
RX64M, RX71M グループ

R01AN2606JJ0104

Rev.1.04

RX Driver Package Ver.1.02

2016.5.16

要旨

本書は、RX64M, RX71M グループ用 RX Driver Package Ver.1.02 のユーザーズマニュアルです。

本ユーザーズマニュアルでは、RX Driver Package の構成、特徴、使用方法と RX Driver Package を利用したサンプルアプリケーションプログラムについて説明します。

動作確認デバイス

RX64M グループ (Renesas Starter Kit+ RX64M)

RX71M グループ (Renesas Starter Kit+ RX71M)

お客様の製品にてご利用される際は、お客様の環境に合わせて十分に評価してください。

また、本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

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 (R01AN1723JU)

RX ファミリ CS+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 動作環境.....	3
2. RX Driver Package とは.....	4
2.1 システム構成.....	4
2.2 RX Driver Package の特徴.....	5
3. RX64M, RX71M グループ用 RX Driver Package の構成.....	6
3.1 フォルダ構成.....	6
3.2 モジュール構成.....	7
3.3 FIT モジュール一覧.....	8
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 プラグインで FIT モジュールをインストールする.....	17
4.3.3 LED 点灯プログラムを作成する.....	20
4.3.4 プログラムをビルドし動作を確認する.....	21
5. RX Driver Package Application について.....	24
5.1 RX Driver Package Application の構成.....	24
6. 補足.....	25
6.1 M3S-T4-Tiny (TCP/IP プロトコルスタックライブラリ) について.....	25
ホームページとサポート窓口.....	26

1. 概要

1.1 適用

本ユーザーズマニュアルは、RX64M, RX71M グループ用 RX Driver Package Ver1.02 に適用します。

1.2 動作環境

本パッケージの動作環境を以下に示します。

表 1-1 動作環境 (RX64M)

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 V4.0.2 以降 CS+ V3.01.00 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 V2.03.00 以降
エミュレータ	E1, E20

表 1-2 動作環境 (RX71M)

対応 MCU	RX71M グループ
評価ボード	Renesas Starter Kit+ RX71M
統合開発環境 (IDE)	e ² studio V4.0.2 以降 CS+ V3.01.00 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 V2.03.00 以降
エミュレータ	E1, E20

2. RX Driver Package とは

RX Driver Package は、開発に必要となる以下のモジュール群を一つのパッケージとしてまとめたソフトウェア・プラットフォーム（フレームワーク）です。複数のモジュールがパッケージングされているため、個別に入手する必要がなく、開発に直ぐに着手できます。

- ボード・サポート・パッケージ(BSP)モジュール
- FIT 周辺機能モジュール（無償版）
- FIT ミドルウェアモジュール（無償版）
- FIT インタフェースモジュール

また、RX Driver Package を活用したサンプルアプリケーションプログラム（RX Driver Package Application）を用いることで、ユーザアプリケーション層の開発が容易になります。

2.1 システム構成

RX Driver Package のシステム構成を以下に示します。

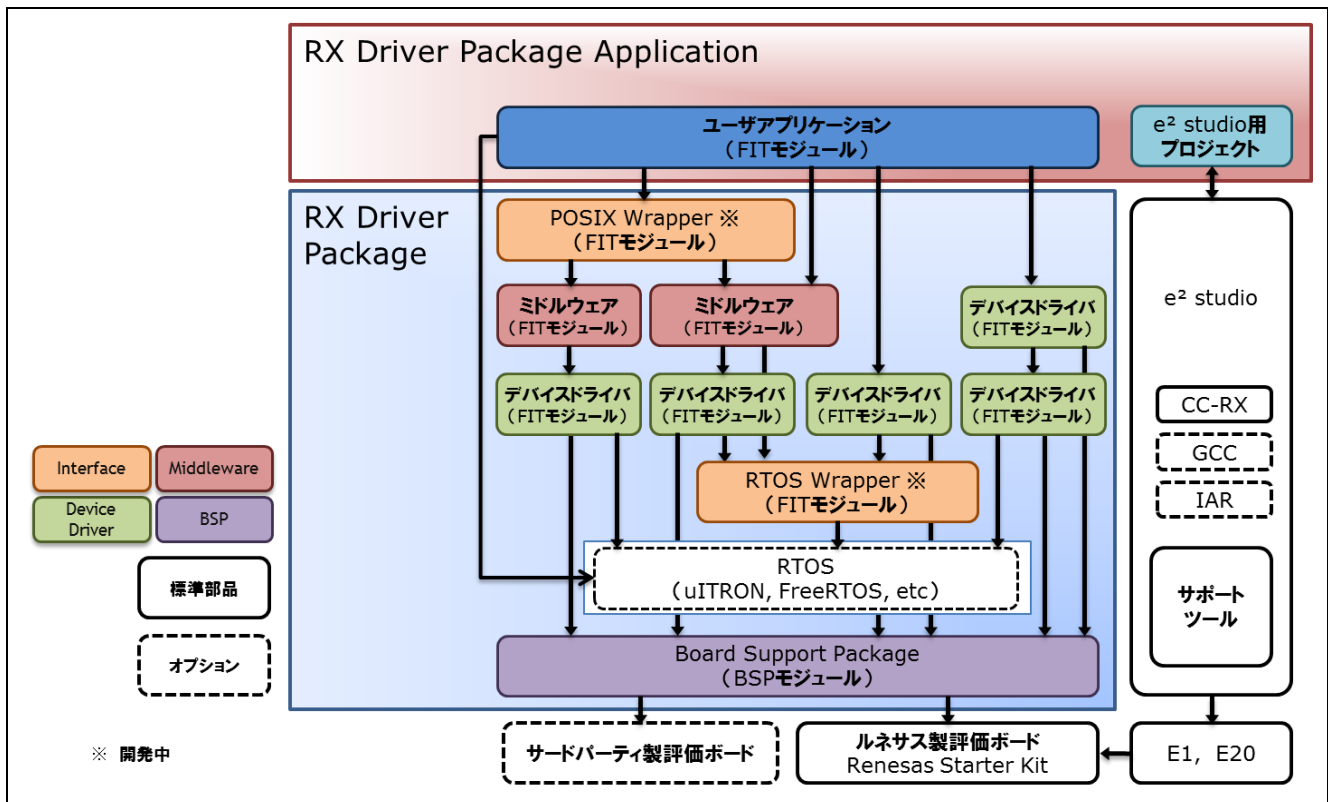


図 2-1 システム構成

2.2 RX Driver Package の特徴

RX Driver Package の特徴を以下に示します。

(a) 必要なモジュールを選択し、すぐにアプリケーションプログラムを開発可能

システムに必要なモジュールをパッケージから選択するだけで簡単にシステムを構築できます。あとはアプリケーションプログラムを開発するだけです。

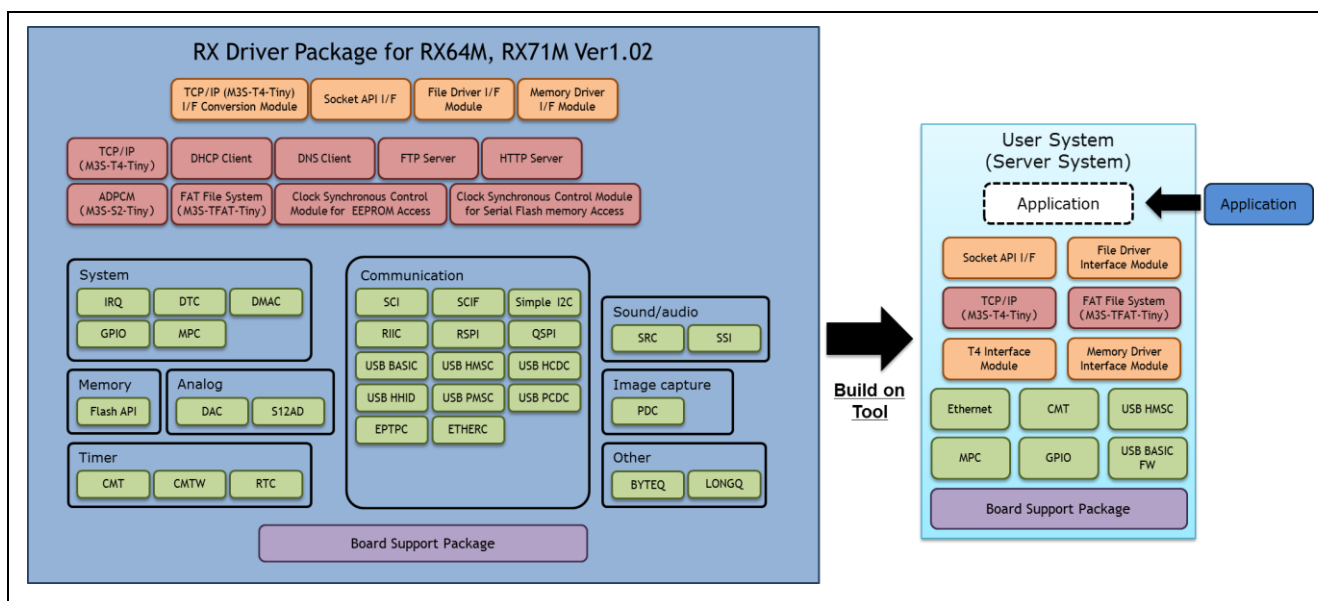


図 2-2 構築イメージ

(b) 無償で利用可能

RX Driver Package 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。

なお、TCP/IP、ファイルシステム等のミドルウェアモジュールも無償版が入っています。

(c) 有償版モジュールへの入れ替えが可能

RX Driver Package に入っている無償版モジュールを製品版（有償）モジュールに置き換えることができます。

製品版（有償）を使用することでモジュールの全機能を使用することができ、また製品版に関するサポートも利用することができます。

(d) ユーザアプリケーションを含めた動作確認が可能

RX Driver Package を用いたユーザアプリケーションのサンプルとして、RX Driver Package Application を提供します。RX Driver Package Application は、RX Driver Package の各種モジュールを動作させるプログラムと、そのプログラムをビルドするためのプロジェクトファイルで構成されています。これにより、すぐにユーザアプリケーションを含めた動作確認を開始することができます。

3. RX64M, RX71M 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 3-1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。

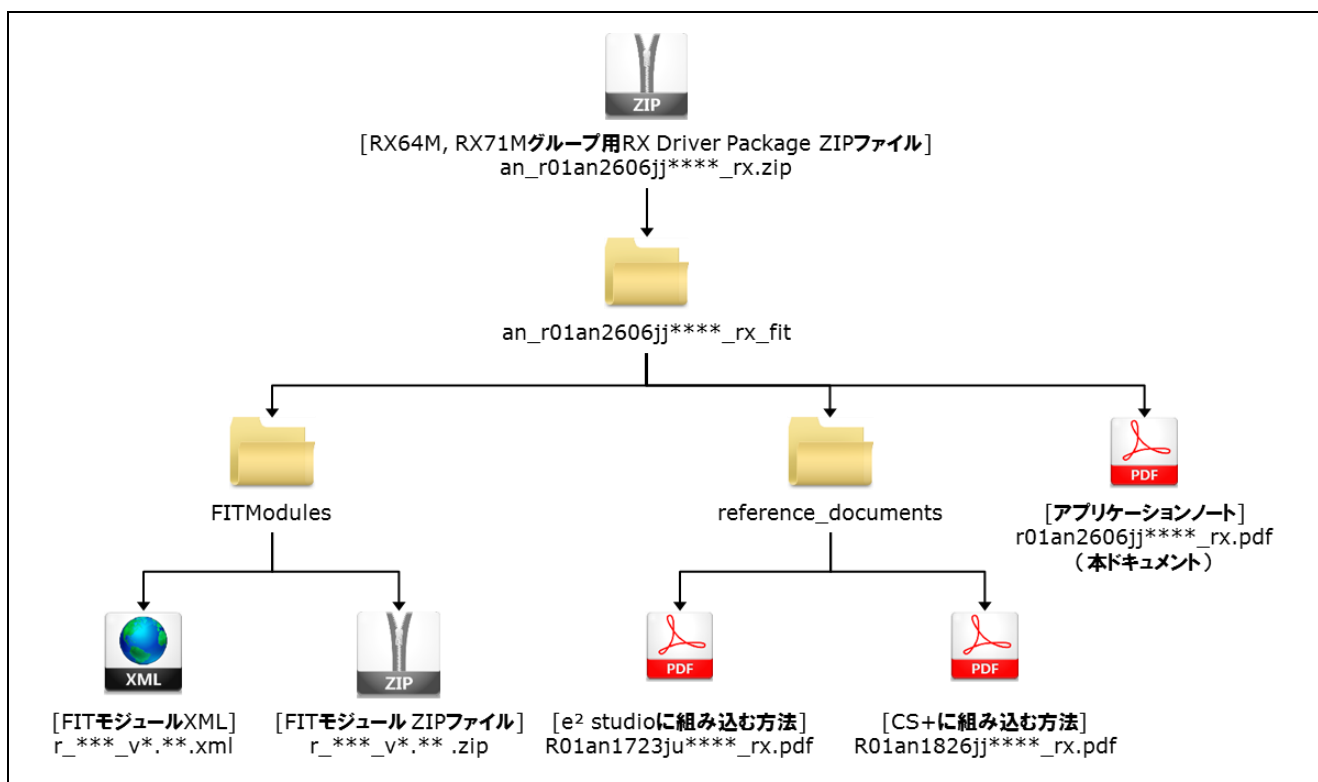


図 3-1 RX64M, RX71M グループ用 RX Driver Package フォルダ構成

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。

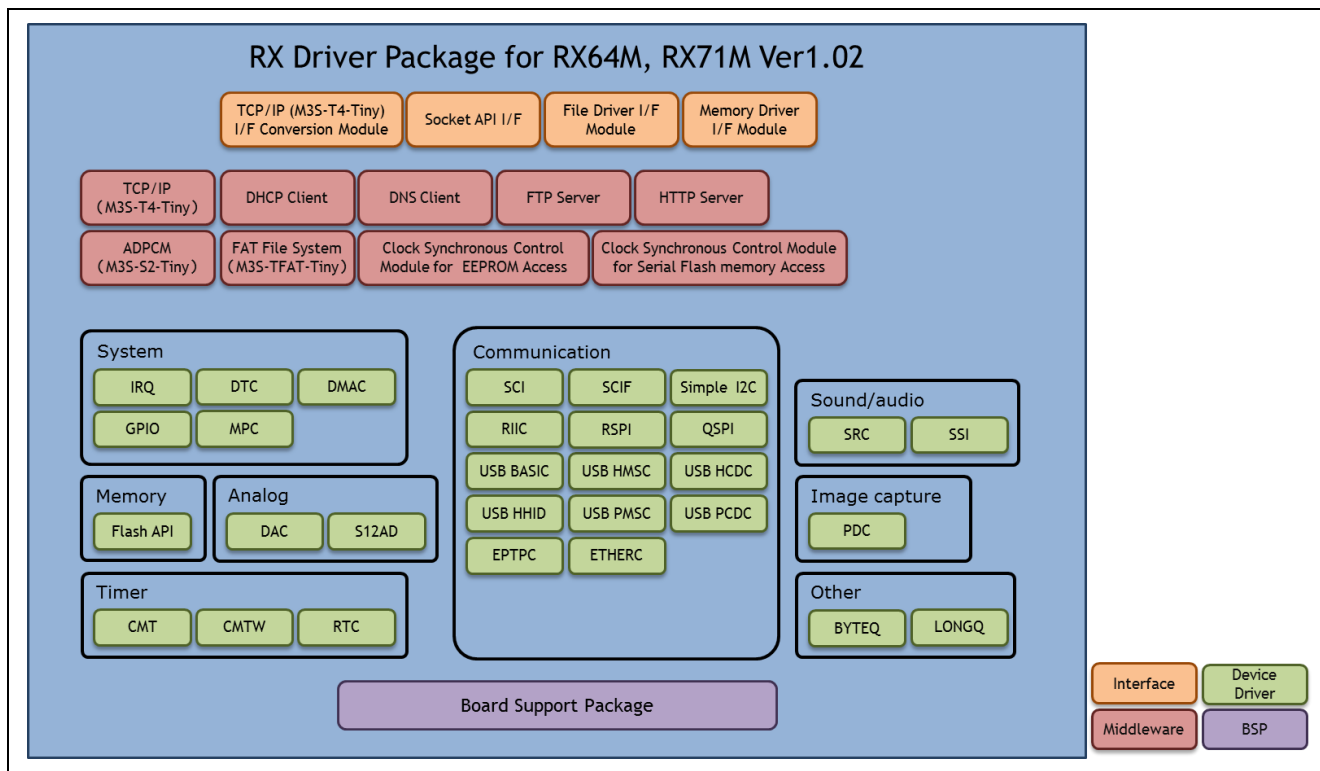


図 3-2 RX64M, RX71M グループ用 RX Driver Package FIT モジュール構成イメージ

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 3-1 RX64M, RX71M グループ用 RX Driver Package FIT モジュール一覧 (○: サポート、-: 未サポート)

種類	モジュール名	FIT モジュール名	RX64M	RX71M	Rev.
Board Support Package	ボードサポートパッケージ(BSP)	r_bsp	○	○	3.00
Device Driver	割り込みコントローラ(IRQ)	r_irq_rx	○	○	1.70
Device Driver	データトランスファコントローラ(DTC)	r_dtc_rx	○	○	2.03
Device Driver	DMA コントローラ(DMAC)	r_dmaca_rx	○	○	1.03
Device Driver	I/O ポート(GPIO)	r_gpio_rx	○	○	1.70
Device Driver	マルチファンクションピンコントローラ(MPC)	r_mpc_rx	○	○	1.70
Device Driver	コンペアマッチタイマ(CMT)	r_cmt_rx	○	○	2.60
Device Driver	コンペアマッチタイマ W(CMTW)	r_cmtw_rx	○	○	1.10
Device Driver	リアルタイムクロック(RTC)	r_rtc_rx	○	○	2.30
Device Driver	シリアルコミュニケーションインタフェース (SCI: 調歩同期式/クロック同期式)	r_sci_rx	○	○	1.70
Device Driver	FIFO 内蔵シリアルコミュニケーションインタフェース(SCIFA: 調歩同期式/クロック同期式)	r_scif_rx	○	○	1.10
Device Driver	FIFO 内蔵シリアルコミュニケーションインタフェース(SCIFA: シリアルメモリ制御用デバイスドライバ)	r_scifa_smstr_rx	○	○	1.08
Device Driver	シリアルコミュニケーションインタフェース (SCI: 簡易 I ² C バス)	r_sci_iic_rx	○	○	1.70
Device Driver	I ² C バスインタフェース(RIIC)	r_riic_rx	○	○	1.70
Device Driver	シリアルペリフェラルインタフェース	r_rspi_rx	○	○	1.30
Device Driver	シリアルペリフェラルインタフェース(RSPI: シリアルメモリ制御用デバイスドライバ)	r_rspi_smstr_rx	○	○	1.09
Device Driver	クワッドシリアルペリフェラルインタフェース (QSPI: シリアルメモリ制御用デバイスドライバ)	r_qsapi_smstr_rx	○	○	1.08
Device Driver	USB Basic Firmware	r_usb_basic	○	○	1.10
Device Driver	USB Host Mass Storage Class	r_usb_hmsc	○	○	1.10
Device Driver	USB Host Communication Device Class	r_usb_hcdc	○	○	1.10
Device Driver	USB Host Human Interface Device Class	r_usb_hhid	○	○	1.10
Device Driver	USB Peripheral Mass Storage Class	r_usb_pmsc	○	○	1.10
Device Driver	USB Peripheral Communications Device Class	r_usb_pcdc	○	○	1.10
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_rx	○	○	1.11
Device Driver	EPTPC Light モジュール	r_ptp_light_rx	○	○	1.10
Device Driver	イーサネットコントローラ(ETHERC)	r_ether_rx	○	○	1.02
Device Driver	パラレルデータキャプチャユニット(PDC)	r_pdc_rx	○	○	1.02
Device Driver	12 ビット A/D コンバータ(S12AD)	r_s12ad_rx	○	○	2.10
Device Driver	D/A コンバータ(DAC)	r_dac_rx	○	○	2.50
Device Driver	フラッシュメモリ(Flash API)	r_flash_rx	○	○	1.30
Device Driver	サンプリングレートコンバータ(SRC)	r_src_api_rx	○	○	1.11
Device Driver	シリアルサウンドインタフェース(SSI)	r_ssi_api_rx	○	○	1.20
Device Driver	Byte Queue Buffer(データ管理)	r_byteq	○	○	1.50
Device Driver	Long Queue Buffer(データ管理)	r_longq	○	○	1.50
Middleware	組み込み用 M3S-T4-Tiny モジュール【注 1】	r_t4_rx	○	○	2.02
Middleware	Ethernet ドライバと組み込み用 TCP/IP	r_t4_driver_rx	○	○	1.02

	M3S-T4-Tiny のインタフェース変換モジュール				
Middleware	組み込み用 TCP/IP M3S-T4-Tiny ソケット API モジュール	r_socket_rx	○	○	1.22
Middleware	組み込み用 TCP/IP M3S-T4-Tiny を用いた DHCP クライアントモジュール	r_t4_dhcp_client_rx	○	○	1.03
Middleware	組み込み用 TCP/IP M3S-T4-Tiny を用いた DNS クライアントモジュール	r_t4_dns_client_rx	○	○	1.02
Middleware	組み込み用 TCP/IP M3S-T4-Tiny を用いた FTP サーバモジュール	r_t4_ftp_server_rx	○	○	1.03
Middleware	組み込み用 TCP/IP M3S-T4-Tiny を用いた Web サーバモジュール	r_t4_http_server_rx	○	○	1.04
Middleware	FTP/Web サーバ用ファイルドライバモジュール	r_t4_file_driver_rx	○	○	1.01
Middleware	音声録音・再生システム(独自 ADPCM コーデック) M3S-S2-Tiny モジュール	r_s2_rx	○	○	3.03
Middleware	オープンソース FAT ファイルシステム M3S-TFAT-Tiny モジュール	r_tfat_rx	○	○	3.02
Middleware	M3S-TFAT-Tiny メモリドライバインタフェースモジュール	r_tfat_driver_rx	○	○	1.02
Middleware	EEPROM アクセス クロック同期制御モジュール	r_eeprom_spi	○	○	2.32
Middleware	Serial Flash memory アクセス クロック同期制御モジュール	r_flash_spi	○	○	2.33

【注 1】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、CS+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 CS+に組み込む方法 Firmware Integration Technology (R01AN1826JJ)」を参照してください。

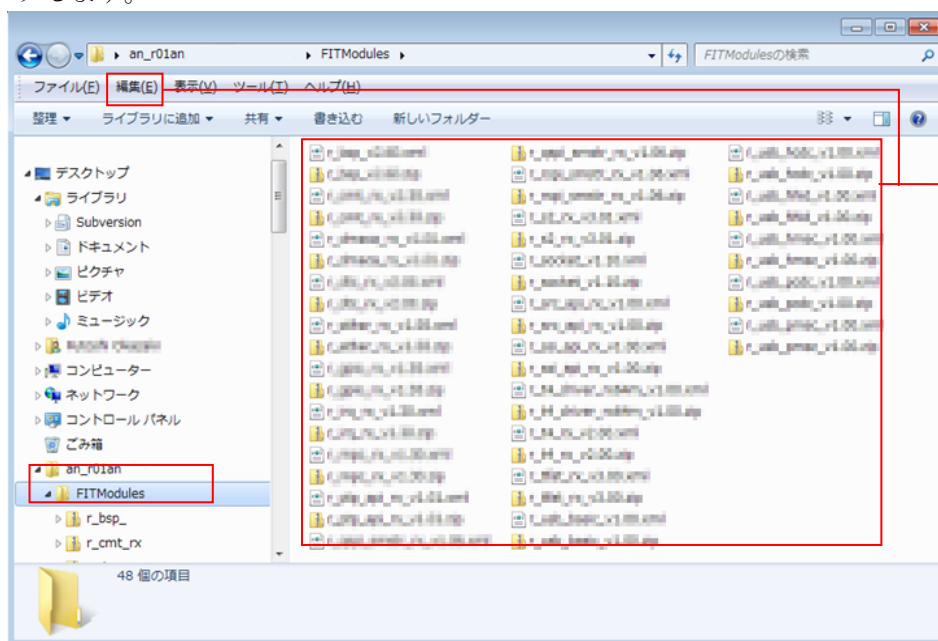
4.1 説明で使用する環境

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

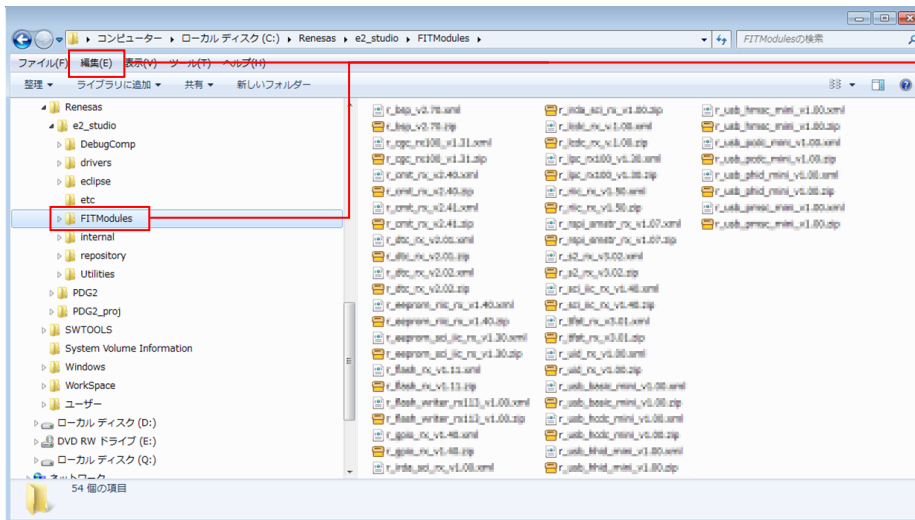
RX Driver Package に入っている FIT モジュールを、e² studio にインストールします。

1. ダウンロードした「an_r01an2606jj****_rx.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(E)」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(E)」メニューから「コピー」をクリックします

4. 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。
 5. 「編集(E)」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(E)」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

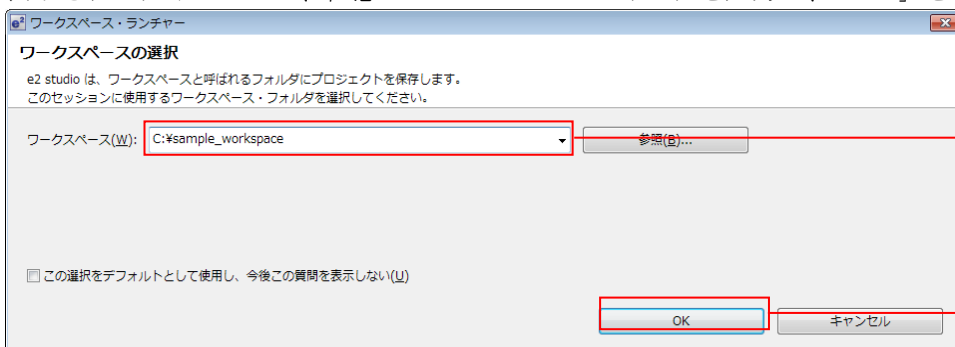
4.3 アプリケーションの作成

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。

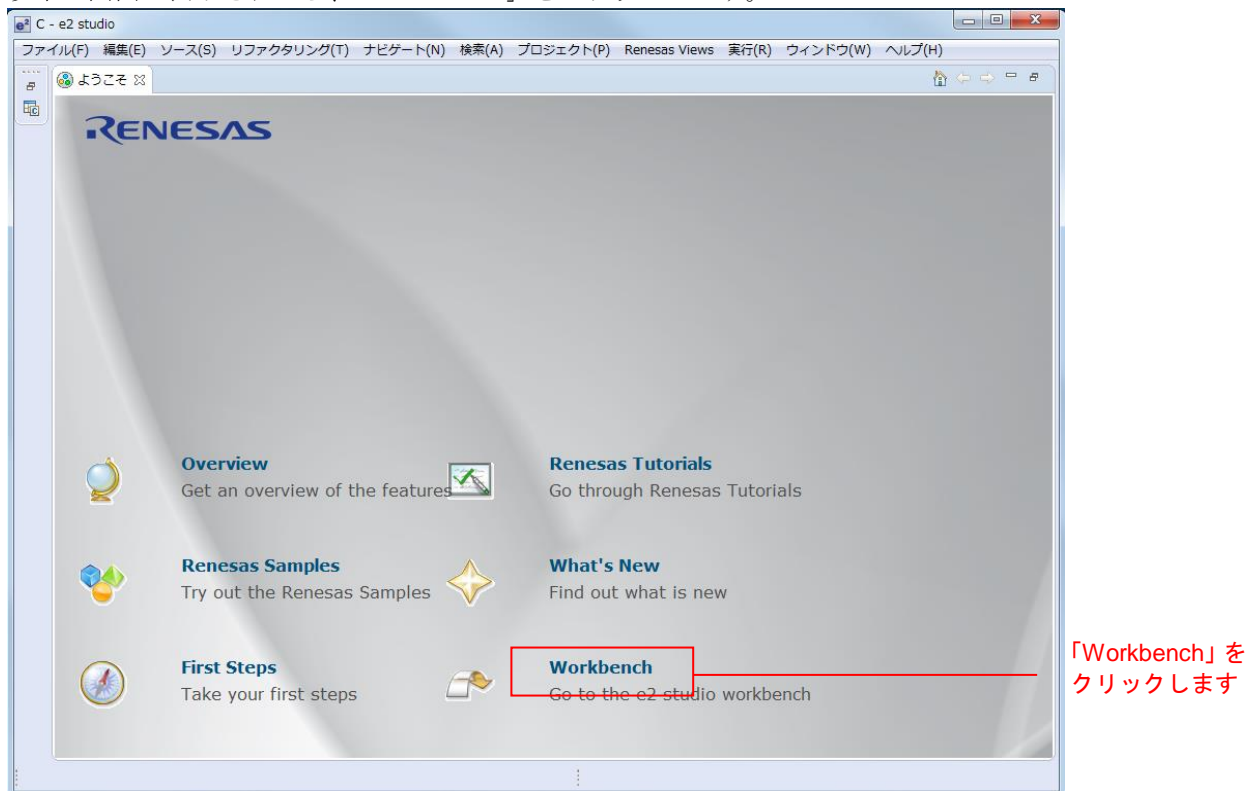
1. e² studio を起動します。
2. 表示されたダイアログに、任意のワークスペースフォルダを入力し、「OK」をクリックします。



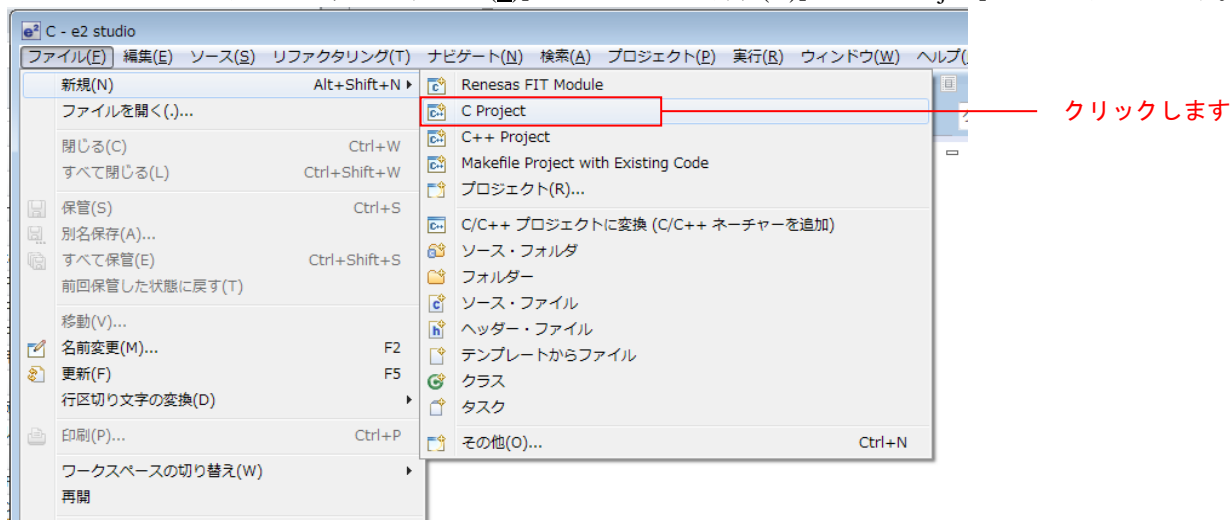
ワークスペースフォルダを入力します

「OK」をクリックします

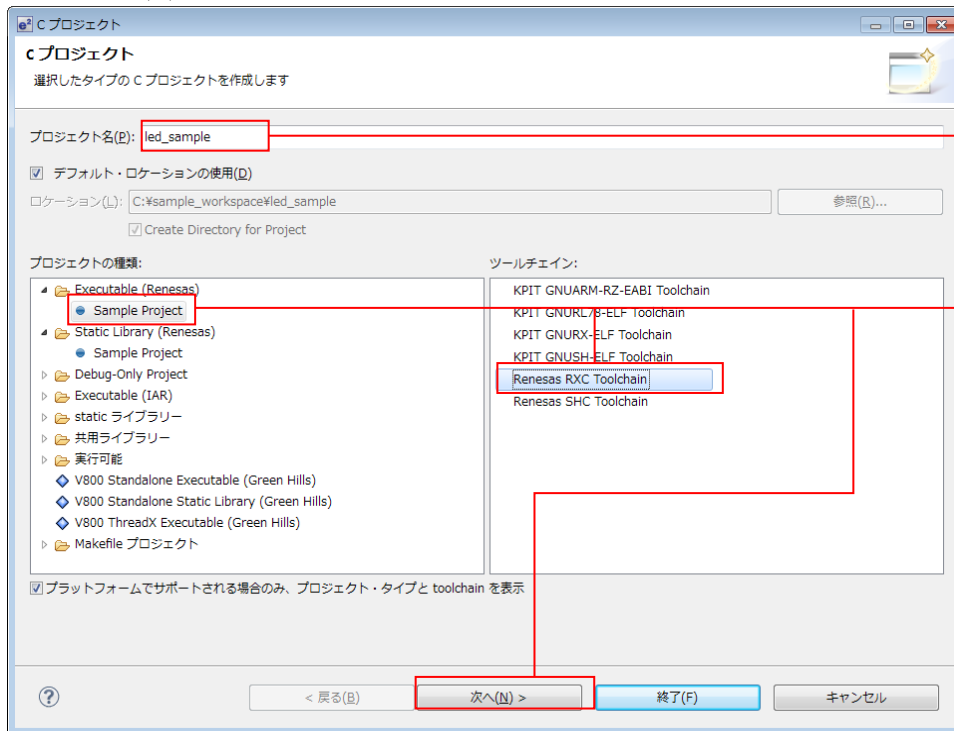
3. 以下の画面が表示されたら、「Workbench」をクリックします。



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



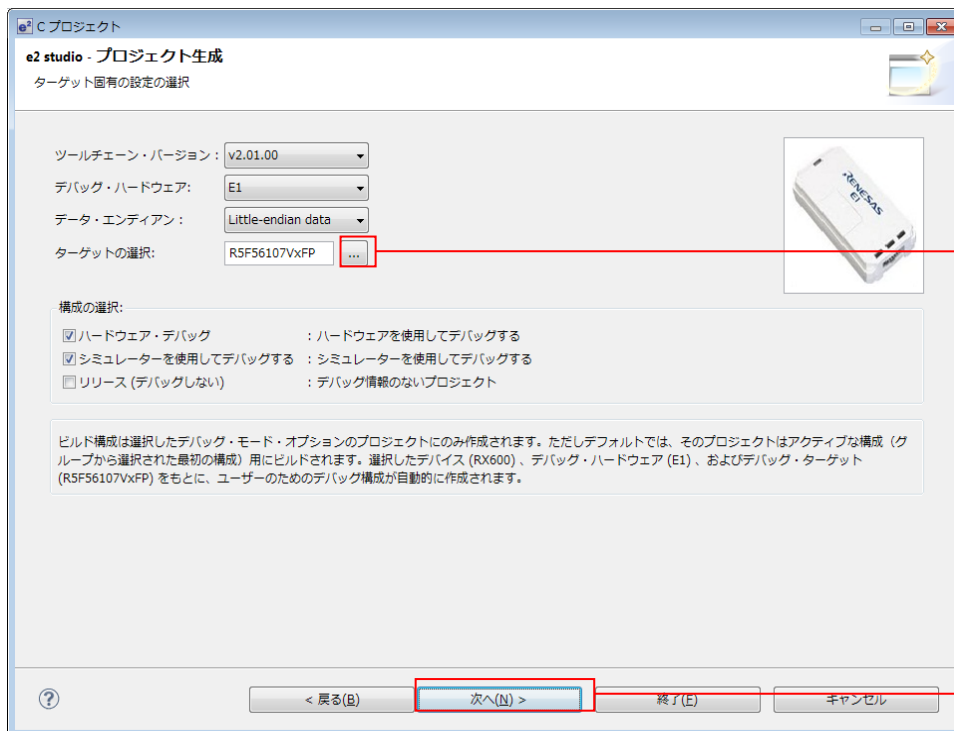
- プロジェクト名(P)を入力します。「プロジェクトの種類:」は、「Executable (Renesas)」の「Sample Project」をクリックします。「ツールチェーン:」は、「Renesas RXC Toolchain」をクリックします。設定が終わったら「次へ(N)」をクリックします。



プロジェクト名
を入力します

クリックします

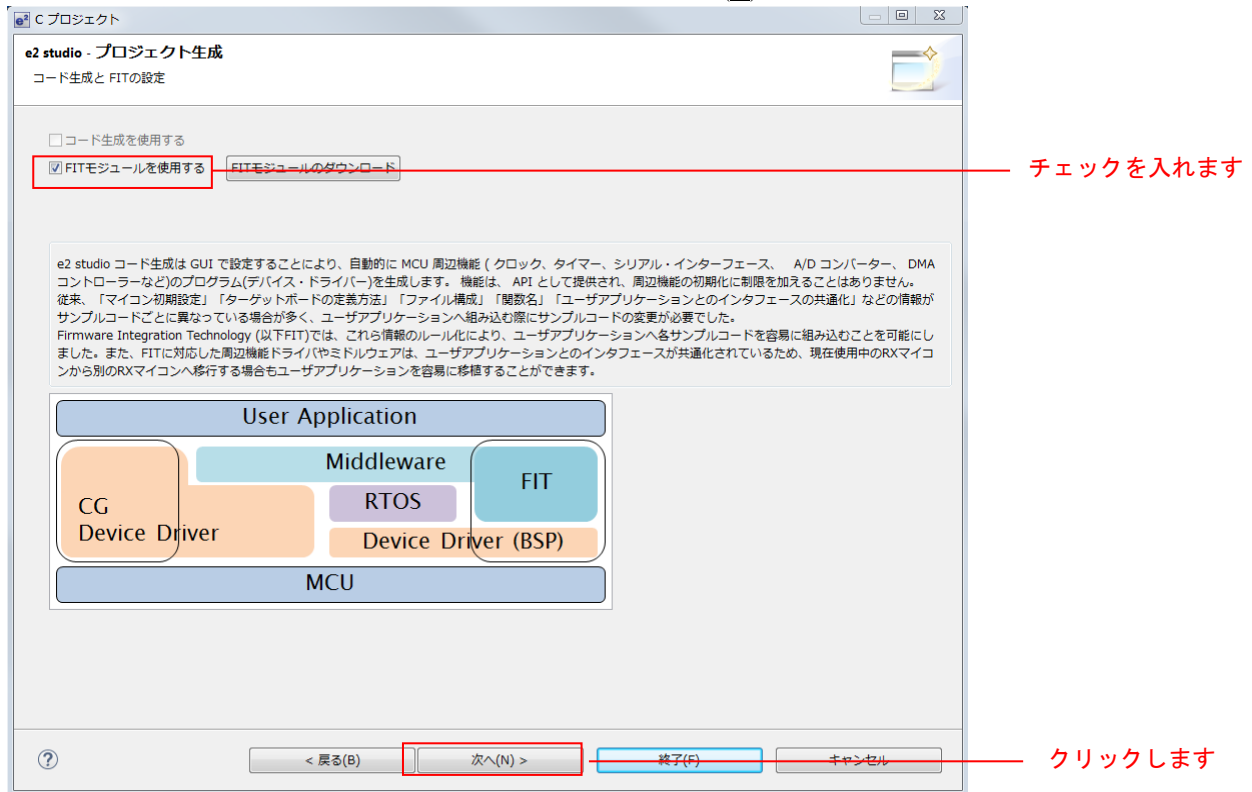
- ターゲットを選択します。「ターゲットの選択:」の「...」ボタンをクリックし、「R5F564MLCxFC」を選択します。設定が終わったら「次へ(N)」をクリックします。



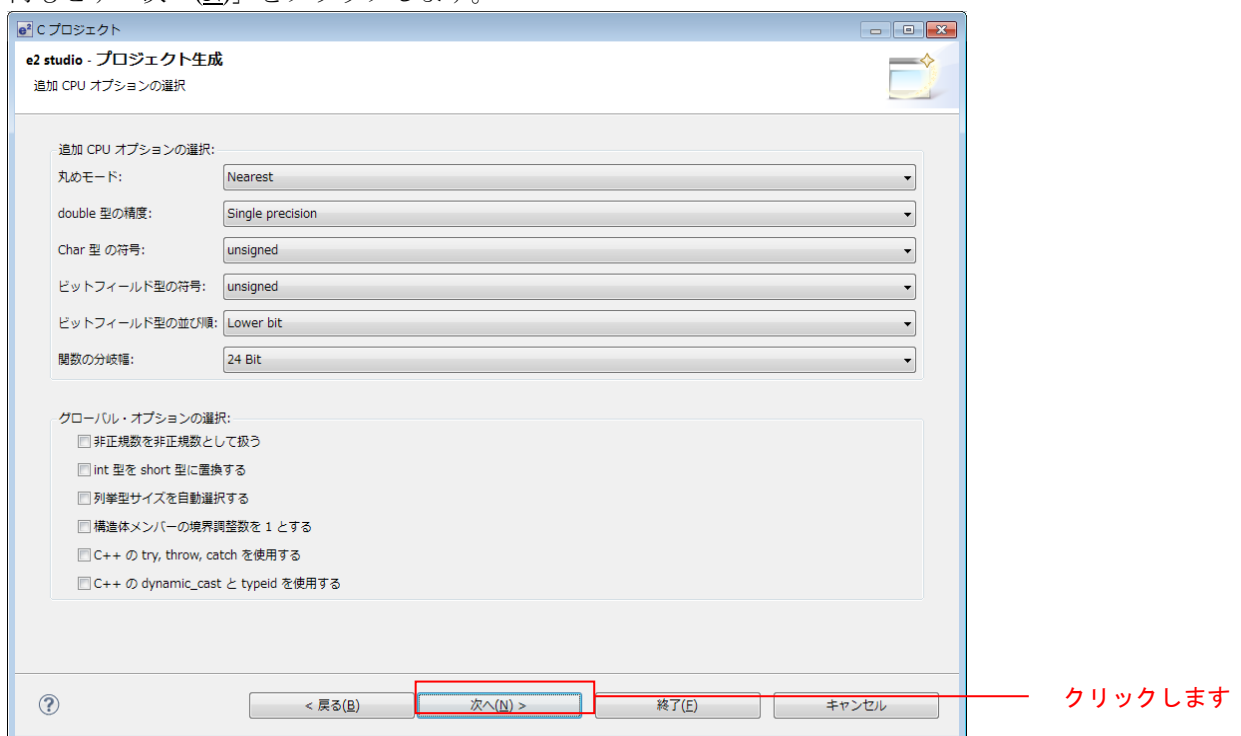
クリックして
「R5F564MLCxFC」
を選択します。

クリックします

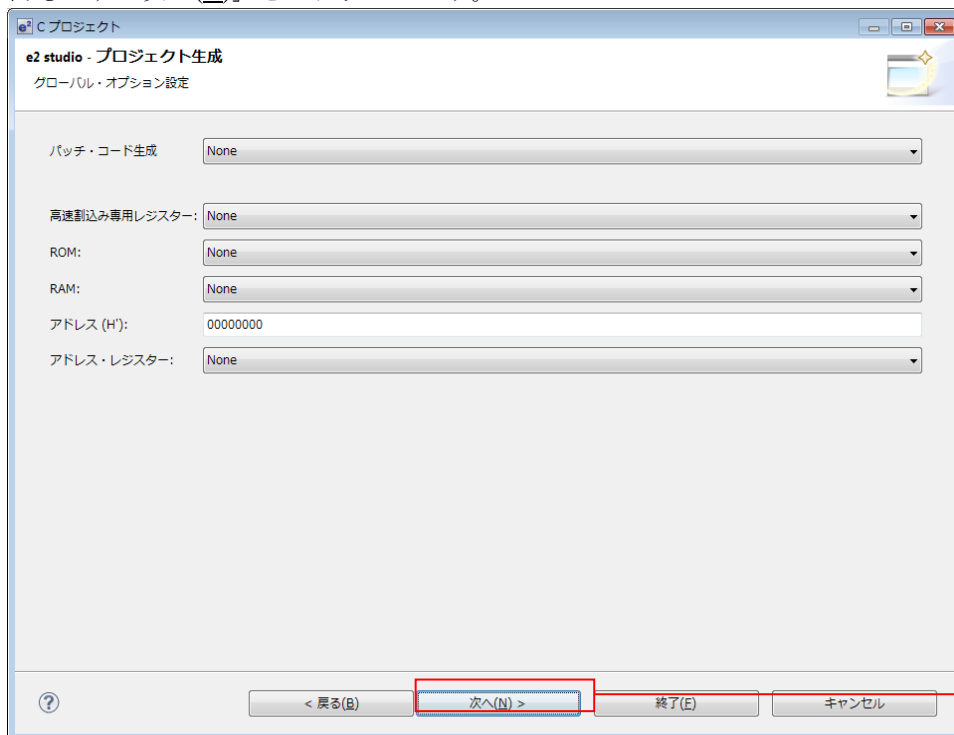
7. 「FIT モジュールを使用する」にチェックを入れて、「次へ(N)」をクリックします。



8. 何もせず「次へ(N)」をクリックします。

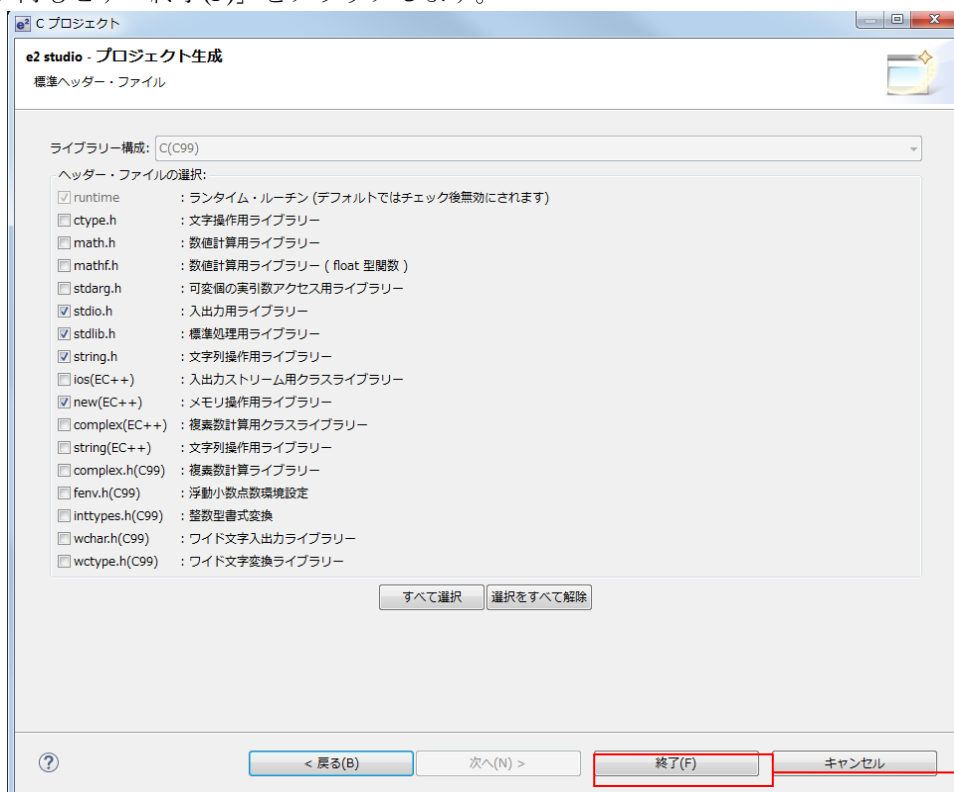


9. 何もせず「次へ(N)」をクリックします。



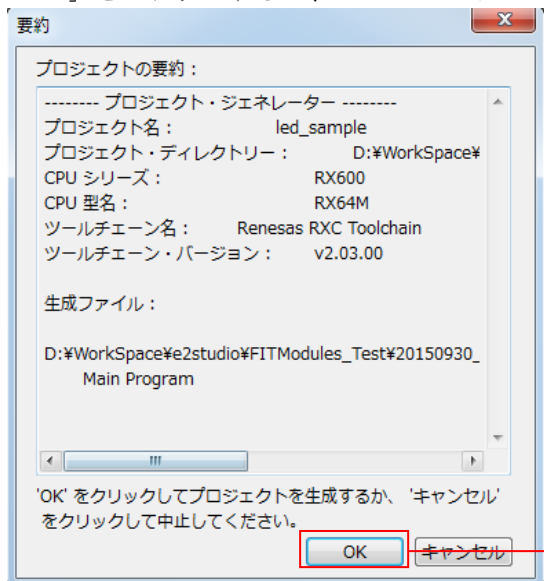
クリックします

10. 何もせず「終了(F)」をクリックします。



クリックします

11. 「OK」をクリックすると、プロジェクトが生成されます。



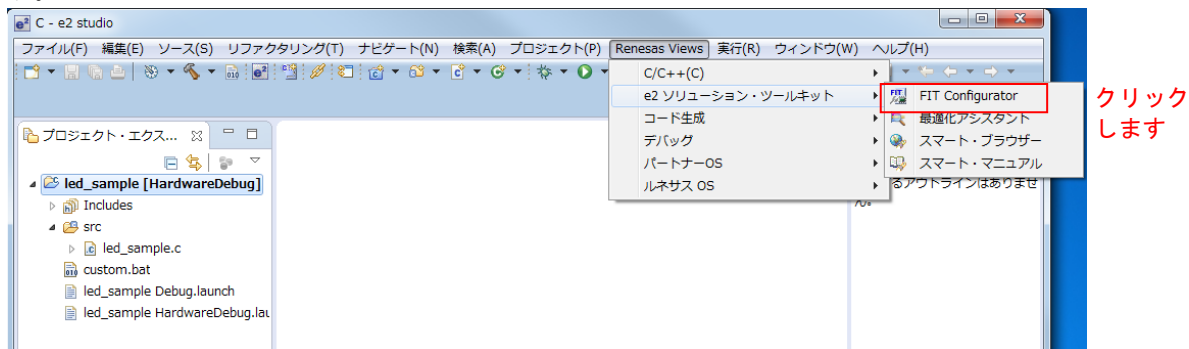
クリックします

4.3.2 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、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. 「Renesas Views」メニューの「e2 ソリューション・ツールキット」の「FIT Configurator」をクリックします。

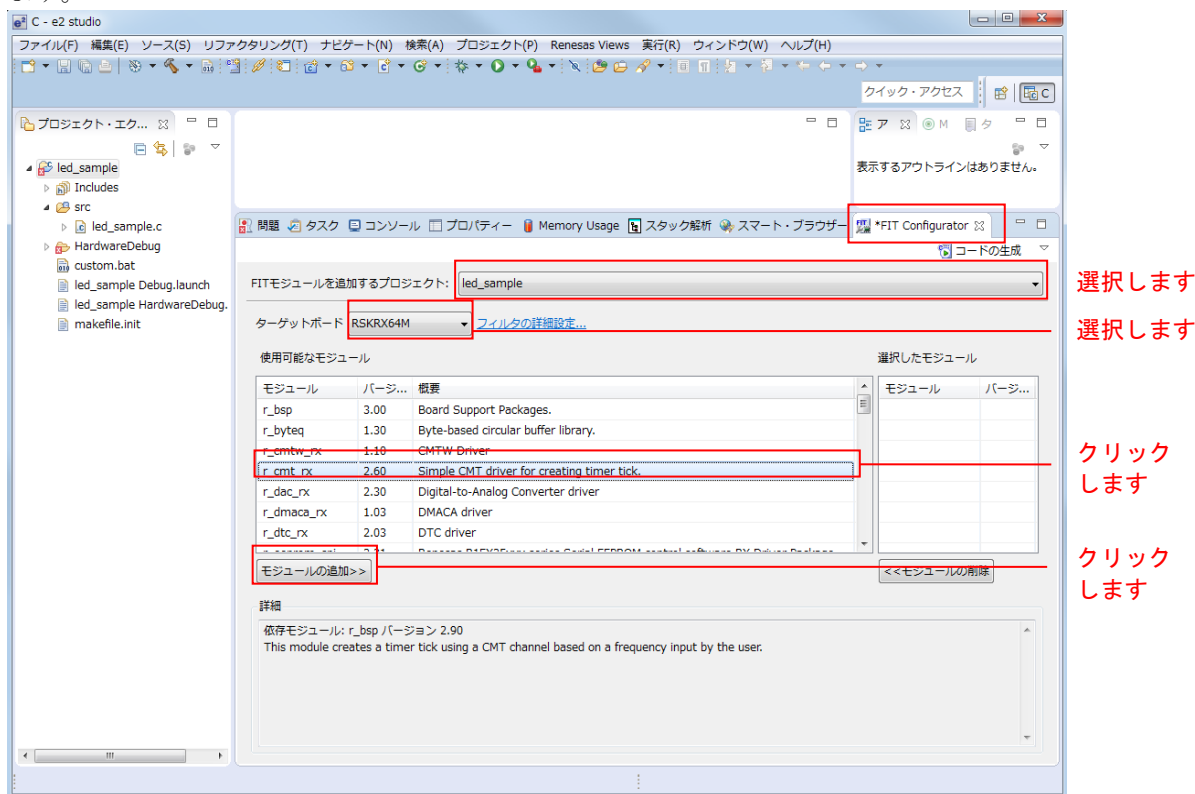


2. 画面右下に「FIT Configurator」が表示されます。

「FIT モジュールを追加するプロジェクト」で作成したプロジェクトを選択します。

次に「ターゲットボード」から「RSKR64M」を選択します。

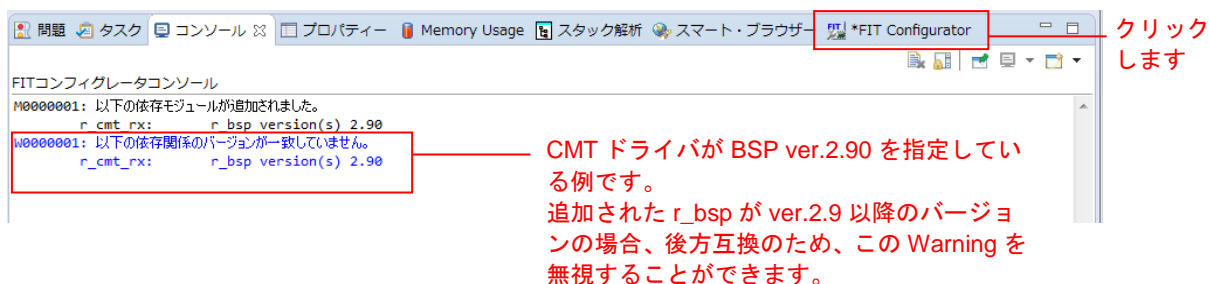
次に「使用可能なモジュール」から「r_cmt_rx」をクリックし、「モジュールの追加>>」をクリックします。



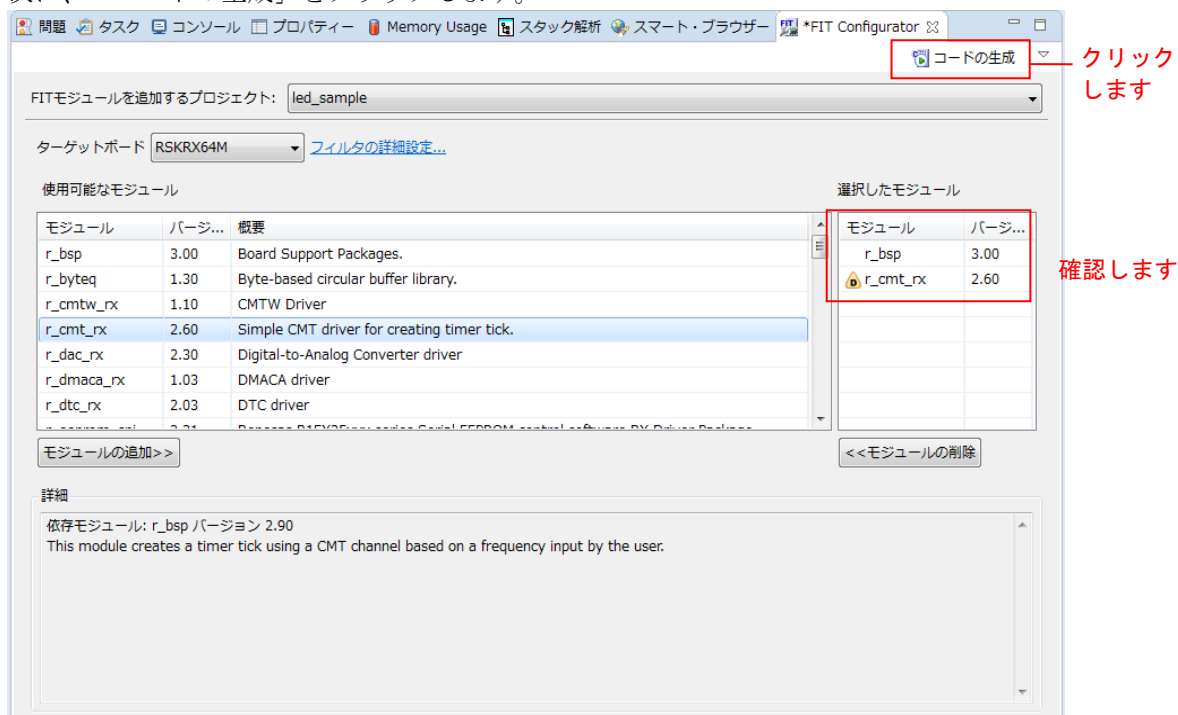
3. 「コンソール」が表示されます。

「r_cmt_rx」と依存関係にある「r_bsp」も同時に追加されます。（※）
再度「FIT Configurator」をクリックします。

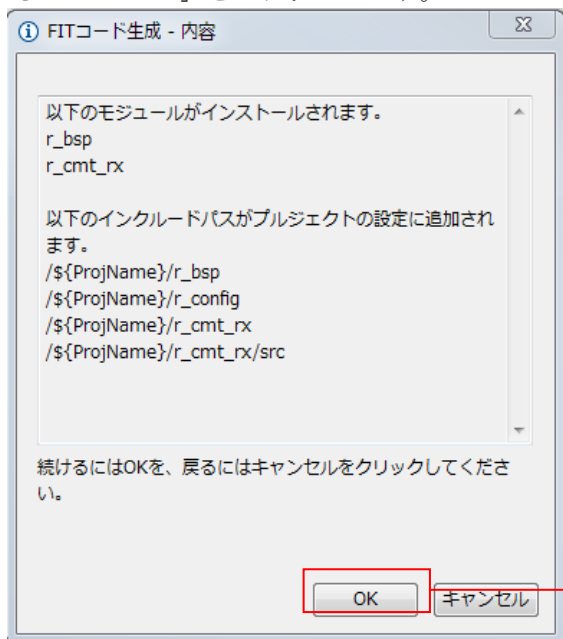
※：追加する FIT ドライバは、適応可能な「r_bsp」のバージョンを指定します。FIT ドライバのリリースタイミングにより、指定した「r_bsp」のバージョンと追加された「r_bsp」のバージョンが不一致の場合、コンソール画面に Warning（W0000001）が出力されます。追加された「r_bsp」のバージョンが、指定した「r_bsp」のバージョン以降の場合、バージョンの新しい「r_bsp」は後方互換を持つため、Warning を無視することができます。



4. 「選択したモジュール」に「r_bsp」と「r_cmt_rx」が追加されていることを確認します。尚「r_cmt_rx」に表示されている ⚠ は上記 3. で説明した Warning の発生を指し示すものです。無視して構いません。次に、「コードの生成」をクリックします。



5 そのまま「OK」をクリックします。



クリックします

4.3.3 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/(プロジェクト名).c を開き、以下のように修正します。

【src/(プロジェクト名).c】

```
#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;

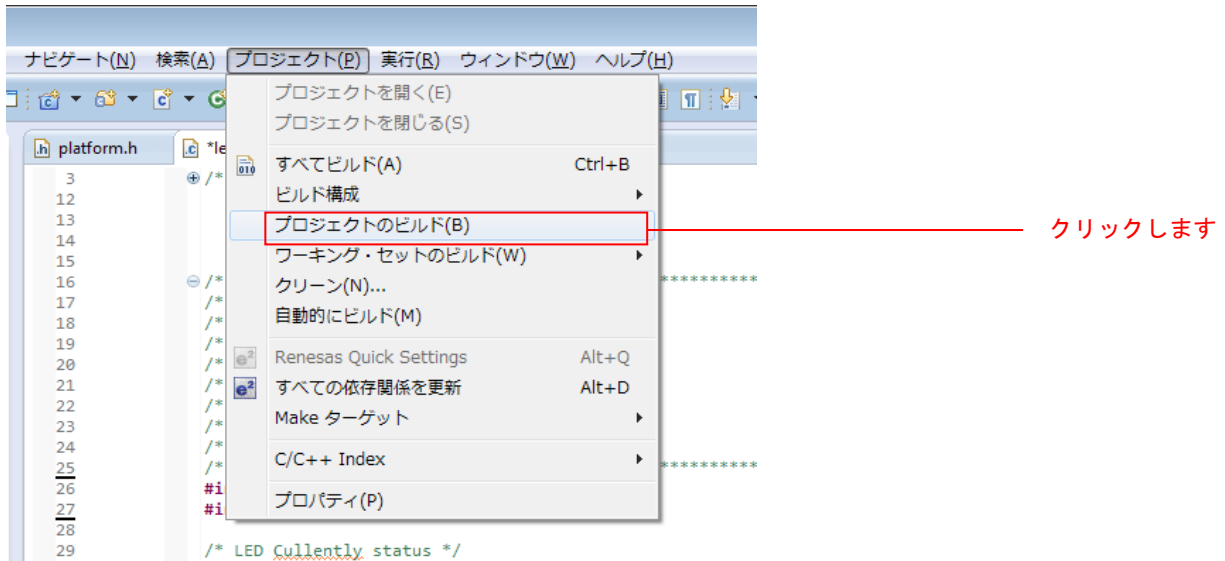
    /* LED0 off */
    LED0 =LED_OFF;
    /* Create of 0.5 second(2Hz) cyclic timer. */
    R_CMT_CreatePeriodic(2, &call_back, &cmt_ch);

    while(1);
}
```

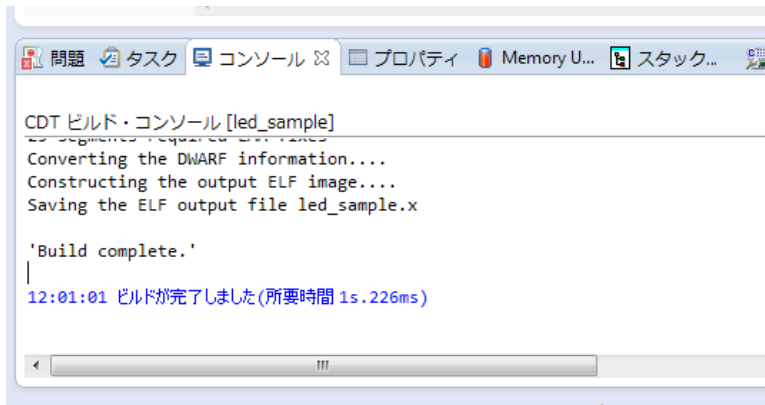
4.3.4 プログラムをビルドし動作を確認する

作成したプログラムをビルドして、動作を確認します。

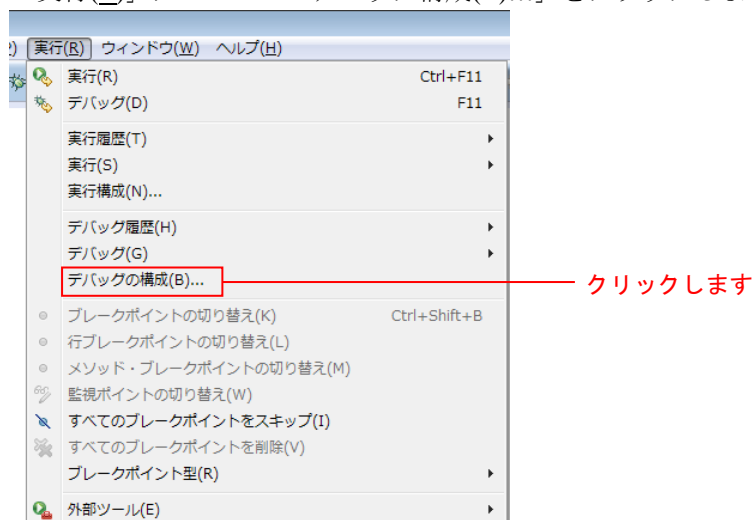
1. 「プロジェクト(P)」メニューの「プロジェクトをビルド(B)」をクリックします。




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。

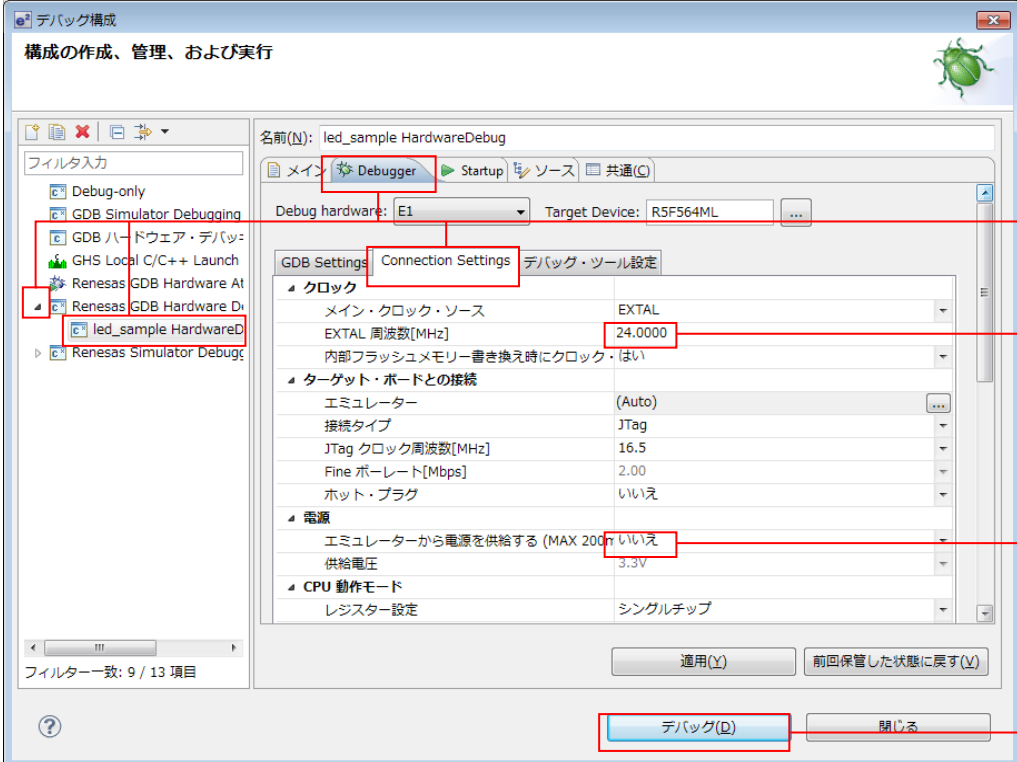


3. 「実行(R)」メニューの「デバッグ構成(B)...」をクリックします。



4. 画面左側「Renesas GDB Hardware Debugging」の  をクリックし、「(プロジェクト名) HardwareDebug」をクリックします。
- 「Debugger」タブをクリックし、「Connection Setting」タブをクリックします。
- 「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(D)」をクリックします。

※：外部電源を使用する場合の設定です。エミュレータから電源を供給する場合は「はい」を選択してください。



名前(N): led_sample HardwareDebug

Debug hardware: E1 Target Device: R5F564ML

GDB Settings Connection Settings デバッグ・ツール設定

▲ クロック		
メイン・クロック・ソース		EXTAL
EXTAL 周波数[MHz]	24.0000	
内部フラッシュメモリー書き換え時にクロック		はい

▲ ターゲット・ボードとの接続		
エミュレータ		(Auto)
接続タイプ		JTag
JTag クロック周波数[MHz]		16.5
Fine ボーレート[Mbps]		2.00
ホット・プラグ		いいえ

▲ 電源		
エミュレータから電源を供給する (MAX 200mA)		いいえ
供給電圧		3.3V

▲ CPU 動作モード		
レジスタ設定		シングルチップ

適用(Y) 前回保管した状態に戻す(V)

デバッグ(D) 閉じる

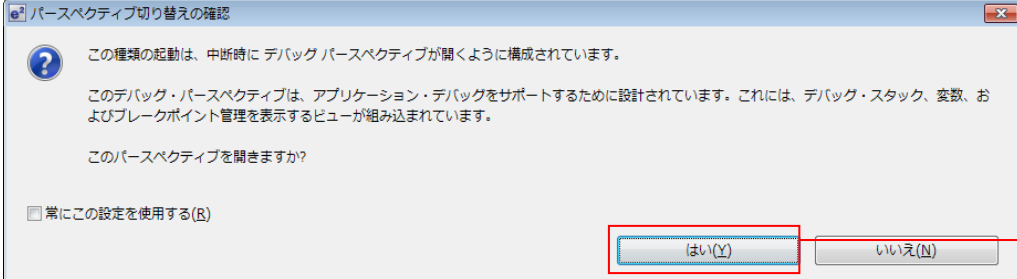
クリックします

「24.0000」に修正します

「いいえ」に変更します

クリックします

5. 以下のメッセージが表示されたら、「はい(Y)」をクリックします。



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。

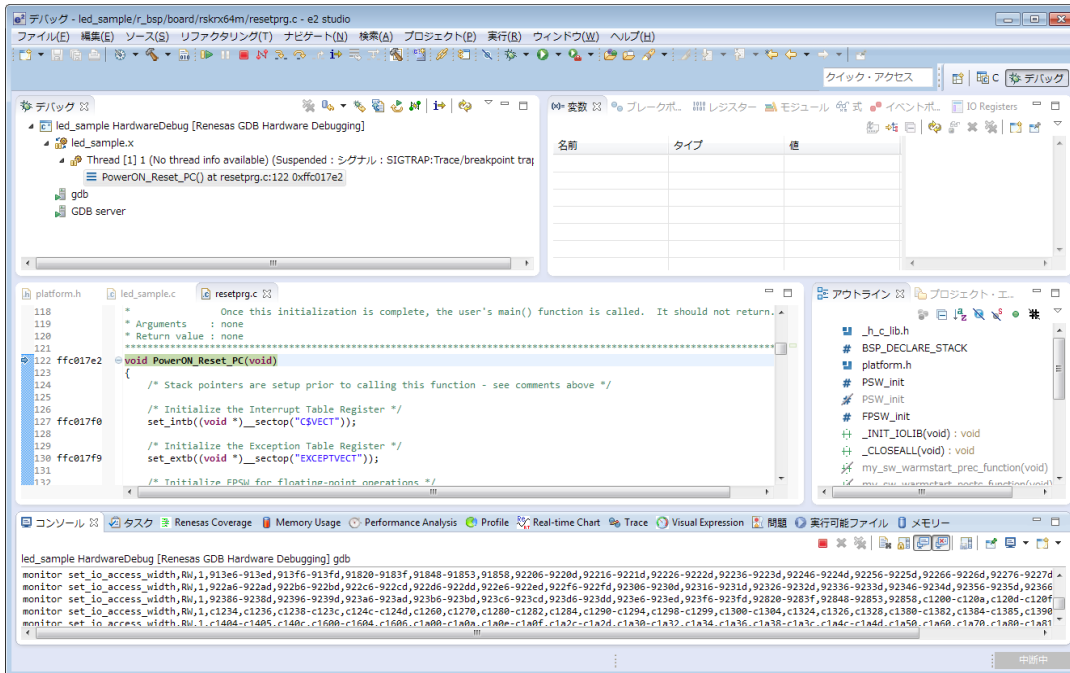
このパースペクティブを開きますか?

常にこの設定を使用する(B)

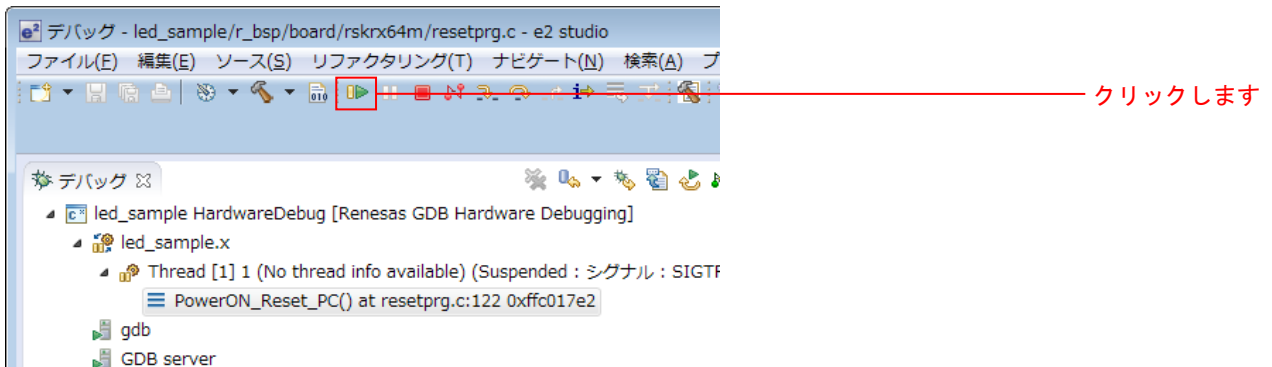
はい(Y) いいえ(N)

クリックします

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ので、すぐに評価を開始することができます。

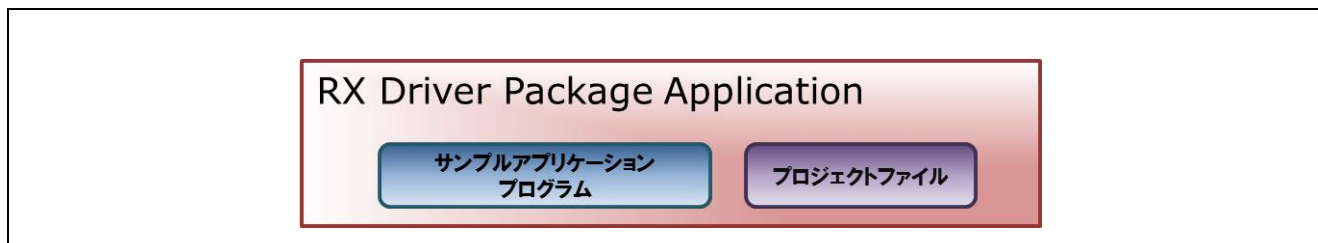


図 5-1 RX Driver Package Application の構成

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て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。

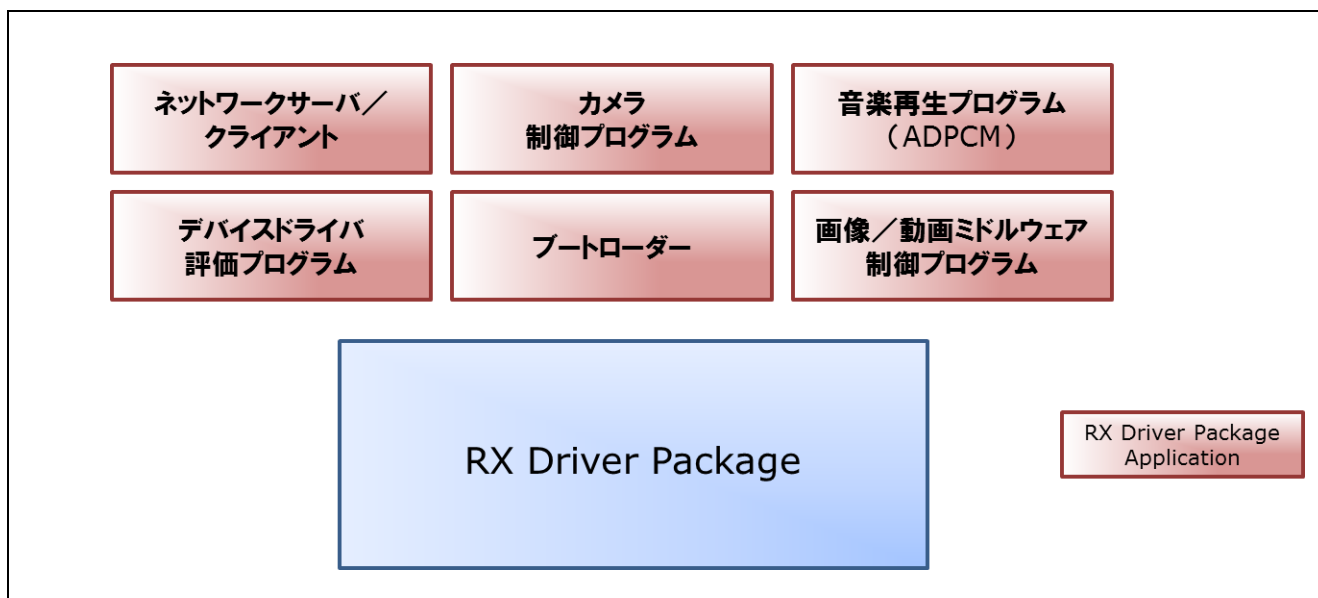


図 5-2 RX Driver Package Application の種類

6. 補足

6.1 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
1.01	2015.01.05	—	既存モジュールを最新バージョンに更新。 新規公開されたモジュールを追加。
1.02	2015.09.30	—	RX71M グループを追加し、既存モジュールを最新バージョンに更新。 e2studio のバージョンを更新。併せて設定手順を修正。
1.03	2016.02.05	5	2.2 RX Driver Package の特徴 図 2-2 構築イメージ にて、 「*1 Japanese edition only」を削除
		7	3.2 モジュール構成 図 3 2 RX64M, RX71M グループ用 RX Driver Package FIT モジュール構成イメージ にて、 「*1 Japanese edition only」を削除
		8	3.3 FIT モジュール一覧 表 3-1 RX64M, RX71M グループ用 RX Driver Package FIT モジュール一覧 にて、 FIFO 内蔵シリアルコミュニケーションインタフェースのモジュール名元は、「r_scif_smster_rx」であった。 「イーサネットコントローラ用 PTP コントローラ(EPTPC)」を削除 「EPTPC Light モジュール」を追加
		9	3.3 FIT モジュール一覧 表 3-1 RX64M, RX71M グループ用 RX Driver Package FIT モジュール一覧 にて、 組み込み用 TCP/IP M3S-T4-Tiny ソケット API モジュールのモジュール名元は、「r_socket」であった。 Serial Flash memory アクセス クロック同期制御モジュールの Rev. 元は、「2.32」であった。
		—	イーサネットコントローラ用 PTP コントローラ(EPTPC)モジュールを削除 EPTPC Light モジュールを追加
1.04	2016.05.16	4	2.1 システム構成 にて、図中の POSIX Wrapper に「※」を追加し、「開発中」に変更した。
		5	2.2 RX Driver Package の特徴 にて、図中から「POSIX API I/F」を削除した。
		6	3.1 フォルダ構成 にて、図を更新した
		7	3.2 モジュール構成 にて、図中から「POSIX API I/F」を削除した。
		8	3.3 FIT モジュール一覧 にて、 イーサネットコントローラ用 PTP コントローラ(EPTPC)モジュールを追加 EPTPC Light モジュールを更新 POSIX ラッパーモジュールを削除
18	4.3.2 FIT プラグインで FIT モジュールをインストールする 3 「コンソール」が表示されます にて、「r_bsp」バージョンの依存説明を変更		

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 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレストシア）

■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/>