

RX64M グループ

R01AN2144JJ0100

RX Driver Package ユーザーズマニュアル

Rev.1.00

2014.09.01

要旨

これは、RX64M グループ用 RX Driver Package Ver1.00 のアプリケーションノートです。

RX Driver Package は、Firmware Integration Technology (以下 FIT と称す) に対応したデバイスドライバやミドルウェアをパッケージしたソフトウェアプラットフォームです。RX マイコンの豊富な周辺機能を手軽に評価するための環境を提供します。

RX Driver Package には、RX マイコンに搭載されている周辺モジュールのデバイスドライバ、RX マイコン用に開発されたミドルウェア、各種インタフェースモジュール、Board Support Package (以下 BSP と称す) モジュールが含まれています。ユーザは、RX Driver Package に入っているモジュールを自由に組み合わせ、それらを使うアプリケーションを作成するだけでシステムを構築することができ、すぐに評価を開始することができます。

本ドキュメントは、RX Driver Package と RX Driver Package Application の基本構成や特徴と、本アプリケーションノートに入っている FIT モジュールの構成について説明します。

RX Driver Package の使用方法や、RX Driver Package を使用したアプリケーションの評価手順については、関連ドキュメントを参照してください。

製品への利用においては、お客様の環境に合わせて、十分に評価を行ってください。

動作確認デバイス

RX64M グループ (Renesas Starter Kit+ RX64M)

関連ドキュメント

RX ファミリ ボードサポートパッケージモジュール Firmware Integration Technology (R01AN1685JU)

Firmware Integration Technology ユーザーズマニュアル(R01AN1833JU)

RX ファミリ e² studio に組み込む方法 Firmware Integration Technology (R01AN1723JJ)

RX ファミリ CubeSuite+に組み込む方法 Firmware Integration Technology (R01AN1826JJ)

RX Driver Package Application に付属するユーザーズマニュアル

目次

1. 概要.....	3
1.1 適用.....	3
1.2 RX64M グループ用 RX Driver Package について.....	3
1.3 動作環境.....	3
2. RX Driver Package とは.....	4
2.1 RX Driver Package の構成.....	4
2.2 RX Driver Package の特徴.....	6
2.3 RX Driver Package のロードマップ.....	6
3. RX64M グループ用 RX Driver Package の構成.....	7
3.1 フォルダ構成.....	7
3.2 モジュール構成.....	8
3.3 FIT モジュール一覧.....	9
4. 使用方法.....	10
4.1 説明で使用する環境.....	10
4.2 e ² studio に RX Driver Package をインストールする.....	10
4.3 アプリケーションの作成.....	11
4.3.1 ワークスペースとプロジェクトを作成する.....	11
4.3.2 FIT モジュールの組み込み準備.....	17
4.3.3 FIT プラグインで FIT モジュールをインストールする.....	20
4.3.4 LED 点灯プログラムを作成する.....	24
4.3.5 プログラムをビルドし動作を確認する.....	25
5. RX Driver Package Application について.....	28
5.1 RX Driver Package Application の構成.....	28
5.2 RX Driver Package Application の特徴.....	28
5.3 RX Driver Package Application の使用例 (e ² studio を使用する場合).....	29
5.4 RX Driver Package Application と組み合わせて使用する場合.....	29
6. システム構成.....	30
7. 補足.....	31
7.1 M3S-T4-Tiny (TCP/IP プロトコルスタックライブラリ) について.....	31
ホームページとサポート窓口.....	32

1. 概要

1.1 適用

本ドキュメントは、RX64M グループ用 RX Driver Package Ver1.00 に適用します。

1.2 RX64M グループ用 RX Driver Package について

RX64M グループ用 RX Driver Package (以下本パッケージ と称す) は、RX64M 用のデバイスドライバや各種ミドルウェアなどが多数収録された RX Driver Package です。本パッケージを使用することで、RX64M の豊富な周辺モジュールを使用した、さまざまな用途のアプリケーションプログラムを開発することができます。

1.3 動作環境

本パッケージの動作環境を以下に示します。

表1.3.1 動作環境

対応 MCU	RX64M グループ
評価ボード	Renesas Starter Kit+ RX64M
統合開発環境 (IDE)	e ² studio V3.0.1.09 以降 CubeSuite+ V2.02.00 以降
クロスツール	RX ファミリ用 C/C++コンパイラパッケージ V2.01.00 以降
エミュレータ	E1, E20

2. RX Driver Package とは

2.1 RX Driver Package の構成

RX Driver Package は、アプリケーションプログラムの開発に必要な下位のソフトウェア部品をまとめた、ソフトウェアプラットフォームです。RX Driver Package 上でさまざまなアプリケーションを開発することができます。

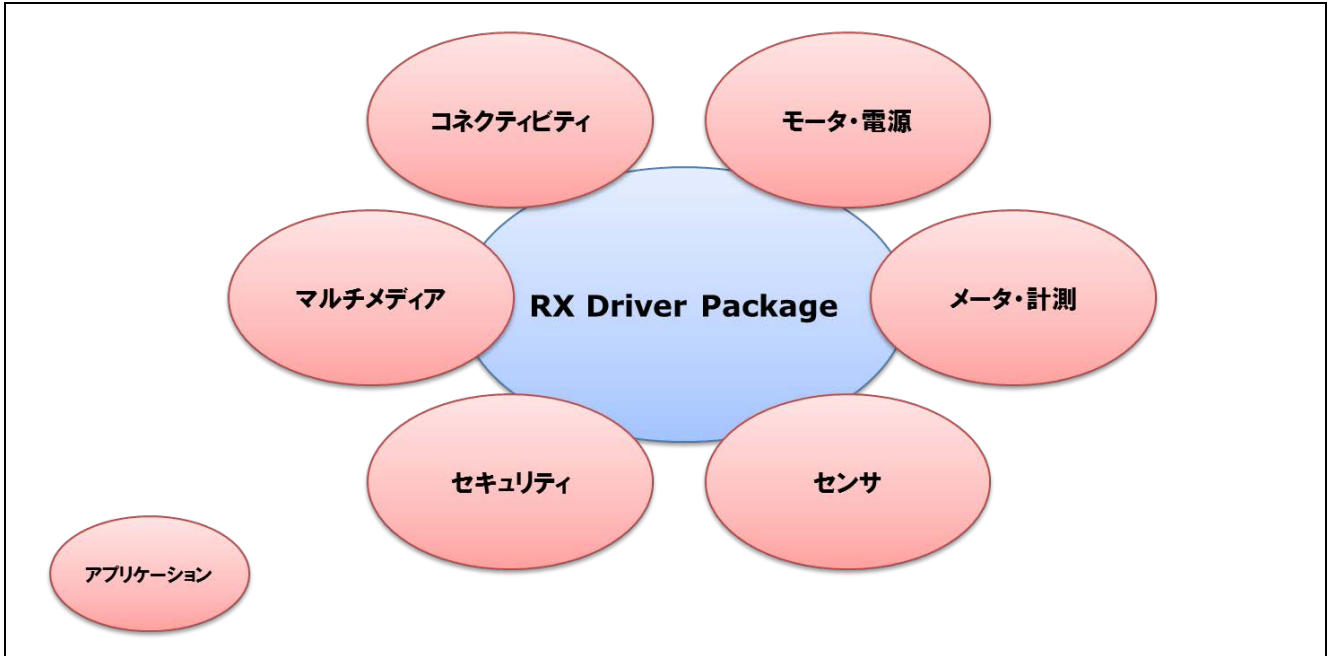


図2.1.1 RX Driver Package Application の種類

RX Driver Package は、以下の図に示すような BSP モジュール、MCU の周辺モジュール用デバイスドライバ、ミドルウェア（TCP/IP、ファイルシステム等）と、各種インタフェースで構成されています。

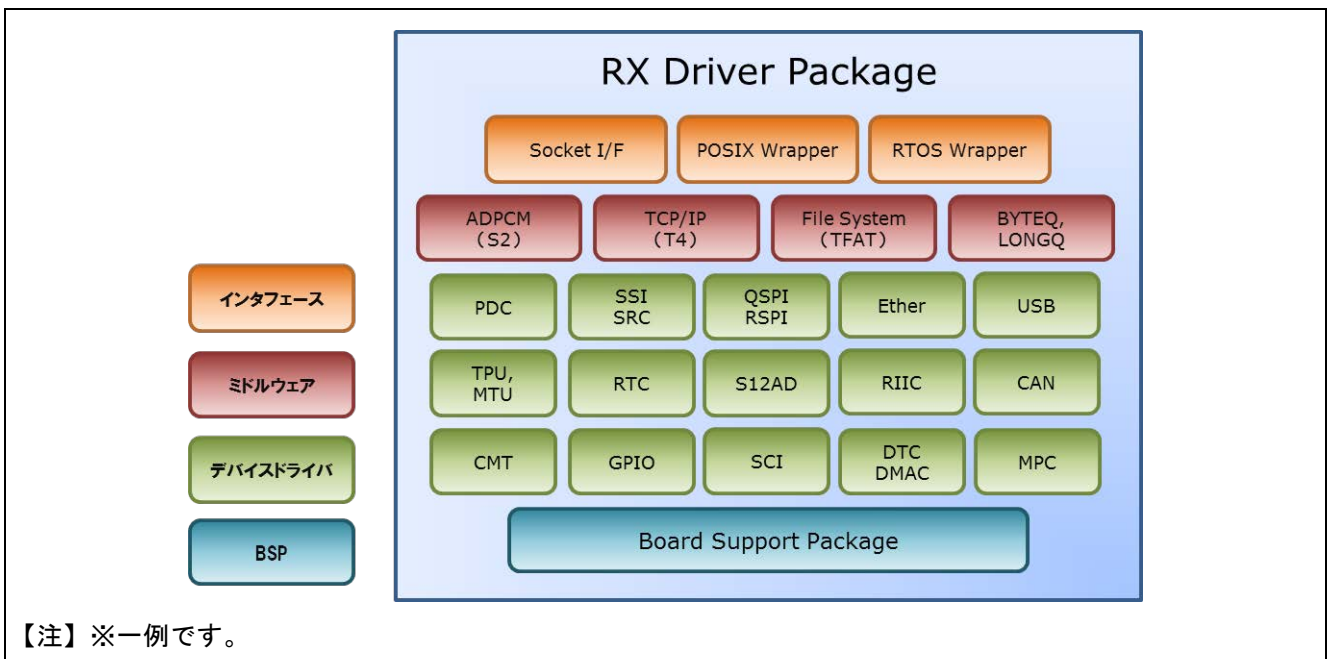


図2.1.2 RX Driver Package の構成例

例えば、以下の図に示すように、RX Driver Package の中から必要な部品を集め、アプリケーションプログラムを用意すれば、システムを構築することができます。

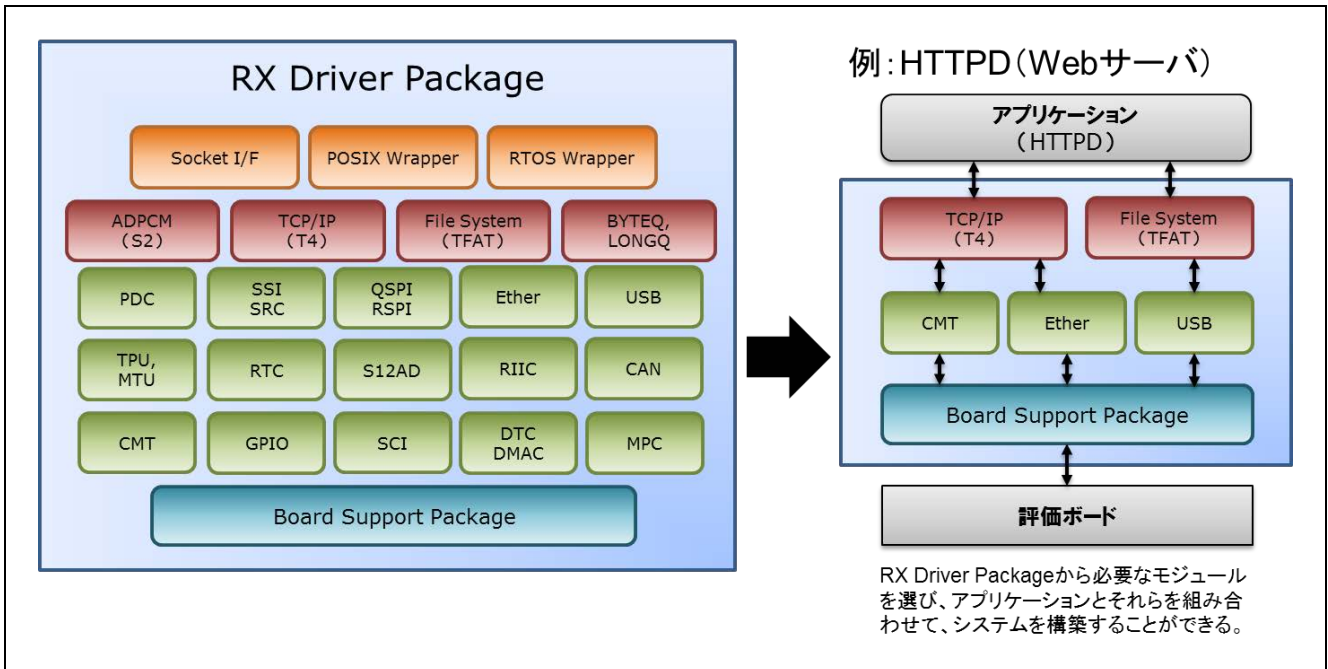


図2.1.3 RX Driver Package の構築例

また、全てのモジュールはFIT仕様に対応し、フォルダ構成やAPIが共通化（一部を除く）されているため、モジュールの入れ替えや、別マイコンへの移植が容易に行える構造になっています。

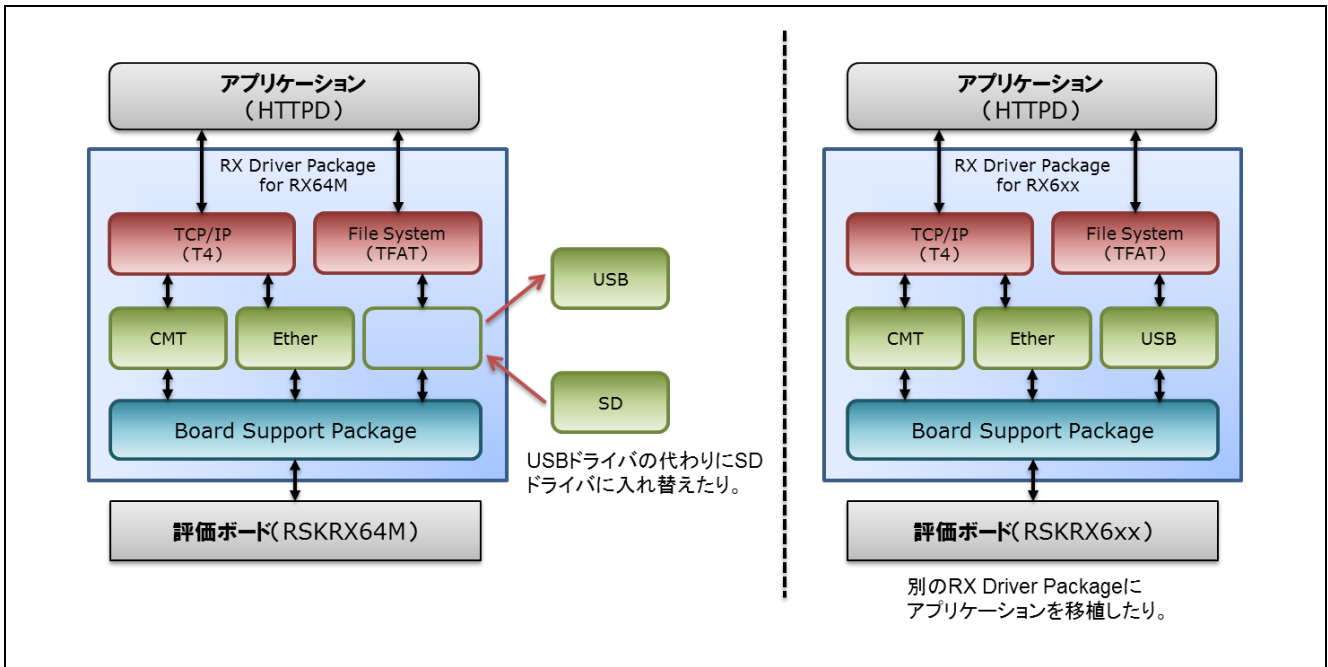


図2.1.4 RX Driver Package の応用例

2.2 RX Driver Package の特徴

RX Driver Package の特徴を以下に示します。

- RX Driver Package は、アプリケーションプログラムの開発に必要なデバイスドライバやミドルウェアを1つのパッケージにしたソフトウェアプラットフォームです。MCU のリリースに合わせて、ルネサスよりタイムリーに提供します。
- RX Driver Package 上で開発したアプリケーションプログラムは、容易に別の MCU の RX Driver Package に移植することができます。
- RX Driver Package に含まれるデバイスドライバやミドルウェアは、全て FIT 仕様に準拠もしくは互換があります。FIT の特徴を生かしたアプリケーションプログラムを開発することができます。
- ミドルウェアやリアルタイム OS の共通インタフェースを提供します。これによって、ソフトウェア資産の流用や、リアルタイム OS 間の移植などが容易になります。
- RX Driver Package と組み合わせてすぐに評価を開始することができる、「RX Driver Package Application」を提供します。
- RX Driver Package は無償で提供されます。パッケージに入っているモジュールは、全て無償で利用することができます。

2.3 RX Driver Package のロードマップ

RX Driver Package は、RX64M を筆頭に、順次展開していく予定です。また、新たなデバイスドライバやミドルウェアも追加される予定です。

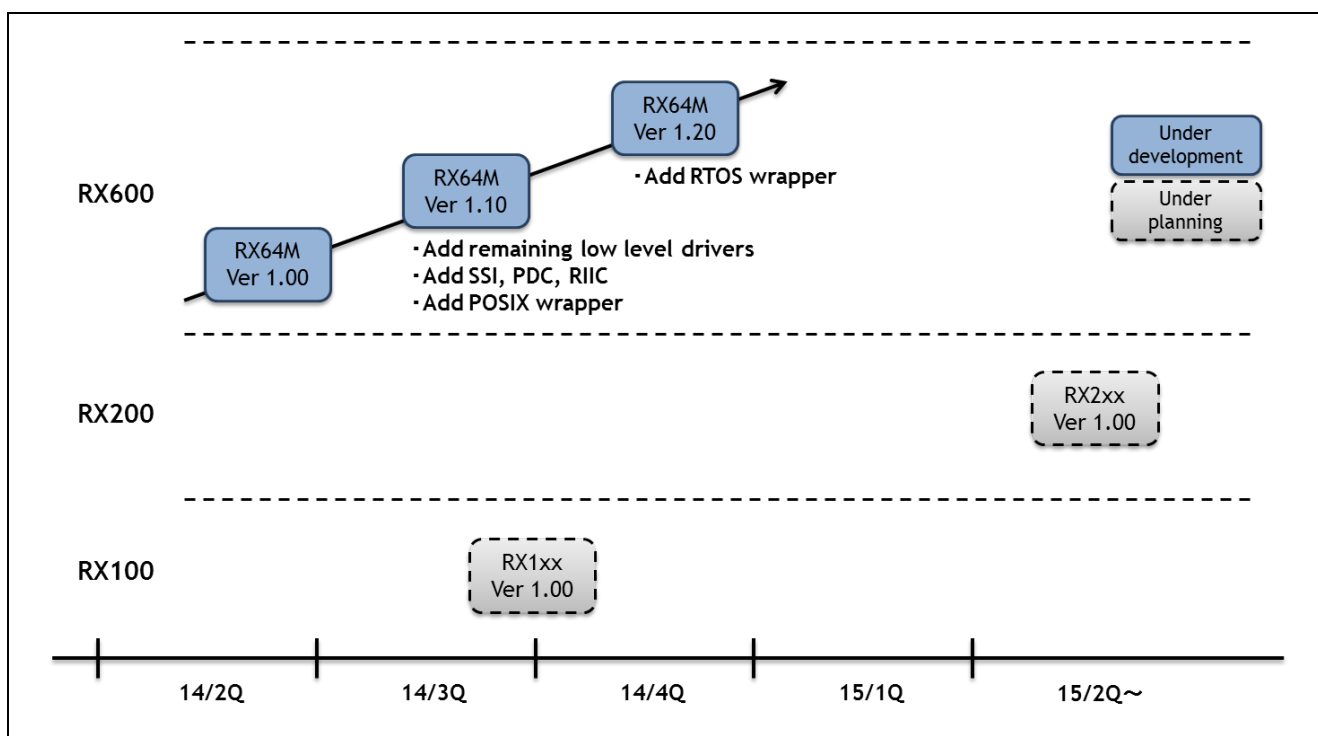


図2.3.1 RX Driver Package のロードマップ

3. RX64M グループ用 RX Driver Package の構成

3.1 フォルダ構成

本パッケージのフォルダ構成を以下に示します。

ルネサスの Web サイトからダウンロードした本パッケージの ZIP ファイルを解凍すると、同名のフォルダがあり、その中に「FITModules」フォルダと「reference_documents」フォルダと本ドキュメントが入っています。

「FITModules」フォルダ内には、表3.3.1 に示すような各モジュールの FIT モジュール (ZIP ファイルと XML ファイル) が複数入っています。

「reference_documents」フォルダには、各開発環境へ適用するためのドキュメントが入っています。

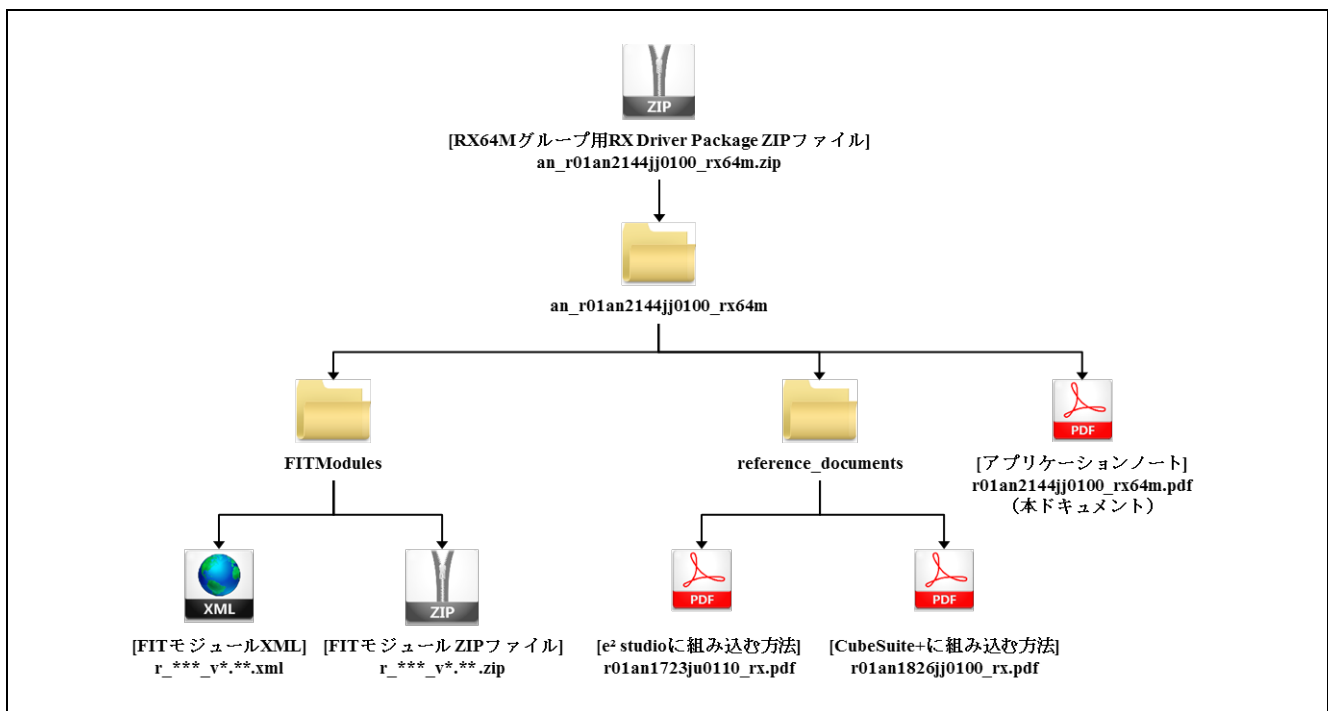


図3.1.1 RX64M グループ用 RX Driver Package フォルダ構成

3.2 モジュール構成

本パッケージに入っている FIT モジュールの種類と構成を以下に示します。

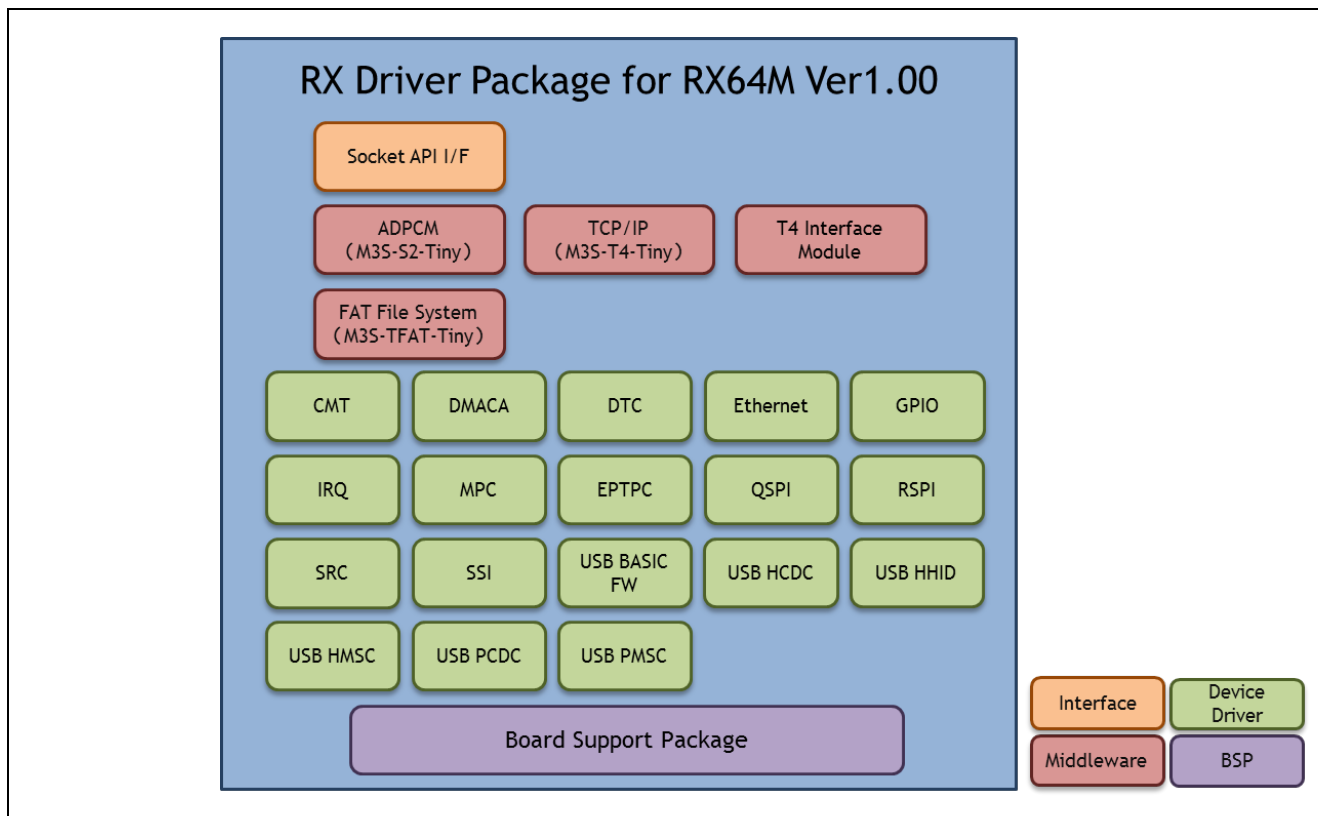


図3.2.1 RX64M グループ用 RX Driver Package FIT モジュール構成

3.3 FITモジュール一覧

本パッケージに入っている FIT モジュール一覧を以下に示します。

表3.3.1 RX64M グループ用 RX Driver Package FIT モジュール一覧

種類	モジュール名	FIT モジュール名	Version
Board Support Package	ボードサポートパッケージ	r_bsp	2.60
Device Driver	コンペアマッチタイマ (CMT)	r_cmt_rx	2.30
Device Driver	DMA コントローラ (DMACA)	r_dmaca_rx	1.01
Device Driver	データトランスファコントローラ(DTC)	r_dtc_rx	2.00
Device Driver	イーサネットコントローラ (ETHERC)	r_ether_rx	1.00
Device Driver	I/O ポート (GPIO)	r_gpio_rx	1.30
Device Driver	割り込みコントローラ (IRQ)	r_irq_rx	1.30
Device Driver	マルチファンクションピンコントローラ (MPC)	r_mpc_rx	1.30
Device Driver	イーサネットコントローラ用 PTP コントローラ(EPTPC)	r_ptp_api_rx	1.01
Device Driver	クワッドシリアルペリフェラルインタフェース (QSPI)	r_qspi_smstr_rx	1.06
Device Driver	シリアルペリフェラルインタフェース (RSPI)	r_rspi_smstr_rx	1.06
Middleware	M3S-S2-Tiny (ADPCM エンコーダ/デコーダライブラリ)	r_s2_rx	3.01
Interface	M3S-T4-Tiny ソケット API モジュール	r_socket	1.10
Device Driver	サンプリングレートコンバータ (SRC)	r_src_api_rx	1.00
Device Driver	シリアルサウンドインタフェース (SSI)	r_ssi_api_rx	1.00
Middleware	M3S-T4-Tiny インタフェース変換モジュール	r_t4_driver_rx64m	1.00
Middleware	M3S-T4-Tiny (TCP/IP プロトコルスタックライブラリ)	r_t4_rx	2.00
Middleware	M3S-TFAT-Tiny (FAT ファイルシステム)	r_tfat_rx	3.00
Device Driver	USB Basic Firmware	r_usb_basic	1.00
Device Driver	USB ホストコミュニケーションデバイスクラス	r_usb_hcdc	1.00
Device Driver	USB ホストヒューマンインタフェースデバイスクラス	r_usb_hhid	1.00
Device Driver	USB ホストマスストレージクラス	r_usb_hmsc	1.00
Device Driver	USB ペリフェラルコミュニケーションデバイスクラス	r_usb_pcdc	1.00
Device Driver	USB ペリフェラルマスストレージクラス	r_usb_pmssc	1.00

【注】本パッケージには、評価版の「M3S-T4-Tiny (TCP/IP プロトコルスタックライブラリ)」が含まれています。製品版については、以下の URL を参照してください。

<http://japan.renesas.com/mw/t4>

4. 使用方法

RX Driver Package は、e² studio に入っている FIT プラグインを使用することで、簡単にプログラムを構築することができます。以下に、e² studio を使用した簡単な使用例を示します。なお、CubeSuite+を使用する場合には、本パッケージに付属している「RX ファミリ CubeSuite+に組み込む方法 Firmware Integration Technology (R01AN1826JJ)」を参照してください。

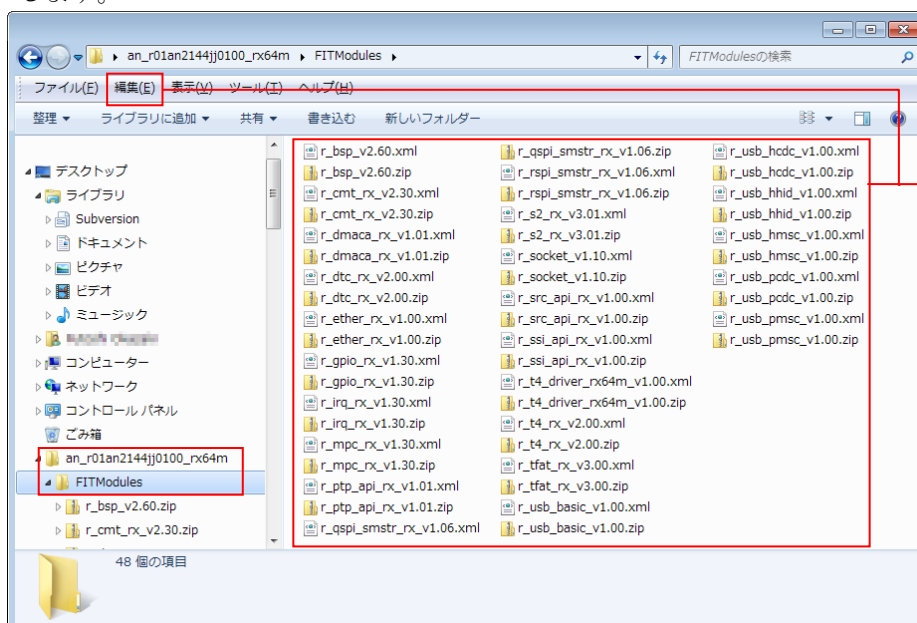
4.1 説明で使用する環境

ターゲット MCU に「RX64M」、ターゲットボードに「Renesas Starter Kit+ RX64M」を使用します。それ以外の環境で行う場合は、使用する環境に合わせて説明を読み替えてください。

4.2 e² studio に RX Driver Package をインストールする

RX Driver Package に入っている FIT モジュールを、e² studio にインストールします。

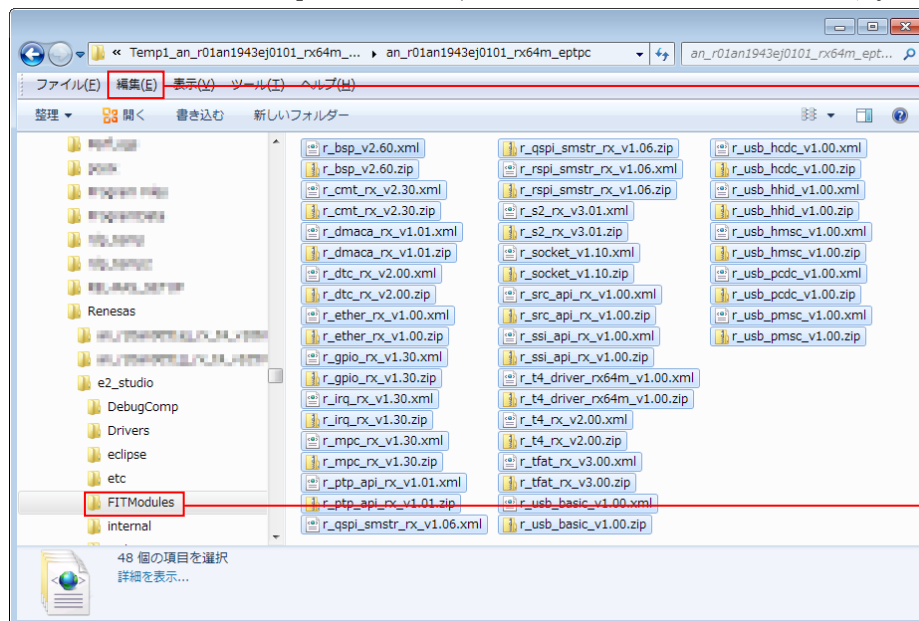
1. ダウンロードした「an_r01an2144jj0100_rx64m.zip」ファイルを、任意のフォルダに解凍します。
2. 解凍してできたフォルダを開き、その中にある「FITModules」フォルダを開きます。
3. 「FITModules」フォルダ内にある全てのファイルを選択し、「編集」メニューから「コピー」をクリックします。



ファイルを全て選択し、「編集」メニューから「コピー」をクリックします

- e² studio のインストールフォルダ (通常は、C:\¥Renesas¥e2_studio です) を開き、その中にある「FITModules」フォルダを開きます。
- 「編集」メニューから「貼り付け」をクリックします。

e² studio の「FITModules」フォルダに、FIT モジュールがコピーされます。



「FITModules」フォルダを開き、「編集」メニューから「貼り付け」をクリックし、このフォルダへコピーします

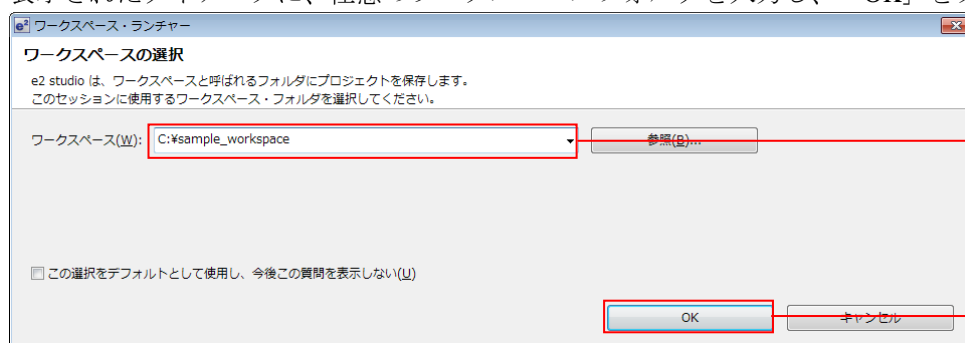
4.3 アプリケーションの作成

LED を光らせる、簡単なプログラムを作成してみます。

4.3.1 ワークスペースとプロジェクトを作成する

まず、ワークスペースとプロジェクトを新規に作成します。

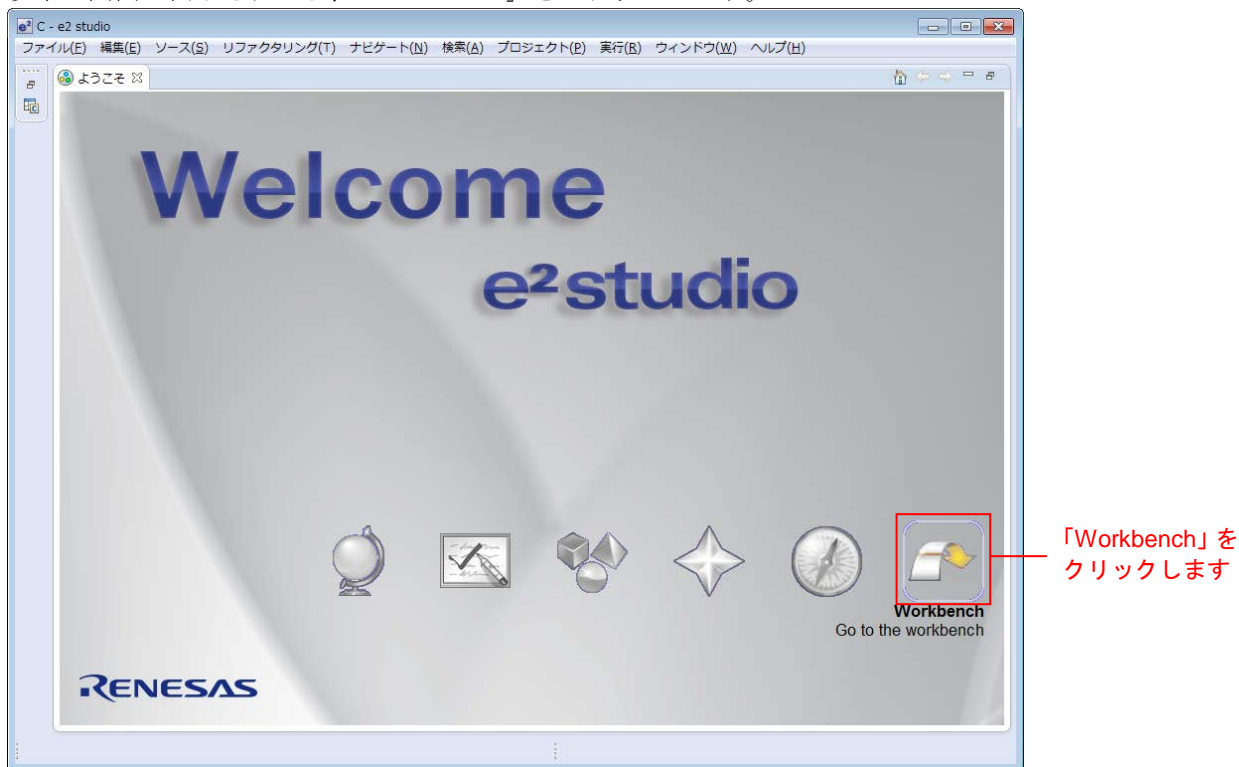
- e² studio を起動します。
- 表示されたダイアログに、任意のワークスペースフォルダを入力し、「OK」をクリックします。



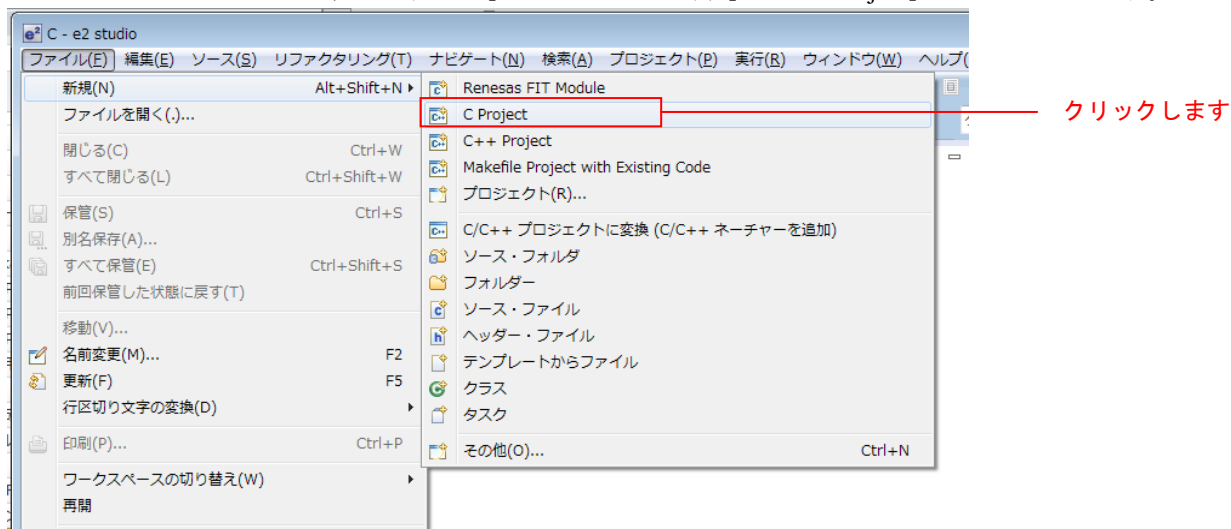
ワークスペースフォルダを入力します

「OK」をクリックします

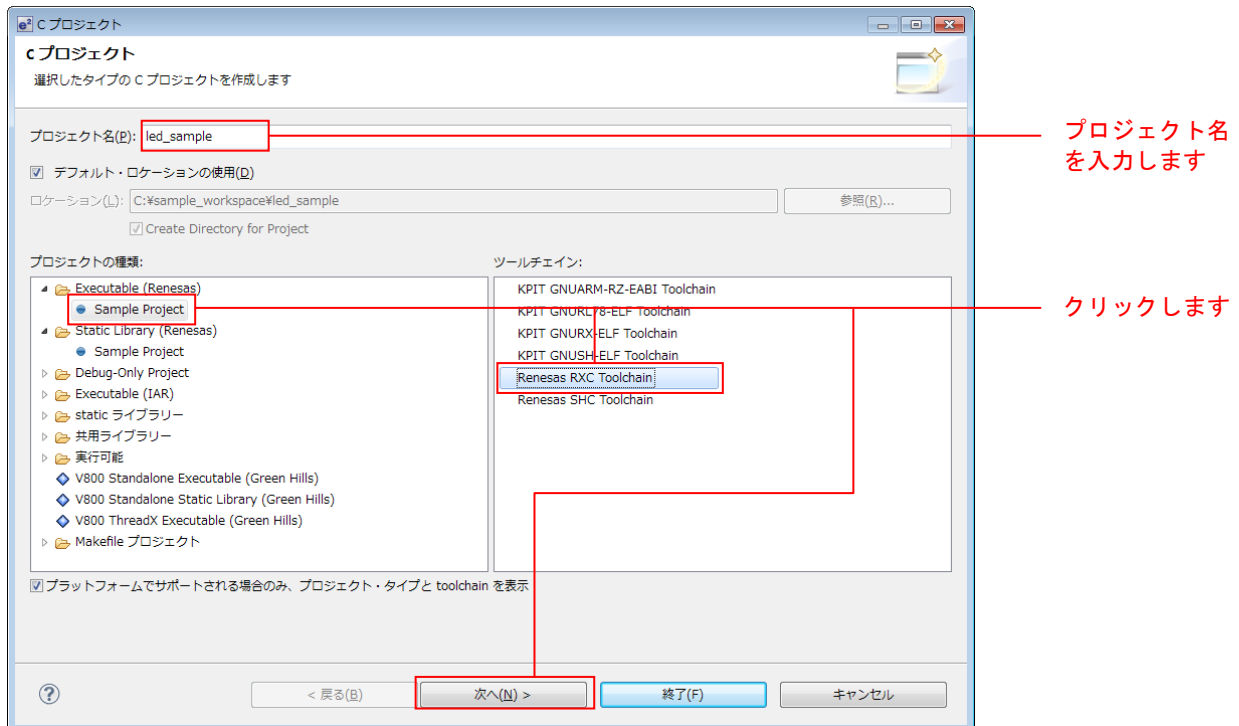
3. 以下の画面が表示されたら、「Workbench」をクリックします。



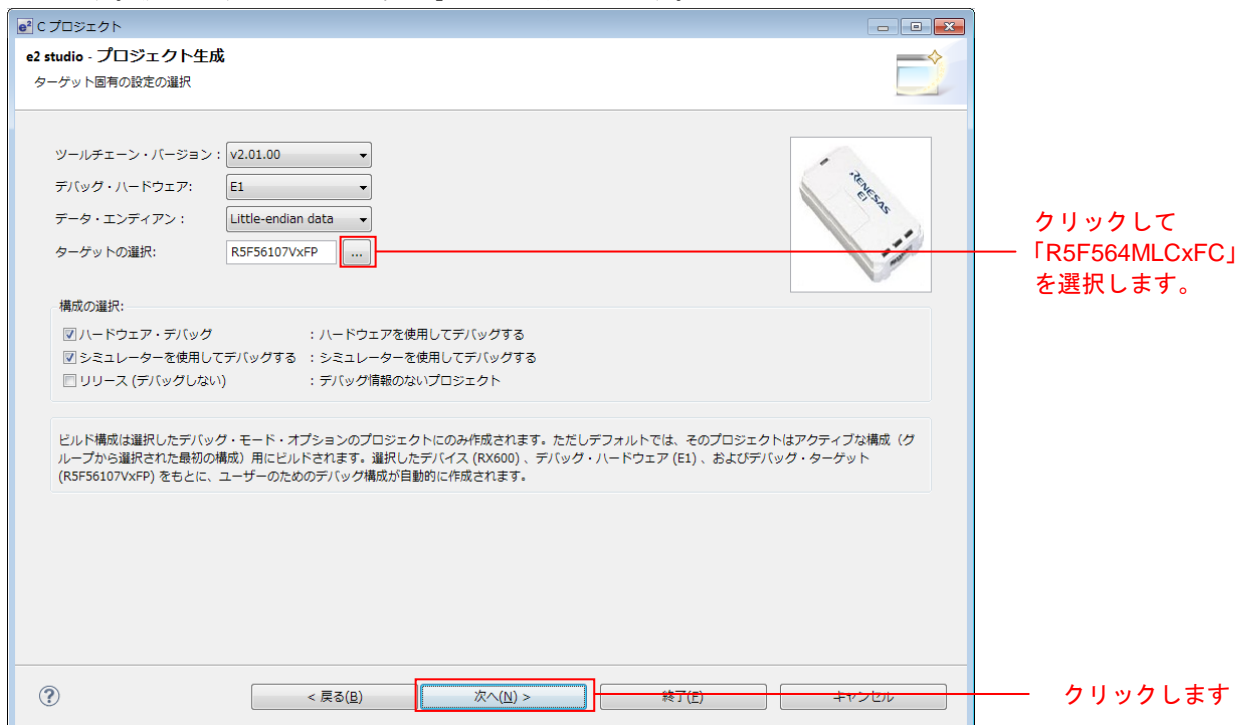
4. ワークベンチが起動したら、「ファイル」メニューの「新規」の「C Project」をクリックします。



5. プロジェクト名を入力します。「プロジェクトの種類」は、「Executable (Renesas)」の「Sample Project」をクリックします。「ツールチェーン」は、「Renesas RXC Toolchain」をクリックします。設定が終わったら「次へ」をクリックします。



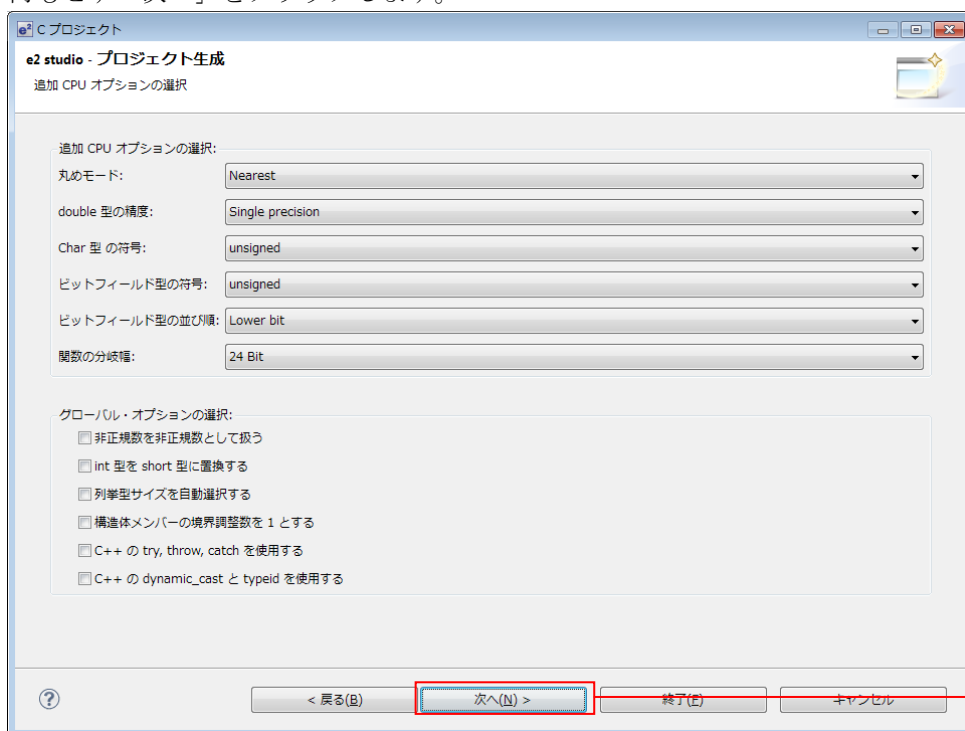
6. ターゲットを選択します。「ターゲットの選択」の「…」ボタンをクリックし、「R5F564MLCxFC」を選択します。設定が終わったら「次へ」をクリックします。



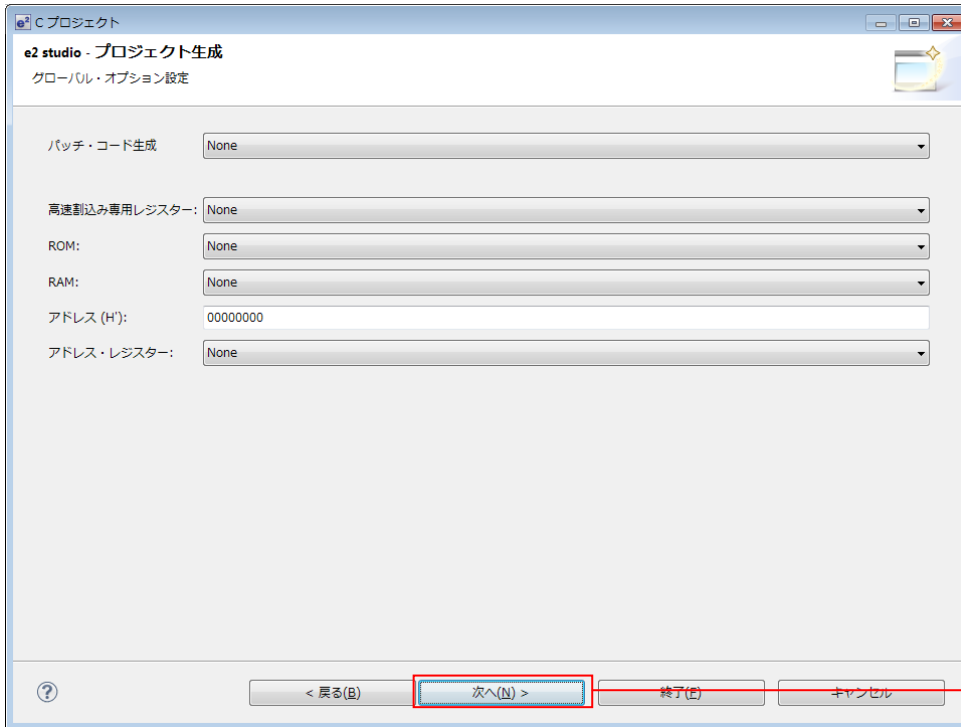
7. 何もせず「次へ」をクリックします。



8. 何もせず「次へ」をクリックします。

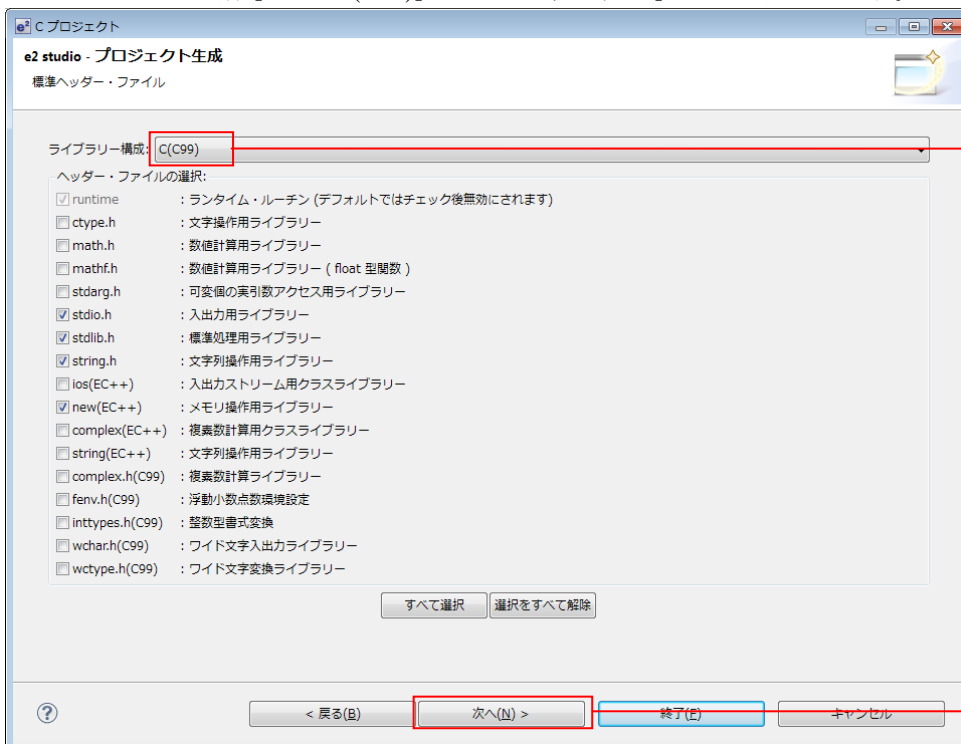


9. 何もせず「次へ」をクリックします。



クリックします

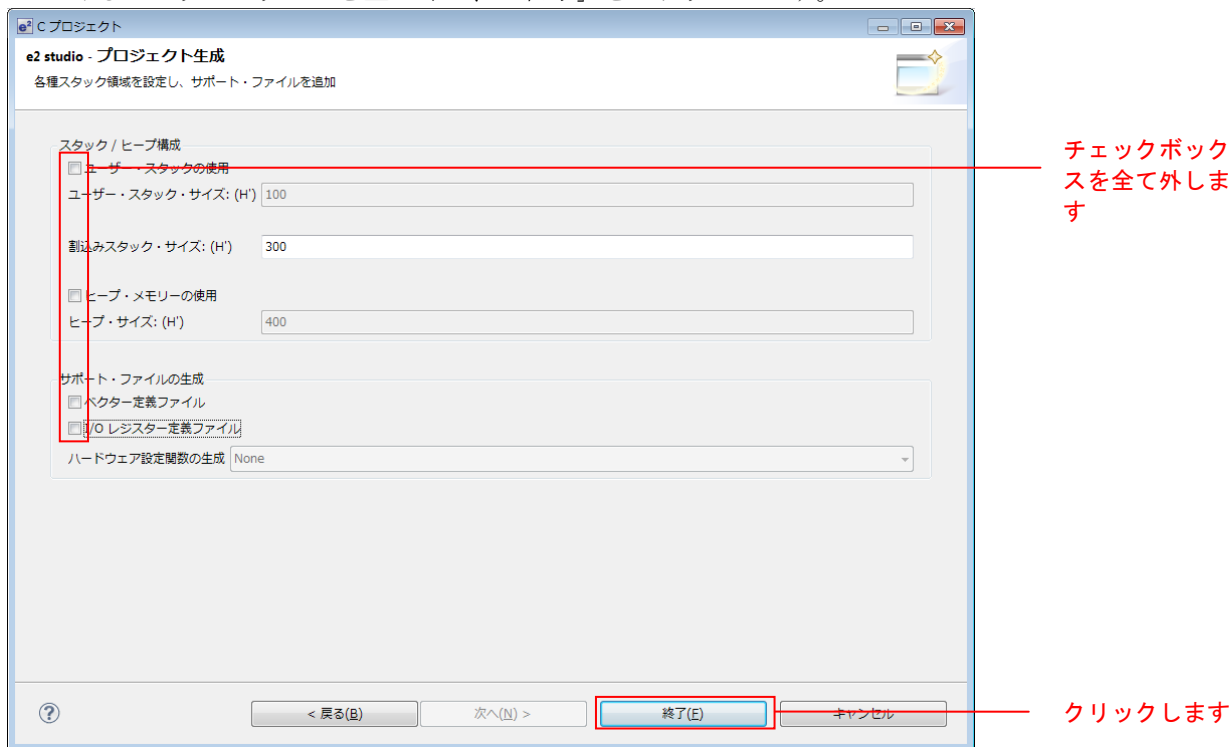
10. 「ライブラリー構成」で「C(C99)」を選択し、「次へ」をクリックします。



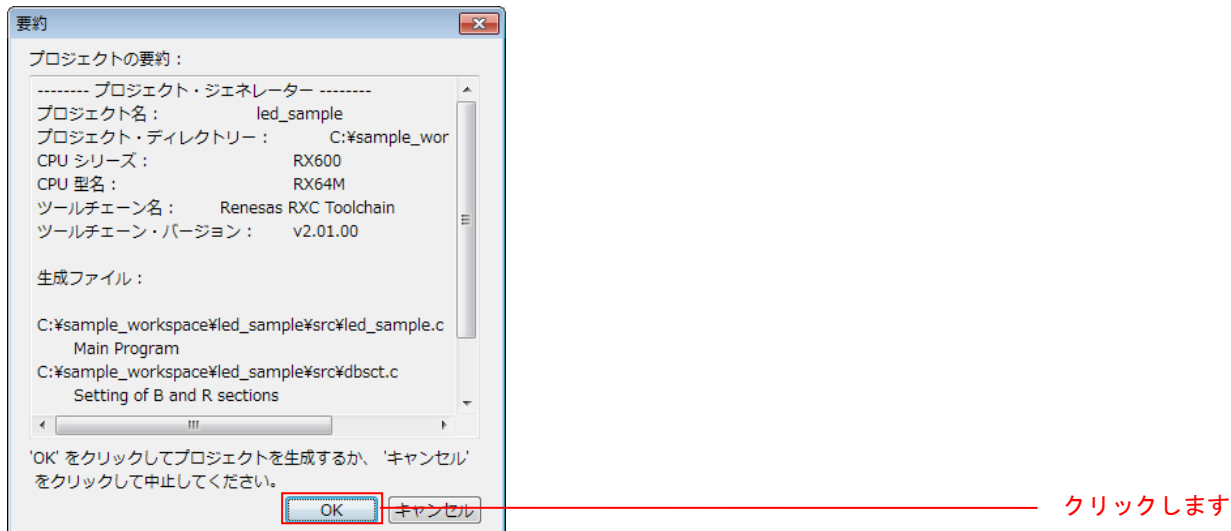
「C(C99)」を選択
します

クリックします

11. 4つあるチェックボックスを全て外し、「終了」をクリックします。



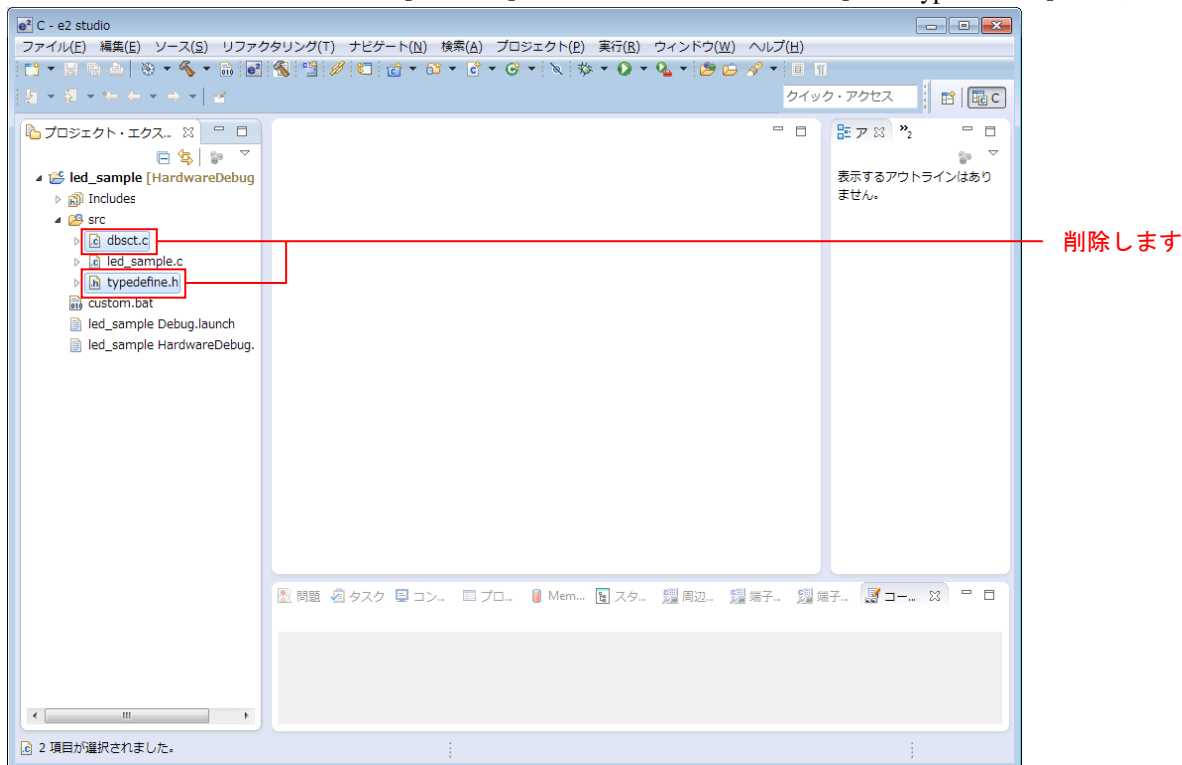
12. 「OK」をクリックすると、プロジェクトが生成されます。



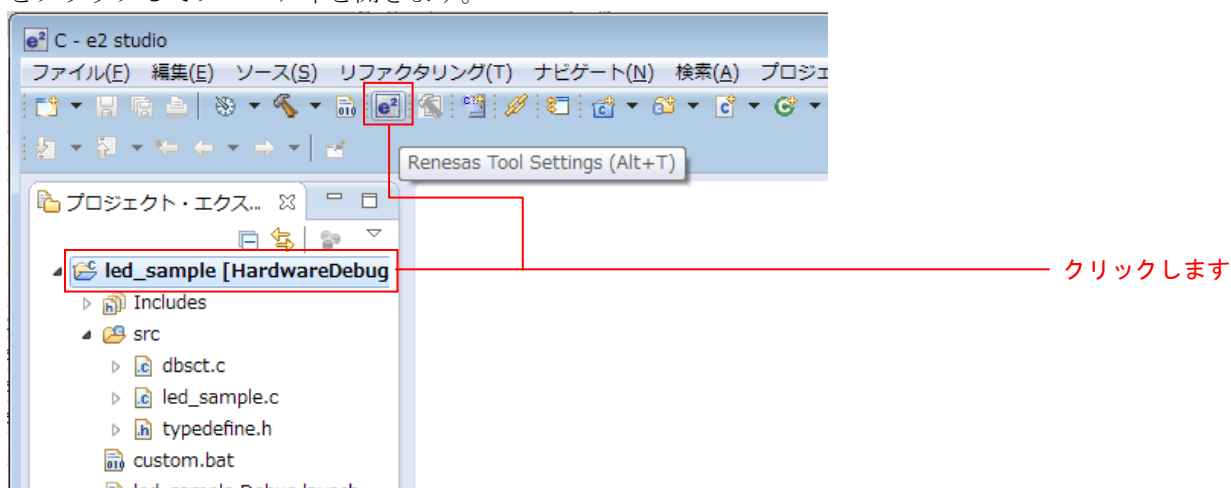
4.3.2 FIT モジュールの組み込み準備

FIT モジュールは BSP にて独自の初期設定を行っているため、e² studio が生成したプロジェクトの一部を変更する必要があります。

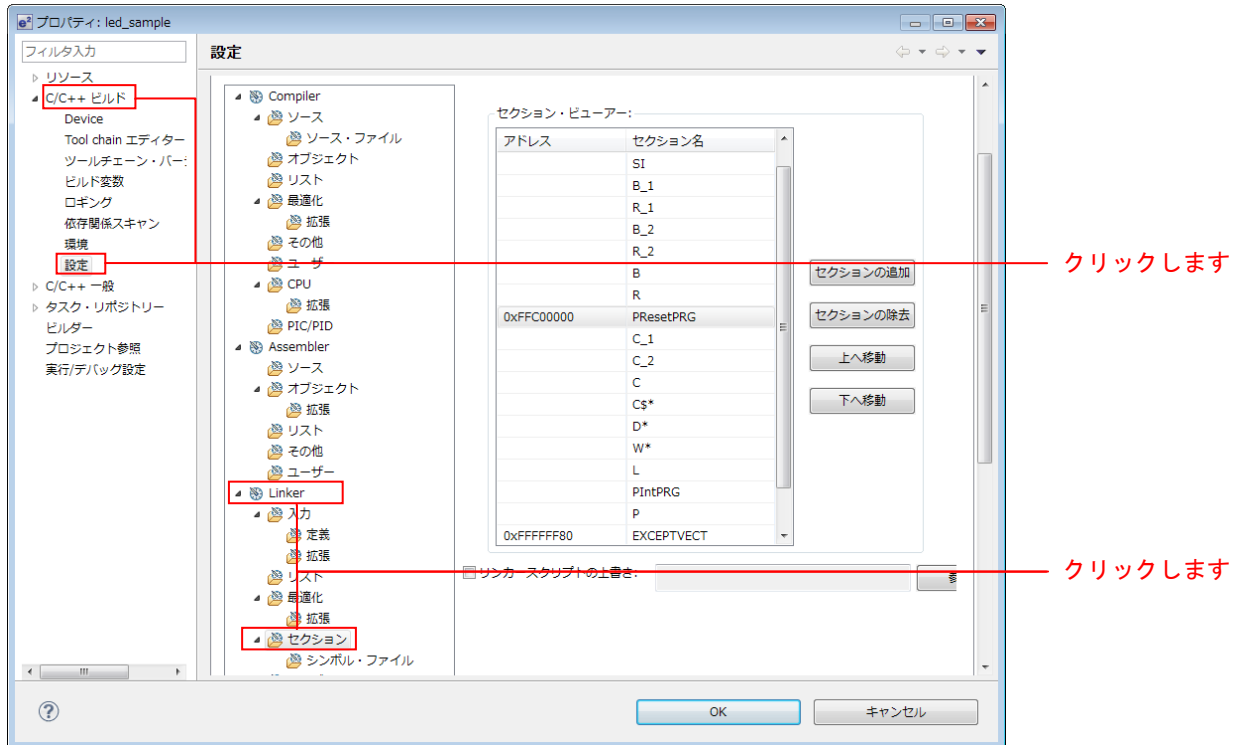
1. 「プロジェクト・エクスプローラ」で「src」フォルダ内にある「dbsct.c」と「typedefine.h」を削除します。



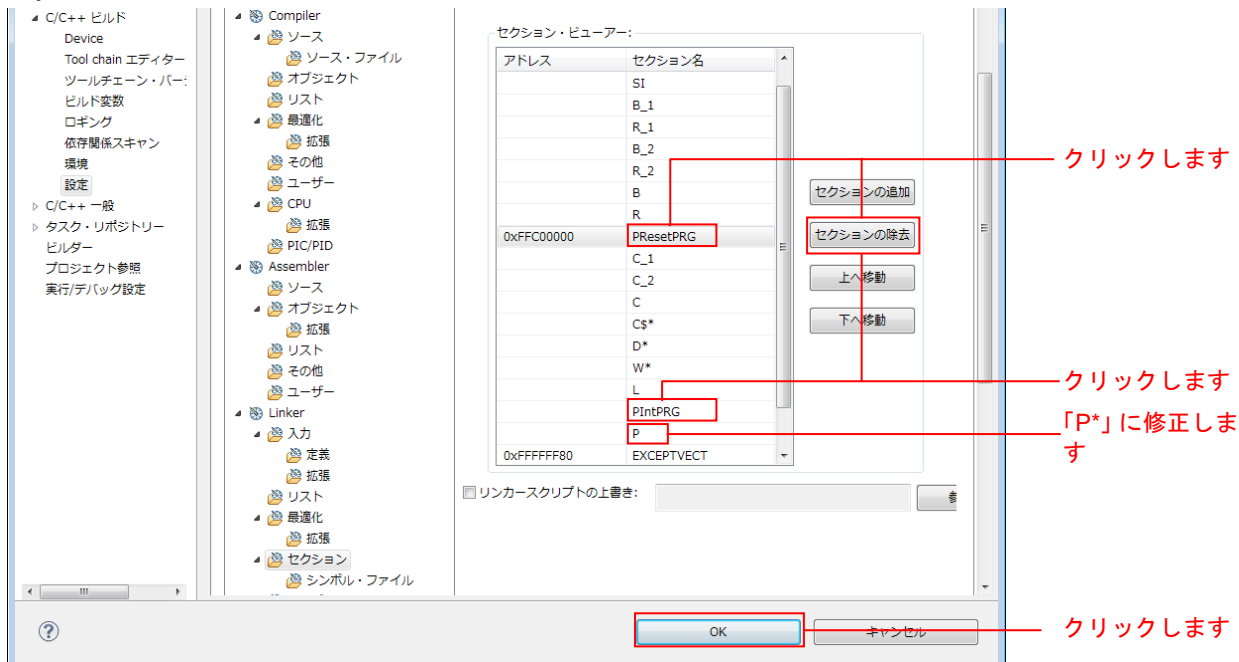
2. 「プロジェクト・エクスプローラ」でプロジェクトをクリックし、ツールバーの「Renesas Tool Setting」をクリックしてプロパティを開きます。



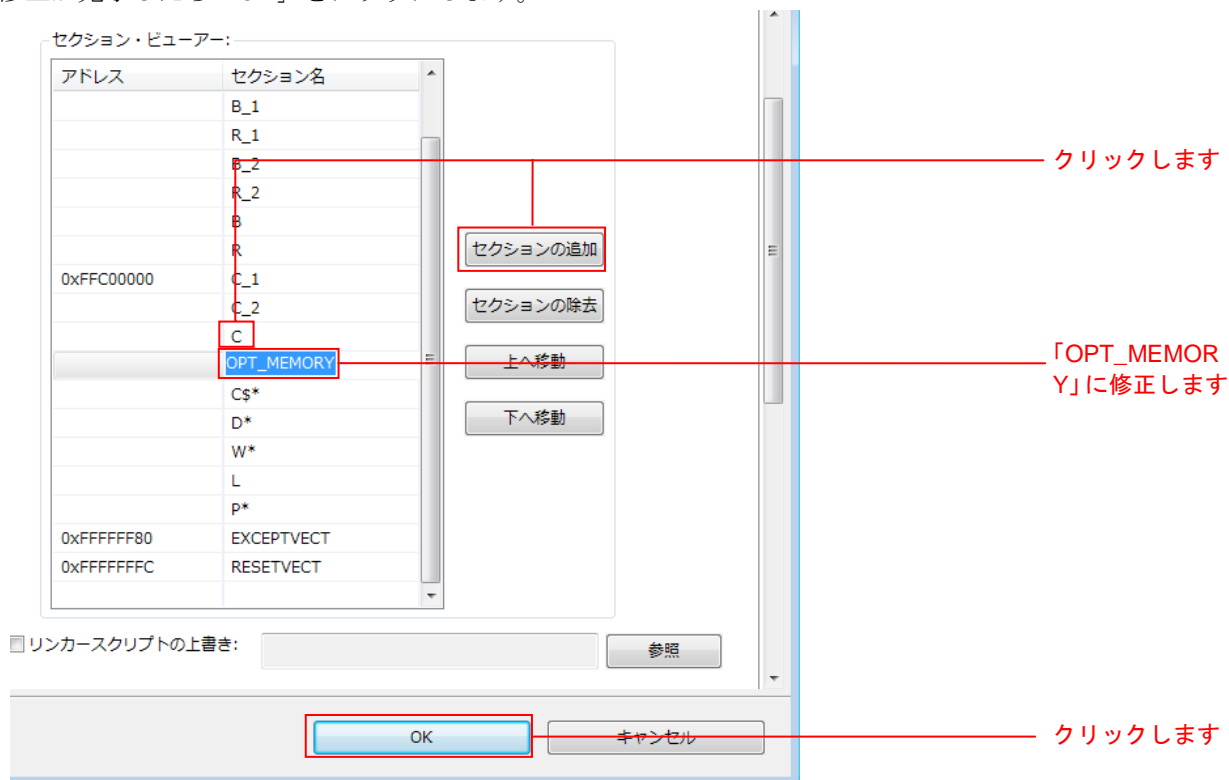
3. プロパティの「C/C++ ビルド」の「設定」をクリックし、「Linker」の「セクション」をクリックします。



4. 「セクション・ビューアー」から「PResetPRG」セクションと「PIntPRG」セクションを削除します。各セクションをクリックし、「セクションの除去」をクリックします。また、「P」セクションを「P*」に変更します。「P」セクションをクリックし、「P*」に修正してください。



5. 「C」セクションの後に「OPT_MEMORY」セクションを追加します。「C」セクションをクリックし、「セクションの追加」をクリックします。追加された「NEW_SECTION_1」セクションをクリックし、「OPT_MEMORY」に修正してください。
修正が完了したら「OK」をクリックします。

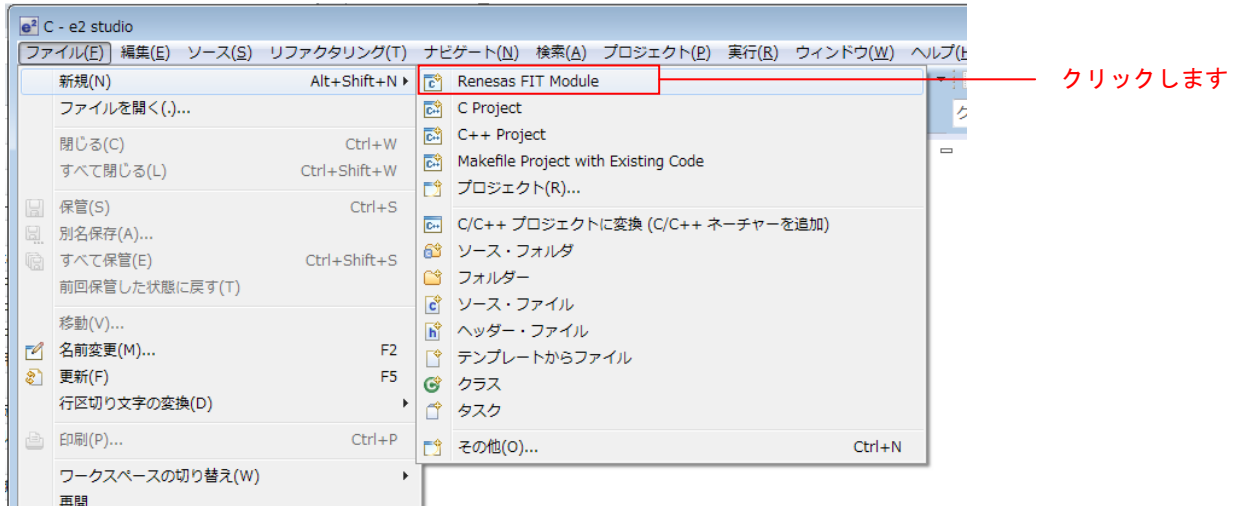


4.3.3 FIT プラグインで FIT モジュールをインストールする

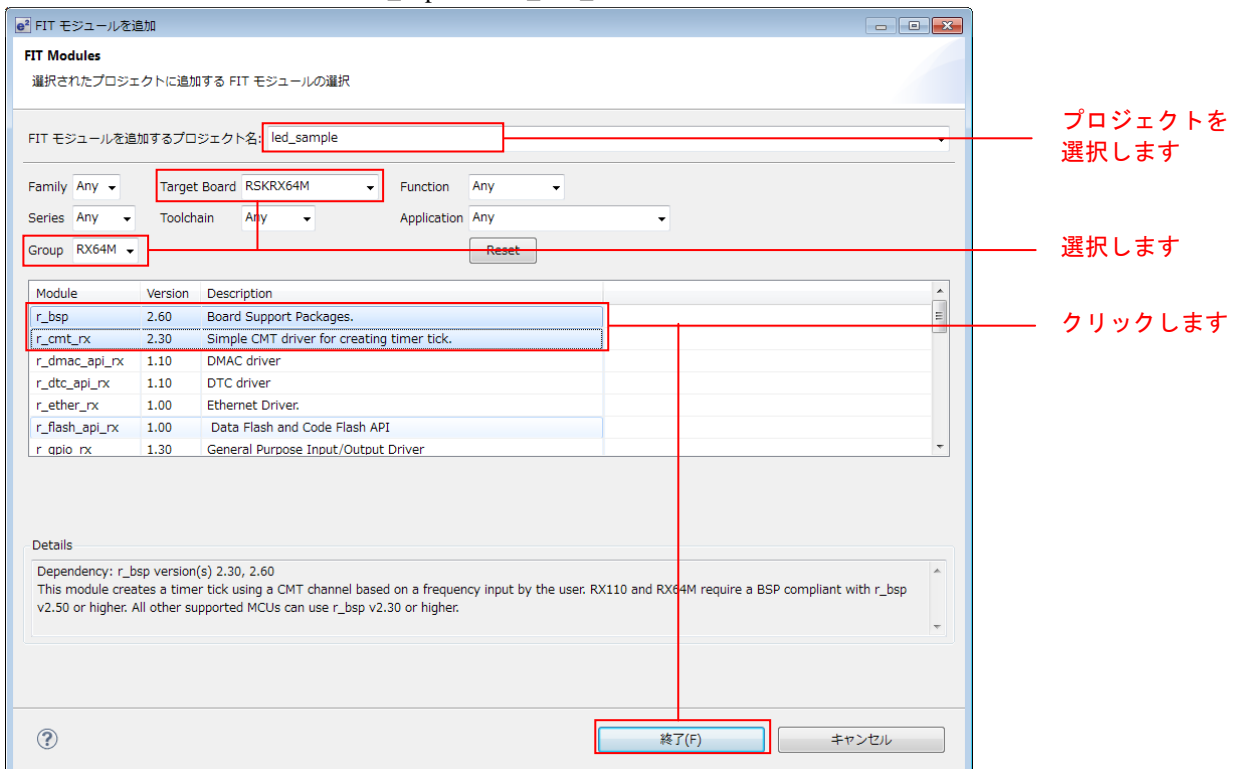
作成したプロジェクトに、FIT プラグインを使って必要なモジュールをインストールします。

ここでは、BSP モジュール (r_bsp) とコンペアマッチタイマドライバ (r_cmt_rx) をインストールします。

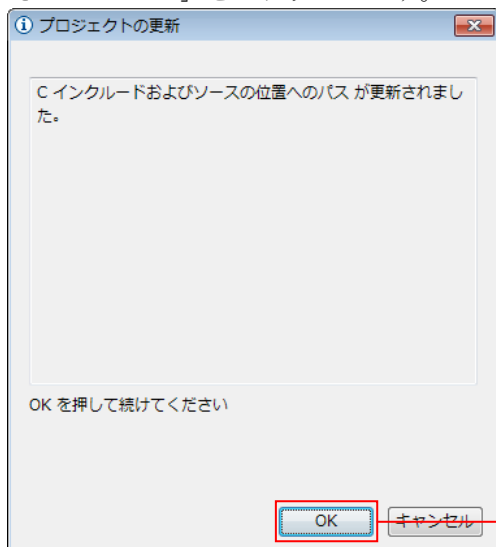
1. 「ファイル」メニューの「新規」の「Renesas FIT Module」をクリックします。



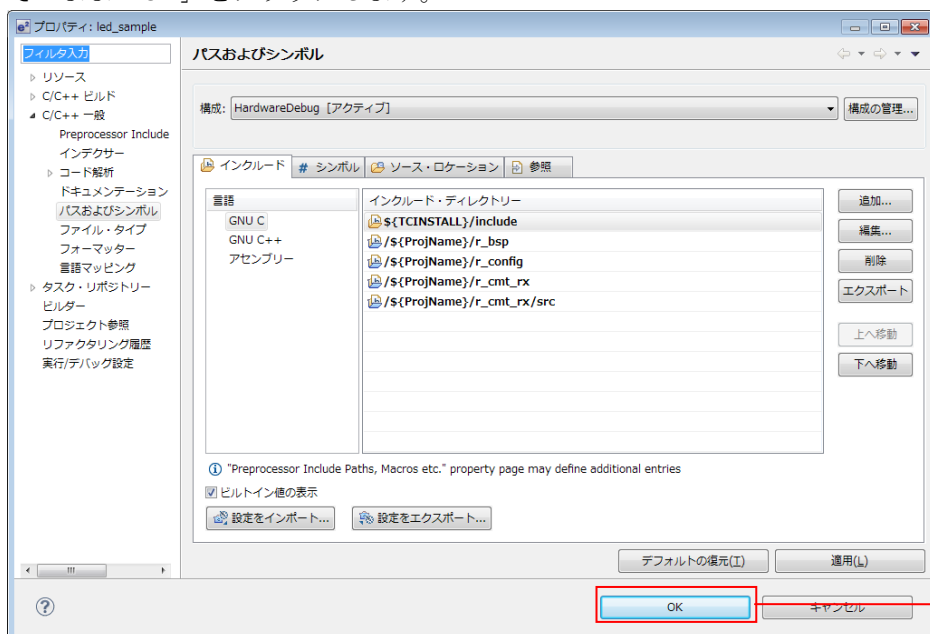
2. 「FIT モジュールを追加するプロジェクト名」で作成したプロジェクトを選択します。
次に、「Group」から「RX64M」を選択し、「Target Board」から「RSKR64M」を選択します。
次に、モジュールリストから「r_bsp」と「r_cmt_rx」をクリックし、「終了」をクリックします。



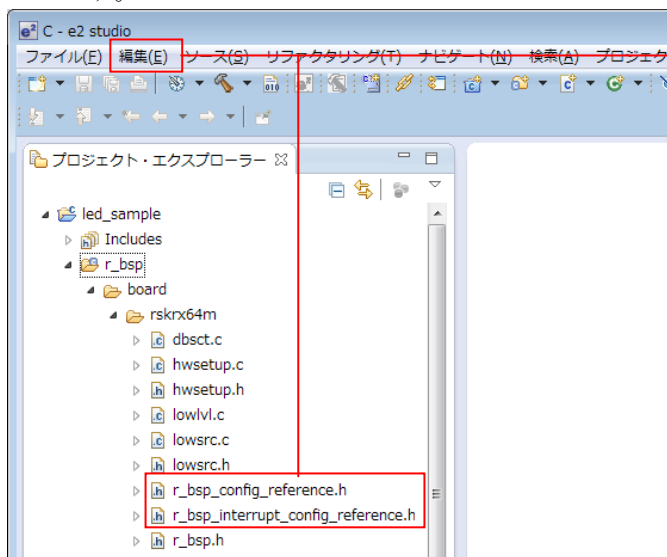
3. そのまま「OK」をクリックします。



4. そのまま「OK」をクリックします。

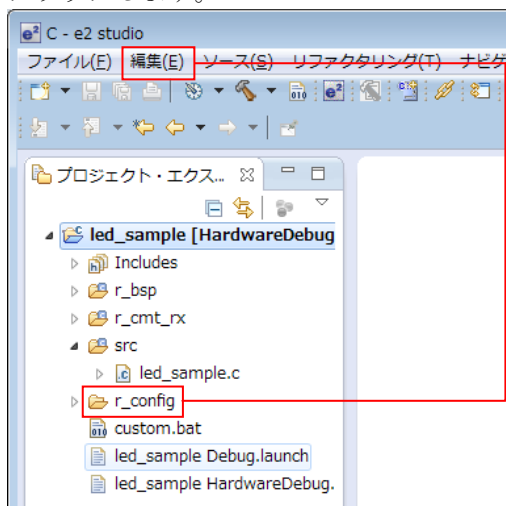


5. e2 studio のプロジェクト・エクスプローラから「r_bsp/board/rskrx64m」を開き、「r_bsp_config_reference.h」と「r_bsp_interrupt_config_reference.h」の2つのファイルを選択し、「編集」メニューの「コピー」をクリックします。



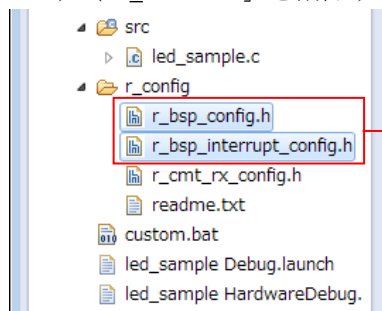
2つのファイルを選択し、「編集」メニューの「コピー」をクリックします

6. e2 studio のプロジェクト・エクスプローラから「r_config」を選択し、「編集」メニューの「貼り付け」をクリックします。



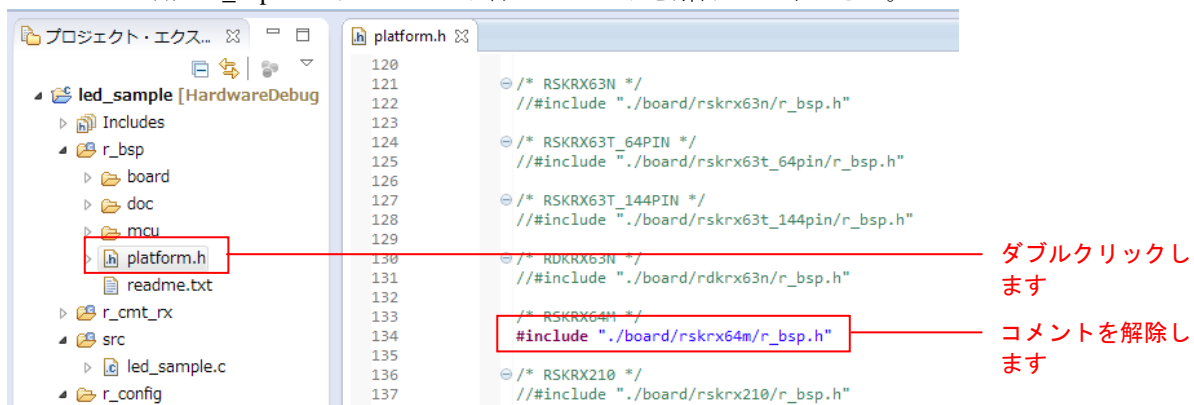
「r_config」フォルダを選択し、「編集」メニューの「貼り付け」をクリックします

7. コピーしたファイルのファイル名を、それぞれ「r_bsp_config.h」と「r_bsp_interrupt_config.h」にリネームします（「_reference」を削除する）。



リネームします

8. 使用するターゲットボードに合わせて `platform.h` を修正します。
e² studio のプロジェクト・エクスプローラから「`r_bsp/platform.h`」をダブルクリックし、エディタ上で RSKRX64M 用の `r_bsp.h` のインクルード行のコメントを解除してください。



4.3.4 LED 点灯プログラムを作成する

コンペアマッチタイマを使用し、0.5 秒間隔で LED0 を点滅させるプログラムを作成します。

src/led_sample.c を開き、以下のように修正します。

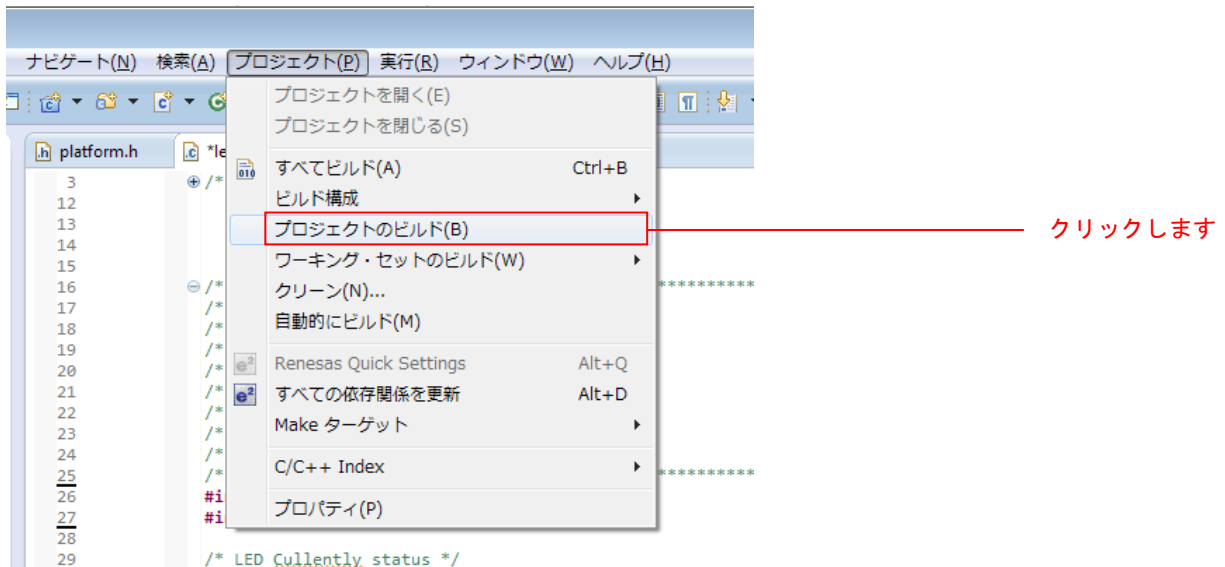
【src/led_sample.c】

```
/******  
/*  
/* FILE      :Main.c or Main.cpp  
/* DATE      :Tue, Oct 31, 2006  
/* DESCRIPTION :Main Program  
/* CPU TYPE   :  
/*  
/* NOTE:THIS IS A TYPICAL EXAMPLE.  
/*  
/******  
#include "platform.h"  
#include "r_cmt_rx_if.h"  
  
/* LED Currently status */  
uint32_t ledstatus = LED_OFF;  
  
void call_back(void *pdata)  
{  
    if (ledstatus == LED_OFF)  
    {  
        /* Turn ON the LED0 If the status is LED_OFF */  
        LED0 = LED_ON;  
        ledstatus = LED_ON;  
    }  
    else  
    {  
        /* Turn OFF the LED0 If the status is LED_ON */  
        LED0 = LED_OFF;  
        ledstatus = LED_OFF;  
    }  
}  
  
void main(void)  
{  
    uint32_t cmt_ch;  
  
    /* LED0 off */  
    LED0 =LED_OFF;  
    /* Create of 0.5 second(2Hz) cyclic timer. */  
    R_CMT_CreatePeriodic(2, &call_back, &cmt_ch);  
  
    while(1);  
}
```

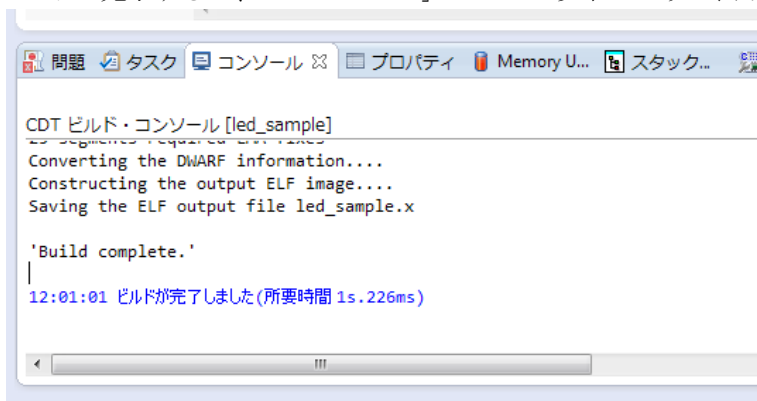

4.3.5 プログラムをビルドし動作を確認する

作成したプログラムをビルドして、動作を確認します。

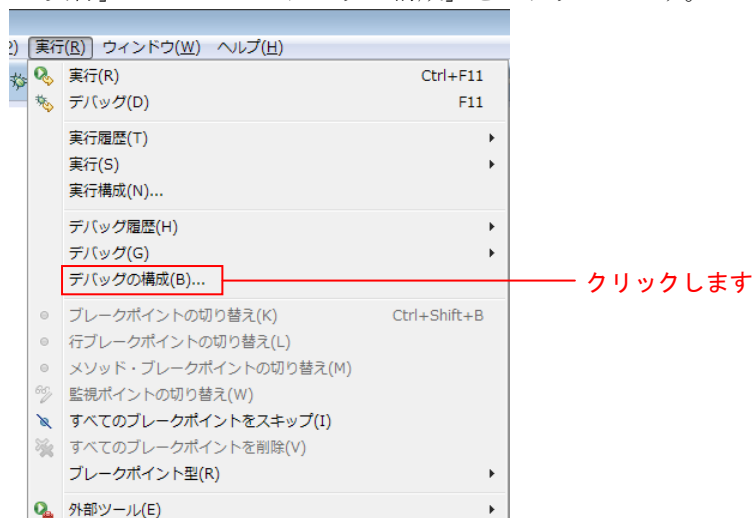
1. 「プロジェクト」メニューの「プロジェクトをビルド」をクリックします。



2. ビルドが完了すると、「コンソール」ビューに以下のように表示されます。



3. 「実行」メニューの「デバッグ構成」をクリックします。



4. 「Renesas GDB Hardware Debugging」の「led_sample HardwareDebug」をクリックします。「Debugger」タブをクリックし、「Connection Setting」をクリックします。
「EXTAL 周波数」を「24.0000」に修正し、「エミュレータから電源を供給する」を「いいえ」に変更します。
完了したら「デバッグ」をクリックします。

このスクリーンショットは、「デバッグ構成」ダイアログボックスの「Debugger」タブの「Connection Settings」サブタブを示しています。以下の設定が確認できます：

- 名前(N): led_sample HardwareDebug
- Debug hardware: E1
- Target Device: R5F564ML
- EXTAL 周波数[MHz]: 24.0000
- エミュレータから電源を供給する (MAX 200MHz): いいえ
- デバッグ(D) ボタンがクリックされています。

5. 以下のメッセージが表示されたら、「はい」をクリックします。

このダイアログボックスは、「パースペクティブ切り替えの確認」を示しています。メッセージは以下の通りです：

この種類の起動は、中断時に デバッグ パースペクティブが開くように構成されています。

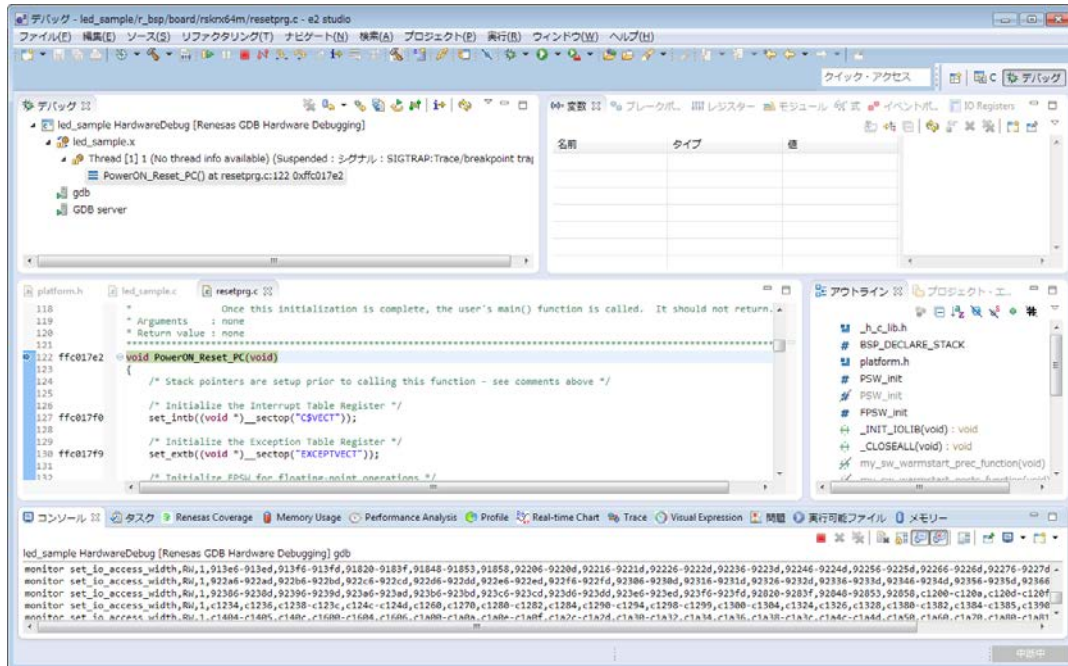
このデバッグ・パースペクティブは、アプリケーション・デバッグをサポートするために設計されています。これには、デバッグ・スタック、変数、およびブレークポイント管理を表示するビューが組み込まれています。

このパースペクティブを開きますか？

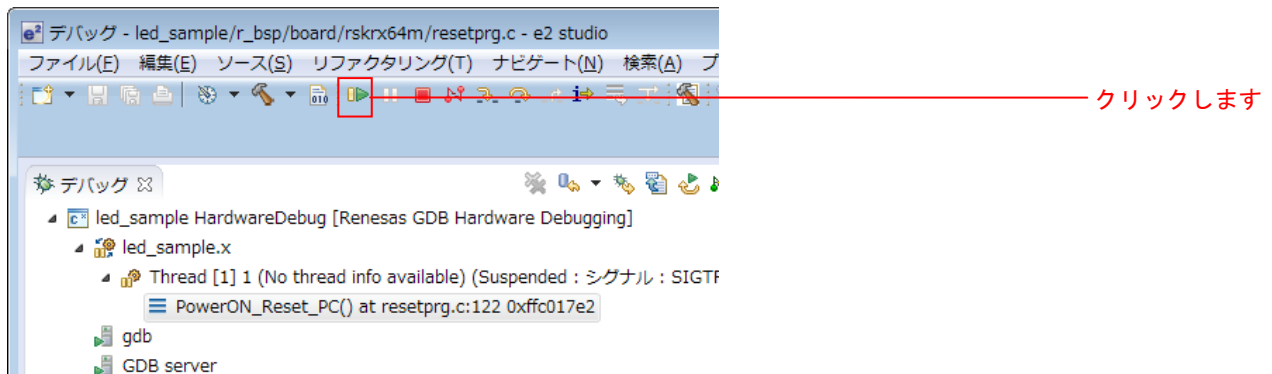
常にこの設定を使用する(R)

はい(Y) / いいえ(N) ボタンがクリックされています。

6. ロードモジュールのダウンロードが完了すると、「デバッグ」パースペクティブが開きます。



7. ツールバーの「再開」をクリックします。プログラムが実行され、main 関数の先頭でブレークします。



8. main 関数の先頭でブレークした後に、もう一度ツールバーの「再開」をクリックします。プログラムが実行され、LED0 が 0.5 秒間隔で点灯と消灯を繰り返します。

5. RX Driver Package Application について

5.1 RX Driver Package Application の構成

RX Driver Package Application は、RX Driver Package を簡単に使って頂くためのサンプルアプリケーションプログラムです。RX Driver Package Application には、RX Driver Package に入っているデバイスドライバやミドルウェアを使って動作するアプリケーションプログラムと、そのアプリケーションをビルドするためのプロジェクトファイルが入っているため、すぐに評価を開始することができます。

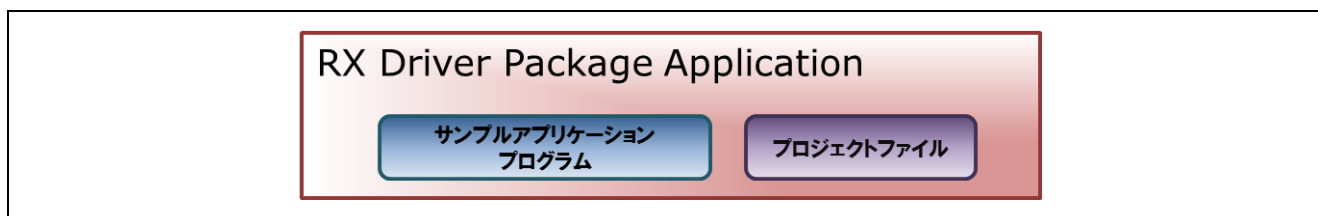


図5.1.1 RX Driver Package Application の構成

RX Driver Package Application には、複数のドライバやミドルウェアを組み合わせで動作するシステムプログラムや、RX Driver Package に入っているモジュール単体の評価プログラムなど、さまざまな種類を順次公開していく予定です。

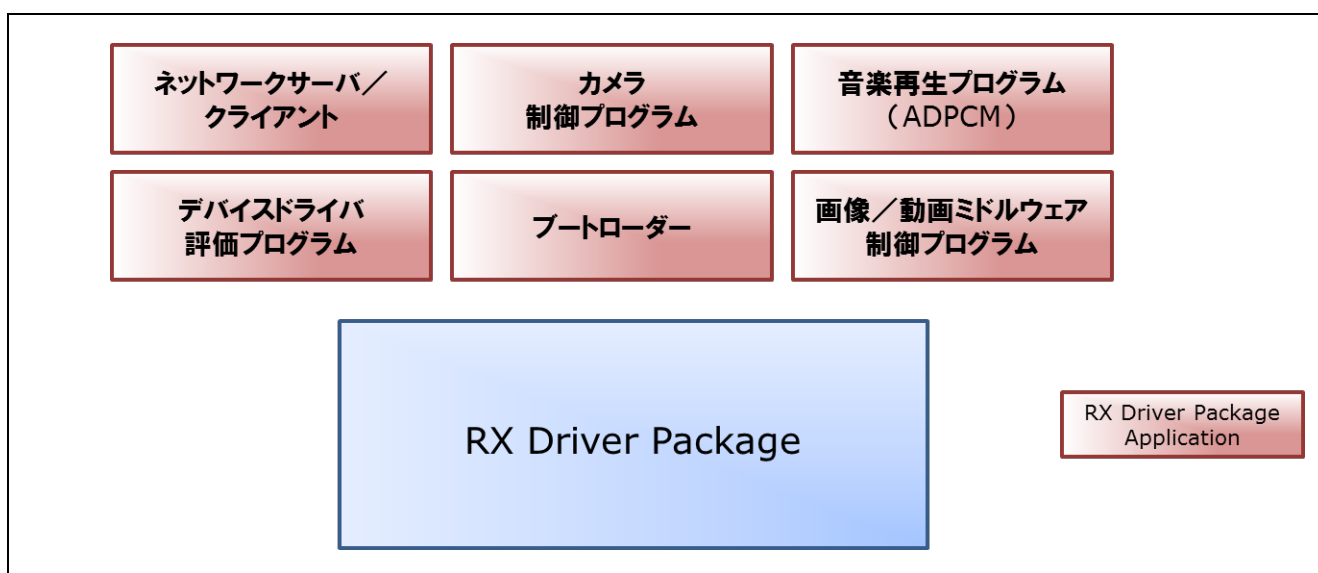


図5.1.2 RX Driver Package Application の種類

5.2 RX Driver Package Application の特徴

RX Driver Package Application の特徴を以下に示します。

- RX Driver Package Application は、RX Driver Package と組み合わせて評価されています。
- RX Driver Package Application には、プロジェクトファイルが付属しています。付属のプロジェクトファイルには、アプリケーションプログラムのビルド構成とデバッグ構成が設定済みなので、プロジェクトをワークスペースにインポートすれば、すぐにビルドと評価を開始することができます。
- 開発環境 (IDE) に e² studio を使用する場合は、e² studio に付属している FIT プラグインを使うことで、アプリケーションプログラムで使用するデバイスドライバやミドルウェアを自動的にプロジェクトに追加することができます。
- RX Driver Package Application は無償で提供されます。

5.3 RX Driver Package Application の使用例 (e² studio を使用する場合)

RX Driver Package Application で必要なデバイスドライバやミドルウェアは、e² studio に付属している FIT プラグインが自動的にプロジェクトに追加します。

RX Driver Package Application に付属のプロジェクトを、e² studio のワークスペースにインポートした後、FIT プラグインで RX Driver Package Application を選択してプロジェクトにインストールするだけで、RX Driver Package の中から必要なデバイスドライバやミドルウェアもインストールされるので、あとはビルドするだけで評価を開始できます。

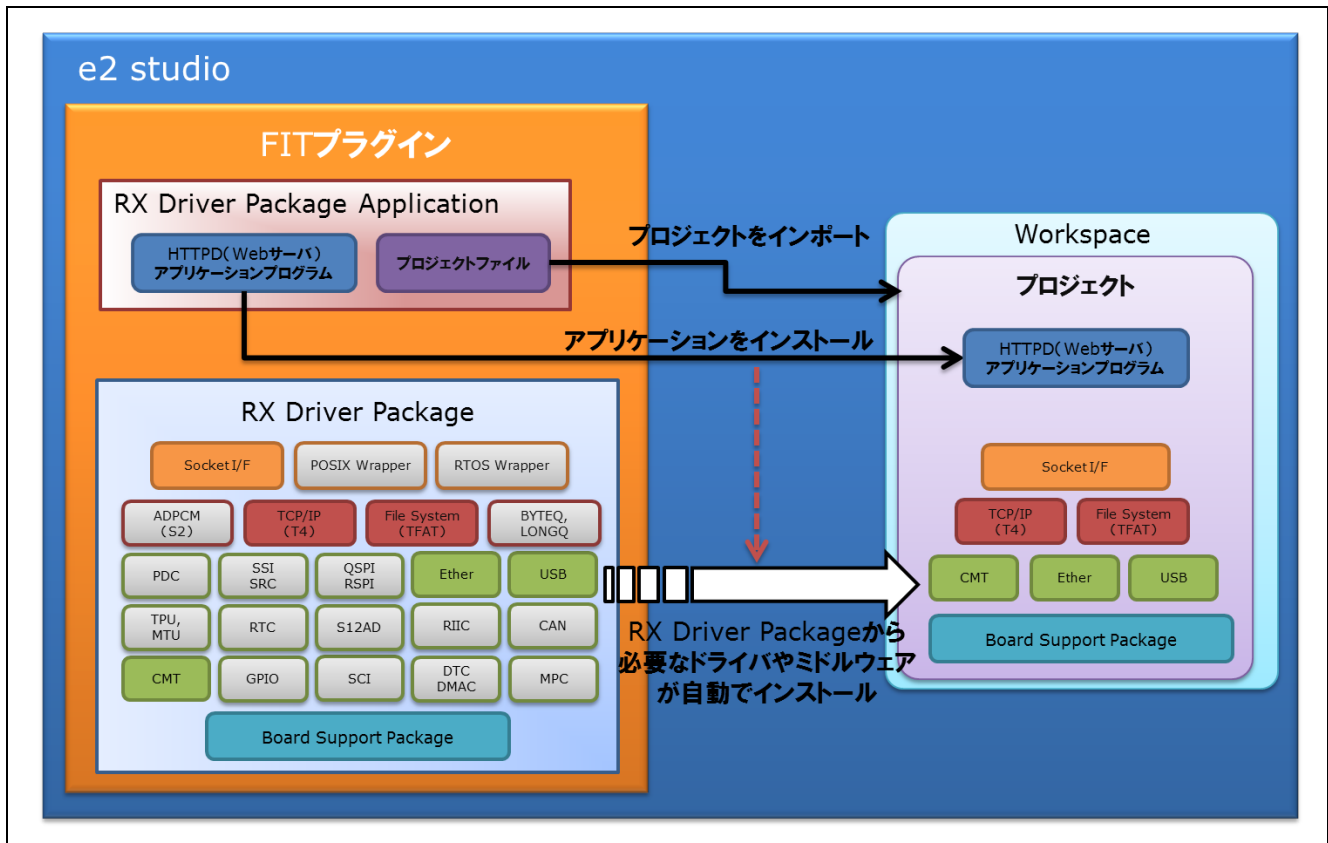


図5.3.1 FIT プラグインによる自動インストール

5.4 RX Driver Package Application と組み合わせて使用する場合

RX Driver Package Application の詳しい使用方法については、各 RX Driver Package Application に付属するドキュメントを参照してください。

6. システム構成

RX Driver Package を使用したシステム構成を以下に示します。

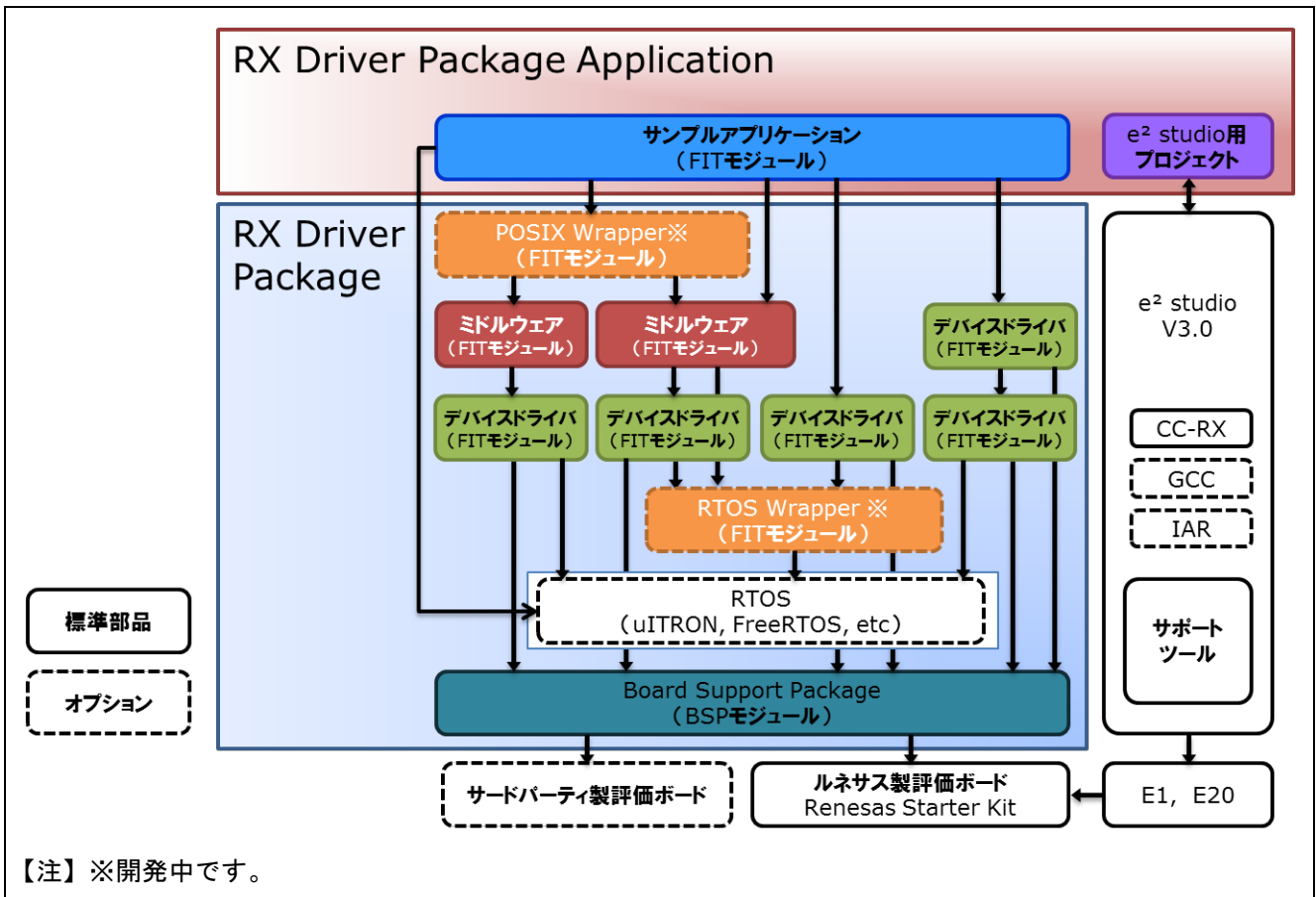


図6.1 システム構成

7. 補足

7.1 M3S-T4-Tiny (TCP/IP プロトコルスタックライブラリ) について

本パッケージには、評価版の「M3S-T4-Tiny (TCP/IP プロトコルスタックライブラリ)」が含まれています。製品版については、以下の URL を参照してください。

<http://japan.renesas.com/mw/t4>

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。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2014.09.01	—	初版発行

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. 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（予約領域）のアクセス禁止

【注意】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、将来の機能拡張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。

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、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っていると、内部ROM、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。お客様の機器・システムの設計において、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て、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。
2. 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。
3. 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に関し、当社は、何らの責任を負うものではありません。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。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製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。
5. 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）、交通用信号機器、
防災・防犯装置、各種安全装置等
当社製品は、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（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図し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損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の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。
7. 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。
8. 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。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。
9. 本資料に記載されている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。また、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に使用しないでください。当社製品または技術を輸出する場合は、「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。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
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禁じます。

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 エレクトロニクス株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

■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/>