RX64M Group

RX Driver Package User's Manual

R01AN2460EJ0101
Rev.1.01
Jan 5, 2015

Introduction

This document is the RX64M Group RX Driver Package User's Manual, version 1.01.

This User's Manual describes basic structures, features and usage of RX Driver Package applications, and about the sample application program using the FIT modules included in this package.

Target Device

RX64M Group (Renesas Starter Kit+ RX64M)

When using this application note with your product, careful evaluation is recommended.

And when using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Related Documents

- RX Family Board Support Package Module Using Firmware Integration Technology (R01AN1685EU)
- Firmware Integration Technology User's Manual (R01AN1833EU)
- RX Family Adding Firmware Integration Technology Modules to Projects (R01AN1723EU)
- RX Family Adding Firmware Integration Technology Modules to CS+ Projects (R01AN1826EJ)
- The User's Manual provided with the RX Driver Package Application.

Contents

1. Overview

1.1 Applicability

This User's Manual applies to the RX64M Group RX Driver Package, version 1.01.

1.2 Operating Environment

This package runs under the operating environment described below.

Table 1.2.1 Operating Environment

Microcontroller	RX64M Group
Evaluation board	Renesas Starter Kit+ RX64M
Integrated development environment (IDE)	e^2 studio, V3.1.2 or later Or: CS+ V3.00.00 or later
Cross tools	RX Family C/C++ Compiler Package V2.02.00 or later
Emulator	E1, E20

2. About RX Driver Package

The RX Driver Package is a software platform (framework) that combines the following modules to be required for development in a single package. Since the package contains multiple modules, you can start developing immediately without having to obtain each module separately.

- Board Support Package (BSP) module
- FIT peripheral function modules (free version)
- FIT middleware modules (free version)
- FIT interface modules

You can develop the user application layer with ease by using the Sample Application Program (RX Driver Package Application) which utilizes the RX Driver Package.

2.1 System Structure

The figure below shows the system structure of the RX Driver Package.

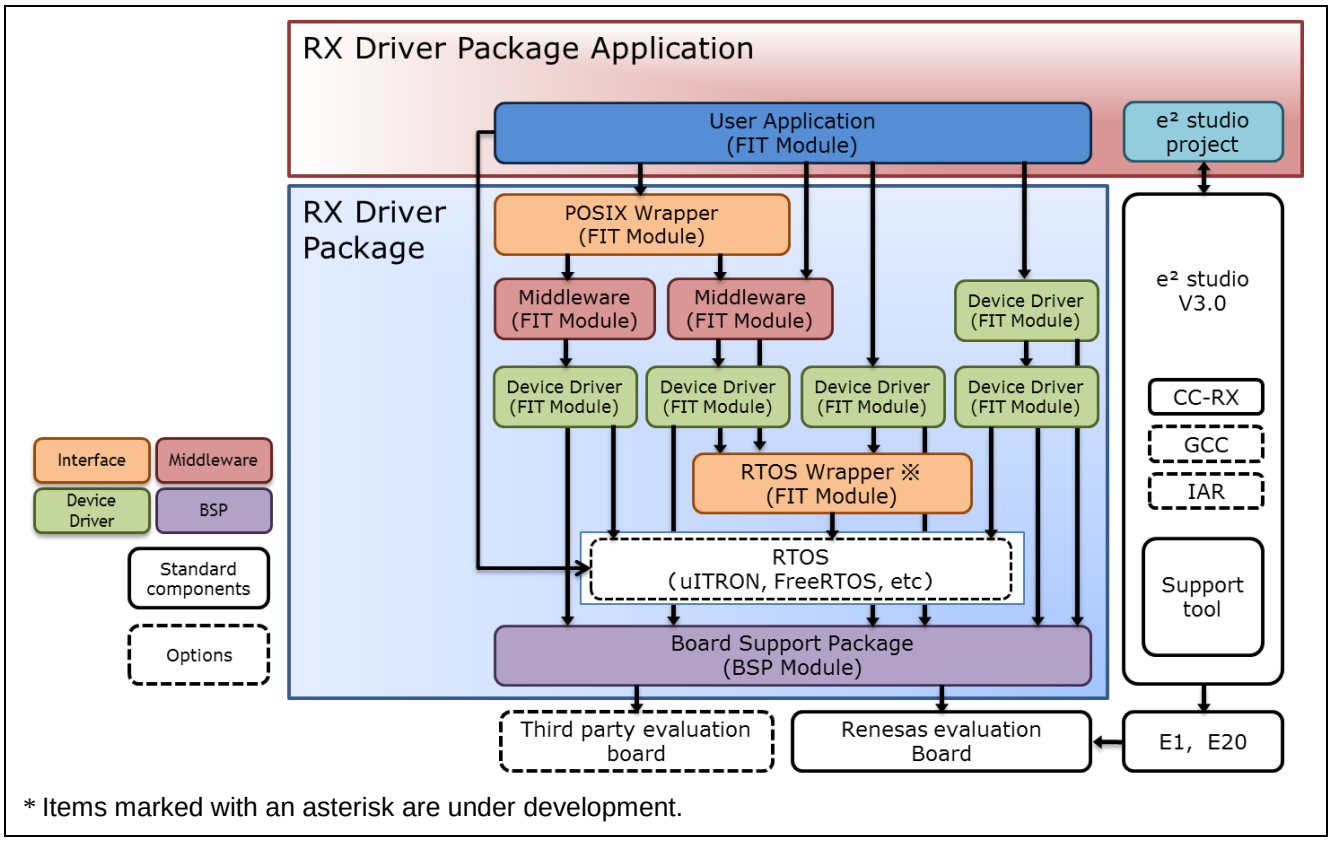


Figure 2.1.1 System Structure

2.2 RX Driver Package Features

The RX Driver Package has the following features.

(a) Select necessary modules and start developing immediately the application program

You can easily build a system simply by selecting the modules you need from the package. After that, all you have to do is develop the application program.

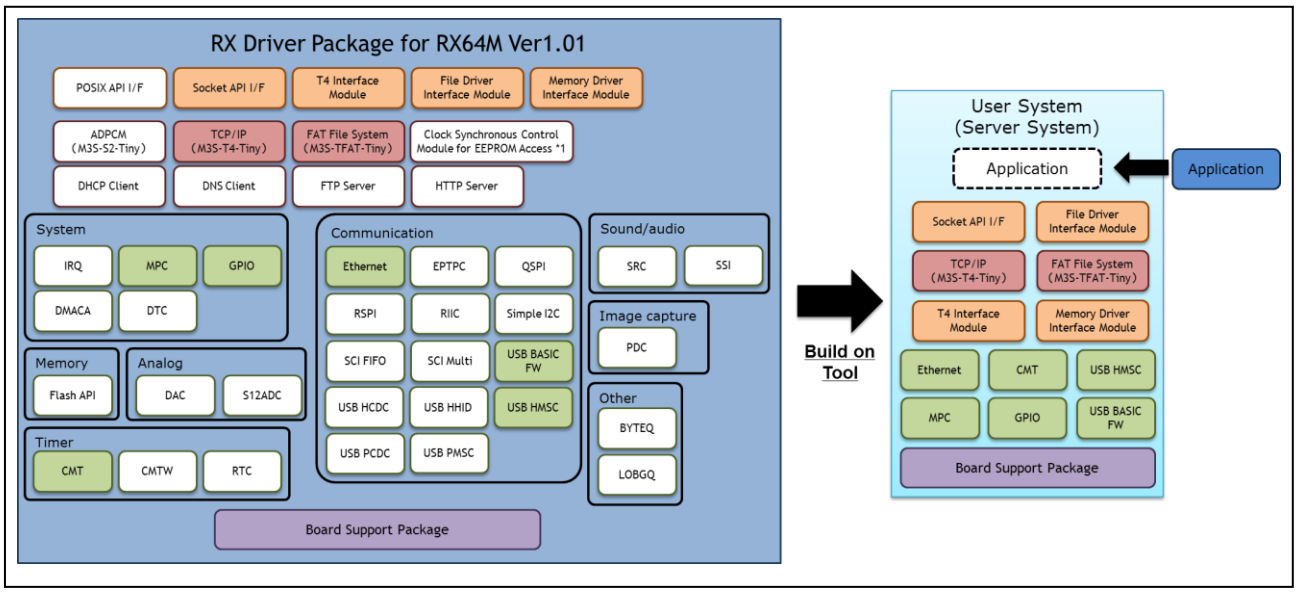


Figure 2.2.1 An example of system build

(b) Free to use

All the modules included in the RX Driver Package can be used free of charge. Free versions of middleware modules such as TCP/IP and file system are included.

(c) Can upgrade to paid versions of modules

The free versions of modules in the RX Driver Package can be replaced with commercial (paid) versions. By using a commercial (paid) version, all the functionality of the module will be available, as well as support about a commercial version.

(d) Check operation including user application

The RX Driver Package Application is provided as a sample user application that uses the RX Driver Package. The RX Driver Package Application consists of programs for operating each module in the RX Driver Package, and the project files for building the programs. It enables you to start checking the operation of your user application immediately.

3. Structure of the RX64M Group RX Driver Package

3.1 Folder Structure

The folder structure used in this package is shown below.

When the ZIP file for this package is downloaded from the Renesas web site and decompressed, a folder of the same name will be present and it will contain a FITModules folder, a reference_documents folder, and this document.

The FITModules folder contains the FIT modules for the modules shown in table 3.3.1 (as ZIP files and XML files).

The reference_documents folder contains the documentation for using this package in various development environments.

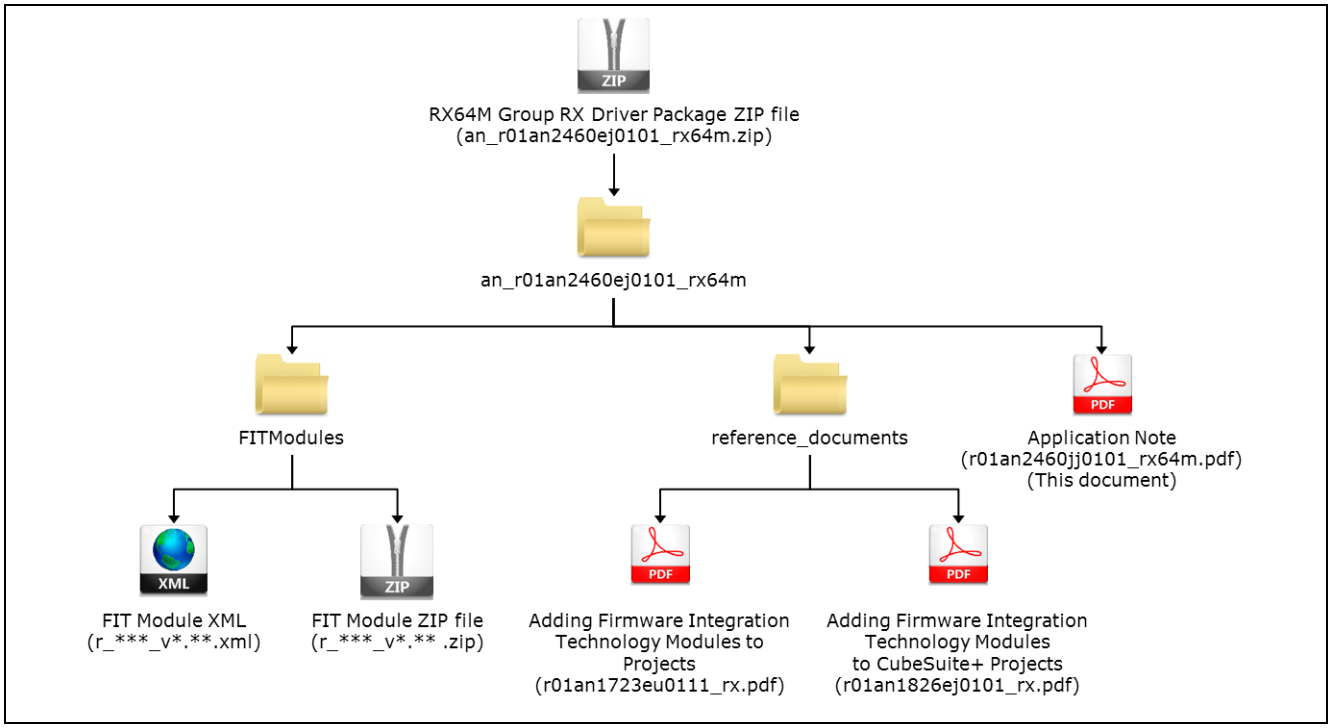


Figure 3.1.1 Folder Structure of the RX64M Group RX Driver Package

3.2 Module Structure

The figure below shows the types and structure of the FIT modules included in this package.

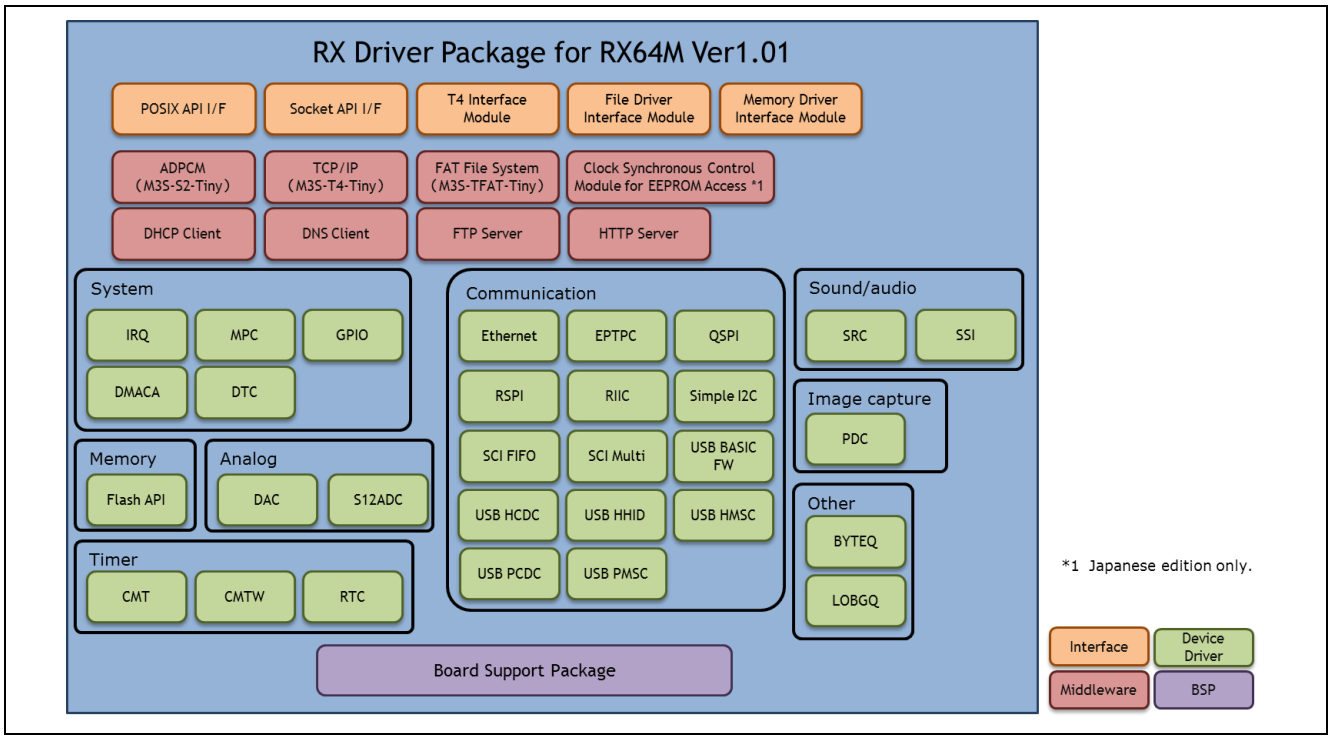


Figure 3.2.1 RX64M Group RX Driver Package FIT Module Structure

3.3 FIT Modules

The table below lists the FIT modules included in this package.

Table 3.3.1 RX64M Group RX Driver Package FIT Modules

Type	Module	FIT Module Name	Version
Board Support Package	Board support package	r_bsp	2.70
Device Driver	Interrupt Controller (IRQ)	r_irq_rx	1.30
Device Driver	Data transfer controller (DTC)	r_dtc_rx	2.01
Device Driver	DMA controller (DMAC)	r_dmaca_rx	1.01
Device Driver	General-purpose I/O	r_gpio_rx	1.30
Device Driver	Multi-function pin controller (MPC)	r_mpc_rx	1.30
Device Driver	Compare match timer (CMT)	r_cmt_rx	2.40
Device Driver	Compare Match Timer W (CMTW)	r_cmtw_rx	1.00
Device Driver	Real-Time Clock (RTC)	r_rtc_rx	2.10
Device Driver	Serial Communications Interface with FIFO (SCI: Asynchronous/Clock Synchronous)	r_scif_rx	1.00
Device Driver	Serial Communications Interface (SCI: Simple I2C Bus)	r_sci_rx64m	1.00
Device Driver	I2C Bus Interface (RIIC)	r_riic_rx	1.05
Device Driver	Serial Peripheral Interface (RSPI: Device Driver for Serial Memory Control)	r_rspi_smstr_rx	1.07

Type	Module	FIT Module Name	Version
Device Driver	Quad Serial Peripheral Interface (QSPI: Device Driver for Serial Memory Control)	r_qspi_smstr_rx	1.06
Device Driver	USB basic firmware	r_usb_basic	1.00
Device Driver	USB host communication device class	r_usb_hcdc	1.00
Device Driver	USB host human interface device class	r_usb_hhid	1.00
Device Driver	USB host mass storage class	r_usb_hmsc	1.00
Device Driver	USB peripheral communication device class	r_usb_pcdc	1.00
Device Driver	USB peripheral mass storage class	r_usb_pmsc	1.00
Device Driver	Ethernet controller PTP controller (EPTPC)	r_ptp_api_rx	1.01
Device Driver	Ethernet controller (ETHERC)	r_ether_rx	1.00
Device Driver	12-Bit A/D Converter (S12AD)	r_s12ad_rx64m	1.00
Device Driver	D/A Converter (DAC)	r_dac_rx	2.10
Device Driver	Flash Memory (Flash API)	r_flash_rx	1.11
Device Driver	Sampling rate converter (SRC)	r_src_api_rx	1.10
Device Driver	Serial Sound Interface (SSI)	r_ssi_api_rx	1.00
Device Driver	Parallel Data Capture Unit (PDC)	r_pdc_rx	1.00
Device Driver	Byte Queue Buffer (Data Management)	r_byteq	1.20
Device Driver	Long Queue Buffer (Data Management)	r_longq	1.20
Middleware	M3S-S2-Tiny (ADPCM encoding/decoding library)	r_s2_rx	3.01
Middleware	M3S-T4-Tiny (TCP/IP protocol stack library)	r_t4_rx	2.01
Middleware	M3S-TFAT-Tiny (FAT file system)	r_tfat_rx	3.00
Middleware	DHCP Client Module	r_t4_dhcp_client_rx	1.01
Middleware	DNS Client Module	r_t4_dns_client_rx	1.01
Middleware	FTP Server Module	r_t4_ftp_server_rx	1.02
Middleware	HTTP Server Module	r_t4_http_server_rx	1.03
Interface	POSIX Wrapper	r_posix	1.00
Interface	Socket API Module for M3S-T4-Tiny	r_socket	1.20
Interface	Interface Conversion for Ethernet Controller Driver Module for M3S-T4-Tiny	r_t4_driver_rx64m	1.01
Interface	File driver for FTP server and Web server Module	r_t4_file_driver_rx	1.00
Interface	M3S-TFAT-Tiny Memory Driver Interface Module	r_tfat_driver_rx	1.00

Note: This package includes the M3S-T4-Tiny (TCP/IP protocol stack library) of evaluation version. For the commercial version, please go to the below URL.
http://www.renesas.com/mw/t4

4. Usage Procedures

The RX Driver Package allows programs to be easily constructed by using the FIT plugin included in e^2 studio. The remainder of this section presents a simple usage example using e^2 studio. To use CS+, see the document "RX Family Adding Firmware Integration Technology Modules to CS+ Projects (R01AN1826EJ)" included in this package.

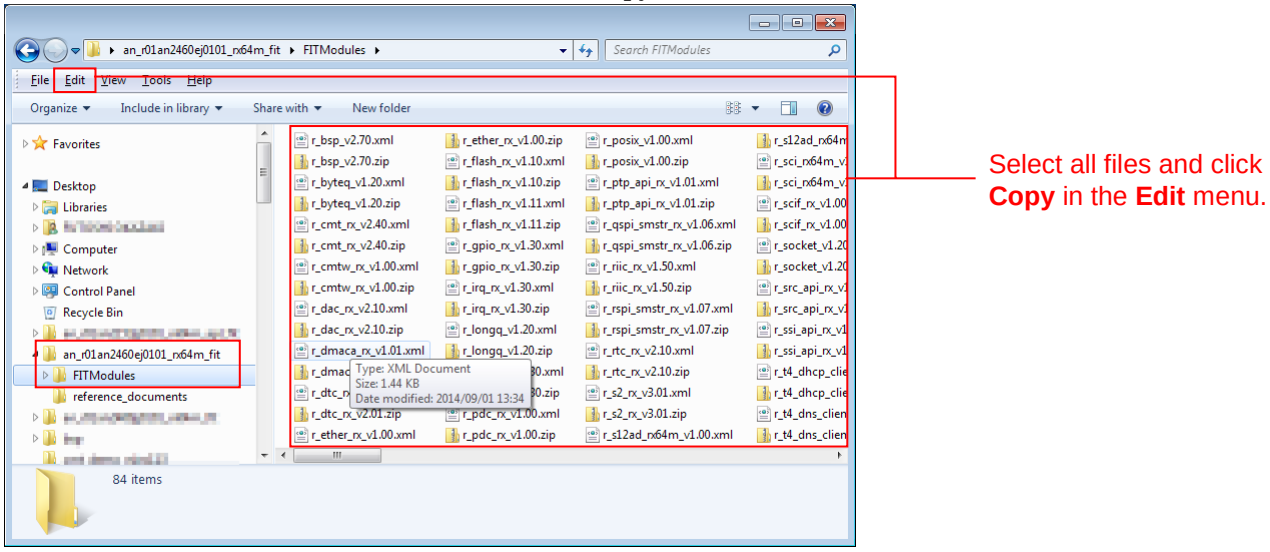
4.1 Environment Used

The RX64M is used as the target microcontroller and the Renesas Starter Kit+ RX64M is used as the target board. If a different environment is used, replace the specifics used in the example with the ones for that environment as you read.

4.2 Install RX Driver Package in e^2 studio

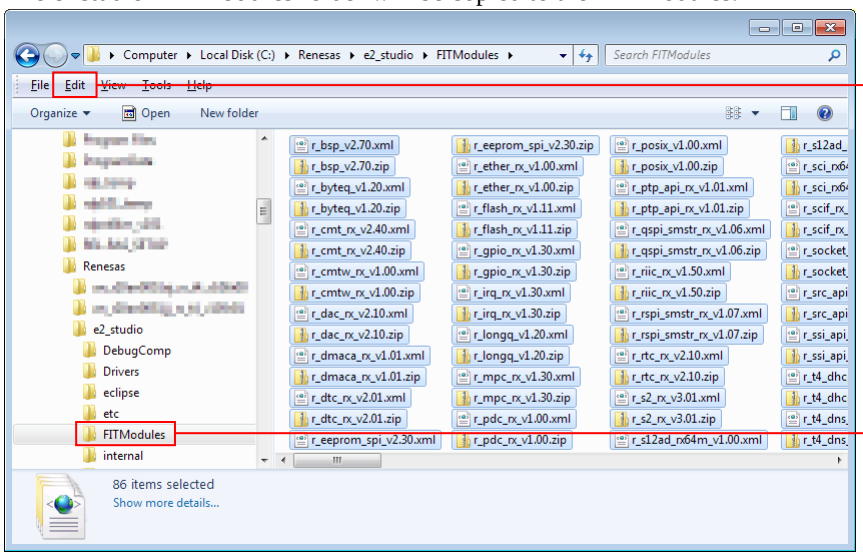
Install the FIT modules in the RX Driver Package into e^2 studio.

1. Decompress the downloaded file **an_r01an2460ej0101_rx64m.zip** into an arbitrary directory.
2. Open the folder that was decompressed and open the **FITModules** folder in that folder.
3. Select all the files in the **FITModules** folder and click **Copy** in the **Edit** menu.



4. Open the e^2 studio install folder (Usually, this will be c:/Renesas/e2_studio.) and open the **FITModules** folder in that folder.

5. Click **Paste** on the **Edit** menu.
   The e^2 studio **FITModules** folder will be copied to the FIT modules.



4.3 Application Creation

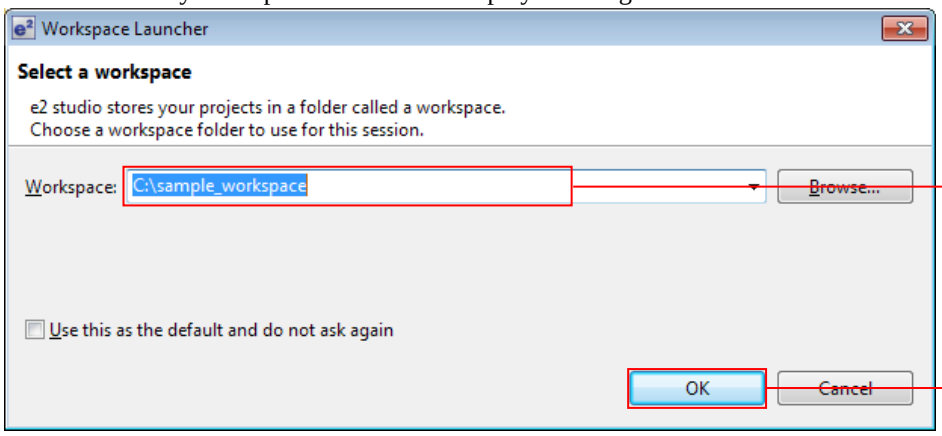
In this section, create a simple application that drives an LED.

4.3.1 Create a Workspace and a Project

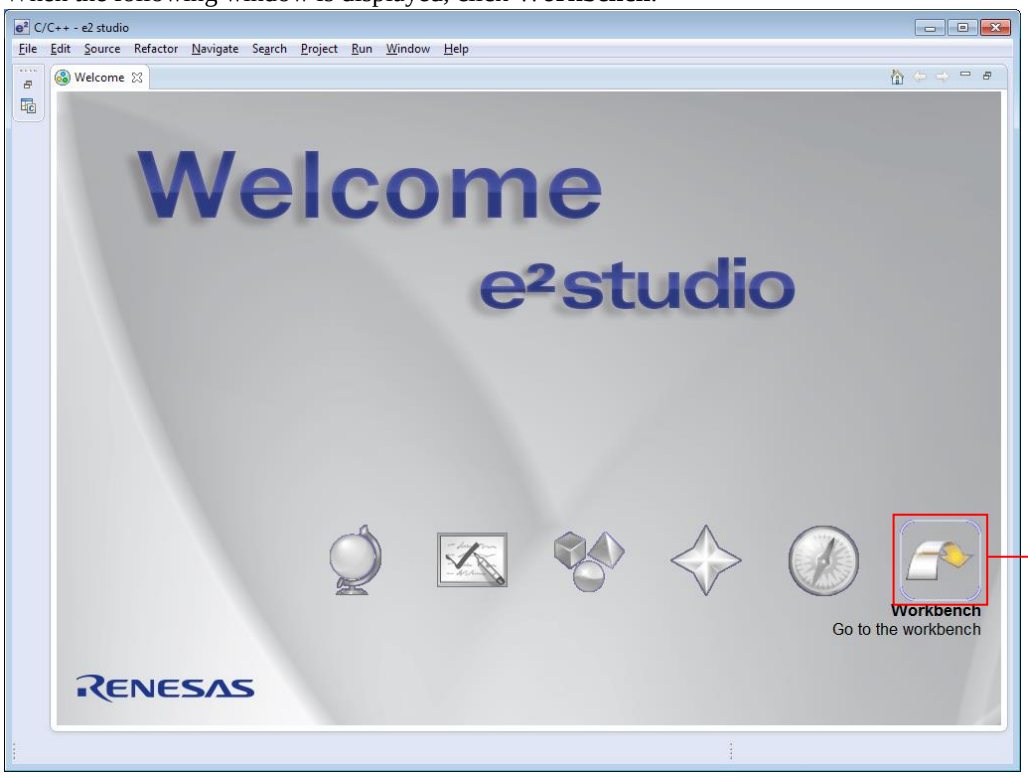
First, create a new workspace and a new project.

1. Start e^2 studio.

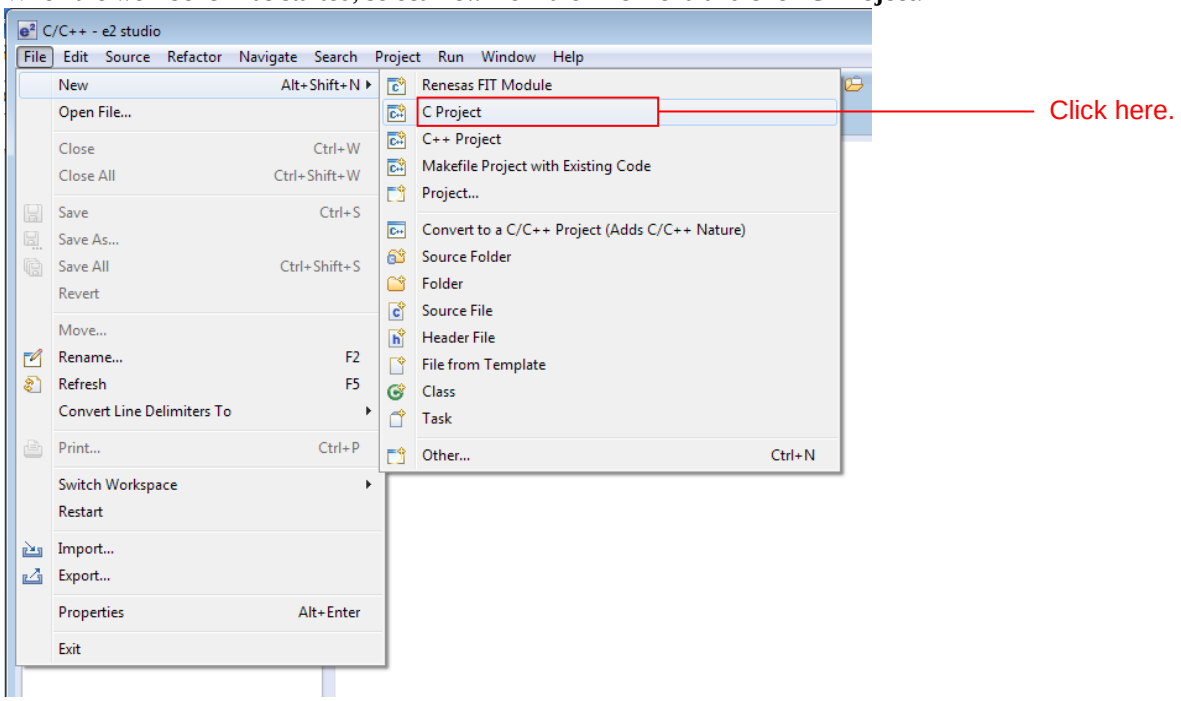
2. Enter an arbitrary workspace folder in the displayed dialog box and click **OK**.



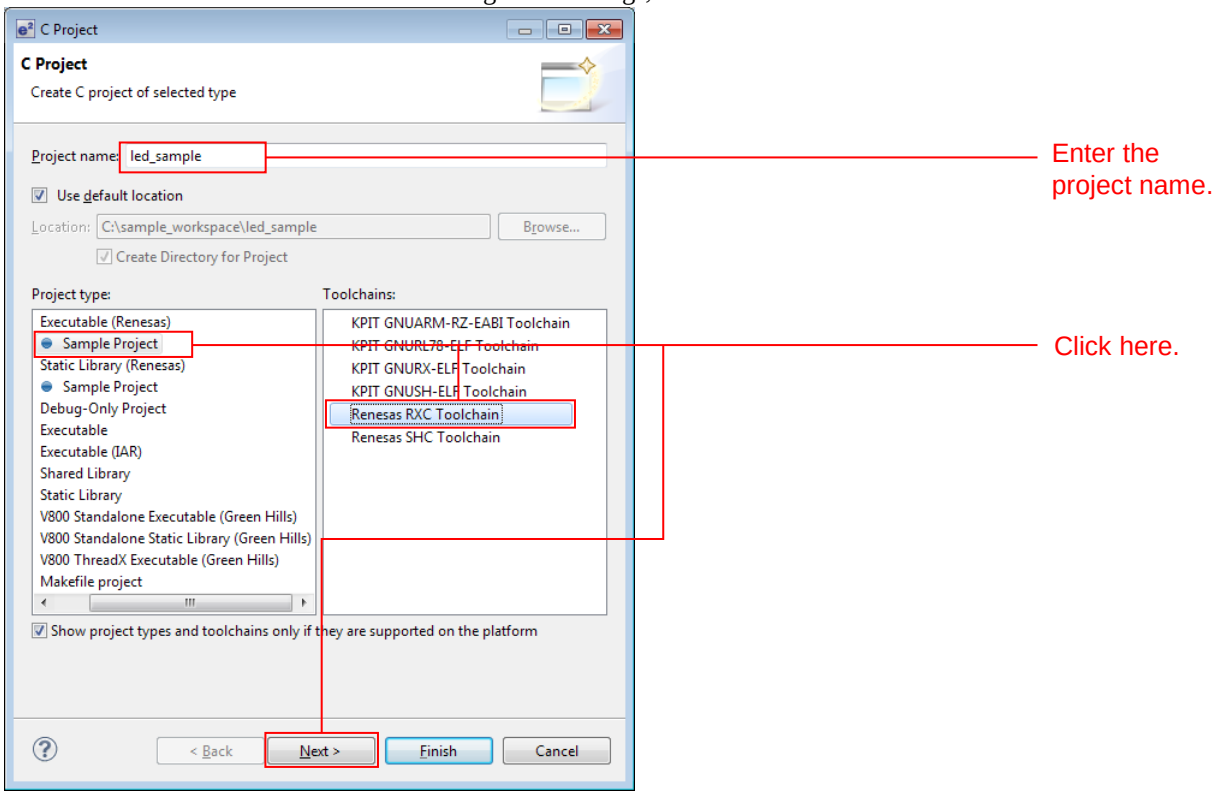
3. When the following window is displayed, click **Workbench**.



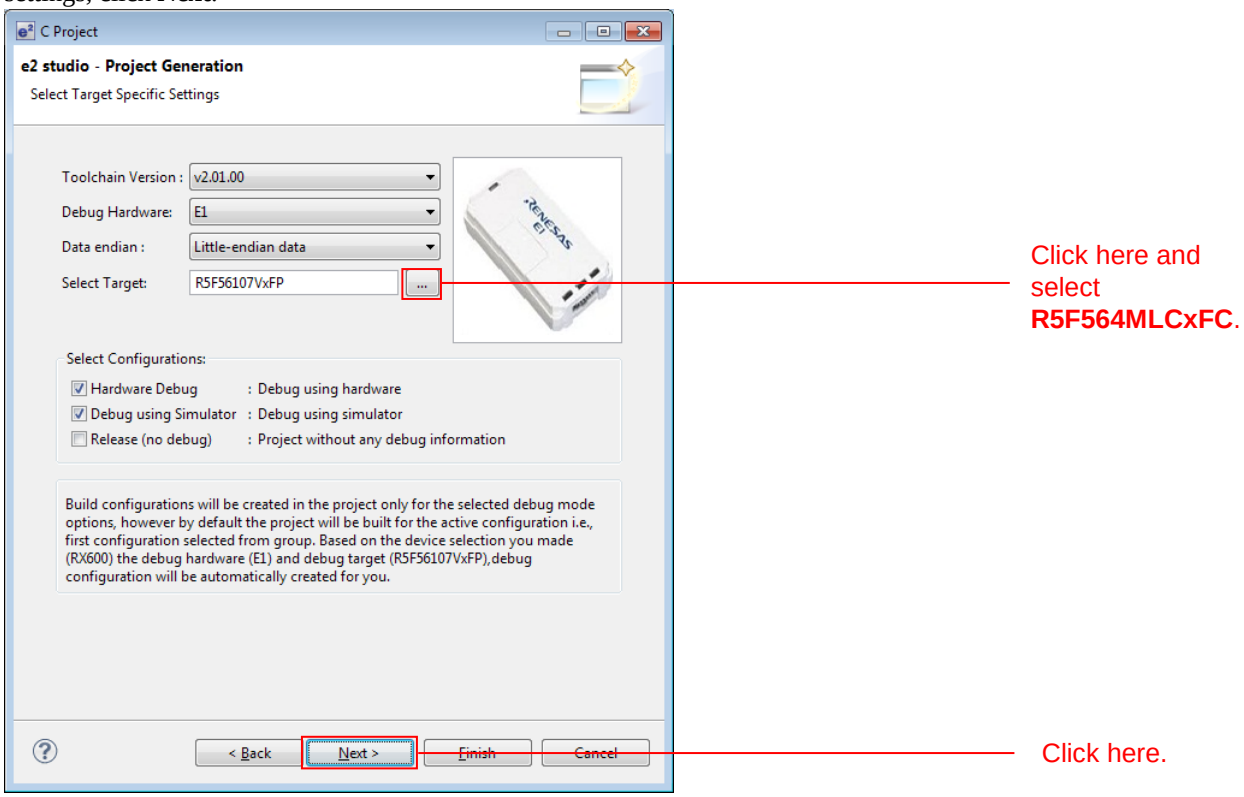
4. When the workbench has started, select **New** from the **File** menu and click **C Project**.



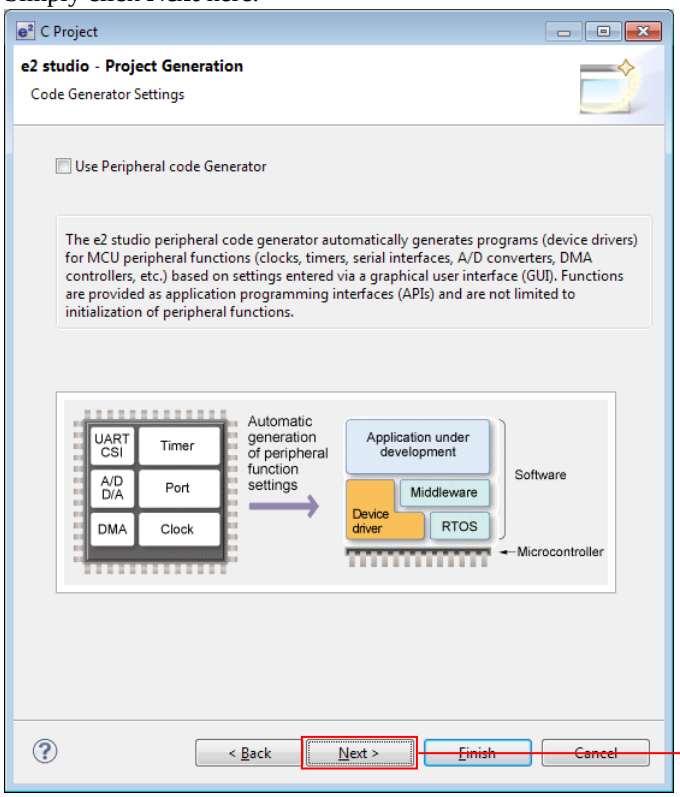
5. Enter the project name. For the project type, click **Sample Project** under **Executable (Renesas)**. For the tool chain, click **Renesas RXC Toolchain**. After making these settings, click **Next**.



6. Select the target. Click the "..." button under **Target Selection** and select **R5F564MLCxFC**. After making these settings, click **Next**.

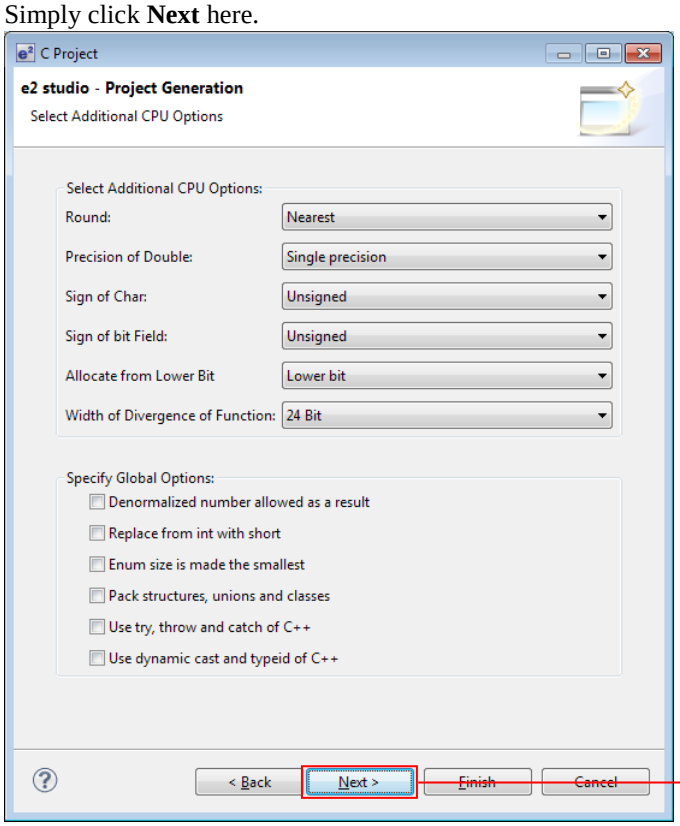


7. Simply click **Next** here.



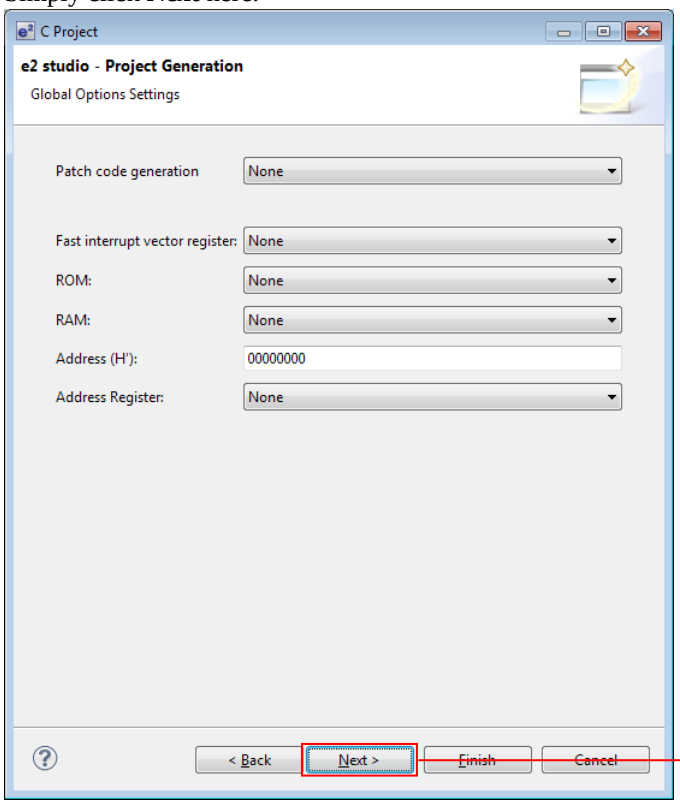
Click here.

8. Simply click **Next** here.

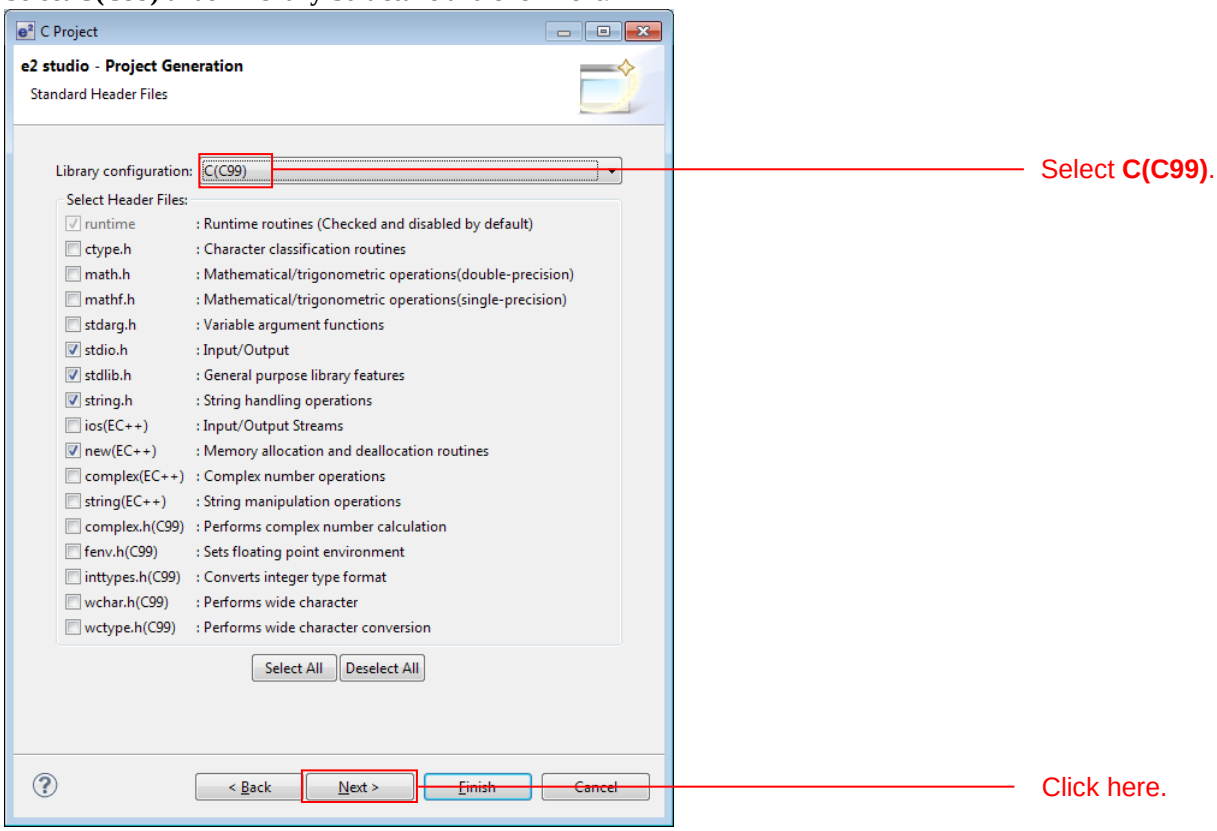


Click here.

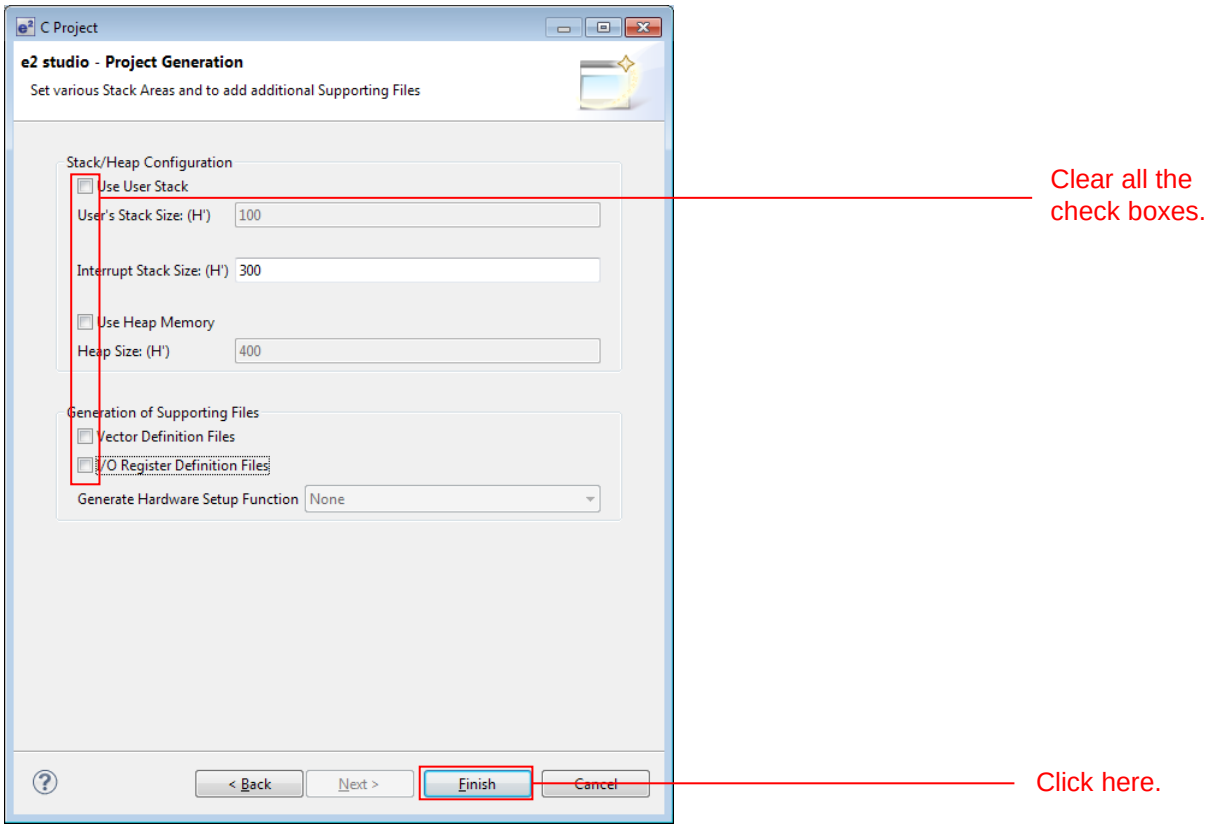
9. Simply click **Next** here.



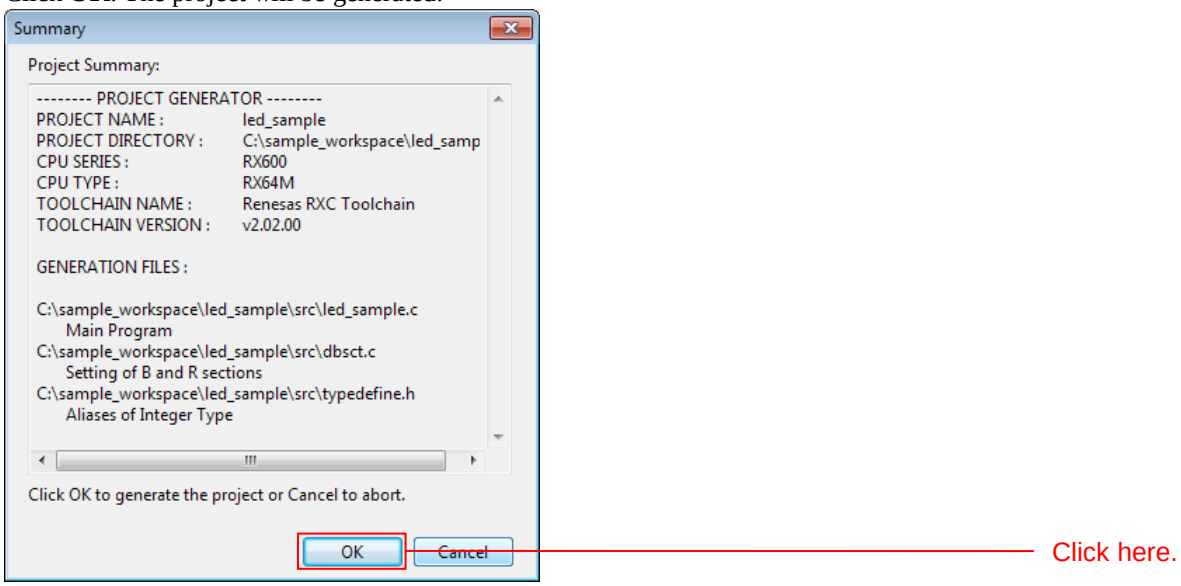
10. Select **C(C99)** under **Library Structure** and click **Next**.



11. Clear all four check boxes and click **Finish**.



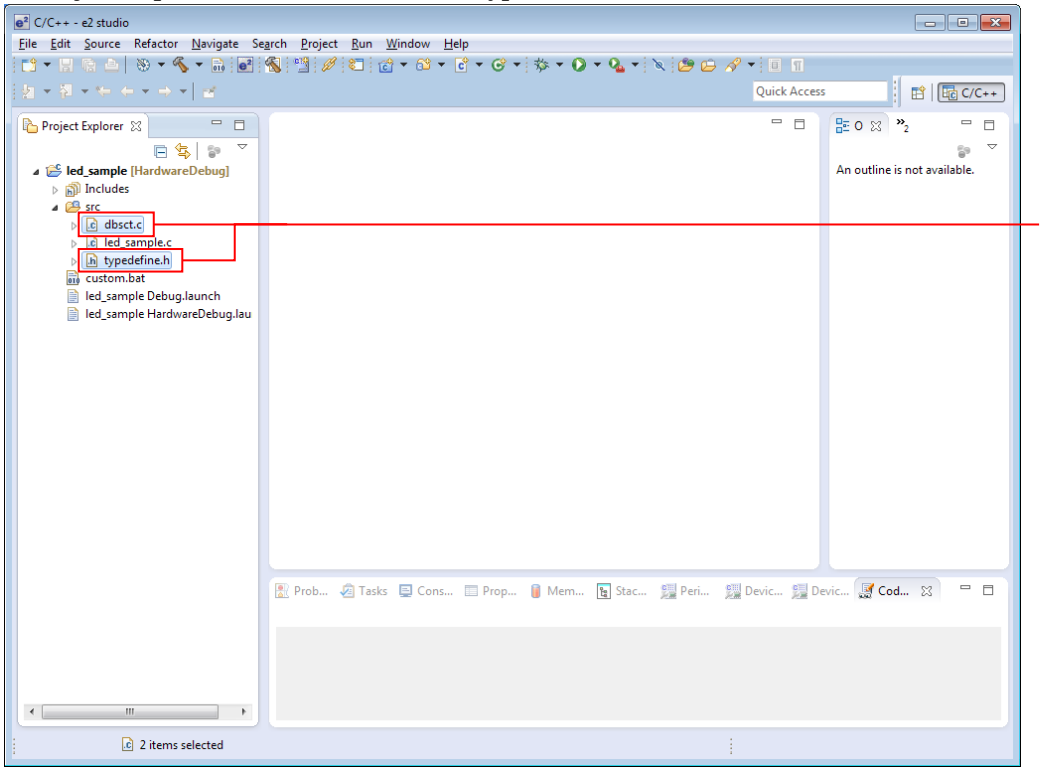
12. Click **OK**. The project will be generated.



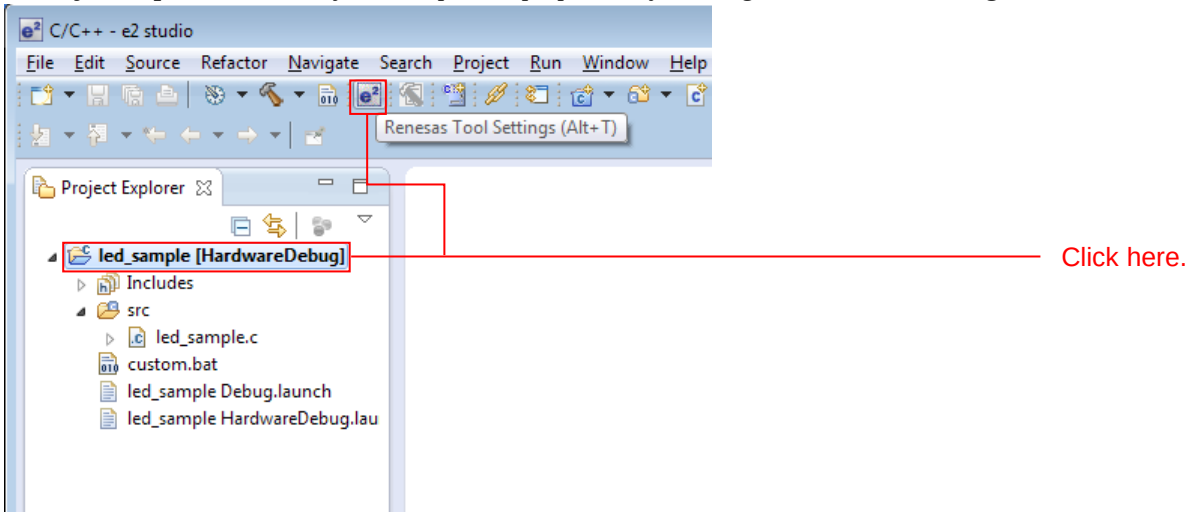
4.3.2 Prepare to Embed the FIT Modules

Since the FIT modules are independently initialized in BSP, it is necessary to modify certain aspects of the project generated by e^2 studio.

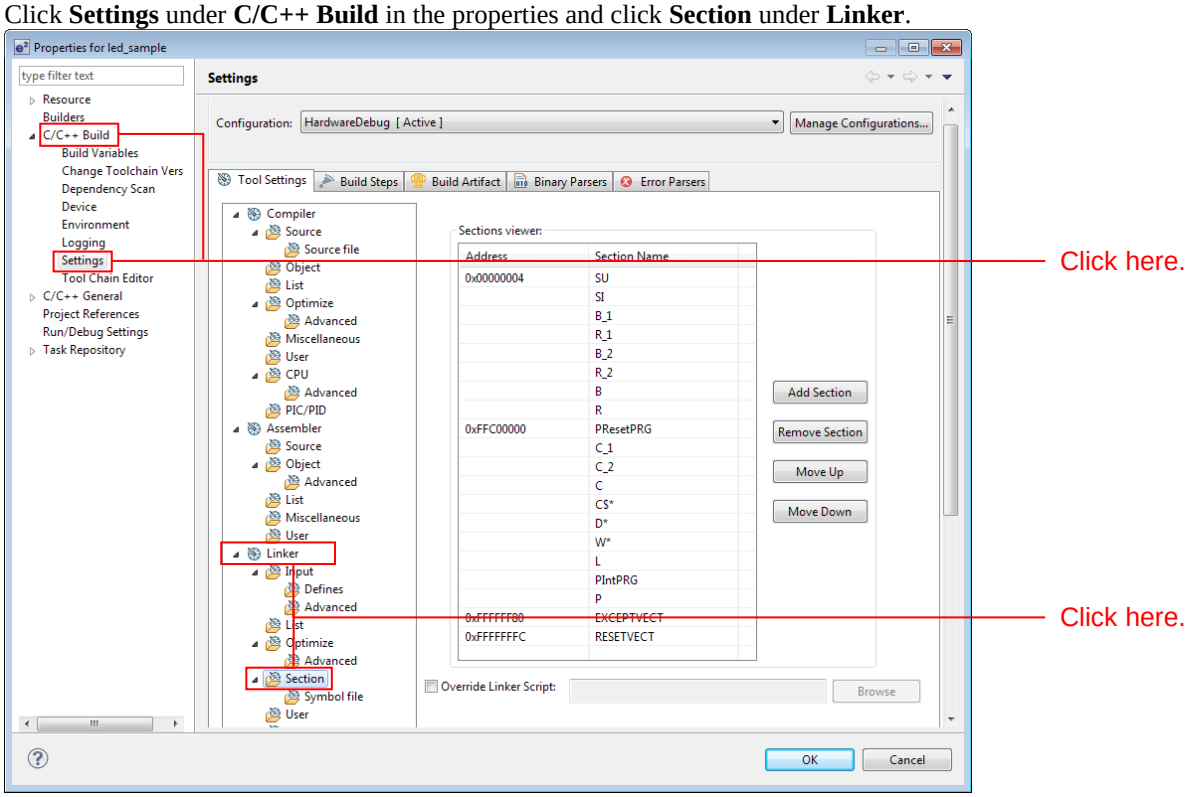
1. In **Project Explorer**, delete the **dbsct.c** and **typedefine.h** files in the **src** folder.



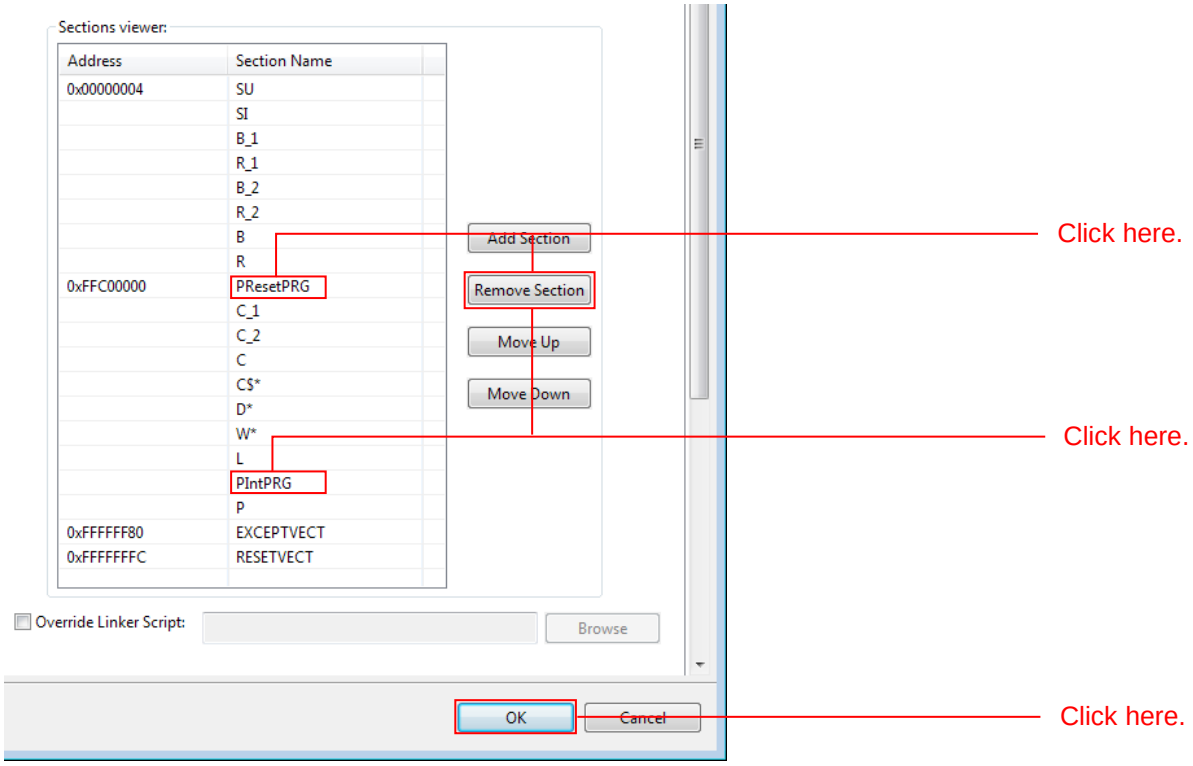
2. In **Project Explorer**, click Project and open the properties by clicking **Renesas Tool Settings** in the toolbar.



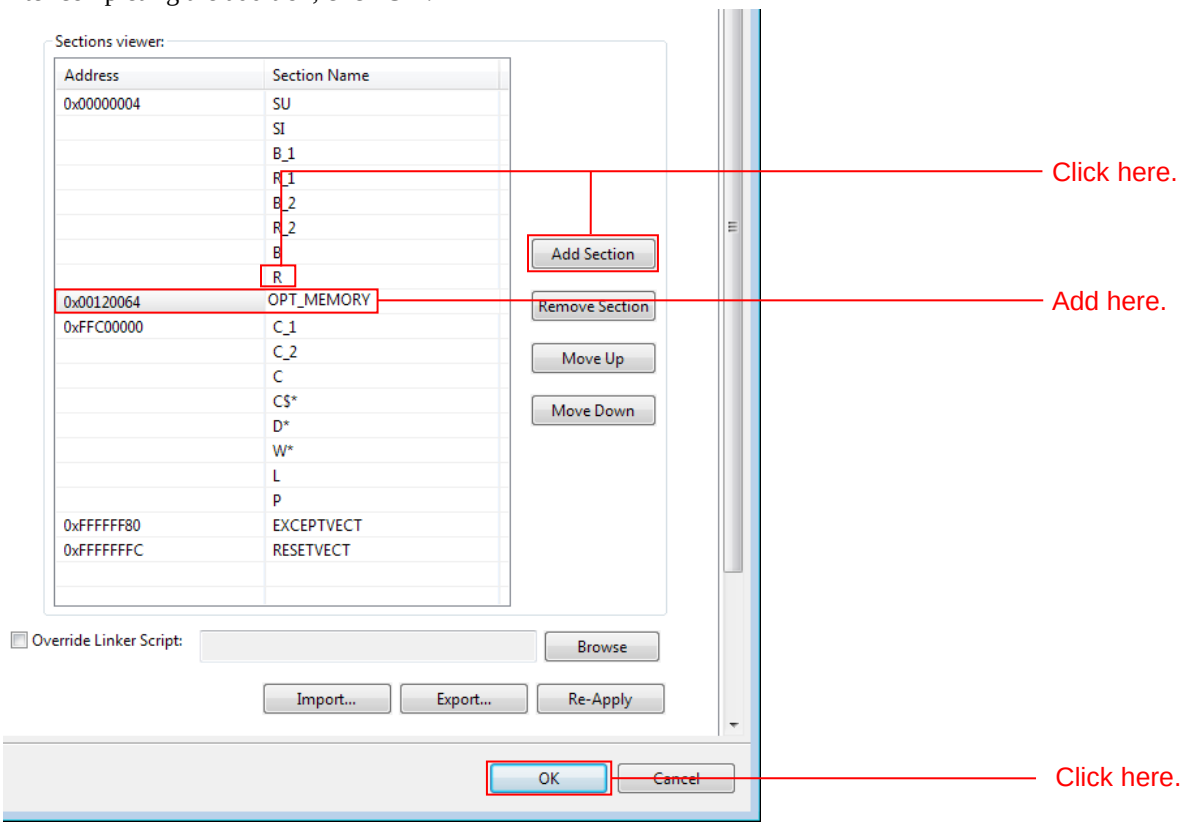
3. Click **Settings** under **C/C++ Build** in the properties and click **Section** under **Linker**.



4. In **Section View**, delete **PResetPRG** and **PIntPRG**. Click each section and click **Delete Section**.



5. Add an **OPT_MEMORY** section after the **R** section. Click the **R** section and click **Add Section**. Click the added section, **NEW_SECTION_1**, and modify the name to be **OPT_MEMORY**, and input the address to be **0x00120064**.
   After completing the addition, click **OK**.

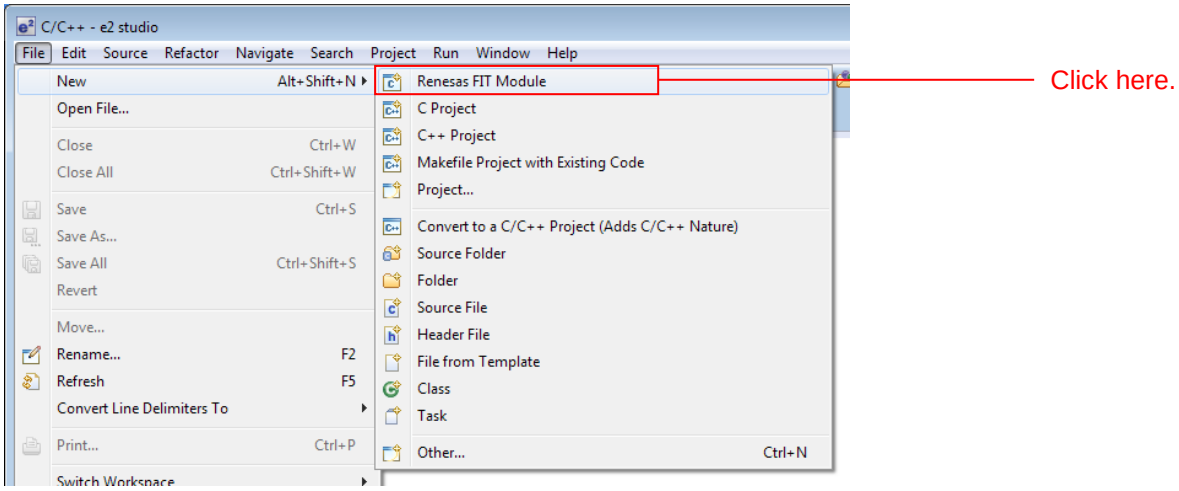


4.3.3 Install the FIT Modules with the FIT Plugin.

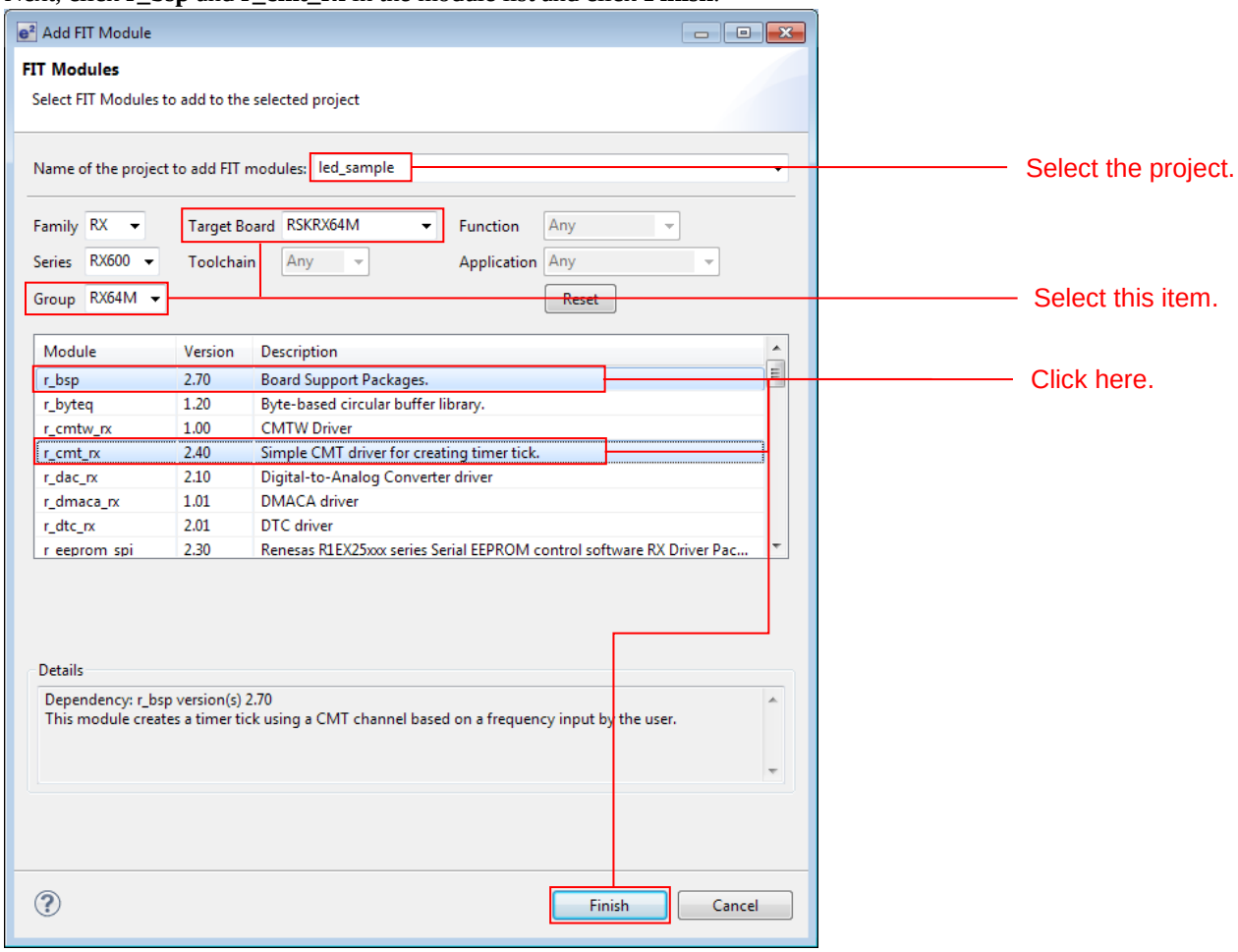
Install the required modules with the FIT plugin into the created project.

Here, install the BSP module (r_bsp) and the compare match timer driver (r_cmt_rx).

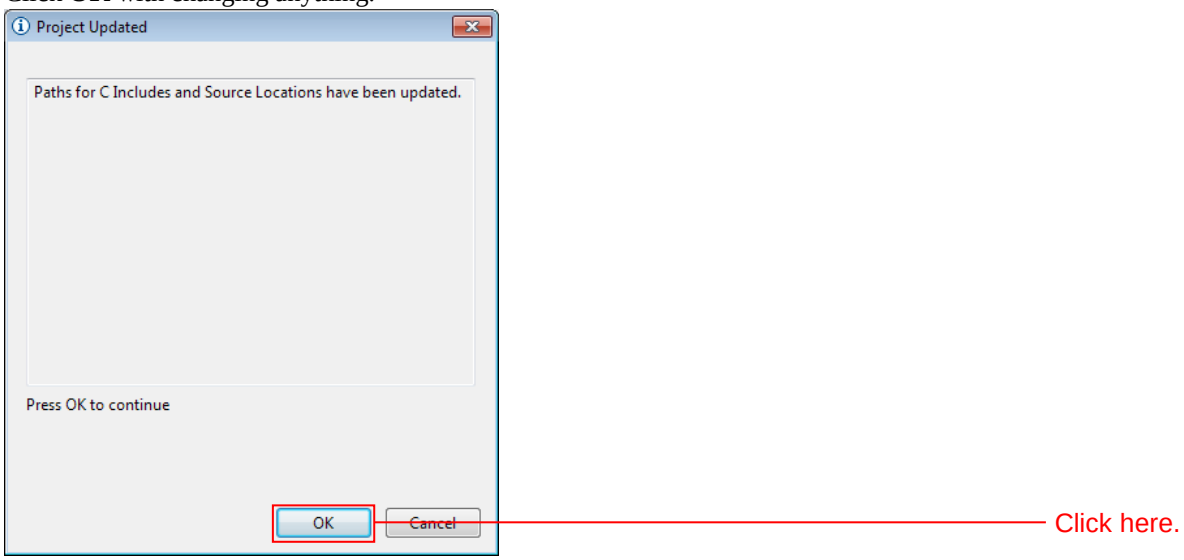
1. In the **File** menu, select **New** and click **Renesas FIT Module**.



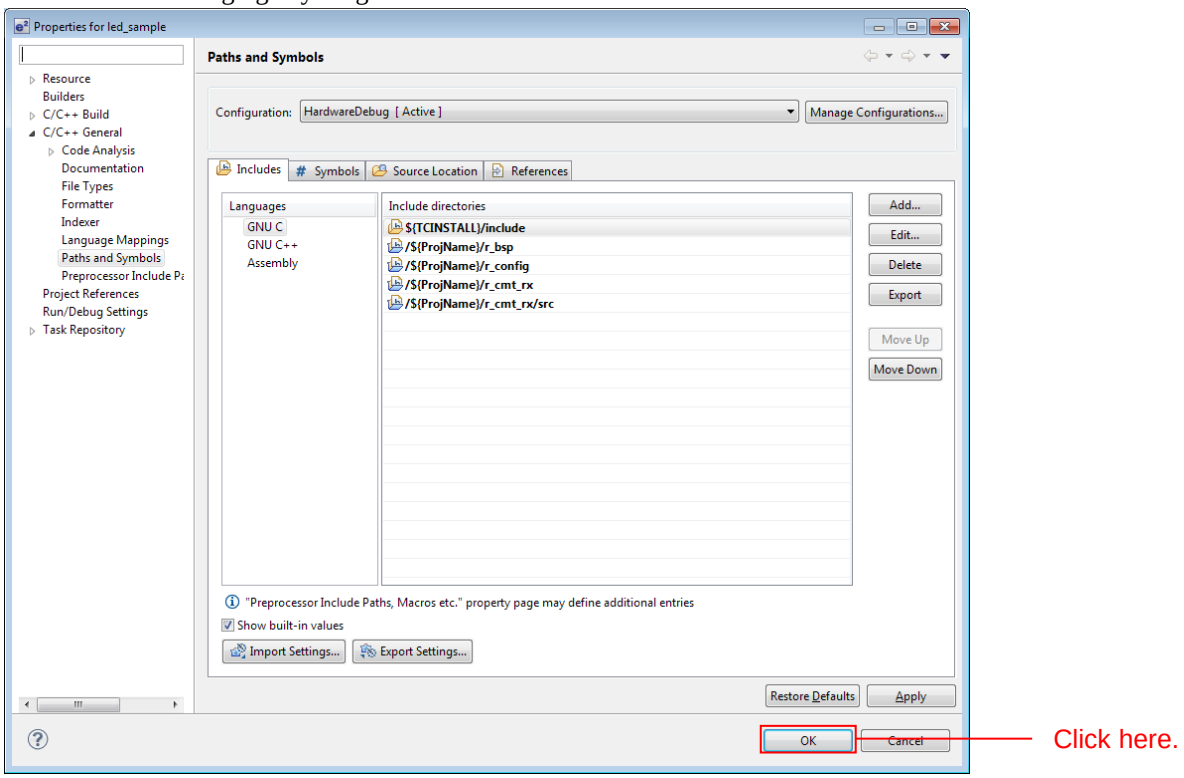
2. Select the created project with **Project to Add FIT Module** to.
   Next, select **RX64M** under **Group** and select **RSKRX64M** from **Target Board**.
   Next, click **r_bsp** and **r_cmt_rx** in the module list and click **Finish**.



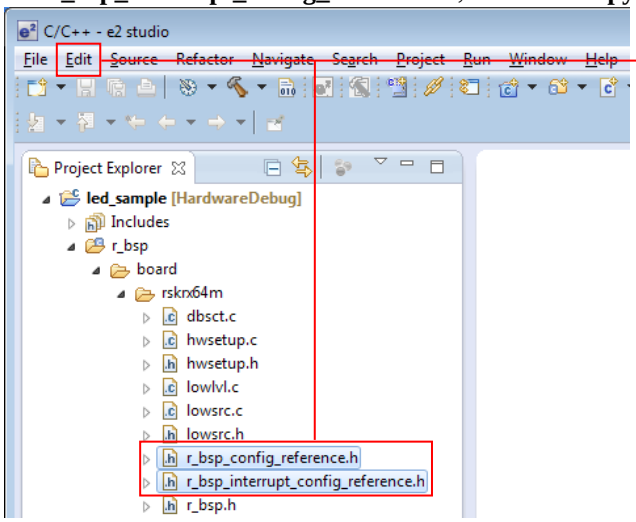
3. Click **OK** with changing anything.



4. Click **OK** with changing anything.

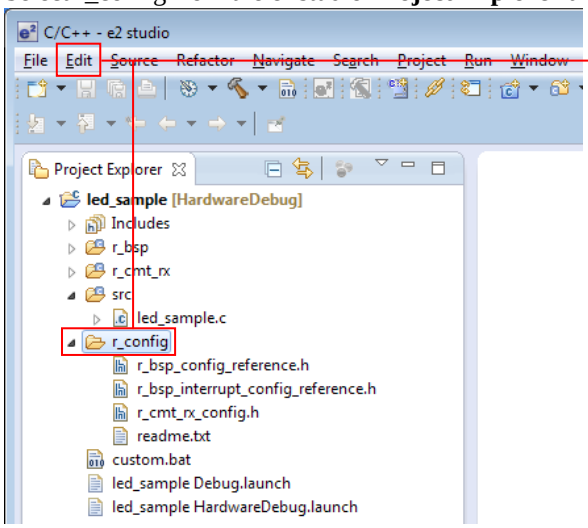


5. Open **r_bsp/board/rskrx64m** from the e^2 studio Project Explorer, select the two files **r_bsp_config_reference.h** and **r_bsp_interrupt_config_reference.h**, and click **Copy** on the **Edit** menu.



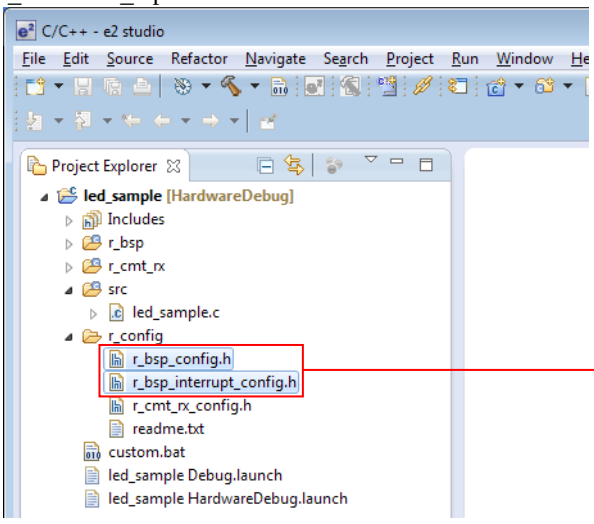
Select these two files and click **Copy** on the **Edit** menu.

6. Select r_config from the e^2 studio **Project Explorer** and click **Paste** on the **Edit** menu.



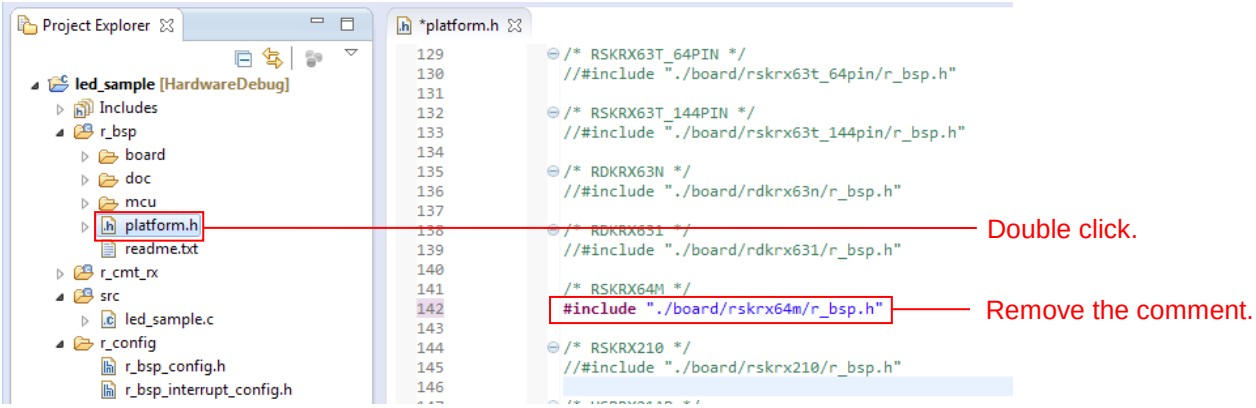
Select the **r_config** folder and click **Paste** on the **Edit** menu.

7. Change the names of the two copied files to **r_bsp_config.h** and **r_bsp_interrupt_config.h**. That is, delete the "_reference_" part of the file names.



Rename these files.

8. Modify platform.h to correspond to the target board used.
   Double click **r_bsp/platform.h** from the e^2 studio **Project Explorer** and, in the editor, remove the comment from the include line for the r_bsp.h file for the RSKRX64M.



4.3.4 Create an LED Driving Program

Create a program that toggles the LED0 on/off state every 0.5 seconds using the compare match timer.

Open the file src/led_sample.c and modify it as shown below.

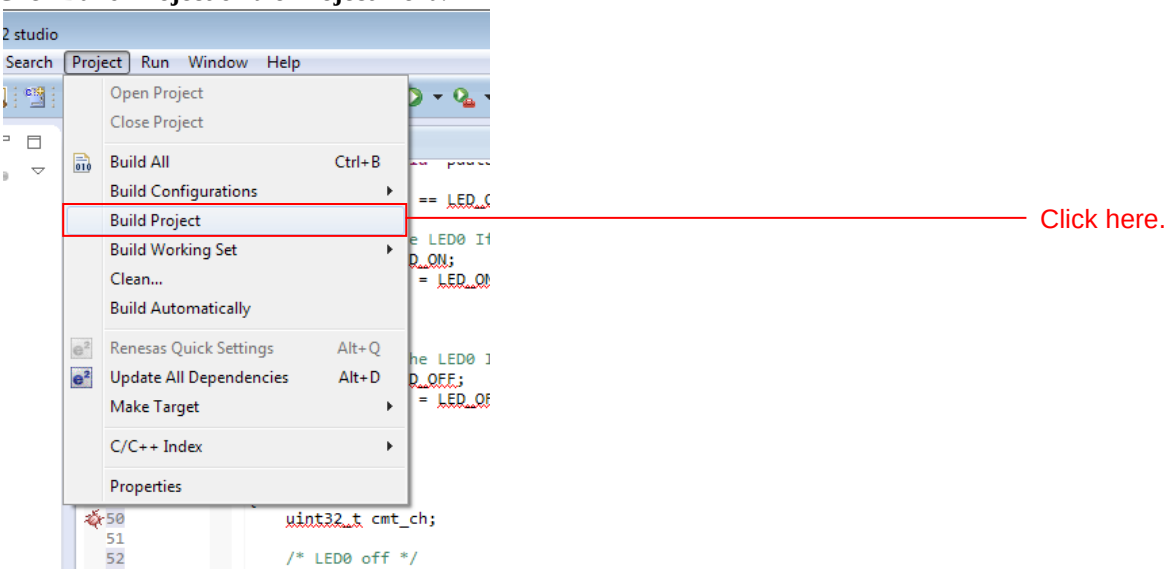
src/led_sample.c

```
/***********************************************************************/
/*                                                                     */
/*  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);
}
```

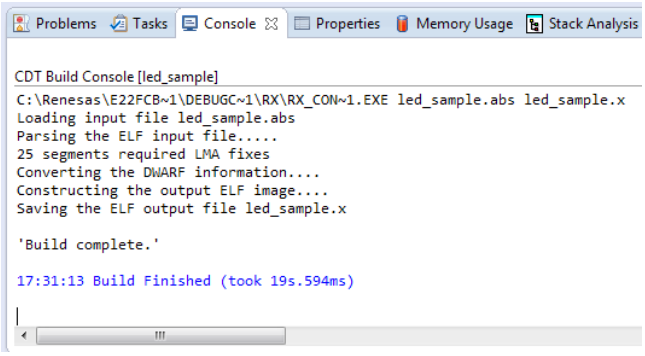
4.3.5 Build and Try Running the Program

Build the program just created and verify that it runs.

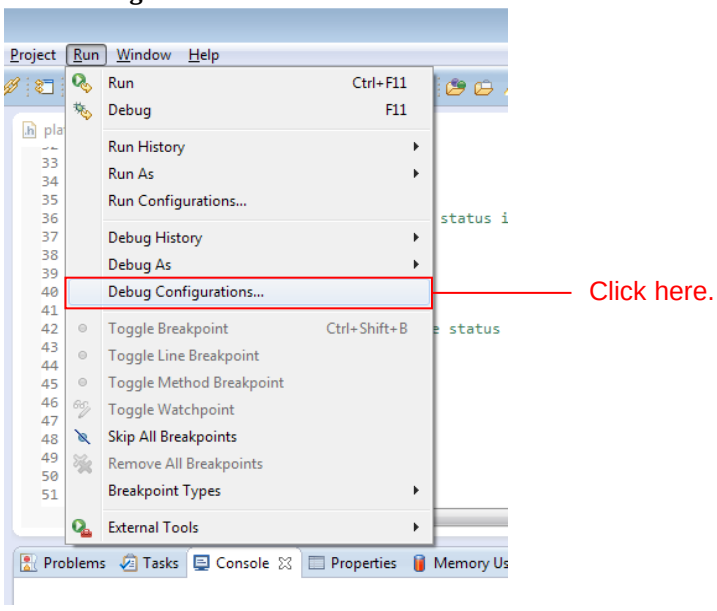
1. Click **Build Project** on the **Project** menu.



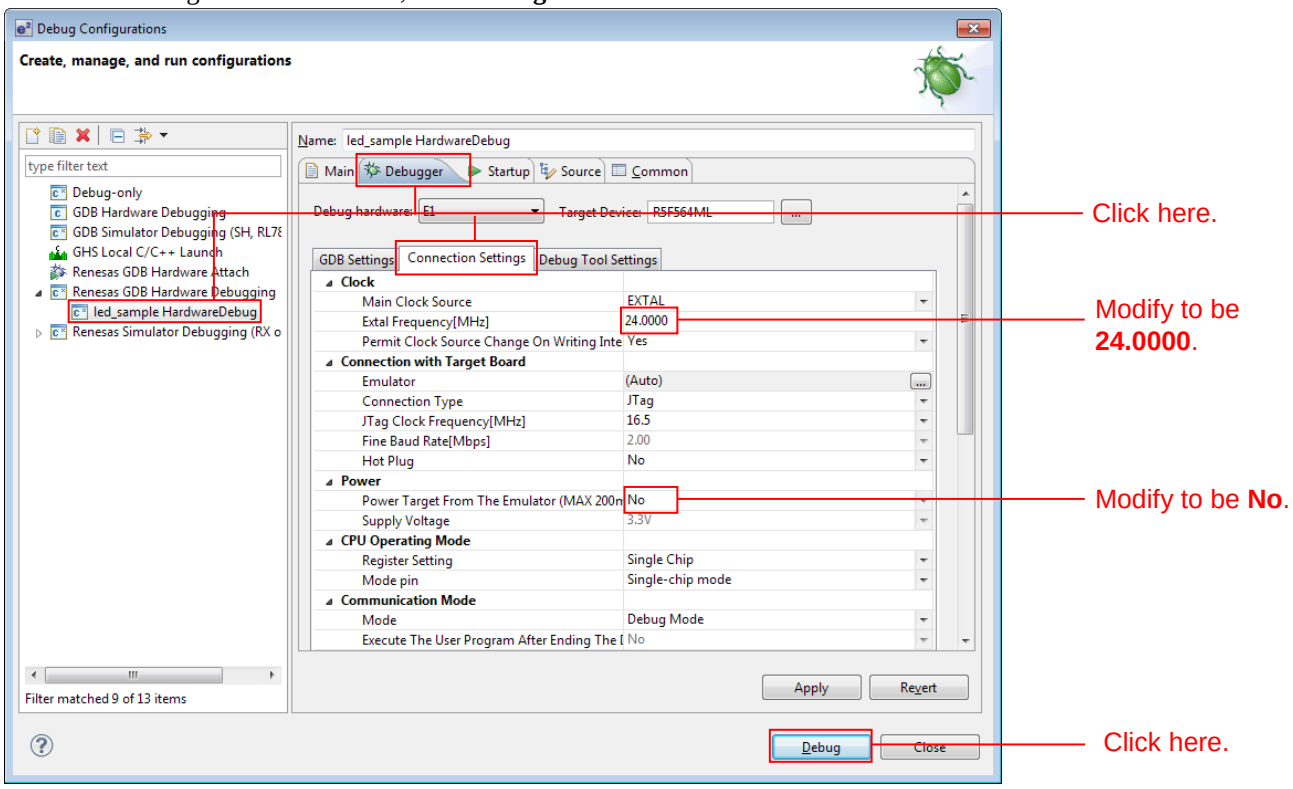
2. When the build completes, the following will be displayed in **Console** view.



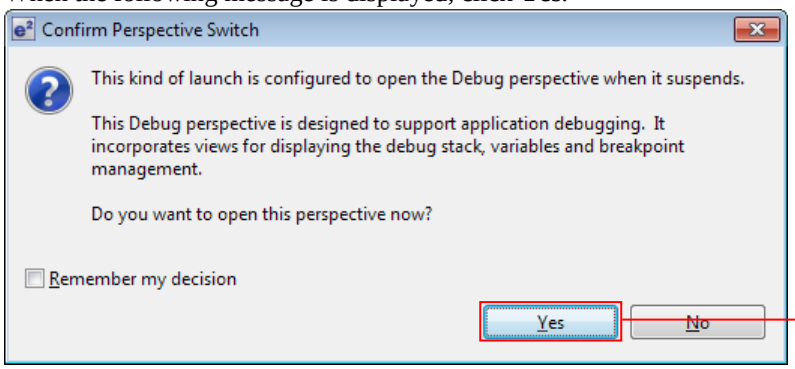
3. Click **Debug Build** on the **Run** menu.



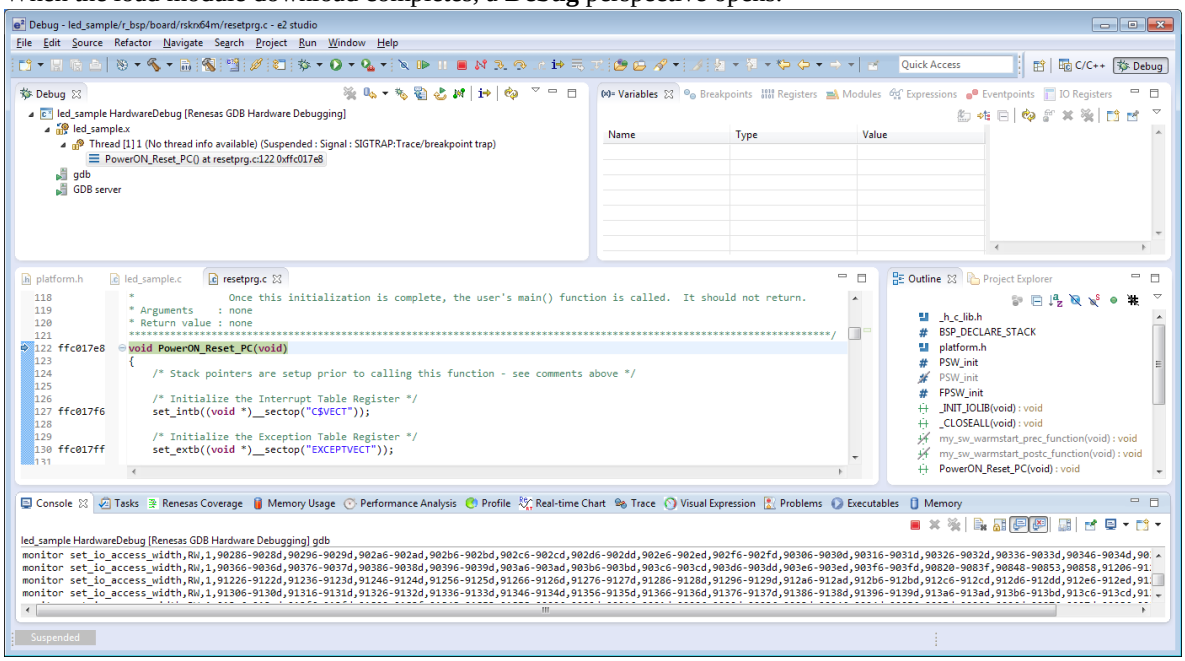
4. Click **led_sample HardwareDebug** under the **Renesas GDB Hardware Debugging**. Click the **Debugger** tab and click **Connection Setting**.
   Modify **EXTAL Frequency** to be **24.0000** and change **Provide Power from Emulator** to **No**.
   When these changes have been made, click **Debug**.



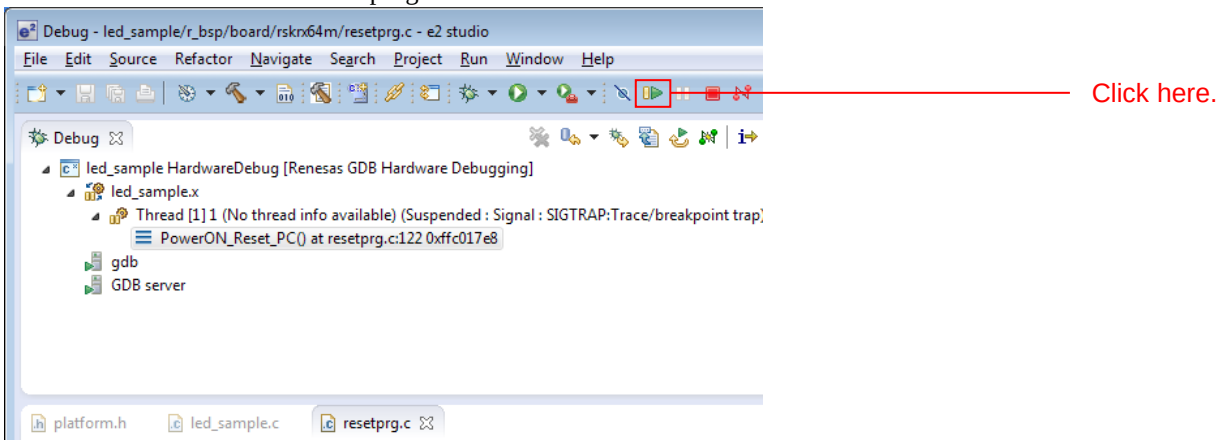
5. When the following message is displayed, click **Yes**.



6. When the load module download completes, a **Debug** perspective opens.



7. Click **Restart** on the toolbar. The program will be executed and a break will occur at the start of the main function.



8. After the break at the start of the main function, click **Restart** on the tool bar again.
The project will be run and the program will iterate toggling LED0 with a period of 0.5 seconds.

5. RX Driver Package Application

5.1 RX Driver Package Application Structure

The RX Driver Package Application is a sample application program provided so that users can use the RX Driver Package easily. The RX Driver Package Application consists of an application program that operates using device drivers and middleware included in the RX Driver Package and a project file for building that application. This allows users to start evaluation quickly.

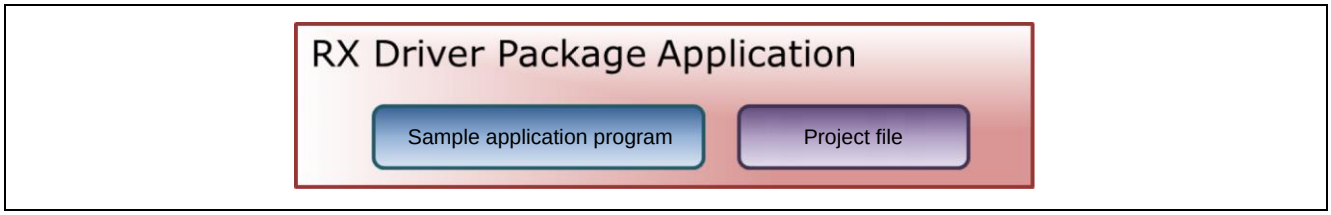


Figure 5.1.1 RX Driver Package Application Structure

Renesas plans to release a variety of types of this RX Driver Package Application in the future, such as system programs that operate using a combination of multiple drivers and middleware and evaluation programs for independent modules from the RX Driver Package.

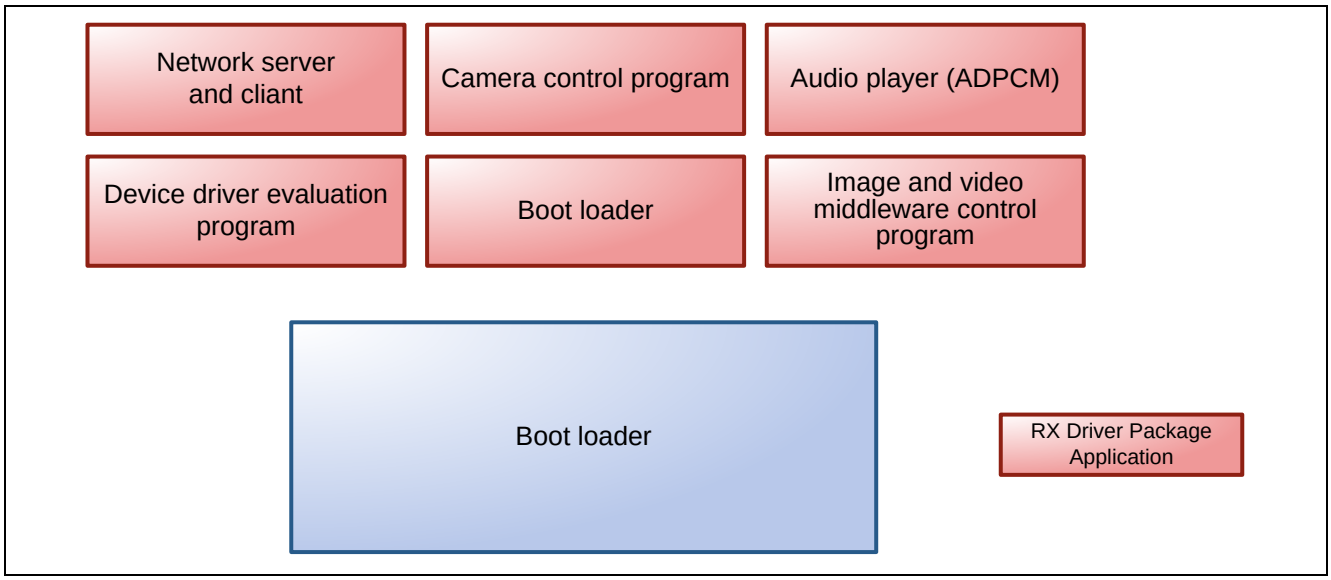


Figure 5.1.2 Types of RX Driver Package Application

5.2 RX Driver Package Application Features

The RX Driver Package Application has the following features.

- The RX Driver Package Application is evaluated in combination with the RX Driver Package.
- Project files are included in the RX Driver Package Application. Since both build and debug configurations for the application project are already set up in the provided project file, the user can quickly begin building and evaluating simply by importing the project into a workspace.
- If e^2 studio is used as the integrated development environment (IDE), the device drivers and middleware used in the application project can be automatically added to the project by using the FIT plugin provided with e^2 studio.
- Renesas provides the RX Driver Package Application without charge.

5.3 RX Driver Package Application Usage Example (when e² studio is used)

The device drivers and middleware required by the RX Driver Package Application are automatically added to the project by the FIT plugin, which comes with the e² studio.

After the project provided with the RX Driver Package Application has been installed in an e² studio workspace, the required device drivers and middleware from the RX Driver Package are also installed simply by installing in the project by selecting the RX Driver Package Application with the FIT plugin. Therefore all that remains is to build the project and start evaluation.

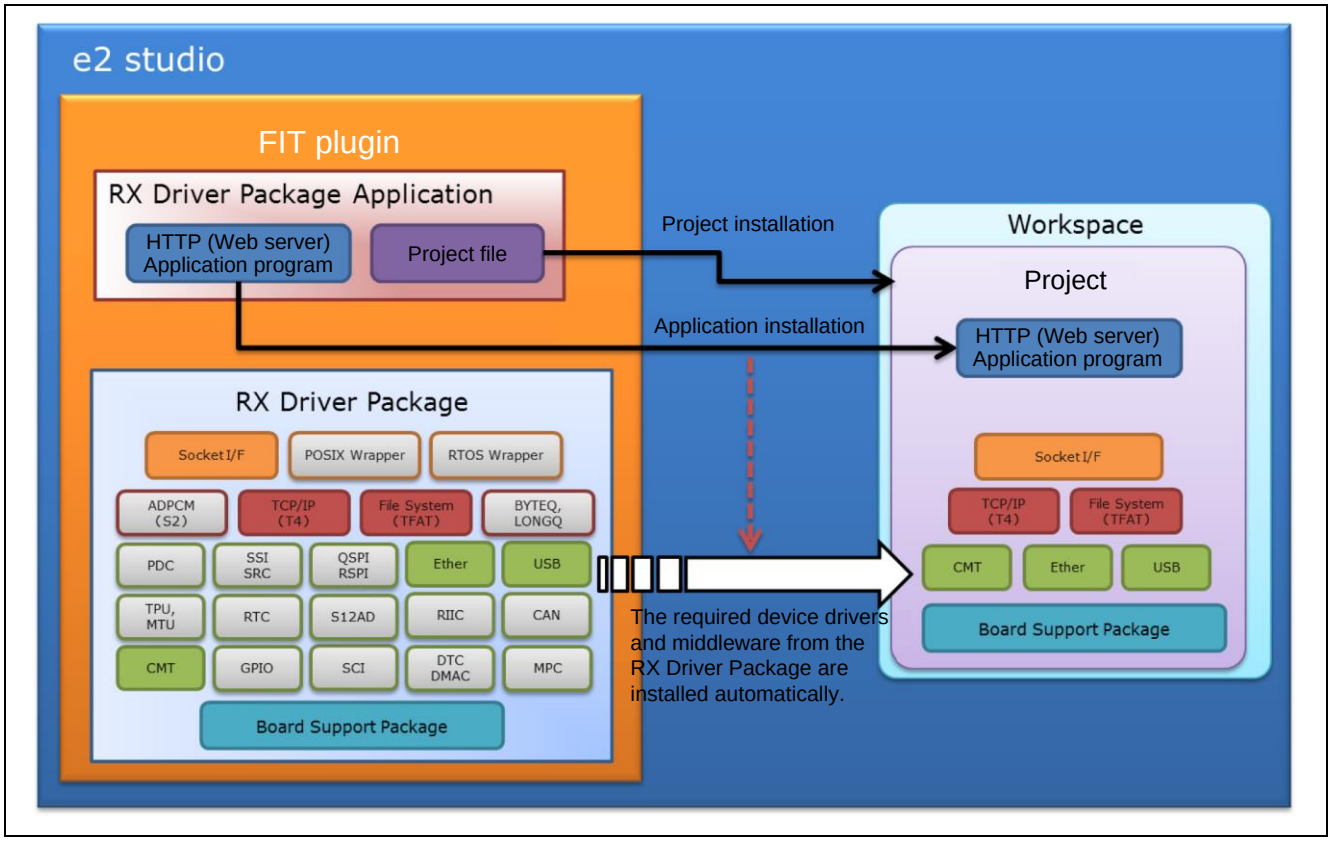


Figure 5.3.1 FIT Plugin Automatic Installation

5.4 When Using in Combination with an RX Driver Package Application

See the document provided with each RX Driver Package Application for detailed usage methods for that RX Driver Package Application.

6. Supplement

6.1 M3S-T4-Tiny（TCP/IP protocol stack）

This package include the M3S-T4-Tiny (TCP/IP protocol stack library) of evaluation version. For the commercial version, please go to the below URL.

http://www.renesas.com/mw/t4

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/contact/

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Sep 1, 2014	-	First edition issued
1.01	Jan 5, 2015	-	Updated existing modules to latest modules. Added new release modules to the package.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.
   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.
5. Differences between Products
   Before changing from one product to another, i.e. to a product with a different type number, confirm that the change will not lead to problems.
   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141