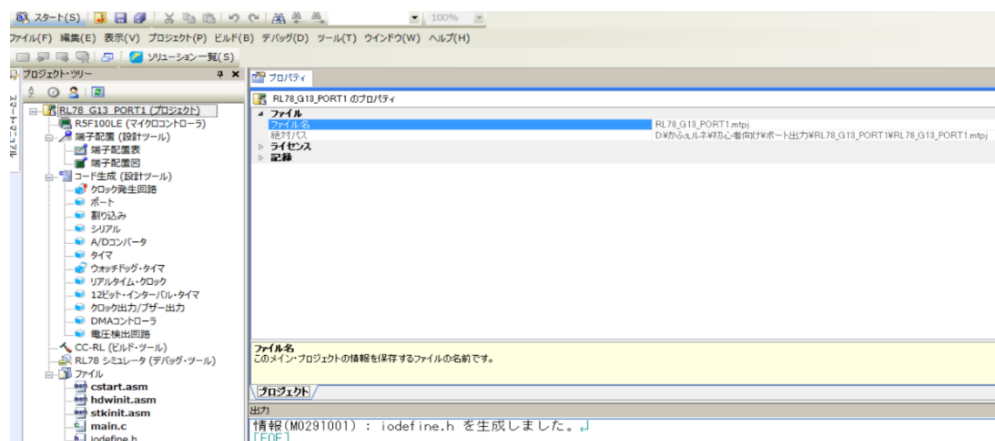
②「プロジェクトの種類(K)」はデフォルトの「アプリケーション(CC-RL)」にします。

③「プロジェクトの名前(N)」はここでは「RL78_G13_PORT1」としておきます。

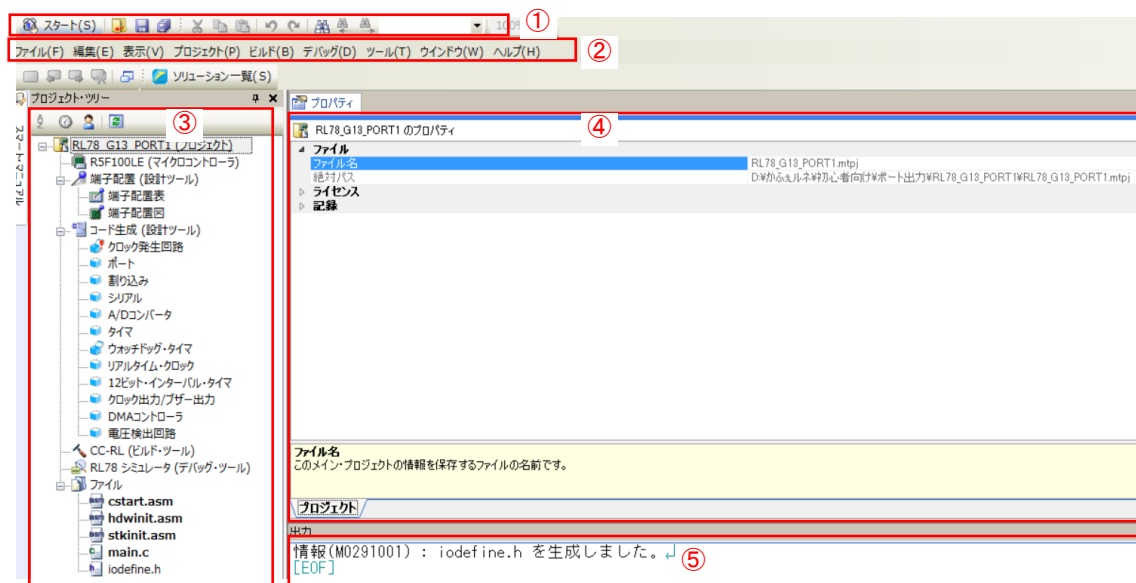
④「作業場所(L)」はプロジェクトを作成するディレクトリを指定します。

⑤「プロジェクト名のフォルダを作成する (A)」をチェックしておきます。

この設定で⑥「作成 (C)」をクリックすると、以下のような画面になります。



通常は、プロジェクトを開くと、このような画面となります。画面の詳細は「ユーザーズマニュアルプロジェクト操作編」を参照してください。ここでは概略だけを示します。

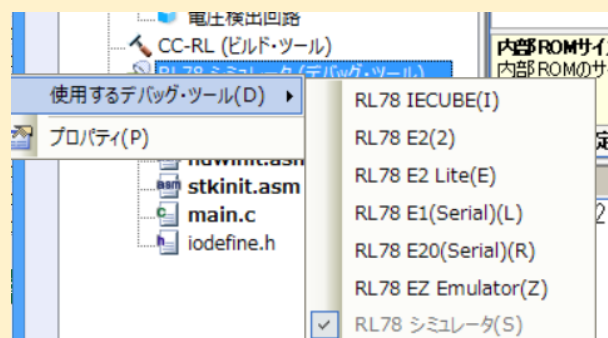


①のツールバーは共通のボタンが並んでいます（カスタマイズで変わります）。
 ②のメニューバーは共通のメニューを表示します（カスタマイズで変わります）。
 ③のプロジェクト・ツリーで項目を選択して、④のウィンドウで内容を確認したり、変更したりします。⑤の出力パネルには、例えば、コード生成でのファイルの生成結果やビルド結果などが表示されます。プロジェクトを作成したときには、ターゲットのデバイスに対応したI/Oを定義したヘッダー・ファイル（iodefine.h）が生成されたことが示されています。

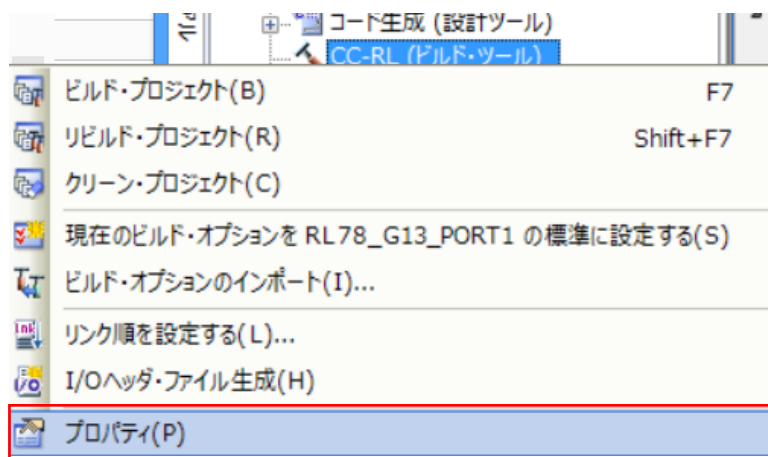
また③のプロジェクト・ツリーには、プロジェクトの名称、ターゲットのデバイス名が表示されます。その下に「端子配置（設計ツール）」「コード生成（設計ツール）」、「CC-RL（ビルド・ツール）」、「RL78 シミュレータ（デバッグ・ツール）」と並び、最後に「ファイル」の順に作成されます。

初期状態で、「デバッグ・ツール」として「シミュレータ」が選択されています。ビルド（コンパイルからリンク）した結果を実行するときに、E1 のつもりでそのままシミュレータに接続して、端子からハイが出ないと悩む人が結構います。実際のハードウェアを動作させるときには、E1 や E2Lite など実際に使用するデバッグ・ツールに確実に変更してください。

下図のように「RL78 シミュレータ（デバッグ・ツール）」を右クリックしてポップアップ・メニューを表示し、「使用するデバッグ・ツール(D)」にマウス・カーソルを合わせるとリストが表示されるので、その中から使用するデバッグ・ツールを選択します。



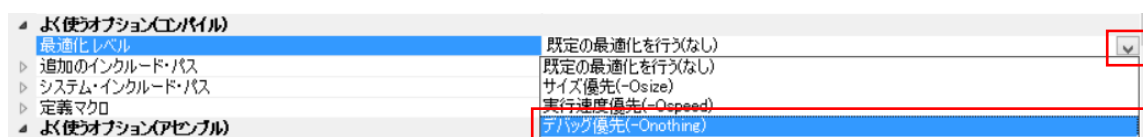
CC-RL（ビルド・ツール）で注意する必要があるのは、最適化です。プロジェクト・ツリーの「CC-RL（ビルド・ツール）」を右クリックすると、下に示すようにプルダウン・メニューが表示されるので、一番下の「プロパティ(P)」を選択します。



「共通オプション」タブの「よく使うオプション（コンパイル）」の「最適化レベル」が「既定の最適化を行う（なし）」になっています。



「最適化レベル」を選ぶと右の方にプルダウン・メニューの表示用の▼が表示されるので、これをクリックしてメニューを表示して、一番下の「デバッグ優先」にします。（デバッグ優先にしておかないと、最適化の影響でデバッグがやり難くなります。）



最後にプロジェクト・ツリー一番下の「ファイル」部分を簡単に説明します。

右に示すのは「ビルド・ツール生成ファイル」の中を展開した状態です。単にプロジェクトを作成しただけで、ビルドが実行されているのが分かります。これは、「ラピッド・ビルド」機能が有効になっているので、その時点でのビルド結果が見えているのです。

「RL78_G13_PORT1.abs」はロードモジュールと呼ばれるビルドの結果のオブジェクトファイルです。

「RL78_G13_PORT1.map」はマップ・ファイルです。

マップ・ファイルはビルド結果がどのようなアドレスに配置されているかを示す情報が出力されたファイルです。

最後の「RL78_G13_PORT1.mot」は、ビルドした結果をモトローラの S タイプの hex ファイルとして出力したものでフラッシュ・ライターなどを使って内蔵のフラッシュにプログラムを書き込むときに使用します。

その下にある「cstart.asm」は、スタートアップ・ルーチンと呼ばれるもので、C で記述したプログラム（の中の main 関数）を動かすために必要な前処理（使用する RL78/G13 に内蔵された周辺機能のハードウェアを初期化したり、変数の初期値を設定したり、初期値のない変数をクリアしたり）を行っているプログラムです。

「hdwinit.asm」は、スタートアップ・ルーチンから呼び出す、使用する RL78/G13 に内蔵された周辺機能のハードウェアを初期化するプログラムです。

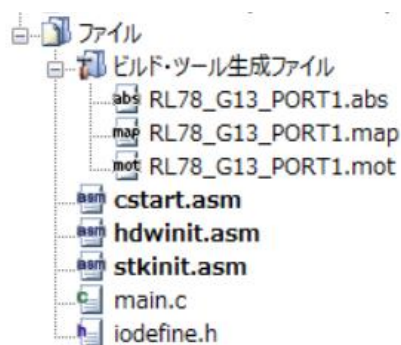
「stkinit.asm」は、スタック領域（作業領域）をクリアするためのプログラムです。

「main.c」は使わないので無視します。

最後の「iodefine.h」は、使用する RL78/G13 に内蔵されているハードウェアを定義したファイルです。内蔵された周辺機を制御するレジスタ（SFR：特殊機能レジスタ）の名前がどのアドレスにあるか等を定義しています。また、RL78 は、組み込み用途で使用するのので、ビット単位でのポート操作を良く使用します。その、ポートでのビット処理を簡単に記述するためにビット・フィールドでの定義を行ってもいます。例えば、ハードウェア・マニュアルで P12 と記述されたポート 1 のビット 2 を操作する場合には、「P1_bit.no2」のように表現します。

プロジェクト・ツリーの上の方にある「コード生成（設計ツール）」には触れませんでした。そこは実際にプログラムを作成する中で触れていきます。

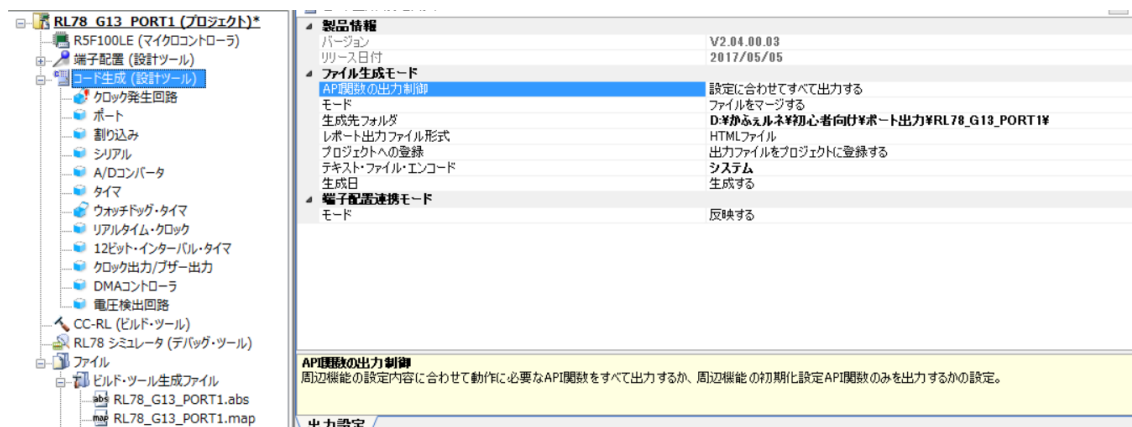
ひとまず、ここまでの共通的な設定です。次はいよいよコード生成の使い方です。



5. コード生成を利用したプログラムの作成方法（共通的な内容）

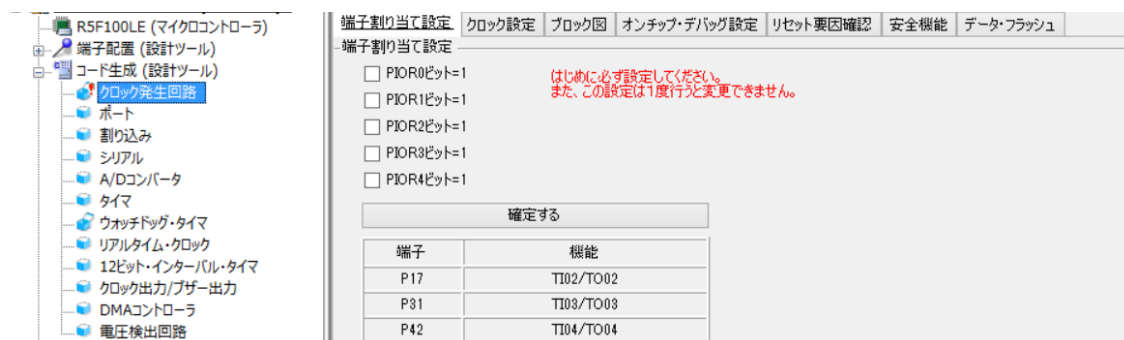
それでは、CS+の大きな特徴であるコード生成機能を使ってみることにします。

コード生成（設計ツール）を選択して、左の+をクリックすると、以下の画面が表示されます。左のプロジェクト・ツリー部には RL78/G13 に内蔵された周辺機能の一覧が表示され、右側には設定が表示されます。



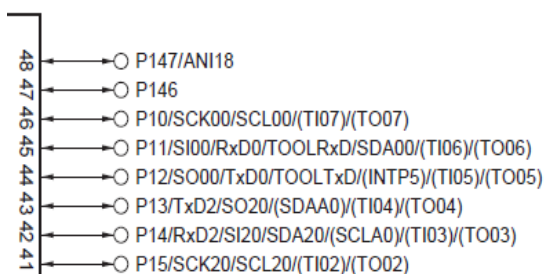
ここでは、設定はデフォルトのままでプログラミングを始めます。プログラミングとは言っても、ここでは、GUIを用いて、画面で使い方を選擇していただくだけです。

まずは、「クロック発生回路」を選択します（というか、最初は「クロック発生回路」しか選択できません）。すると、以下の画面となります。



この画面は端子機能の「周辺 I/O リダイレクション・レジスタ（PIOR）」の指定を行うものです。RL78/G13 では一つの端子に複数の機能が割り当てられています。一部の機能については複数の端子に割り当てることができるようになっています。

右に示すのは端子接続図の一部を抜き出したものです。ここで、機能名称が（）で囲まれたものがリダイレクションすることで、その端子で使えるようになる機能です。ここで、P10 は、通常はシリアルクロック（SCK00 等）だったものがタイマの入出力（TI07/TO07）に変更できます。



それをどこに割り当てるかを指定するのがこの画面です。基本的に、端子の機能は一度割り当てると変更はしません。そこで、この画面の設定は一度確定すると、変更できません。(実際は、レジスタの設定を変更するとはできますが、トラブルの原因になるので、変更はしないようにしましょう。)

普通は、そのまま「確定する」をクリックします。すると、後は自由に内蔵機能が選択できるようになりますが、通常は「クロック設定」から行います。

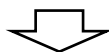
デフォルトでは「高速メイン・モード (3.6V) ≤」となっていますが、「高速メイン・モード (2.7V) ≤」に変更します。すると、「EVDD 設定」が自動的に切り替わります。今回はこれだけにしておきます。

次は、「オンチップ・デバッグ設定」です。デフォルトでは下に示すように「使用しない」になっていますが、これを「使用する」に変更します。

また、RRM（リアルタイム RAM モニタ）機能は使用しません。オンチップ・デバッグを使用するに設定すると、セキュリティ ID や認証失敗時の処理を必要に応じて変更します。

セキュリティ ID は変更しても、プロジェクトとして保持されるので、問題はありませんが、他の環境で使用する時には変更した値を間違えるとフラッシュ・メモリに書かれた内容（プログラム）が消去されてしまうことになるので、注意してください。

次の「リセット要因確認」は、使わないのでチェックを外しておきます。

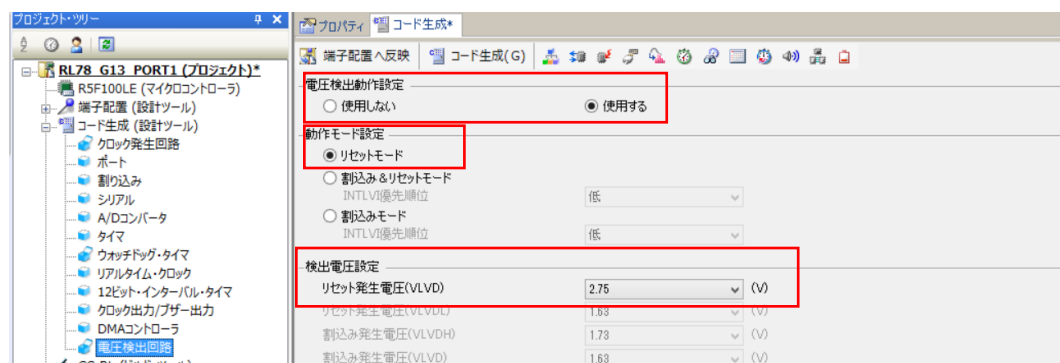


「クロック設定」ではこれだけにしておきます。

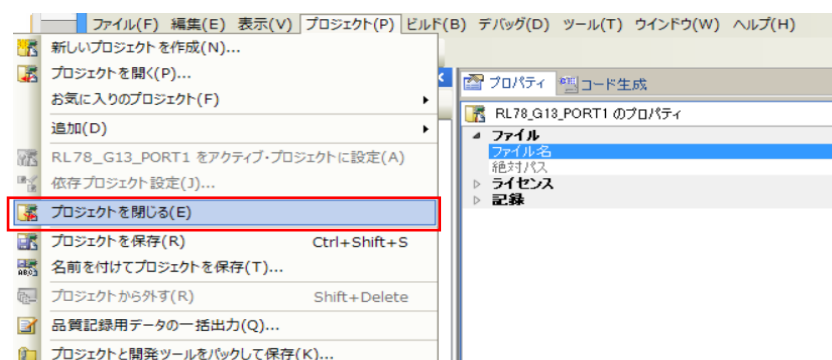
次に「ウォッチドッグ・タイマ」の設定です。ウォッチドッグ・タイマはコード生成のディフォルトでは、「使用する」となっていますので、これを「使用しない」に変更します。

（なぜ、デフォルトが「使用する」なのかが理解できません。一種のトラップですね。）

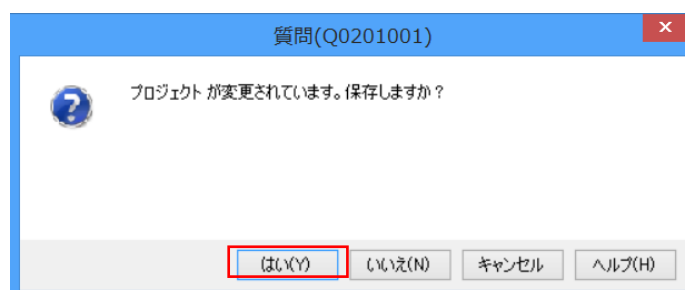
次に、「電圧検出回路」の設定です。RL78 のハードウェア マニュアルでは、電圧検出回路を設定することが推奨されています。コード生成のデフォルトでは、「使用しない」になっているので、これを「使用する」に変更し、「リセットモード」を選択し、検出電圧を 2.75V にしておきます。



共通的な設定は、ここまでです。これから、いよいよプログラミングに入ります。その前に、このプロジェクトを一旦閉じて、ここまでの変更を保存しておきます。画面上方のメニューバーで「プロジェクト (P)」をクリックしてプルダウン・メニューを開きます。メニューの中から、「プロジェクトを閉じる (E)」をクリックしてプロジェクトを閉じます。



プロジェクトを保存するか聞いてくるので、「はい (Y)」をクリックして保存します。



保存したプロジェクトを含むフォルダのコピーを「RL78_G13_PORT 共通部」にリネームして保存しておきましょう。CS+CC-RL も閉じておきます。

続く