

2019 年 5 月 15 日 第 1 版

組込みソフトの安全設計

ー基礎から二足歩行ロボットによる実践まで

付録

C S + の操作手順

開発手順の概要

ワンチップマイコンのプログラムはパソコンで作成し、ビルドし、マイコンに実行形式のプログラムをダウンロードします。この一連の作業を支援するソフトが統合開発環境 CS+です。

【プログラムのビルドとは】

ビルドとは人間が作成したプログラム（C 言語等）をマイコンが実行できるプログラムに変換する作業です。

ビルドではコンパイル（ソースファイル毎に機械語に翻訳）やリンク（各ソースファイルのコンパイル結果を連結）などを行い、一つの実行可能なファイルを作成します。

パソコンでビルドを行い、実行可能なコードをマイコンに転送して実行します。

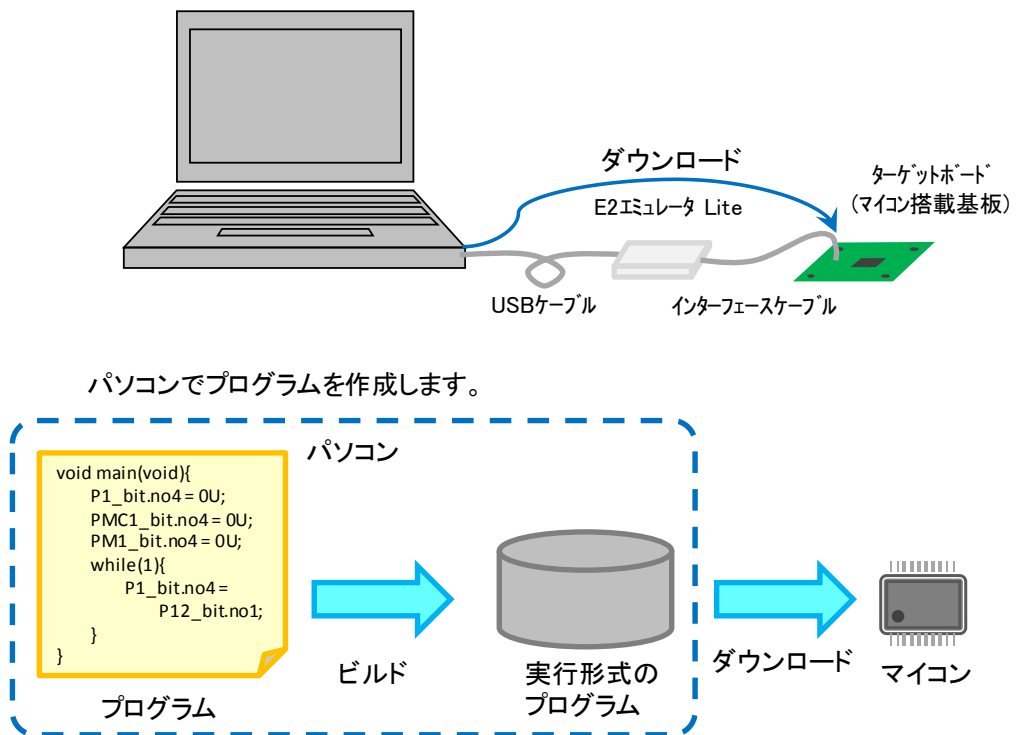


図1 ワンチップマイコンのプログラム作成のイメージ

プロジェクトの作成・実行の基本的流れは以下の通りです。

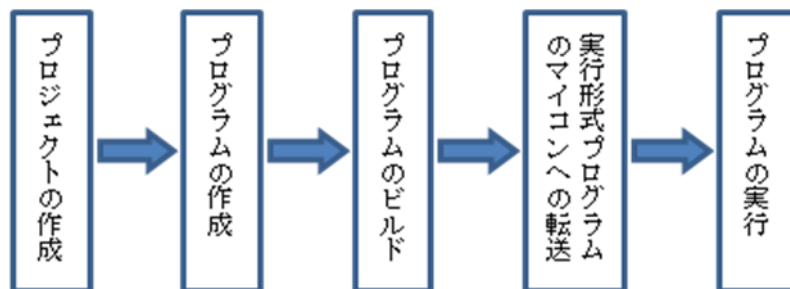


図2 ワンチップマイコンのプログラム作成の流れ

以下、操作手順の詳細を説明します。

【プロジェクトの作成】

「プロジェクトの作成」では、以下の設定をします。

- ・ プロジェクト名の設定
- ・ マイコンやエミュレータ、電源供給方法の選択
- ・ オンチップ・デバッグやウォッチドッグ・タイマ、システムクロックの設定

統合開発環境の起動

パソコンのスタートメニューから、または、アイコンをクリックして CS+ for CC (RL78, RX, RH850) を起動します。CS+ for CC - [スタート] ウィンドウが表示されます。

以下、操作手順の詳細を説明します。

プロジェクトの作成

- ① [スタート] の「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ます。

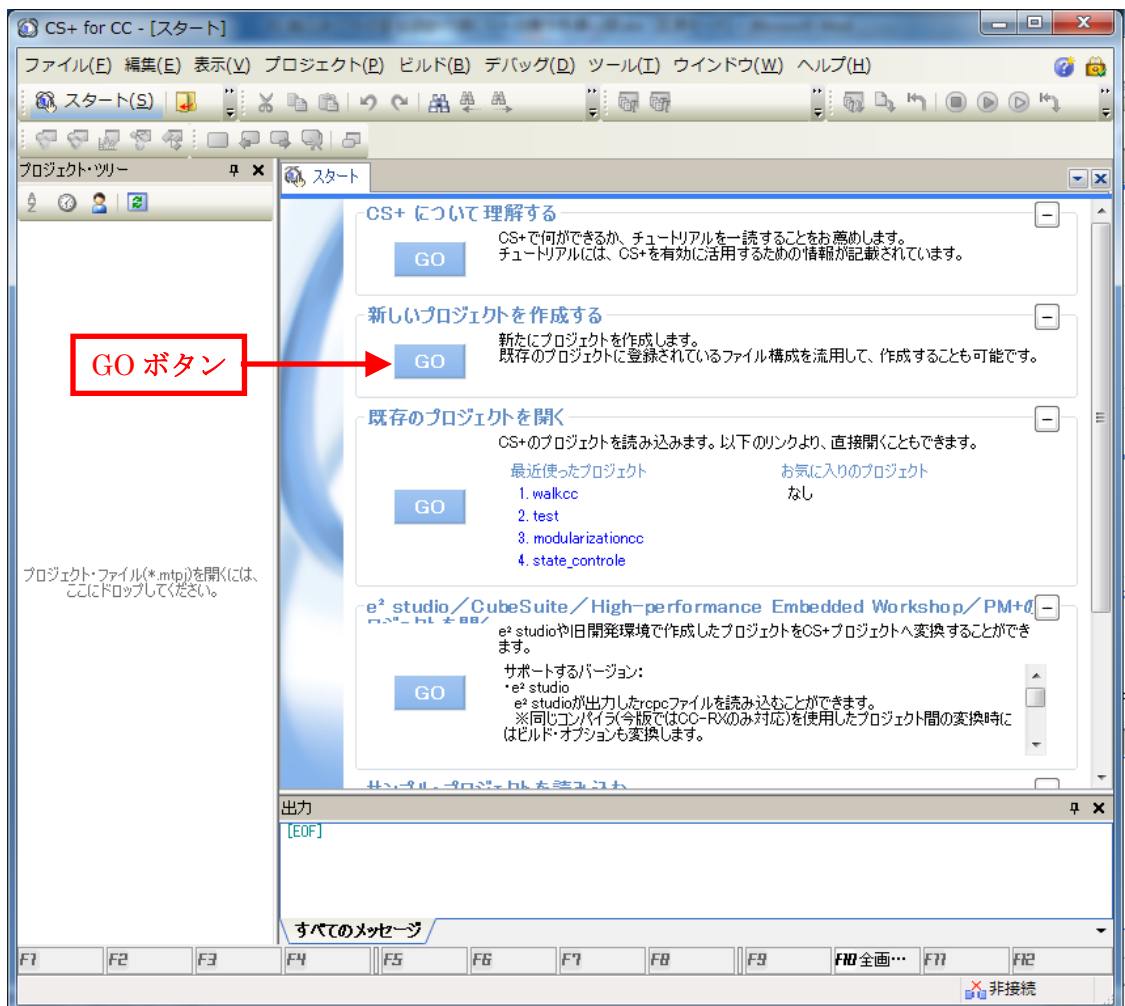


図 3 プロジェクトの作成開始

プロジェクトの作成ウィンドウが表示されますので、プロジェクトの各設定を指定します。

- ① 「マイクロコントローラ (T)」 のプルダウンメニューから 「RL78」 を選択します。
- ② 使用する 「マイクロコントローラ (M)」 の設定画面から 「RL78/G12 (ROM:16KB)」 の 「R5F1026A (20pin)」 をクリックして選択します。
- ③ 「プロジェクトの種類 (K)」 のプルダウンメニューから 「アプリケーション (CC-RL)」 を選択します。
- ④ 「プロジェクト名 (N)」 に設定したいプロジェクト名を入力します。
- ⑤ 「作成場所 (L)」 の 「参照 (R)」 ボタンをクリックしてプロジェクトを保存したいフォルダを選択します。
- ⑥ 「プロジェクト名のフォルダを作成する (A)」 のチェックボックスを選択します。
- ⑦ 「既存のプロジェクトのファイルの構成を流用する (S)」 のチェックボックスは選択しません。
- ⑧ 内容を確認して 「作成 (C)」 ボタンをクリックします。

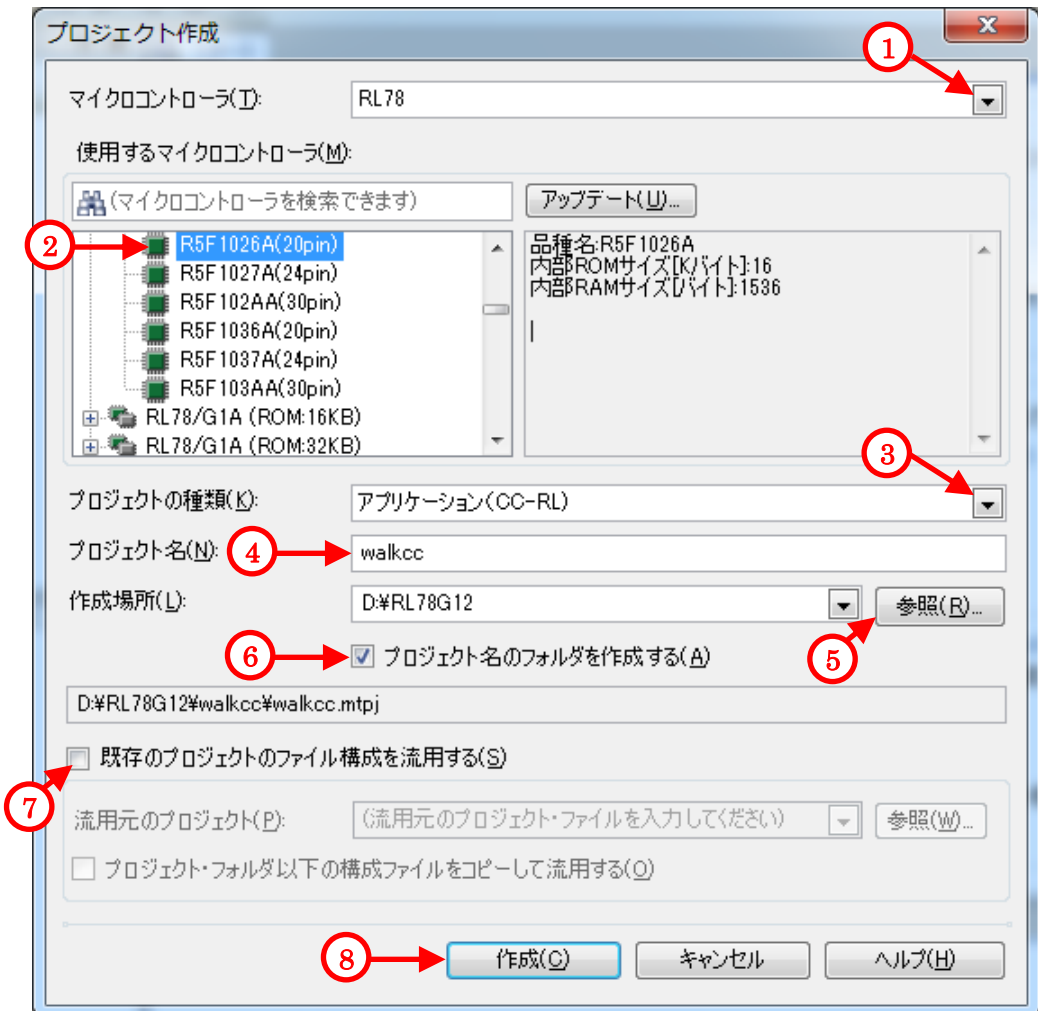


図 4 プロジェクトの作成

デバッグツールの設定

使用するデバッグ・ツールを設定します。

- ① 「RL78 シミュレータ (デバッグ・ツール)」を右クリックして「使用するデバッグ・ツール(D)」の「RL78 E2 Lite(E)」をクリックして選択します。

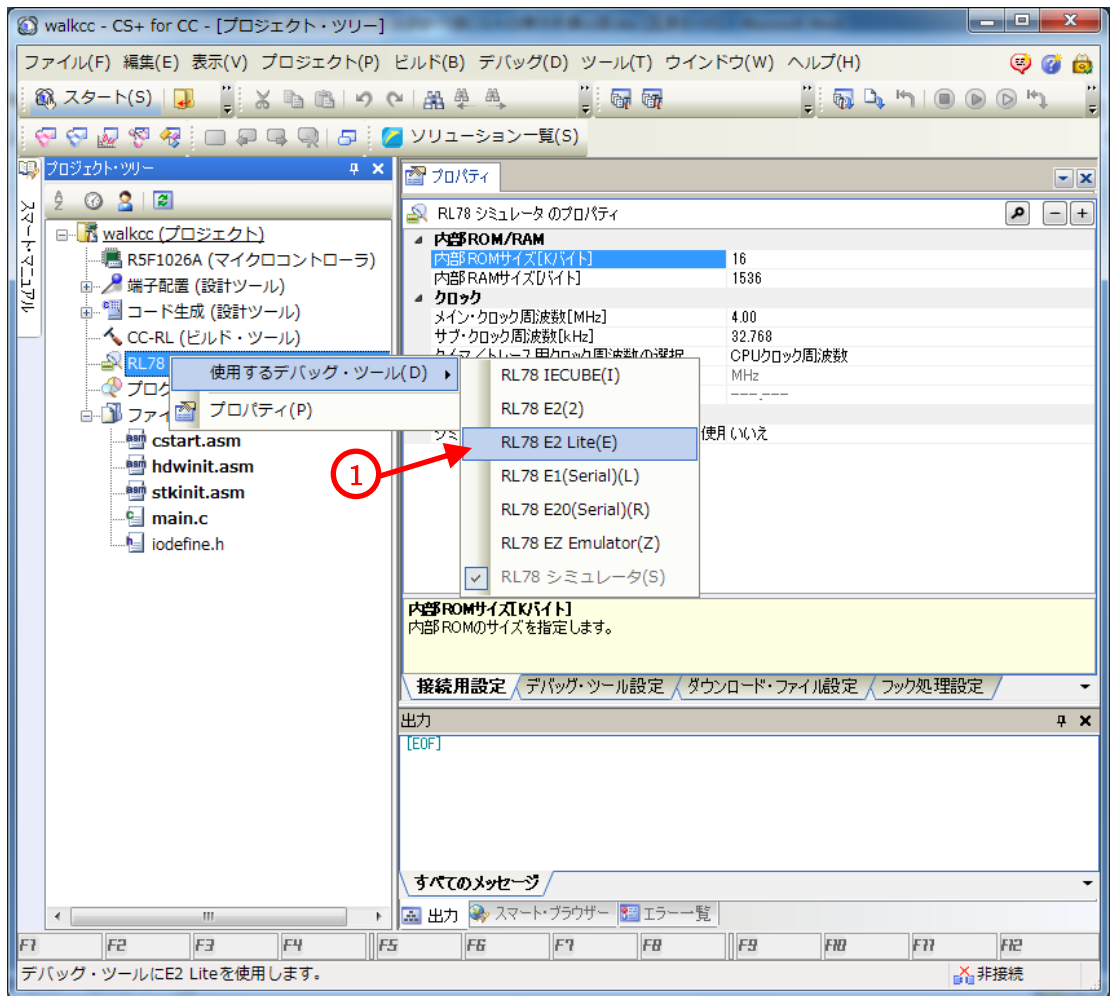


図 5 デバッグツールの設定

マイコンへの電源供給の設定

エミュレータからの電源供給の設定・確認のしかたは次の通りです。「RL78 E2 Lite (デバッグ・ツール)」をダブルクリックして、左側に、「RL78 E2 Lite のプロパティ」ウィンドウを表示します。その、「ターゲット・ボードとの接続」の「エミュレータから電源を供給する(最大200mA)」の右側が「はい」または「いいえ」になっています。必要に応じて、マウス操作で▼を表示し、設定してください。

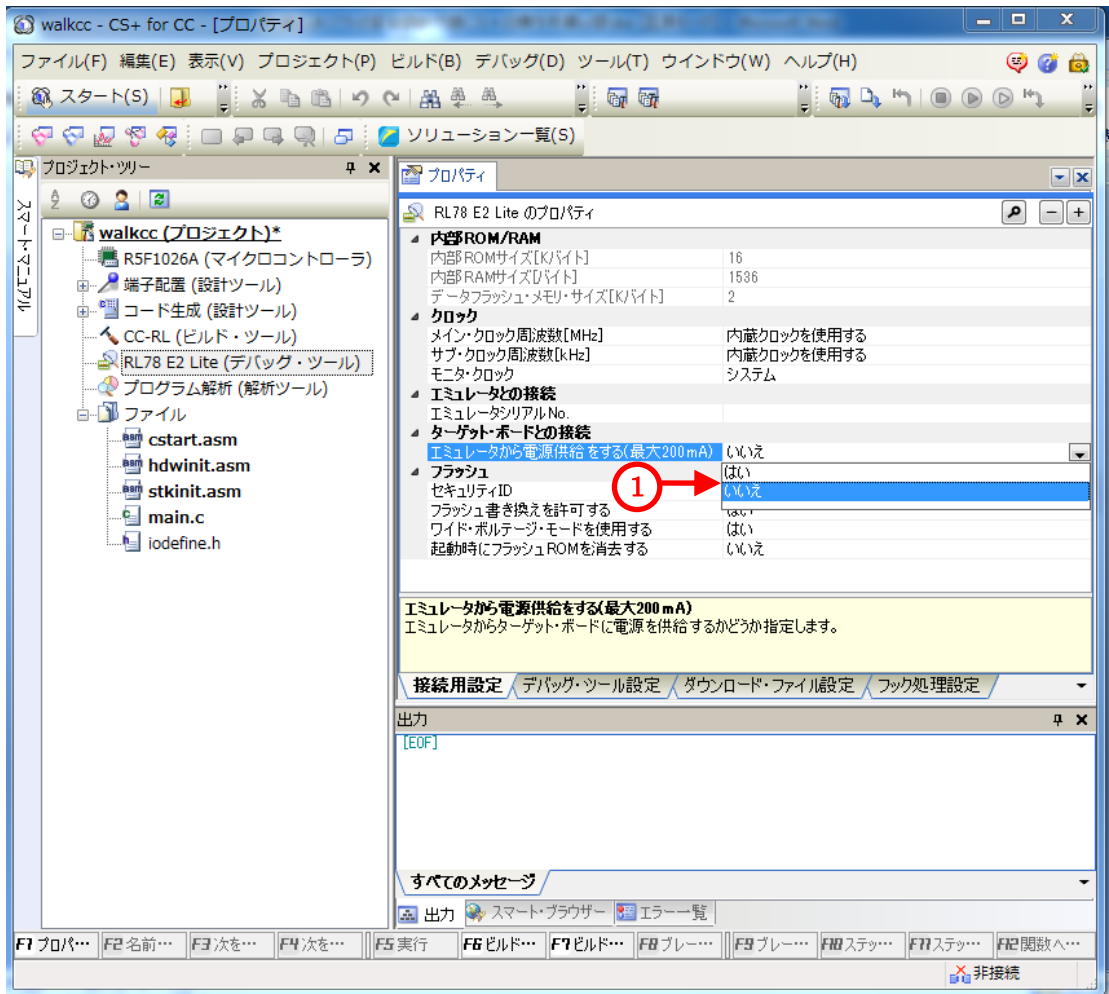


図 6 マイコンへの電源供給の設定

サンプルプログラムのエミュレータからマイコンへの電源供給の設定は、次の通りです。

6 章までのサンプルプログラム

- ・マイコンへ電源を供給する。

7 章の 2 足歩行ロボットのサンプルプログラム

- ・マイコンへ電源を供給しない。

サンプルプログラムのエミュレータの電源供給の設定を変更する必要はありません。ただし、基板上のジャンパー用ピンヘッダー (JP) へのジャンパーピンの接続と電池からの電源供給は

次のようにしてください。

6 章までのサンプルプログラム

- ・ジャンパー用ピンヘッダー（JP）のジャンパーピンを外す。
- ・電池ケースのスイッチとスライドスイッチ（SW1）を OFF にする。

7 章の 2 足歩行ロボットのサンプルプログラム

- ・ジャンパー用ピンヘッダー（JP）へジャンパーピンを接続する。
- ・電池ケースのスイッチとスライドスイッチ（SW1）を ON にする。

注：2 足歩行ロボットのサンプルプログラムはエミュレータから電源を供給しない設定にします。RC サーボモータは大きな電流を消費しますので、モータを動作させる場合は、必ずニッケル水素電池からモータへ電源を供給する必要があります。エミュレータからモータへ電源を供給するとエミュレータが故障する恐れがあります。また、エミュレータと電池の双方から電源を供給すると、エミュレータや制御基板が故障するおそれがあります。

オンチップ・デバッグの設定

オンチップ・デバッグをできるようにします。

- ① プロジェクト・ツリーの「CC-RL (ビルド・ツール)」をクリックして選択します。
- ② 「プロパティ」の「リンク・オプション」タブをクリックして選択します。
- ③ 「リンク・オプション」タブの「デバイス」の「オンチップ・デバッグの許可／禁止をリンク・オプションで設定する」のプルダウンメニューから「はい(-OCDBG)」を選択します。
- ④ 「オンチップ・デバッグ・オプション・バイト制御値」を「84」に設定します。
- ⑤ 「デバッグ・モニタ領域を設定する」の「はい (-DEBUG_MONITOR)」を選択します。

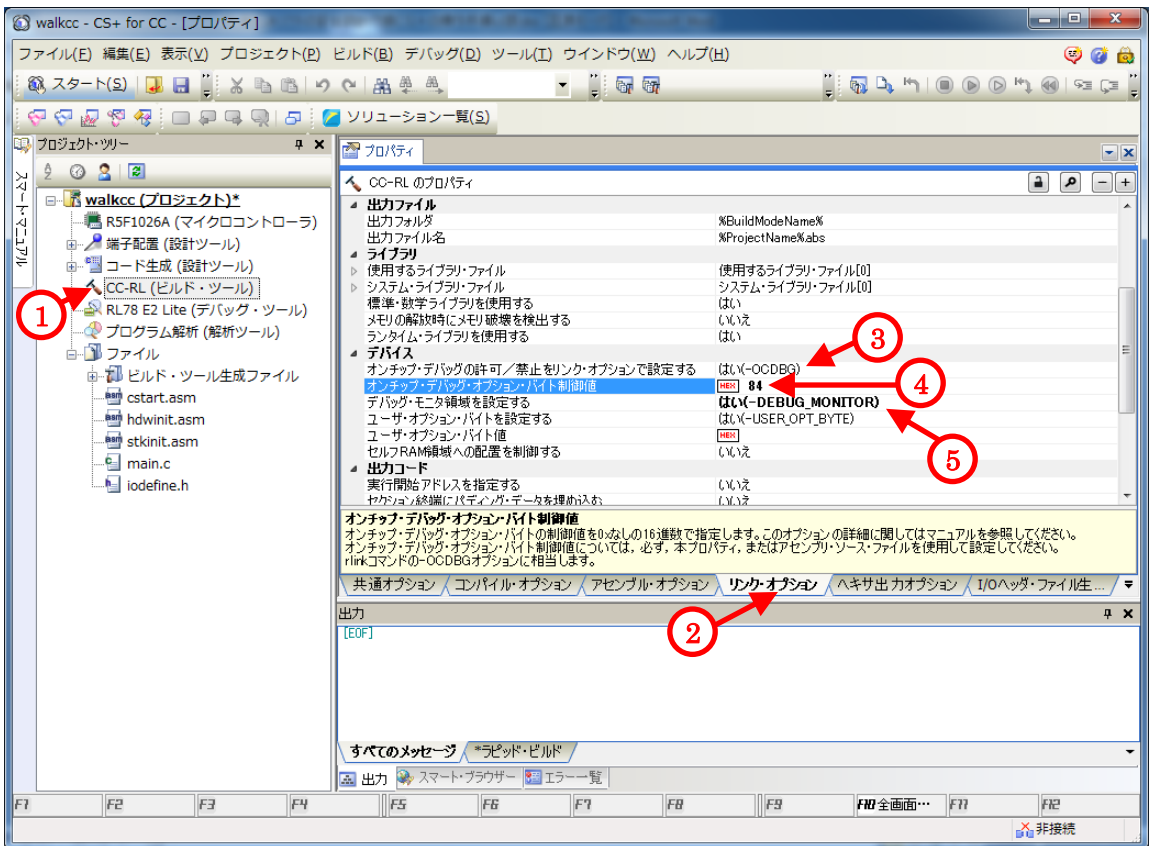


図7 オンチップ・デバッグの設定

ウォッチドッグ・タイマとシステムクロックの設定

ウォッチドッグ・タイマとシステムクロックを設定します。

6章までのサンプルプログラムでは、ウォッチドッグ・タイマを停止し、電圧検出回路を不使用とし、システムクロックとして内蔵の高速オンチップ・オシレータを24 MHzで使用します。

7章のおもちゃの二足歩行ロボットのサンプルプログラムでは、ウォッチドッグ・タイマのオーバーフロー時間を29.68 msとして使用します。

- ① オンチップ・デバッグの設定と同じ「デバイス」の設定画面で、「ユーザ・オプション・バイトを設定する」のプルダウンメニューから「はい(-USER_OPT_BYTE)」を選択します。
- ② 6章までのプログラムでは、マイコンの「ユーザ・オプション・バイト値」をEEFFE0とします。

7章のプログラムでは、マイコンの「ユーザ・オプション・バイト値」を76FFE0とします。

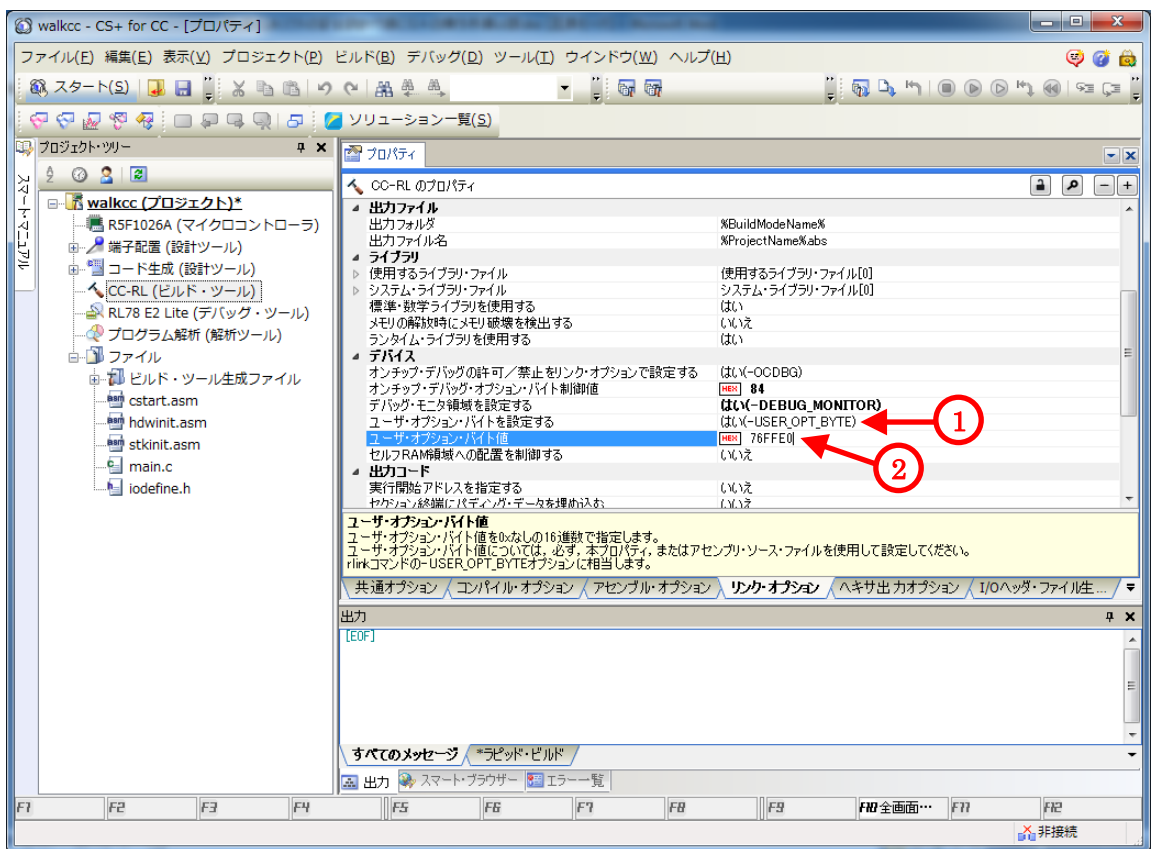


図8 ウォッチドッグ・タイマとシステムクロックの設定

これでプロジェクトの設定はできました。

【プログラムの作成】

プログラムの作成では、以下の作業を説明します。

- ・ 新しいファイルの追加
- ・ プログラムの入力
- ・ プログラムの保存
- ・ ファイルの依存関係の更新

プロジェクトを作成すると既に、プロジェクト・ツリーに統合開発環境が配置したファイルが表示されます。

cstart.asm

スタートアップルーチンです。スタックを初期化し、hdwinit.asm を呼出し、RAM を初期化し main 関数を呼び出します。必要に応じて変更しますが、今回は変更しません。

hdwinit.asm

ハードウェアの初期化ルーチンです。出力設定などを電源投入時の早い時期に設定する必要がある場合等に記述しますが、今回は変更しません。

今回は、main 関数の最初に、C 言語で記載したハードウェア初期化関数を呼出し、ハードウェアの初期化を実施します。

stkinit.asm

スタックエリアの初期化ルーチンですが、デフォルトでは使用しません。

main.c

メイン関数を記述するためのファイルです。空のメイン関数が設定されています。ここに、メインルーチンを記述していきます。

iodefine.h

マイコン (RL78/R5F1026A) の周辺回路を利用するための特殊機能レジスタ (SFR) を記号で指定できるようにするヘッダーファイルです。マイコンの「ユーザーズマニュアル ハードウェア編」で記載している記号とほぼ同じ記号で指定できます。

新しいファイルの追加

更に、新しいファイルを追加したいときは以下のように操作します。

- ① 新しいファイルを追加します。「プロジェクト・ツリー」の「ファイル」を右クリックし、「追加(D)」の「新しいファイルを追加(n)...」をクリックして選択します。

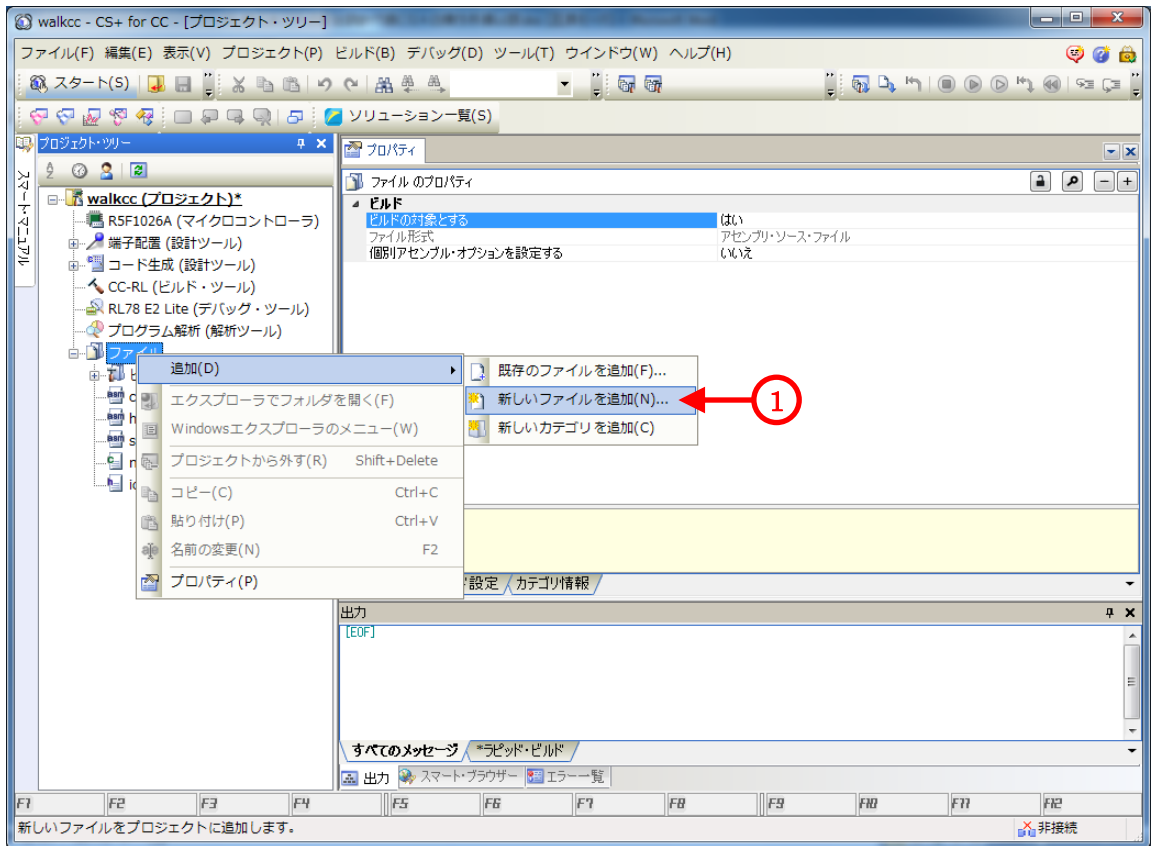


図 9 新しいファイルの追加

- ① ファイル追加ダイアログが表示されますので、「ファイルの種類(T)」で「C ソースファイル(*.C)」をクリックして選択します。
- ② 「ファイル名(N)」に作成したいファイル名を入力します。
- ③ 「作成場所(L)」は、プロジェクトの作成で指定したフォルダとなっています。
- ④ 「OK」ボタンをクリックすれば、プロジェクト・ツリーにファイル名が表示され、プロジェクト・ツリーの横にソースコード（プログラム）の内容が表示されます。

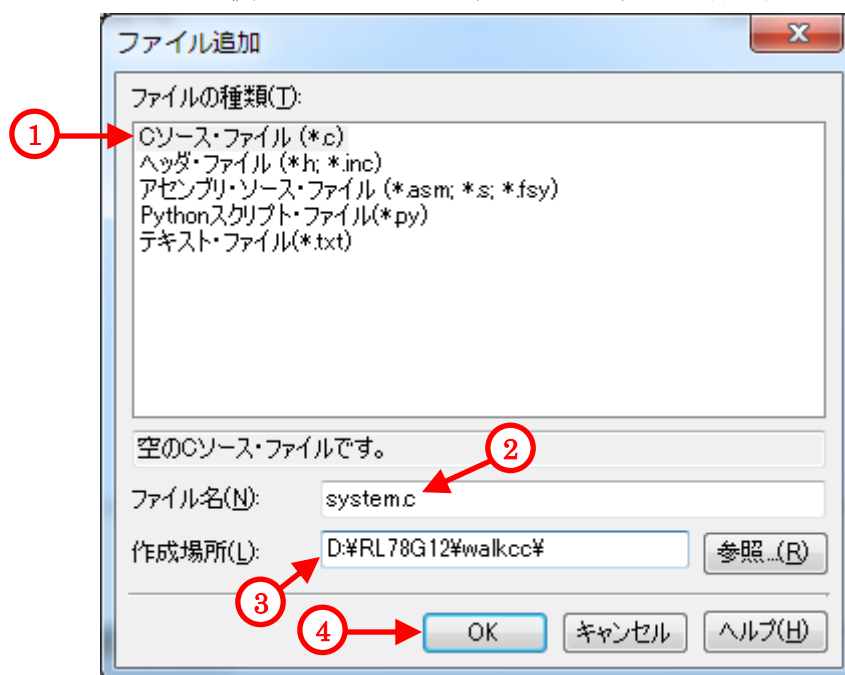


図 10 ファイル名の指定

プログラムの入力

何も入力していないので、エディタ画面には空白が表示されます。

プログラムを入力します。

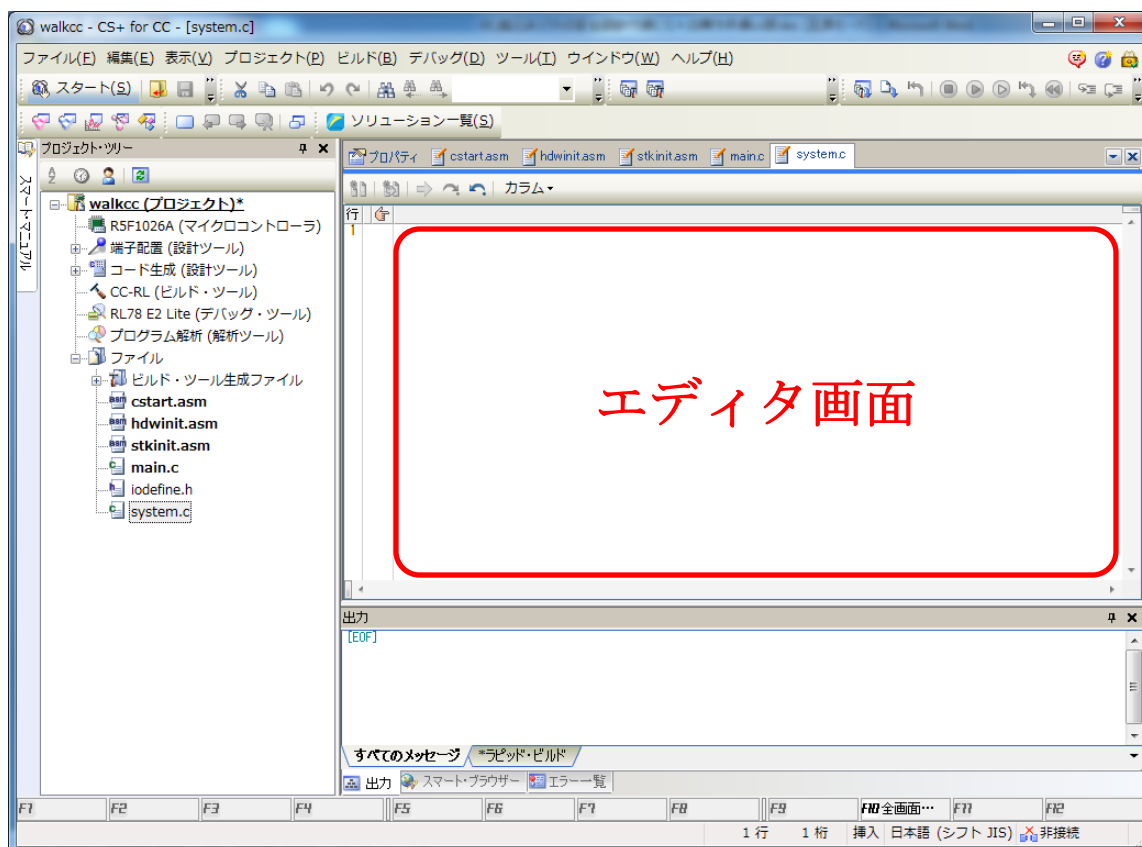


図 11 プログラムの入力

プログラムの保存

- ① 入力し終われば、メインメニューから「ファイル(F)」の「すべてを保存(L)」をクリックして保存します。

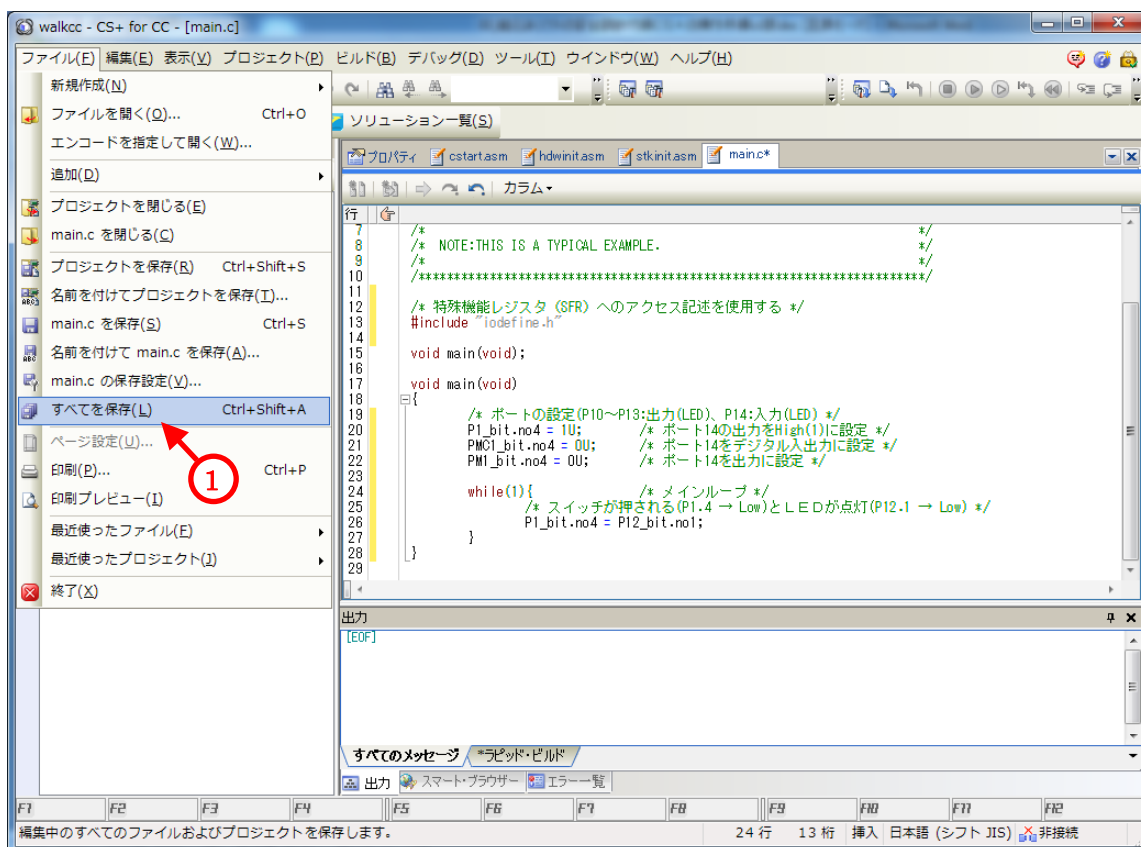


図 12 プログラムの保存

ファイルの依存関係の更新

#include 文で指定されるファイルの依存関係を更新します。

ファイルを追加・変更し、#include 文が追加・変更されたら、依存関係を更新します。

① メインメニューから、「ビルド(B)」の「依存関係の更新(P)」をクリックします。

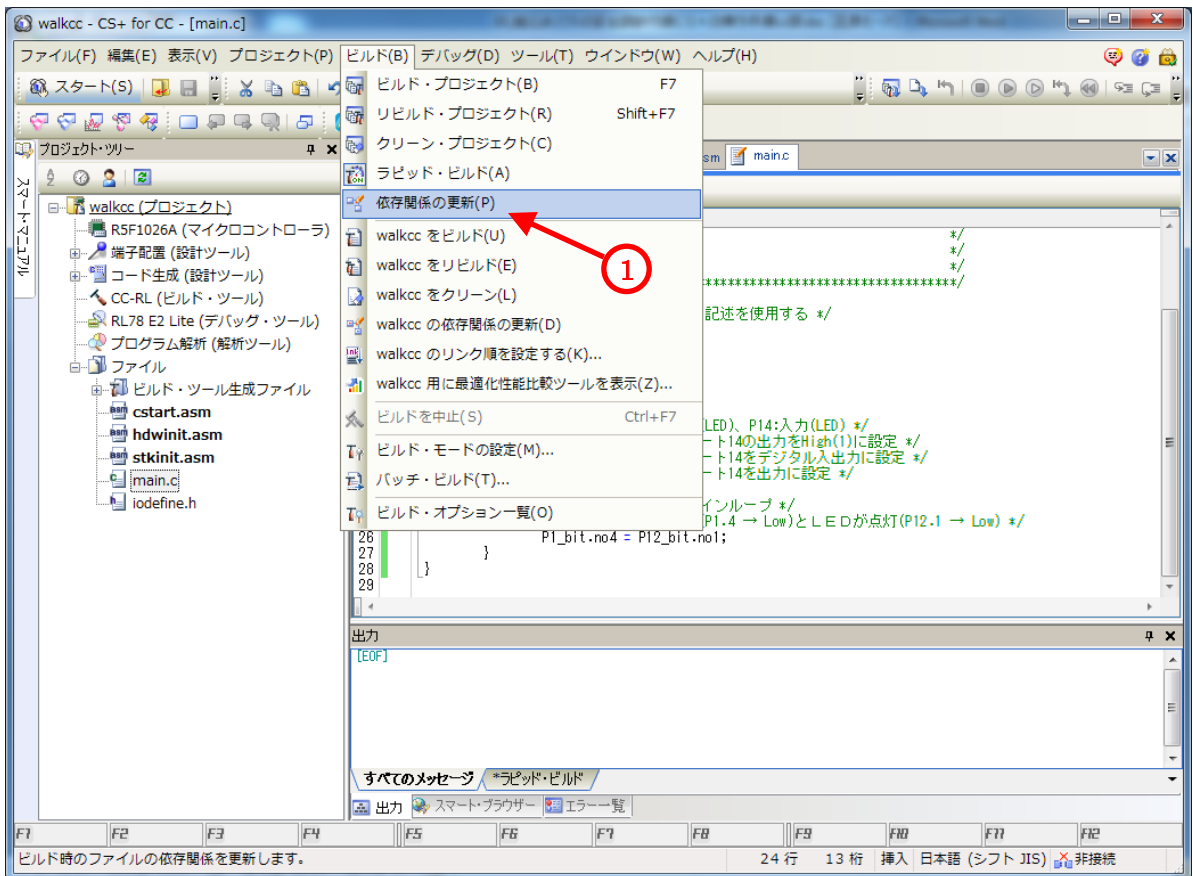


図 13 ファイルの依存関係の更新

【プログラムのビルド】

プログラムのビルドでは、既に説明したように実行可能なファイルを作成します。

ビルドは次のように行います。

- ① メインメニューから、「ビルド(B)」の「ビルド・プロジェクト(B)」をクリックします。

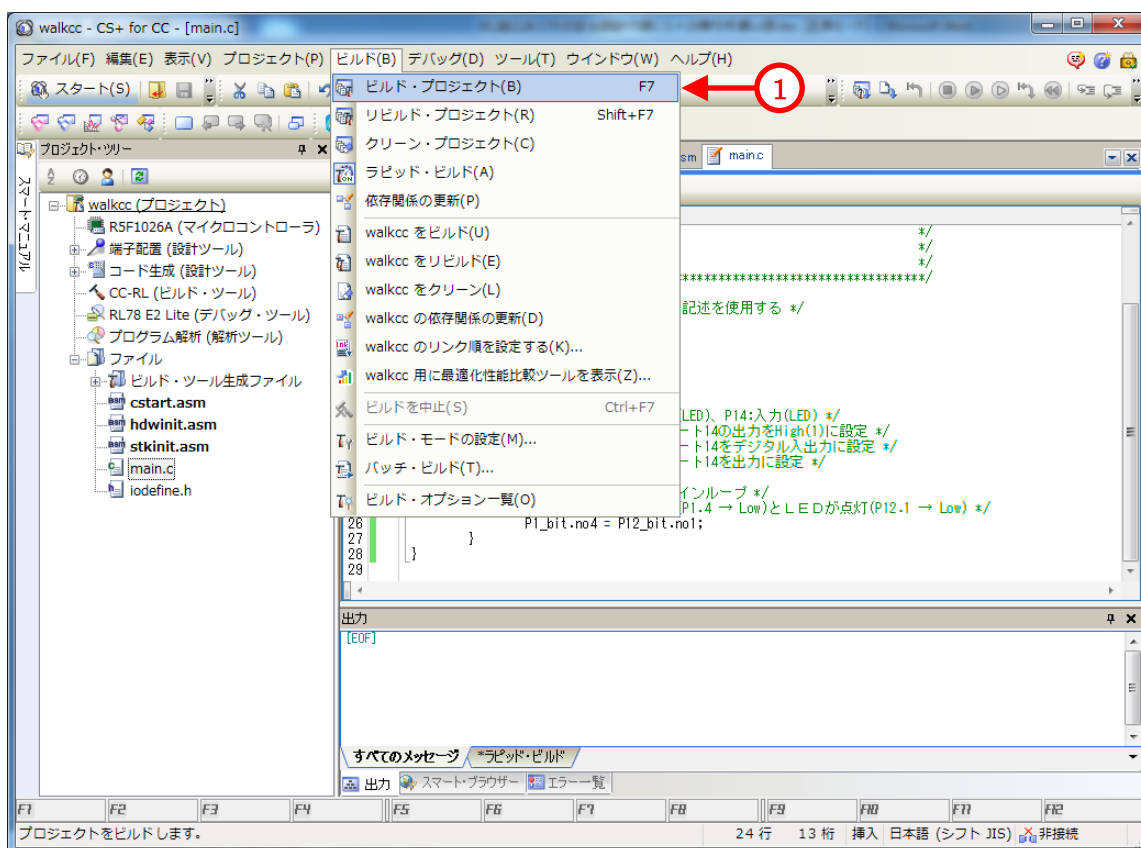


図 14 プログラムのビルド

プログラムのビルド成功

- ② 文法に誤りがなければ、ビルドが成功し、「出力ウインドウ」の「すべてのメッセージ」タブに「エラー：0 個」と表示されます。

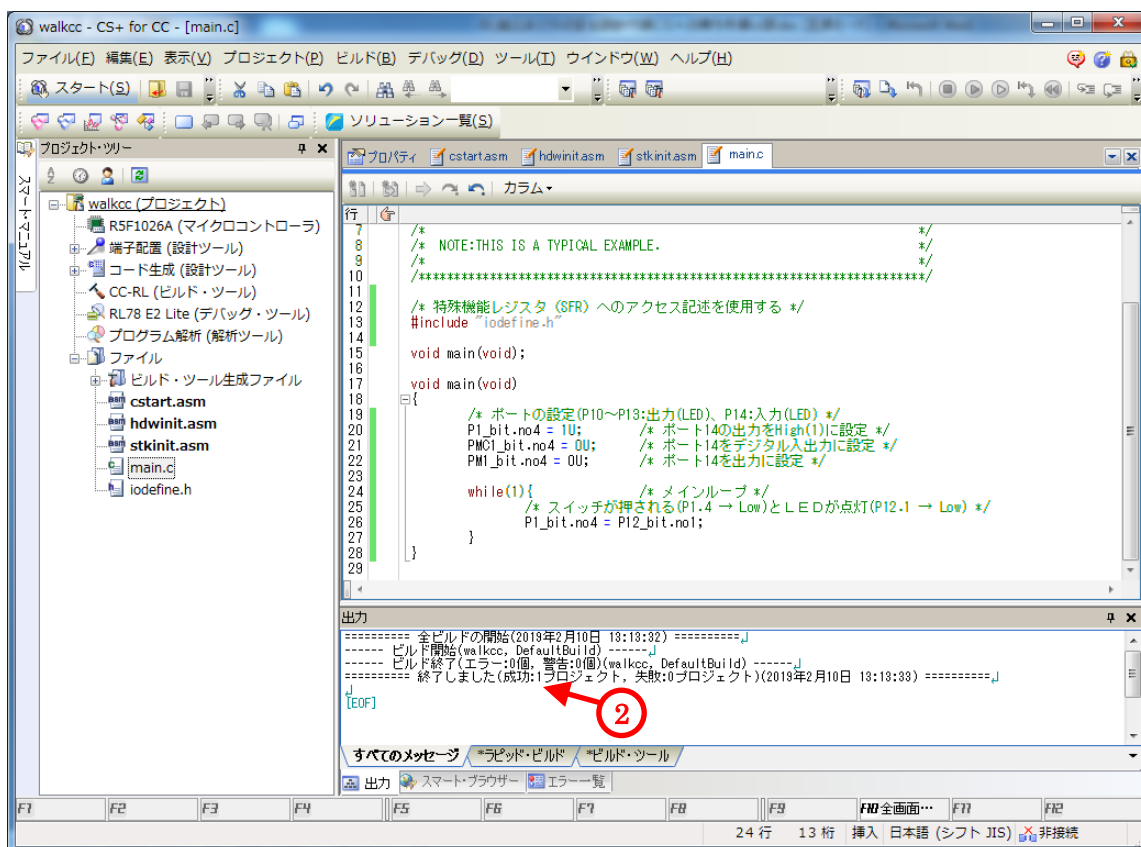


図 15 プログラムのビルドの完了

プログラムのビルド失敗

- ③ 文法に誤りがあれば、アウトプットウインドウの「エラー一覧」タブにエラー情報が表示されます。
- ④ エラー情報の表示をダブルクリックすると、カーソルがソースコードのエラーの箇所に移動します。

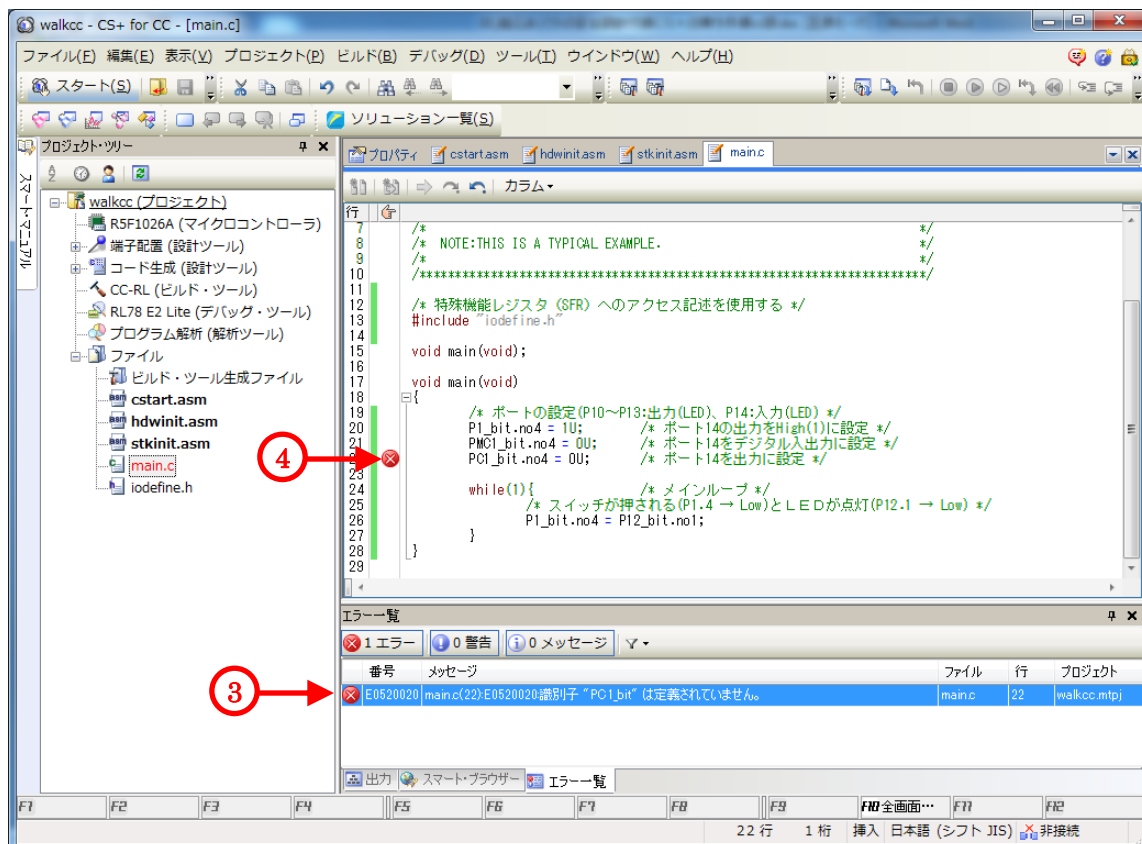


図 16 プログラムのビルドエラー

【実行形式プログラムのマイコンへの転送】

実行可能なプログラムをマイコンへ転送します。

- ・ RL78 E2 Lite をマイコンが実装されたプリント基板にケーブルで接続します。
- ・ RL78 E2 Lite をパソコンと USB ケーブルで接続します。

初回の接続時は、パソコンに RL78 E2 Lite の USB ドライバーがダウンロードされ、インストールされるので時間がかかります。

- ① プリント基板の電源スイッチをオンにし、メインメニューから「デバッグ(D)」の「デバッグ・ツールヘダウンロード(D)」をクリックします。

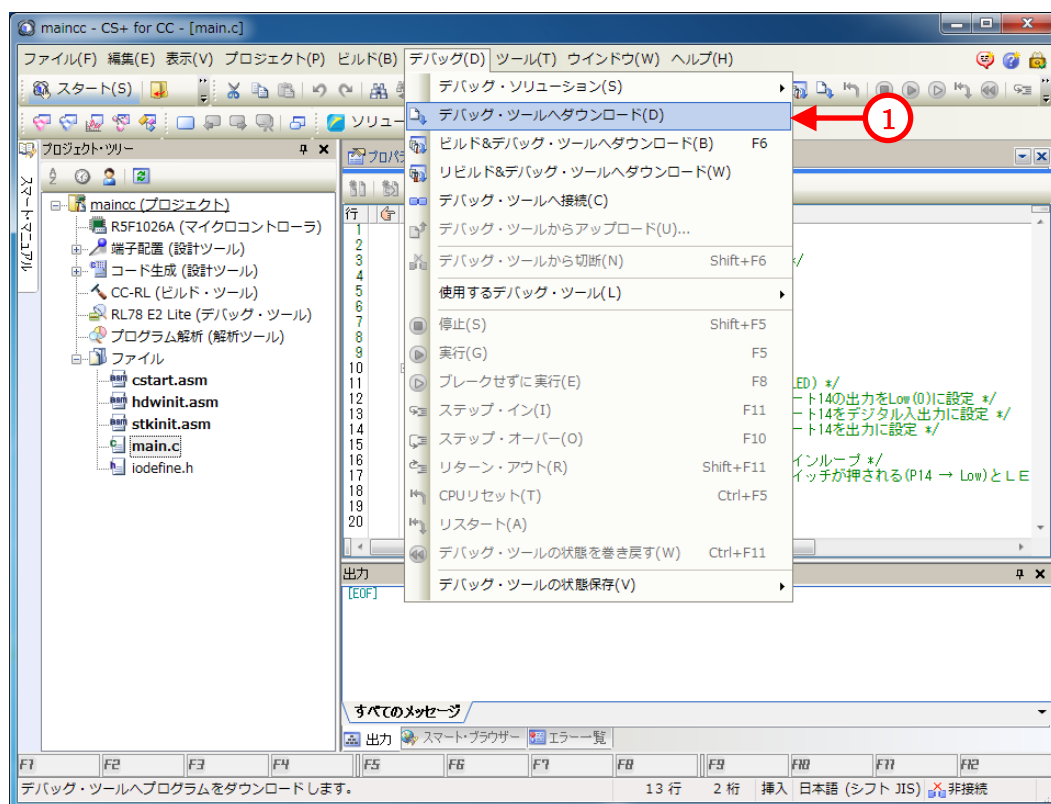


図 17 実行形式プログラムのマイコンへの転送

マイコンへの実行可能なコードの転送が始まると、進捗状況が表示されます。

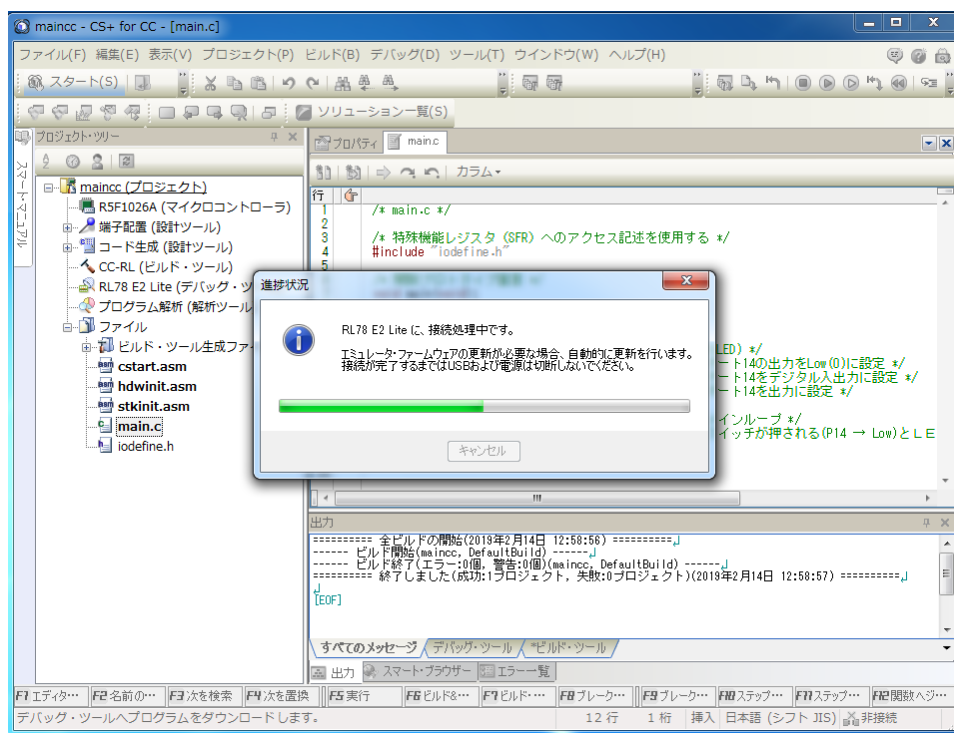


図 18 マイコンへのプログラムのダウンロード

マイコンへのダウンロードが完了するとメイン関数の先頭位置が表示されます。

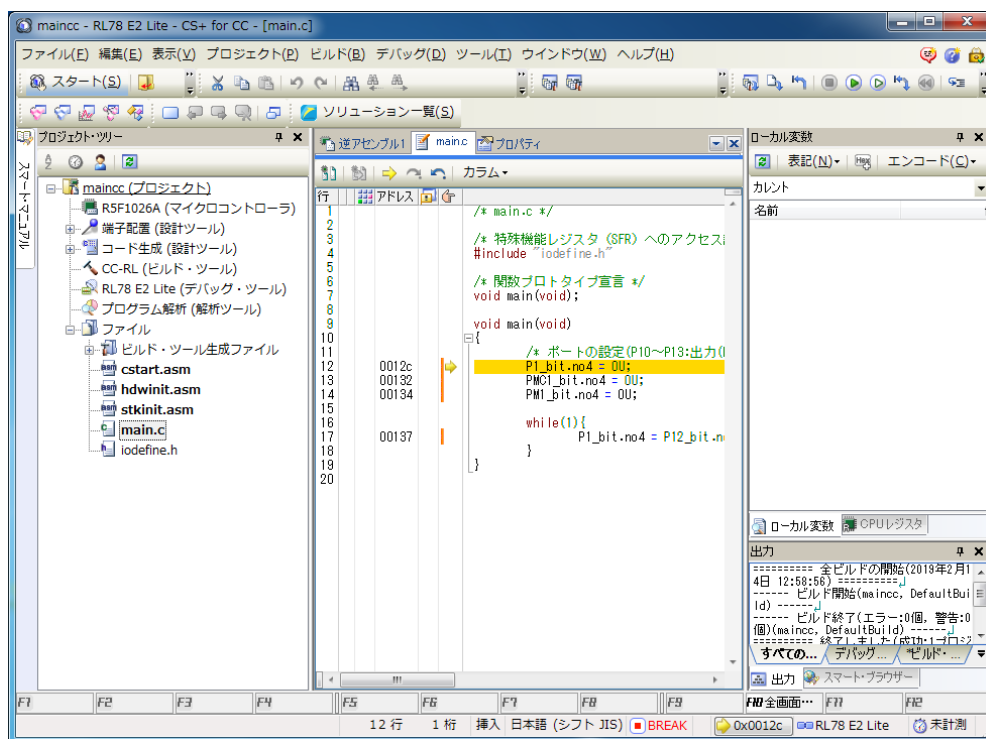


図 19 マイコンへのプログラムのダウンロード完了

【プログラムの実行と停止】

プログラムの実行では、プログラムを実行、停止する方法を説明します。

プログラムを実行します。

- ① プログラムを実行するにはメインメニューから「デバッグ(D)」の「実行(G)」をクリックします。
- ② ウィンドウ右上のツールバーの「実行ボタン」をクリックする方法やファンクションキーF5を押下する方法もあります。

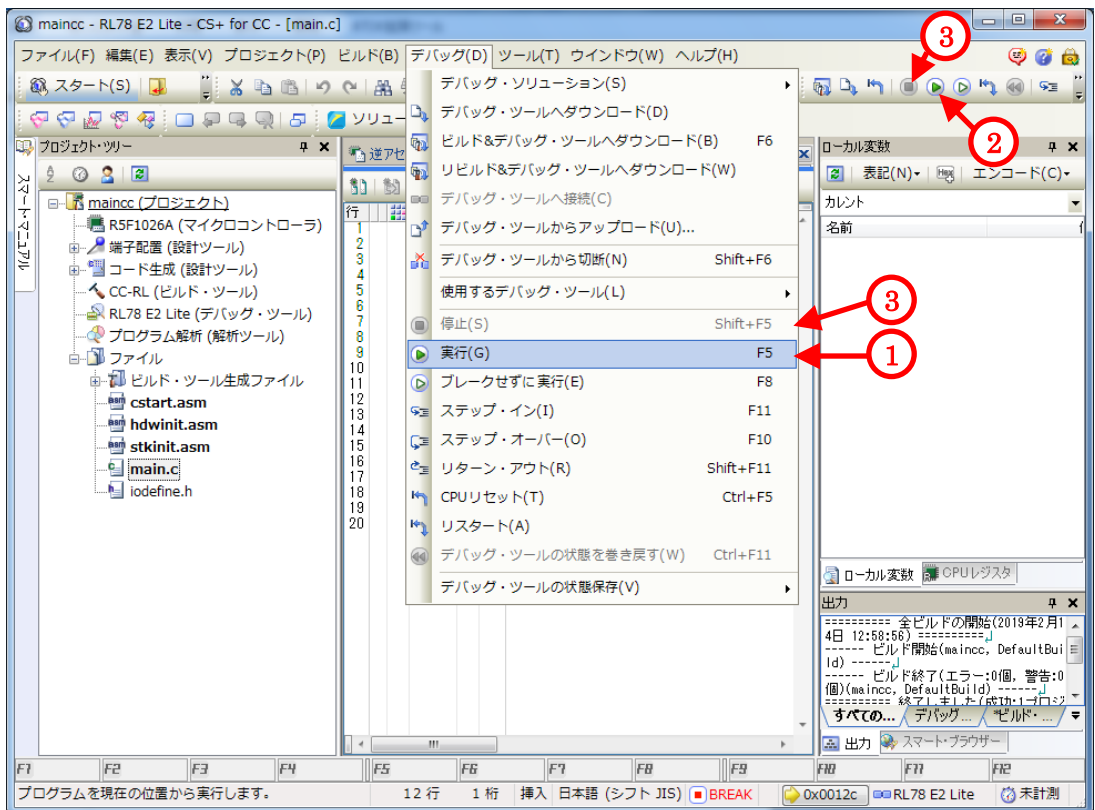


図 20 プログラムの実行

プログラムを停止します。

- ③ プログラムを停止するには、メインメニューから「デバッグ(D)」の「停止(S)」をクリックするか、ツールバーの停止ボタンをクリックします。

【統合開発環境の終了】

プログラムを修正した場合は、終了する前にメインメニューから「ファイル(F)」の「すべてを保存(L)」をクリックして、修正したプログラムを保存するのを忘れないようにしてください。

統合開発環境 CS+を終了します。

CS+を終了する前に、プログラムを停止して、デバッグツールとの接続を解除する必要があります。

