RX ファミリ用 C/C++ コンパイラパッケージ

アプリケーションノート：＜導入ガイド＞ルネサス統合開発環境 スタートアップガイド　RX 編

RJJ06J0096-0100
Rev.1.00
2010.04.20

本書では、RX ファミリ用 C/C++コンパイラパッケージを初めてご使用になる方を対象に、製品インストールからプロジェクト構築、ビルド、デバッグを開始するまでの一連の流れについて説明します。

目次

1. はじめに

RX ファミリ用 C/C++コンパイラパッケージは、RX ファミリの機能・性能を活かしたプログラムを、統合開発環境 ＋ C/C++コンパイラを用いて効果的に作成できるようにしたパッケージです。シミュレータも含まれており、実機が無い状態でもプログラムのデバッグ、性能評価を行うことができます。

ルネサス開発環境の構成を図 1 に示します。
連携ツール(ソフトウェア)の有色部はコンパイラパッケージに付属されているツールです。

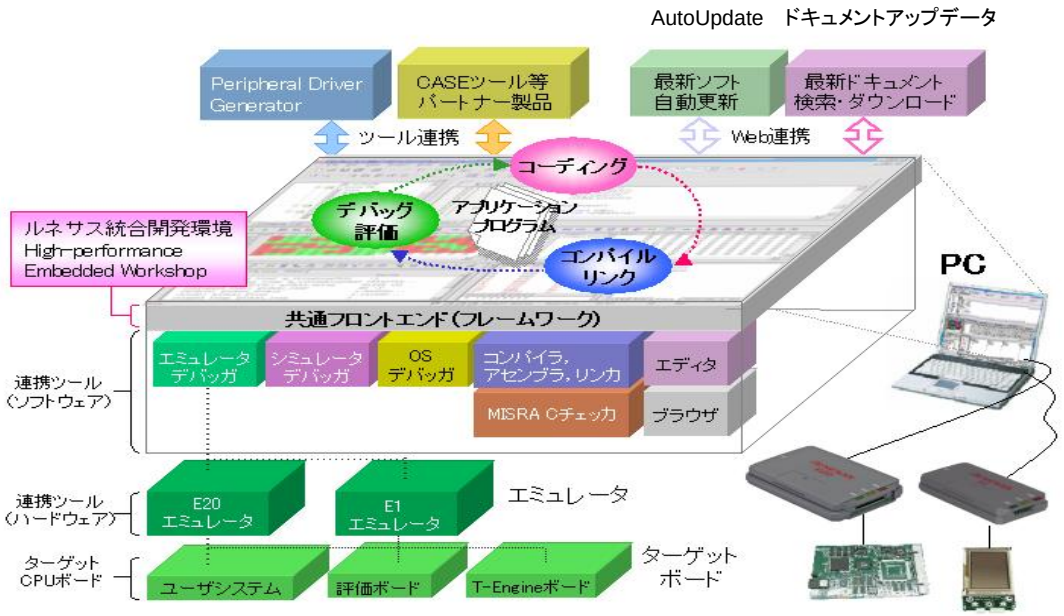


図 1 ルネサス開発環境の構成

• High-performance Embedded Workshop (以下、HEW と略します)

ルネサス統合開発環境。開発ツールを操作する、共通フロントエンドです。

エディタ、ブラウザ、プロジェクト構築機能のほか、多くの機能を搭載しており、コーディングからデバッグまでの一連の作業を行うことができます。

• コンパイラ、アセンブラ、リンカ

RX ファミリの性能を最大限に引き出すことのできる、ルネサス製コンパイラです。
コンパイラ、アセンブラ、リンカを纏めて、「ツールチェイン」と呼びます。HEW のビルド機能を用いることにより、ツールチェインを操作し、マイコン上で動作するプログラムを作成することができます。

[補足] ツールチェインを操作して、ソースプログラムからマイコン上で動作するプログラムを作成する操作を「ビルド」と呼びます。

• シミュレータ

RX の動作を PC 上でシミュレーション、評価することができます。ブレーク、トレースなどのデバッグ機能のほか、擬似割込み機能やタイマ機能などもあります。

図 1 における無色部は、コンパイラパッケージ以外のルネサス開発ツールです。
追加インストールを行うことで、図 1 に示すように HEW を用いて使用可能な機能が拡張されます。
例えば E20 エミュレータを追加インストールすると、図 1 における「連携ツール（ソフトウェア）」の「E20 エミュレータ制御」が追加インストールされ、HEW を用いて、E20 エミュレータを用いた評価を行うことができるようになります。
このように、ルネサス開発環境では、統合開発環境 High-performance Embedded Workshop(HEW)を中心として、シンプルかつ高機能な開発環境を構築することができます。

2. インストール方法

製品 CD-ROM をセットして、インストーラの指示に従ってください。既に HEW がインストールされている場合は、インストーラは既存 HEW へ上書きインストールを行います。
インストール時にインストールマネージャ（図 2）が起動する製品では、既に HEW がインストールされている場合でも、別フォルダに HEW 環境を新しく構築できます。

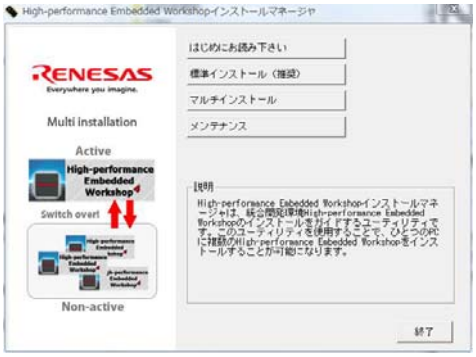


図 2 インストールマネージャ

3. プロジェクトの構築

RX610 を例に、プロジェクトの構築方法を説明します。

3.1 High-performance Embedded Workshop の起動

Windows スタートメニューより「High-performance Embedded Workshop」を選択してください。「ようこそ！」ダイアログボックス（図 3）が表示されます。

図 3 ようこそ! ダイアログ

[注]スタートメニューの表示内容は、ツールのインストール状況により異なる場合があります。
[注]インストールマネージャにより複数の HEW 環境を構築している場合、アクティブな HEW が起動されます。

3.2 ワークスペースの作成

3.2.1 ワークスペースとプロジェクト

HEW では、ワークスペースとプロジェクトという概念があります。

- **ワークスペース**

HEW を用いてプログラムを作成する場合の、一番大きな管理単位です。ワークスペースには、最低 1 つのプロジェクトが必要で、この 1 つのプロジェクトは、ワークスペース作成時に自動的に作成されます。また、ワークスペースは複数のプロジェクトを持つことができます。

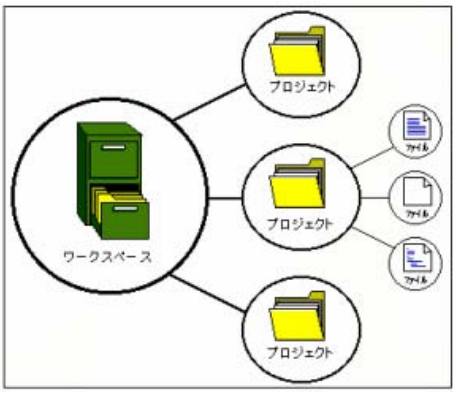


図 4 ワークスペースとプロジェクト

• プロジェクト

プログラムを作成する場合、特定機能をライブラリ化してモジュールを階層化する場合があります。このような場合、ライブラリ用のプロジェクトを作成・追加することができます。

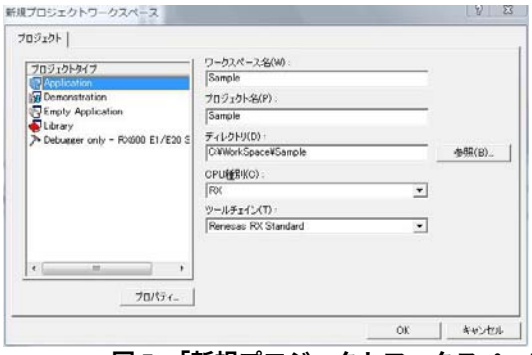


図 5 「新規プロジェクトワークスペース」ダイアログボックス

3.2.2 ワークスペースの作成

図 3 で「OK」ボタンを押すと、「新規プロジェクトワークスペース」ダイアログボックスが表示されます。（図 5）「プロジェクトタイプ」に「Application」を選択、「ワークスペース名」を入力してください。「プロジェクト名」を変更したい場合には、「ワークスペース名」を入力後、「プロジェクト名」を変更してください。

[注]指定した「ディレクトリ」が存在する場合、ワークスペースを作成することはできません。

[注]複数のコンパイラパッケージをインストールされている場合は、「CPU 種別」に「RX」を指定してください。

[注]RX ファミリ用 C/C++コンパイラパッケージ を使用する場合は、「ツールチェイン」に「Renesas RX Standard」を選択してください。

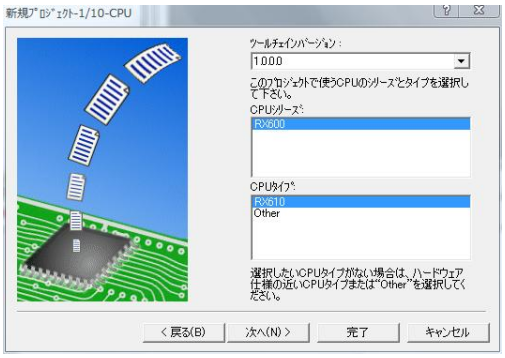


図 6 対象 MCU の選択

3.2.3 対象 MCU の選択

使用するツールチェインバージョンと対象 MCU を選択します。（図 6）「CPU シリーズ」に「RX600」を、「CPU タイプ」に「RX610」を選択してください。

[注]所望の MCU が表示されない場合は、コンパイラパッケージを最新版へリビジョンアップしてください。MCU が追加される場合があります。

図 7 MCU オプションの選択 1

3.2.4 MCU オプションの選択

「オプション」ダイアログボックス（図 7、図 8）では、コンパイラ、アセンブラ、リンカで共通のオプションを選択します。選択するオプションは、プロジェクト構築後も変更することができます。そのまま「次へ」を押してください。

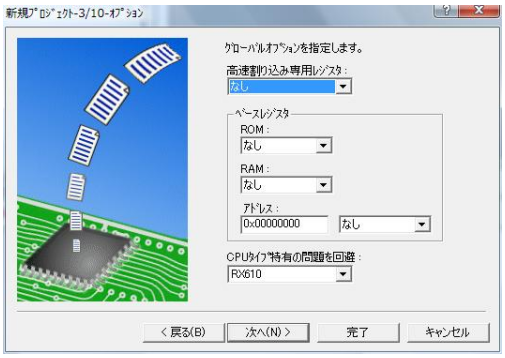


図 8 MCU オプションの選択 2

3.2.5　自動生成ファイルの設定

「生成ファイル」ダイアログボックス（図 9）では、HEW が
自動生成するファイルの設定を行います。

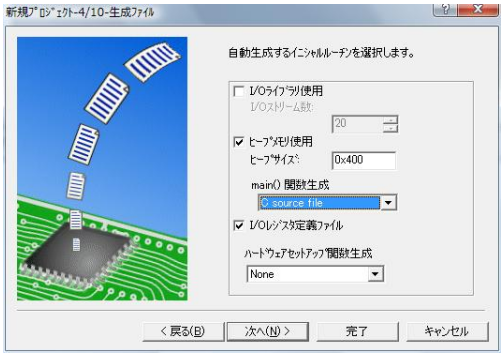


図 9 自動生成ファイルの設定

• I/O ライブラリの使用

標準入出力ストリーム数を指定します。

[注]標準入出力ストリームを使用する場合は、低水準インタフェースルーチンを実装する必要があります。低水準インタフェースルーチン実装方法の詳細は、コンパイラマニュアルを参照してください。

• ヒープメモリ使用

malloc, realloc,calloc 関数および C++(new 演算子)を使用する場合にチェックを付けてください。

• main( )関数生成

アセンブリ、C、C++言語から選択可能です。デフォルトは C 言語です。

• I/O レジスタ定義ファイル

C/C++言語で使用可能な、各 MCU のヘッダファイルを生成します。

• ハードウェアセットアップ関数生成

MCU の初期設定を行なうための HardwareSetup 関数を生成します。

[注]MCU の初期設定はユーザシステムに依存します。自動生成された HardwareSetup 関数の実装はユーザにて行ってください。

ここでは、「ハードウェアセットアップ関数を生成」から「C/C++ source file」を選択、「次へ」を押してください。

3.2.6　標準ライブラリの選択

「標準ライブラリ」ダイアログボックス（図 10）では、使用する標準ライブラリの選択を行います。ライブラリの選択は、プロジェクト構築後も可能です。そのまま「次へ」を押してください。

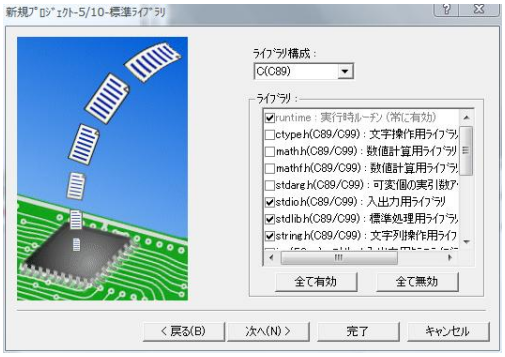


図 10 標準ライブラリの選択

3.2.7　スタック領域の設定

「スタック領域」ダイアログボックス（図 11）では、ユーザスタックサイズと割り込みスタックサイズを指定します。スタックポインタの初期値は、各 MCU の内蔵 RAM の最後の領域付近に設定されています。スタックポインタ値とスタックサイズは、プロジェクト構築後も変更可能です。そのまま「次へ」ボタンを押してください。

[注]スタック領域へのアクセスが遅いと、性能が非常に悪くなります。アクセスの速い内蔵 RAM 領域内へ設定することを推奨します。

図 11　スタック領域の設定

3.2.8　リセットベクタの設定

「ベクタ」ダイアログボックス（図 12）では、リセットベクタを生成するかどうかを指定します。各 MCU に最適な設定が行われています。そのまま「次へ」を押してください。

以上により、自動生成するファイルの設定は完了です。

図 12　リセットベクタの設定

3.2.9　デバッグターゲットの選択

「デバッガ」ダイアログボックス（図 13）では、生成したプログラムをどのターゲット上でデバッグするかを選択します。3.2.3（図 6）で選択した RX ファミリに適合するシミュレータ、エミュレータが自動的に表示されます（エミュレータは該当エミュレータ製品がインストールされている場合のみ、自動的に表示されます）。「RX600 Simulator」にチェックを付け、「次へ」を押してください。

[補足]ターゲットを複数選択することもできます。その場合は、デバッグ時に何れか 1 つのターゲットを「デバッグセッション」(5.1 章にて後述)より選択してデバッグを行うことになります。

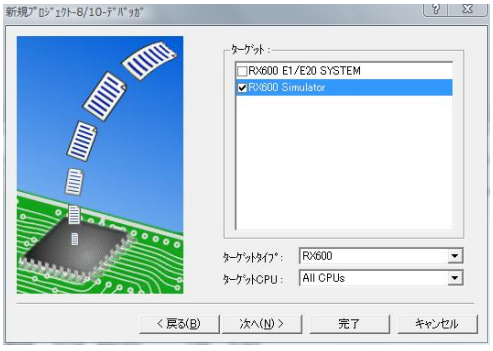


図 13　デバッグターゲットの選択

3.2.10　デバッガオプションの選択

「デバッガオプション」ダイアログボックス（図 14）では、図 13 で指定したターゲットに対するオプション設定を行ないます。本設定は特に変更する必要はありません。そのまま「次へ」を押してください。

[注]ターゲットを複数選択した場合には、各ターゲット毎にオプションダイアログボックスが表示されます。何れも設定を変更する必要はありません。

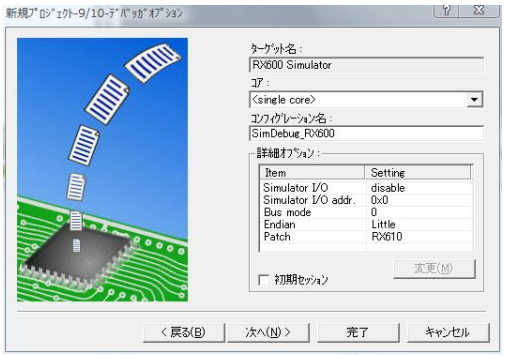


図 14 デバッガオプションの選択

3.2.11　ワークスペース作成の完了

以上により、ワークスペース作成のために必要な設定は完了です。「生成ファイル名」ダイアログボックス（図 15）では、HEW が自動生成するファイル名と各々のファイルの概要を一覧表示します。

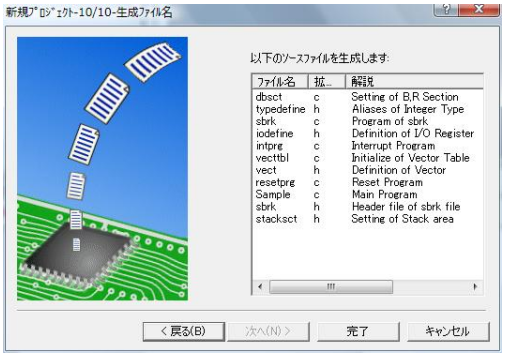


図 15 生成ワークスペースの一覧表示

図 15 で「完了」ボタンを押すと、「概要」ダイアログボックス
（図 16）が表示します。本章で構築されたワークスペースの
詳細情報を確認することができます。図 16 の下部「サマリの・・・という名前で保存する」にチェックが付いている場合、「概要」ダイアログボックスに示されている詳細情報をプロジェクトフォルダ（3.3 参照）内にある ReadMe.txt ファイルに保存します。

図 16 生成ファイルの詳細

[注]ReadMe.txt に保存される情報は、プロジェクト構築時の情報です。プロジェクト構築後に変更した情報は含まれません。

図 16 で「OK」ボタンを押すと、HEW がワークスペースを開いた状態で起動し、図 17 に示す状態になります。図 17 で示される用語は以下のとおりです。

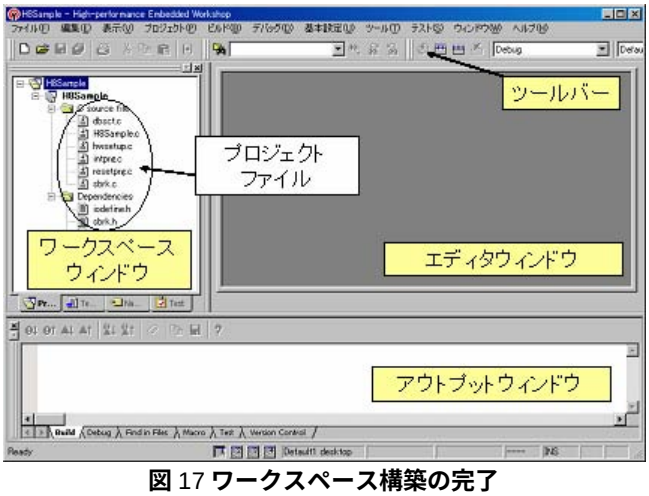


図 17 ワークスペース構築の完了

• ツールバー

各操作を簡単に行うことのできるボタンです。

• ワークスペースウィンドウ

ワークスペースに含まれるプロジェクトと、各プロジェクトに含まれるファイルを一覧表示します。ファイルをダブルクリックすると、エディタウィンドウにファイル内容を表示します。ワークスペースウィンドウは 4 つのタブで構成されており、上記のほかにブラウザ機能、テスト機能などを設定、使用することができます。

• エディタウィンドウ

ソースファイル内容を表示、編集を行うことができます。
デバッグ時は C ソースレベルデバッガとなりますので、「デバッグ」→「ソースプログラム編集」→「再ビルド」→
「デバッグ」の一連の流れをスムーズに行うことができます。

• アウトプットウィンドウ

ビルド結果や実行結果などを表示するウィンドウです。

3.3　ワークスペース内のフォルダ構成

3.2 で自動生成されるフォルダ・ファイルの構成を図 18 に示します。自動生成されるソースファイルは、プロジェクトフォルダ内に配置されます。

[注]ワークスペース／プロジェクトフォルダ内には、HEW が自動生成する設定ファイル（*.hws,*.hwp,*.hsf）などがあります。設定ファイルは、手動で編集を行わないでください。

[注]ユーザプログラムを任意フォルダ内に作成、プロジェクトへ追加することができます。新しいフォルダを追加する場合は、ワークスペースフォルダ以下の階層へ追加することを推奨します。

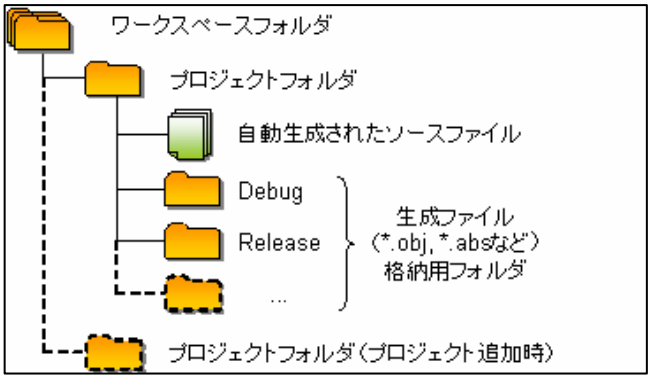


図 18 自動生成ファイルのフォルダ構成

4. ビルド

4.1 ビルドとは

ツールチェインを操作して、ソースプログラムからマイコン上で動作するプログラムを作成する操作を、「ビルド」と呼びます。ビルドの流れを図 19 に示します。

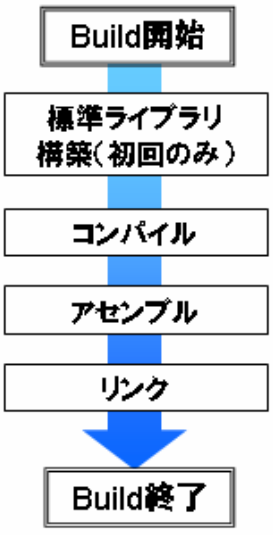


図 19 ビルドの流れ

- **標準ライブラリの構築**

  RX ファミリ用 C/C++コンパイラパッケージでは、ライブラリに対しても最適化の設定を行うことができます。そのため、3.2.6 で指定したライブラリを最初のビルド時に構築する仕様になっています。

  [注]ライブラリ構築には多少時間がかかります。
  [注]ビルドオプション（4.2）の「CPU」タブ、または「標準ライブラリ」タブで設定されるオプションを変更すると、ライブラリを再構築します。

- **コンパイル**

  コンパイラを起動し、C/C++ソースプログラム(*.c/*.cpp)をコンパイルします。

- **アセンブル**

  アセンブラを起動し、アセンブリ言語ソースプログラム(*.src,*.s)をアセンブルします。

- **リンク**

  中間ファイル(*.obj)およびライブラリ(*.lib)等をリンクし、マイコン上で動作するプログラムを作成します。デバッグ段階で使用するロードモジュールファイル(*.abs)と同時に、ROM ライタなどで製品に組み込むための S タイプファイル(*.mot)、HEX ファイル(*.hex)形式などの出力を行うことができます。

4.2 ビルドオプション

4.2.1 ビルドオプションの設定

「ビルド」→「RX Standard Toolchain…」を選択してください。

「RX Standard Toolchain」ダイアログボックス（図 20）が表示されます。ツールチェインに対する全てのオプション設定は、本ダイアログで行うことができます。

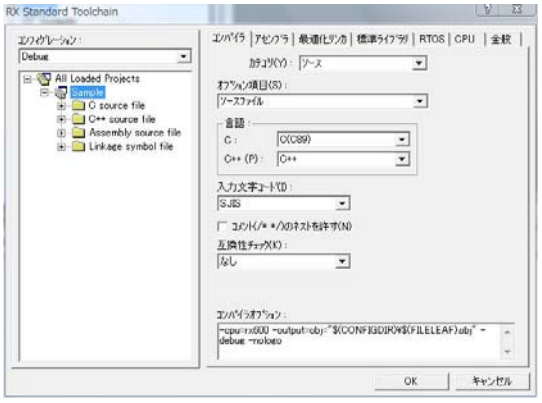


図 20 ビルドオプションの設定

- **「コンパイラ」タブ**

  コンパイラオプションを設定することができます。

- **「アセンブラ」タブ**

  アセンブラオプションを設定することができます。

- **「最適化リンカ」タブ**
  リンカオプションを設定することができます。

  [補足]ルネサス製 RX リンカは、最適化機能を搭載しています。
  そのため、「最適化リンカ」とも呼びます。

- **「標準ライブラリ」タブ**
  ライブラリ構築時のオプションを設定することができます。

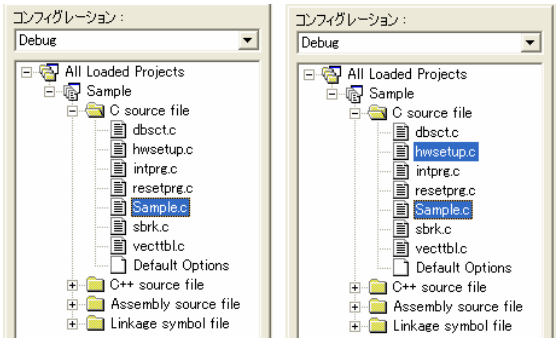
- **「RTOS」タブ**
  RTOS のオプションを設定することができます。

- **「CPU」タブ**
  ビルドを通して共通となる MCU オプション(3.2.4)を設定することができます。

- **「全般」タブ**
  HEW のオプションを設定することができます。

4.2.2　オプション設定対象

初期設定では、プロジェクトファイル全体に共通のオプション設定が行われていますが、ファイル個別にオプションを設定することもできます。個別に指定する場合、設定対象は、図 21 の左側の画面で指定します。指定例を図 21 に示します。単一ファイル指定だけでなく、Ctrl や Shift キーを用いて、複数ファイルに対して、同じオプションを同時に設定することもできます。ここでは、特にオプション等の変更は行わず、オプションダイアログを閉じてください。



（単一ファイル指定）　（複数ファイル指定）

図 21　オプション設定対象の選択

4.3　コンフィグレーション

4.2 で設定したオプション設定を保存しておき、様々なオプション設定でのプログラム評価を素早く行えると便利です。

HEW では、プロジェクトのオプション設定情報を「コンフィグレーション」という単位で登録・管理することができます。

初期状態では、「Debug」、「Release」コンフィグレーションが存在しています。各コンフィグレーション（オプション設定）でのビルド結果は、図 18 の「Debug」「Release」で示されるように、別々のフォルダへ保存・管理されます。　様々なオプション設定に対して「コンフィグレーション」を生成、保存することで、様々なオプション設定の評価を容易に行うことができます。「コ

図 22　「ビルド」ツールバー

ンフィグレーション」の切替えは、「ビルド」ツールバーのコンフィグレーション」ドロップダウンリスト（図 22）により行います。

[注]初期状態の「Release」コンフィグレーションは、「Debug」コンフィグレーションのデバッグ情報出力を抑止したものです。その為、「Release」コンフィグレーションと「Debug」コンフィグレーションは同じプログラムが出力されます。

4.4　ビルドの実行

ビルドを行うには、「F7」キーを押すか、HEW メニュー「ビルド」→「ビルド」を選択するか、ツールバー上部の「ビルド」ボタン（図 23）を押してください。
ビルド後、アウトプットウィンドウ（図 17）にエラーが発生しなければプログラムの作成は終了です。

図 23 「ビルド」ボタン

[注]ビルドを行うには、オプションダイアログ（図 20）を閉じる必要があります。オプションダイアログを閉じることでツールチェインへの設定オプションが確定します。

[補足]エラーやウォーニングが出力された場合、「F8」キーを押すと、該当行をエディタに表示します（コンパイラ、アセンブラエラー時）。また、アウトプットウィンドウのエラー／ウォーニング行にカーソルを置いて「F1」キーを押すと、エラーヘルプが表示（図 24）されます。

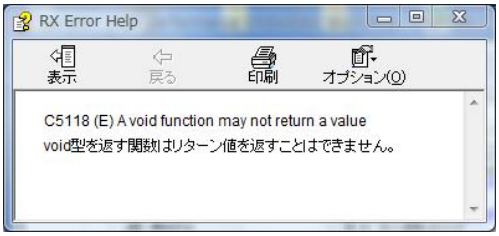


図 24　エラーヘルプ（「F1」キーで表示）

以上により、HEW によるプログラムの自動生成は終了です。

5.　デバッグ

本章では、前章で作成したプログラムをターゲット上へ転送し、プログラムのデバッグを開始するまでの手順を説明します。

5.1　デバッグセッション

オプション設定を「コンフィグレーション」で管理できたのと同様に、HEW では、デバッグ環境を「デバッグセッション」という単位で登録・管理することができます。デバッグセッションの切替えは、「ビルド」ツールバーの「デバッグセッション」ドロップダウンリスト（図 25）で行うことができます。「デバッグセッション」に登録・保存される情報は以下の通りです。

（2つあるドロップダウンリストのうち、右側がデバッグセッション）

図 25「ビルド」ツールバー

• 接続ターゲット

プログラムのデバッグを行うターゲットの指定です。シミュレータ、エミュレータ（インストール済製品）を選択することができます。初期状態では、3.2.9 で選択したターゲット先が登録されています。

• デバッグ対象プログラム

ロードモジュール（*.abs、*.mot など）ファイル。初期状態では、選択されたコンフィグレーションの*.abs ファイルが登録されています。

• デバッグ時のウィンドウ情報

デバッグ時に開いている各ウィンドウ（レジスタ、メモリ、ウォッチなど）情報を使用中のイメージで保存します。つまり、デバッグセッションを登録・保存しておくと、デバッグを再開する場合に「デバッグセッション」を選択するだけで、前回保存したデバッグ環境をすぐに復元（図 26）することができます。

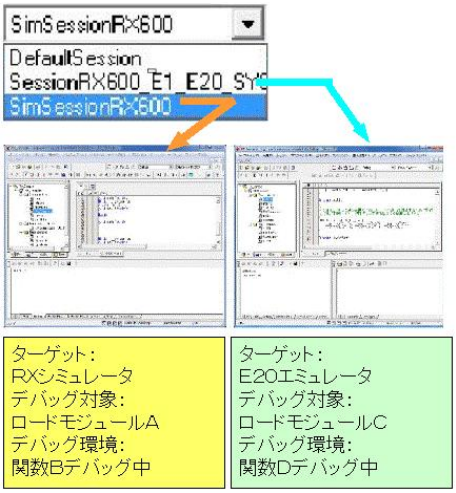


図 26 デバッグ環境の復元

5.2 ターゲットの接続

ターゲットとの接続を行うためには、「デバッグセッション」ドロップダウンリスト（図 25）から所望のターゲットを選択します。「SimSessionRX600」デバッグセッションが、プロジェクト構築時の指定(3.2.9)により登録されていますので、「SimSessionRX600」デバッグセッションを選択してください。デバッグセッションを選択すると、HEW は対象となる連携ソフトウェア（図 1）を起動し、接続を開始します。シミュレータを起動した場合は、「シミュレータの設定」ダイアログ（図 27）が表示されます。「シミュレータの設定」ダイアログは、シミュレータのＣＰＵの構成やサポートする周辺モジュール機能の設定を行うためのダイアログボックスです。「Module Name」に示されたモジュール（図 27 ではコンペアマッチタイマと割り込みコントローラ）にチェックを付けると、その周辺モジュール機能が使用可能になります。

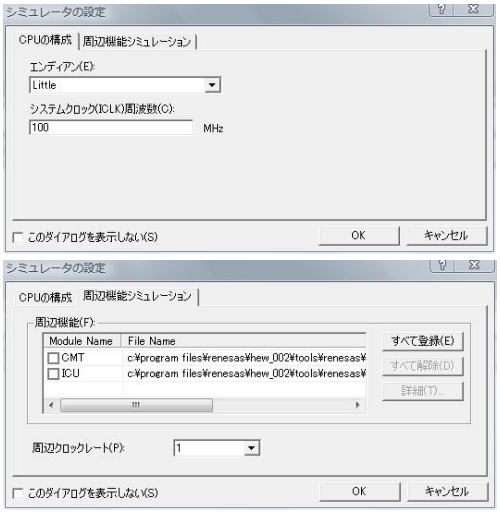


図 27 「シミュレータの設定」ダイアログボックス

ここでは、特にチェックを付けずに「OK」ボタンを押してください。HEW 上部のツールバーにデバッグ操作用のツールバー群が追加（図 28）され、シミュレーション（プログラム実行）およびデバッグを行うことができるようになります。

[注]エミュレータをご使用になる場合、接続時にエミュレータ固有の設定が必要になる場合があります。エミュレータ接続時の設定は、エミュレータユーザーズマニュアルまたはアプリケーションノートを参照してください。

アウトプットウィンドウ（図 17）の「Debug」タブに「Connected」が表示されれば、ターゲット（シミュレータ）との接続は完了です。

図 28 シミュレータ起動前後のツールバー表示（上：起動前、下：起動後）

5.3　デバッグ対象プログラムのダウンロード

次に、デバッグ対象プログラムのダウンロードを行います。ターゲットとの接続時、ワークスペースウィンドウ（図 17）には「Download modules」フォルダおよびロードモジュールファイルが追加（図 29 上図）されています。ロードモジュールをダブルクリックすると、ダウンロードが行われ、ダウンロード済みであることを示す「↓」マークがロードモジュールファイルに表示（図 29 下図）されます。

[注]ロードモジュールファイルの右側の数値は、「オフセットアドレス」です。「オフセットアドレス」には「0」を指定（初期値は 0）するようにしてください。

[注]エミュレータをご使用の場合で、外部メモリ領域へのダウンロードを行う場合には、予め当該領域へのアクセスが正しく行えるように、ポートやバスコントローラ等の設定を行う必要があります。

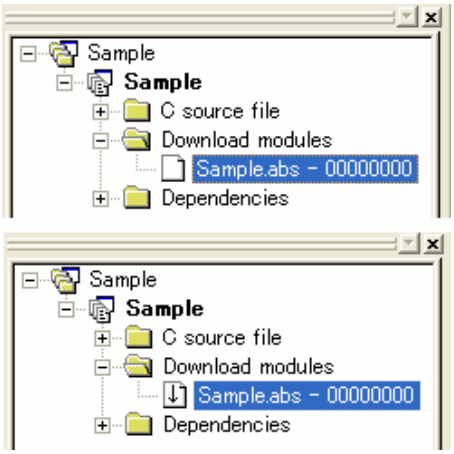


図 29 ロードモジュールのダウンロード
（上：ダウンロード前、下：ダウンロード後）

5.4　プログラムの実行とデバッグ

前節で、プログラムの実行およびデバッグに必要な準備は完了しました。本節では実際に簡単な実行とデバッグを行います。

5.4.1　リセット

まず最初にリセットを行い、リセットベクタへ分岐することを確認してみましょう。リセットを行うためには、ツールバーの「CPU リセット」ボタン（図 30）を押してください。リセットベクタからパワーオンリセットアドレス、スタックポインタ初期値を取得し、PowerON_Reset_PC 関数の先頭で停止します（図 31）。

図 30 「CPU リセット」ボタン

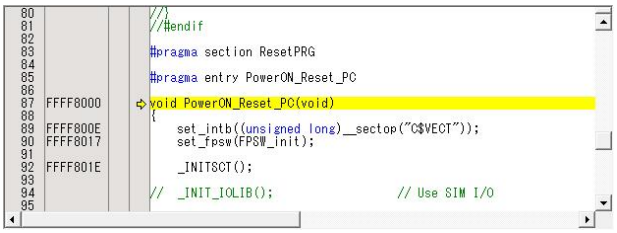


図 31 リセット後の状態

このように、HEW ではプログラムが停止した際、該当するソース行があれば、自動的に該当するソースプログラムを表示します。

[注]・ ソースプログラムが表示されない場合は、以下の事項を確認してください。
- **ビルド時、コンパイラ、リンカの両方にデバッグオプションを指定していない場合は、指定してください**
- **ソースファイルがビルド時と同じ場所に存在していない場合は、再度ビルドを行ってください。**

5.4.2　ソースプログラム表示モードの変更

「表示モード」をエディタウィンドウ左上のボタン（図 32）により切替えることができます。

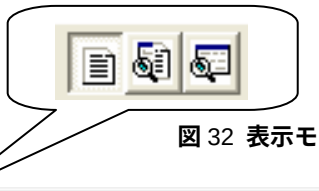


図 32 表示モード切替えボタン

- **「ソース表示モード」ボタン　（図 32 左）**

ソースプログラムを表示・編集を行うモードです。（図 31 参照）

[注]ソースプログラムの編集は、「ソース表示モード」のみ可能です。

- **「混合表示モード」ボタン　（図 32 中）**

ソースプログラムと、アセンブリ言語コードを混合表示するモードです。ソースプログラムに対応するアセンブリ言語コードが表示されるので、オブジェクトコードの詳細を解析するのに便利なモードです（図 33）。

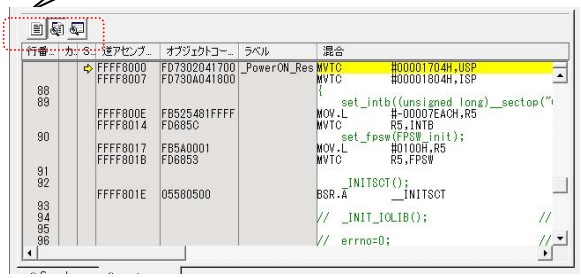


図 33 混合表示モード

- **「逆アセンブリ表示モード」ボタン　（図 32 右）**

逆アセンブリ表示のみを行うモードです。ライブラリ等ソースプログラムが無い場合でも、オブジェクトコード内容を確認することができます。

5.4.3　ステップ実行（ステップイン、ステップオーバー）

ステップ実行ボタンを押すと、プログラムのステップ実行を行います。

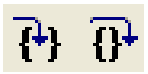


図 34 「ステップ実行」ボタン
左：ステップイン、
右：ステップオーバー

- **「ステップイン」ボタン　（図 34 左ボタン）**
  ステップ実行を行います。関数入口で押すと、関数内の次の行で停止します。

- **「ステップオーバー」ボタン　（図 34 右ボタン）**
  ステップ実行を行います。関数入口でボタンを押すと、関数実行後の次の行で停止します。

[注]ステップ実行時、停止箇所のソースプログラムが無い場合（ライブラリなど）は、「逆アセンブルウィンドウ」に切り替わります。

5.4.4　レジスタ値の参照・設定

レジスタボタン（図 35）を押すと、レジスタウィンドウ（図 36 左）が表示されます。例えばレジスタウィンドウを開いた状態でステップ実行を行うと、PC（プログラムカウンタ）値が更新されることが確認できます。レジスタ値の変更も可能です。変更したいレジスタをダブルクリックすると「レジスタ値設定」ダイアログボックス（図 36 右）が表示されます。

図 35 「レジスタ」ボタン

図 36 レジスタ値の参照・設定

5.4.5　関数・変数定義の参照

ワークスペースウィンドウ（図 17）には、全部で 4 つのタブがあります。左から 3 番目「Navigation」のタブを選択してください。プログラム内の関数および大域変数の一覧が表示されますので、「C Functions」の「main(void)」（図 37）を選択ダブルクリックしてください。main 関数の定義部分のソースプログラムが表示されます。

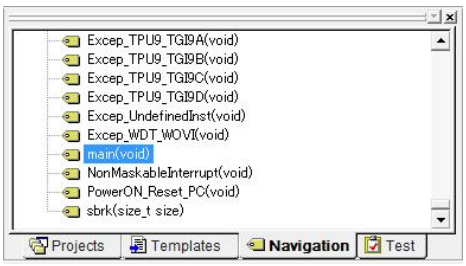


図 37 関数・変数定義の参照

5.4.6　ブレークポイントの設定

次に、ブレークポイントの設定を行なってみましょう。5.4.5 の操作を行うと、main 関数の先頭にマウスカーソルが置かれている状態になりますので、そのまま「F9」キーを押すか、エディタウィンドウの「S/W（ソフトウェア）ブレークポイント」のカラムをダブルクリックしてください。「S/W ブレークポイント」カラムに●マークが表示されます（図 38）。

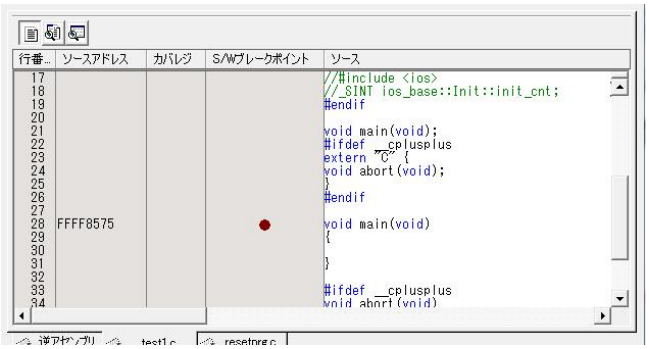


図 38 ブレークポイントの設定

5.4.7　プログラムの実行

実行を行うためには、「F5」キーを押すか、「実行」ボタン（図 39）を押してください。停止位置からプログラムが実行され、ブレークポイント設定箇所で停止します。

図 39 「実行」ボタン

[注]複数行でソースプログラムのアドレスが同じ場合（図 39）、ブレークポイントと停止位置（矢印部）が異なって表示される場合があります。

5.4.8　ウォッチ

変数の参照・編集を行う場合には、「ウォッチ」ボタン（図 40）を押してください。ウォッチウィンドウ（図 41）が表示されますので、参照・編集を行いたい変数名を登録してください。なお、エディタウィンドウ上で変数をダブルクリック（反転表示されます）、反転表示された変数部分を直接ウォッチウィンドウ上へドラッグ&ドロップして登録することもできます。

図 40 「ウォッチ」ボタン

[注]3 章で説明した手順に従ってプロジェクトを作成した場合自動生成されるプロジェクト内のソースファイルには、変数は定義されておりません。

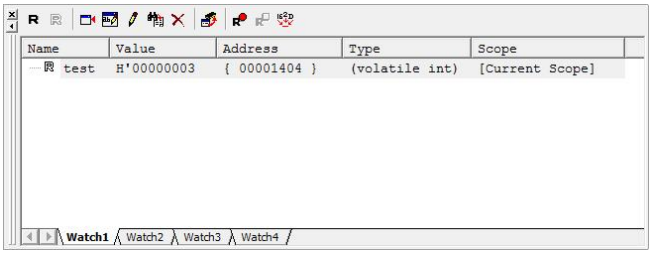


図 41 「ウォッチ」ウィンドウ

5.4.9　プログラムの修正と再評価

デバッグ中、プログラムの修正が必要になった場合には、以下の手順により簡単に再評価を行うことができます。

- **該当ソースを「ソースウィンドウ」モード(5.4.2 参照)で修正してください。**
- **「F7」キーを押すか、「ビルド」ボタンを押してください（4.4 参照）。再ビルドが行われます。**
- **5.3 の手順により、再度プログラムのダウンロードを行ってください。（再ビルド後、自動的にダウンロードが行われる設定にされている場合には、手動でダウンロードを行う必要はありません）**

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/
- お問合せ先
  http://japan.renesas.com/inquiry

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2010.4.20	—	初版発行

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ものですが、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、海底中継機器、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て、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が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/inquiry

Colophon 1.0