RL78 Family
EEPROM Emulation Library Pack02 Ver.1.01
Release Note

R20UT2645EJ0100
Rev.1.00
Mar 28, 2014

Thank you for using the RL78 Family EEPROM Emulation Library Pack02 Ver.1.01.
This document contains notes and cautions for using the EEPROM Emulation Library Pack02 Ver.1.01.
Please read this document before use.

Contents

Chapter 1 Target Product

This shows the target product of release note.

Product Name	Ver.	ZIP File Name	Zip Ver.
RL78 Family EEPROM Emulation Library Pack02	V1.01	JP_R_EEL_RL78_P02_V1.01_A_E	V1.01A

Chapter 2 User's Manual

The following User's Manual is available for this library:

Title	Document Number
RL78 Family EEPROM Emulation Library Pack02 User's Manual	R01US0068EJ0100

Chapter 3 Operating Environment

Using the EEPROM Emulation Library Pack02 Ver. 1.01 requires the following environment, which supports the development tools.

IBM PC/AT-compatibles (Windows™ based)

OS : WindowsXP, WindowsVista, Windows7

Chapter 4 Supported Tools

Use the following tool version when using tools in combination with this library:

Tool Used	Version
Integrated development environment CubeSuite+	V1.00.00 or later

Chapter 5 Installation

This chapter describes how to install and uninstall the EEPROM Emulation Library Pack02.

5.1 Installation

Install the EEPROM Emulation Library Pack02 by using the following procedure:

(1) Start Windows.
(2) Decompress the folder that contains the EEPROM Emulation Library Pack02 files and copy the extracted folders to any location.

5.2 Uninstallation

Uninstall the EEPROM Emulation Library Pack02 by using the following procedure:

(1) Start Windows.
(2) Delete the folder that contains the EEPROM Emulation Library Pack02 files.

5.3 File organization

The file organization after this library is installed is shown below.

```
Installation folder
└ EELRL78 Pack02
   └ V1.01
      ├ Read me
      │  └ TERMS AND CONDITIONS FOR USING THE SOFTWARE.pdf :
      │                                   Terms and conditions for using the software
      ├ doc
      │  ├ r01us0068ej0100_rl78.pdf      : User's Manual
      │  └ r20ut2645ej0100_rl78.pdf      : Release Note (this document)
      ├ librl78
      │  ├ eel.lib                       : EEPROM emulation library (EEL)
      │  └ fdl.lib                       : Data flash library (FDL)
      ├ incrl78
      │  ├ eel.h                         : EEL header file for C program
      │  ├ eel.inc                       : EEL header file for assembler
      │  ├ eel_types.h                   : EEL header file that specifies definitions for C program
      │  ├ eel_types.inc                 : EEL header file that specifies definitions for assembler
      │  ├ fdl.h                         : FDL header file for C program
      │  ├ fdl.inc                       : FDL header file for assembler
      │  └ fdl_types.h                   : FDL header file that specifies definitions for C program
      └ smprl78
         ├ asm
         │  ├ eel_descriptor.inc         : EEL descriptor header file
         │  ├ eel_descriptor.asm         : EEL descriptor source file
         │  ├ eel_sample_linker_file.dr  : EEL sample directive file
         │  ├ fdl_descriptor.inc         : FDL descriptor header file
         │  └ fdl_descriptor.asm         : FDL descriptor source file
         └ C
            ├ eel_descriptor.h           : EEL descriptor header file
            ├ eel_descriptor.c           : EEL descriptor source file
            ├ eel_user_types.h           : EEL user-defined header file
            ├ eel_sample_linker_file.dr  : EEL sample directive file
            ├ fdl_descriptor.h           : FDL descriptor header file
            ├ fdl_descriptor.c           : FDL descriptor source file
            ├ r_eel_sample_c.c           : EEL sample program file
            └ r_eel_sample_c.dr          : EEL sample Link directive file
```

Note: When you use the sample program, use the combination of the program file (*.c) and the link directive file (*.dr).

Chapter 6 How To Build a Program

This chapter describes how to build a program using the EEPROM Emulation Library Pack02.

6.1 Software to be used

Below are the system requirements for building programs using the EEPROM Emulation Library Pack02.

- CubeSuite+ integrated development environment Version 1.00.00 or later

6.2 Building using CubeSuite+

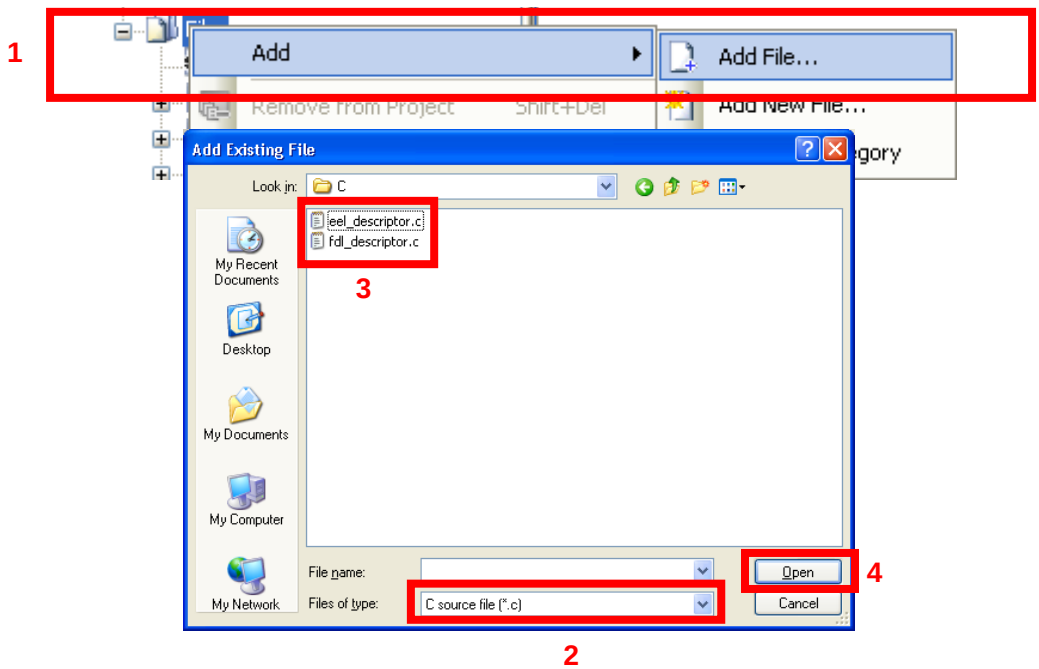
This section describes how to include the EEPROM Emulation Library Pack02 in a user-created program and build the user program by using CubeSuite+.

6.2.1 Building a C program

(1) Creating a project and specifying the source files

Create a project by using CubeSuite+. In the **Project Tree** window displayed on the left, right-click the **File** node, click **Add**, and then click **Add File**. The **Add Existing File** dialog box is displayed (as shown in Figure 6-1). Click the **Files of type** drop-down list, select **C source file (*.c)**, and then register the user-created program file and the descriptor files for the EEPROM emulation library and data flash library (eel_descriptor.c and fdl_descriptor.c) as the source files.

Figure 6-1. Specifying the Source File



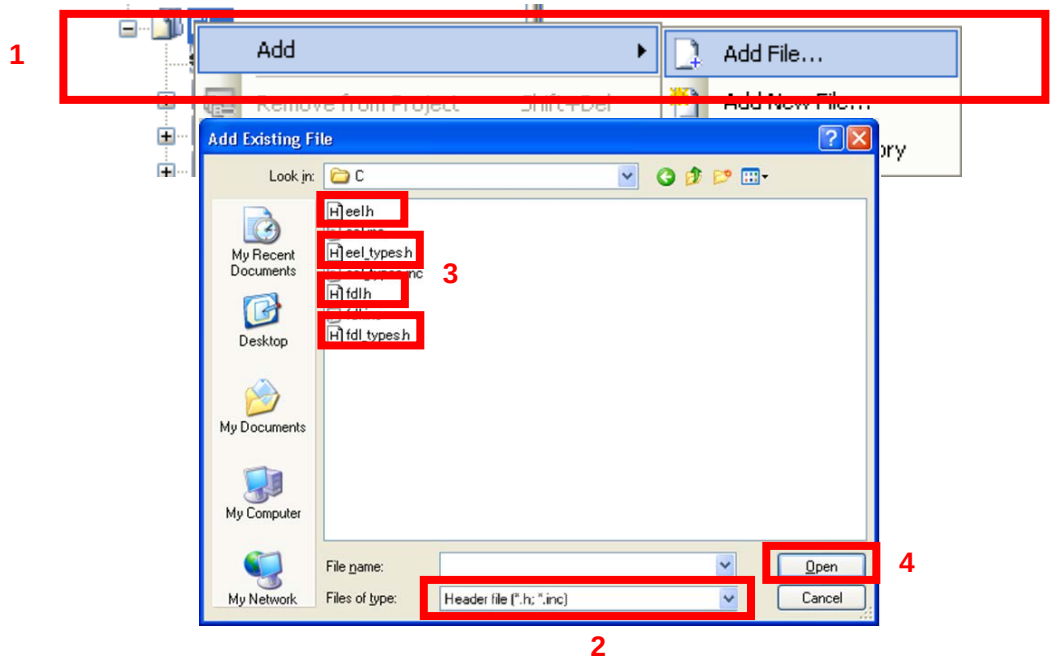
(2) Specifying the include file

In the CubeSuite+ **Project Tree** window, right-click the **File** node, click **Add**, and then click **Add File**.

The **Add Existing File** dialog box is displayed (as shown in Figure 6-2).

Click the **Files of type** drop-down list, select **Header file (*.h;*.inc)**, and then register the header files and descriptor header files for the EEPROM emulation library and data flash library (eel.h, eel_types.h, fdl.h, fdl_types.h, eel_descriptor.h, fdl_descriptor.h, eel_user_types.h).

Figure 6-2. Specifying the include Files

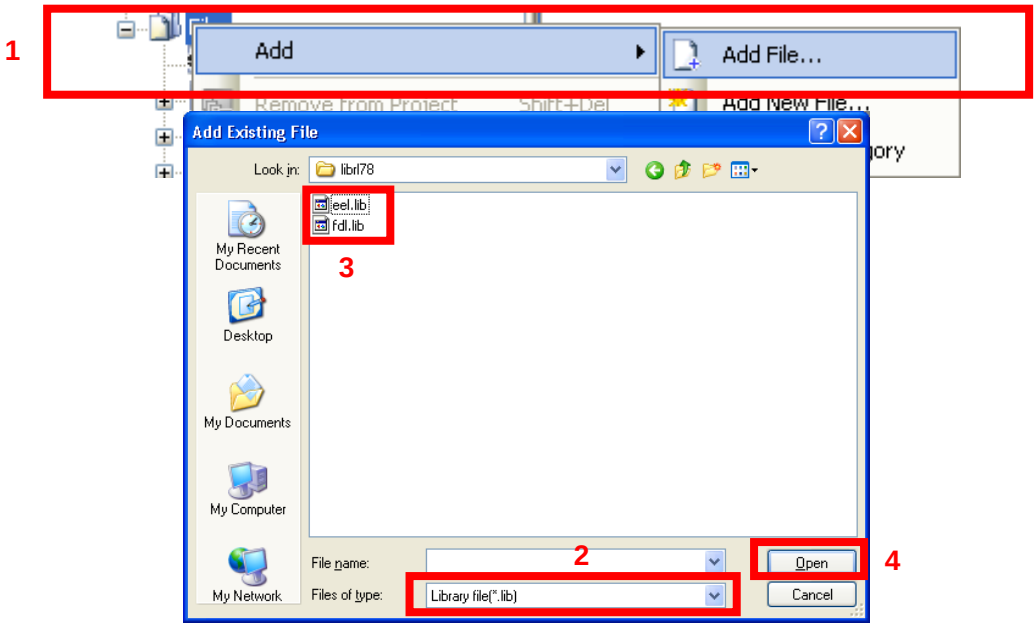


(3) Specifying the library file

In the CubeSuite+ **Project Tree** window, right-click the **File** node, click **Add**, and then click **Add File**. The **Add Existing File** dialog box is displayed (as shown in Figure 6-3).

Click the **Files of type** drop-down list, select **Library file (*.lib)**, and then register the EEPROM emulation library and data flash library files (eel.lib, fdl.lib).

Figure 6-3. Specifying the library File

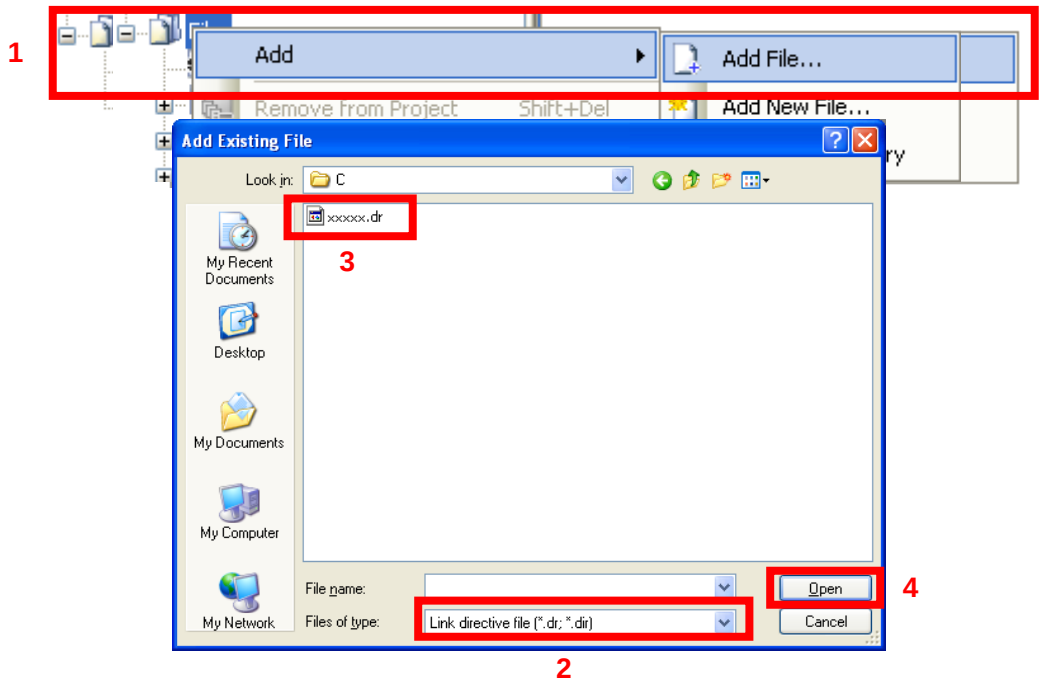


(4) Specifying the link directive file

In the CubeSuite+ **Project Tree** window, right-click the **File** node, click **Add**, and then click **Add File**. The **Add Existing File** dialog box is displayed (as shown in Figure 6-4).

Click the **Files of type** drop-down list, select **Link Directive File (*.dr;*.dir)**, and then register the link directive file that has the same name as the user-created program.

Figure 6-4. Specifying the Link Directive File



(5) Building

On the CubeSuite+ **Build** menu, click **Build Project** to build the project.

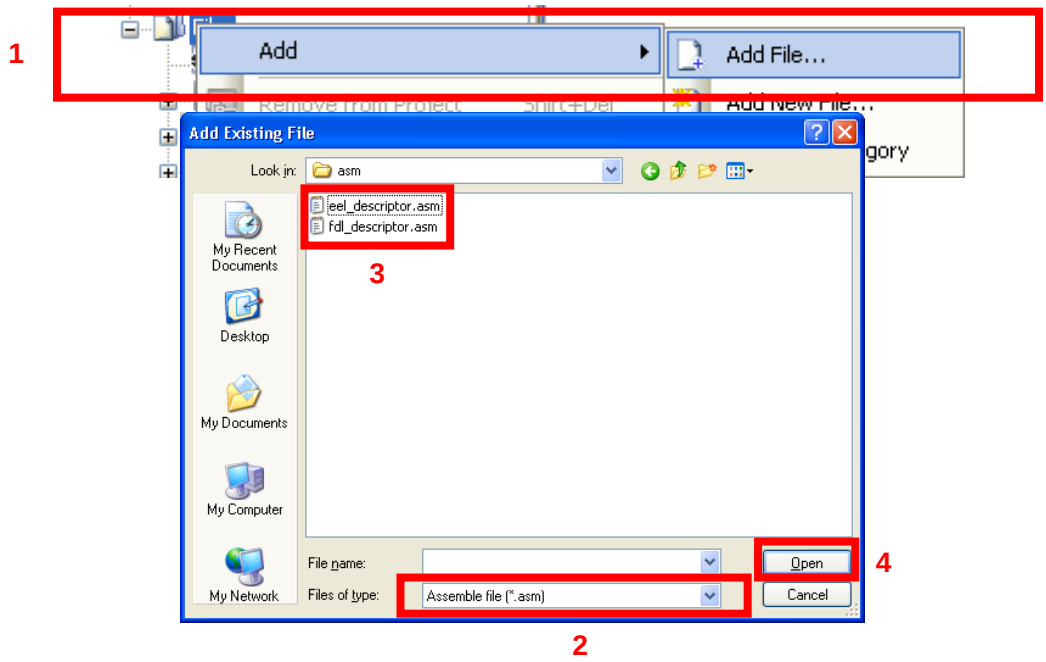
Note: You may need to edit the provided sample directive file before using it.

6.2.2 Building an assembly language program

(1) Creating a project and specifying the source files

Create a project by using CubeSuite+. In the **Project Tree** window displayed on the left, right-click the **File** node, click **Add**, and then click **Add File**. The **Add Existing File** dialog box is displayed (as shown in Figure 6-5).
Click the **Files of type** drop-down list, select **Assemble file (*.asm)**, and then register the user-created program file and the descriptor files for the EEPROM emulation library and data flash library (eel_descriptor.asm and fdl_descriptor.asm) as the source files.

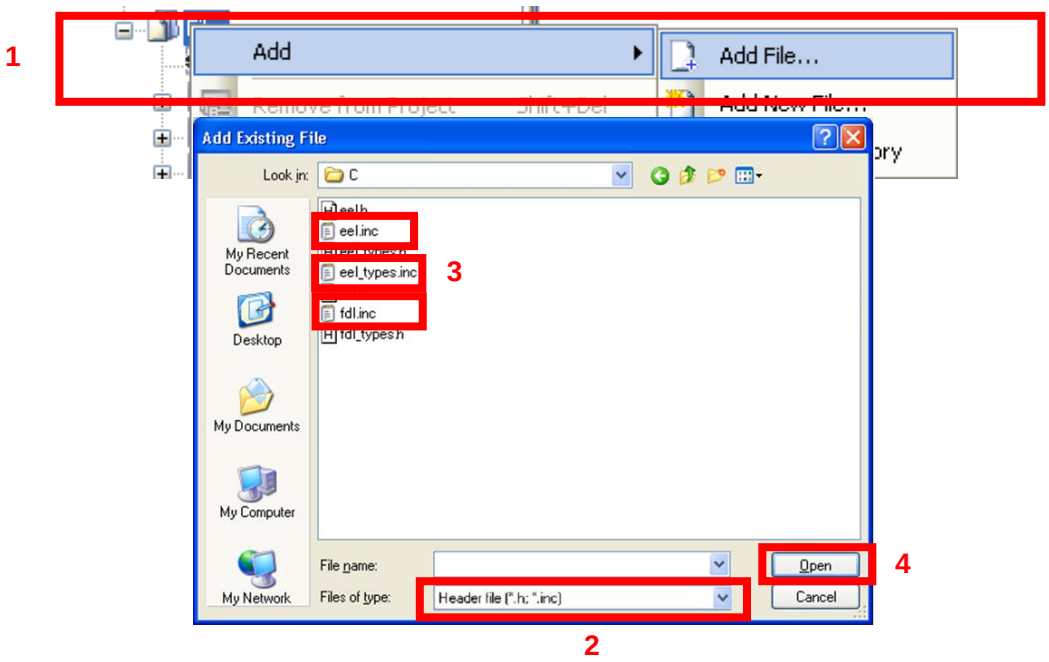
Figure 6-5. Specifying the Assemble File



(2) Specifying the include file

In the CubeSuite+ **Project Tree** window, right-click the **File** node, click **Add**, and then click **Add File**.
The **Add Existing File** dialog box is displayed (as shown in Figure 6-6).
Click the **Files of type** drop-down list, select **Header file (*.h;*.inc)**, and then register the include files and the descriptor include files for the EEPROM emulation library and data flash library (eel.inc, eel_types.inc, fdl.inc, eel_descriptor.inc, fdl_descriptor.inc).

Figure 6-6. Specifying the include Files

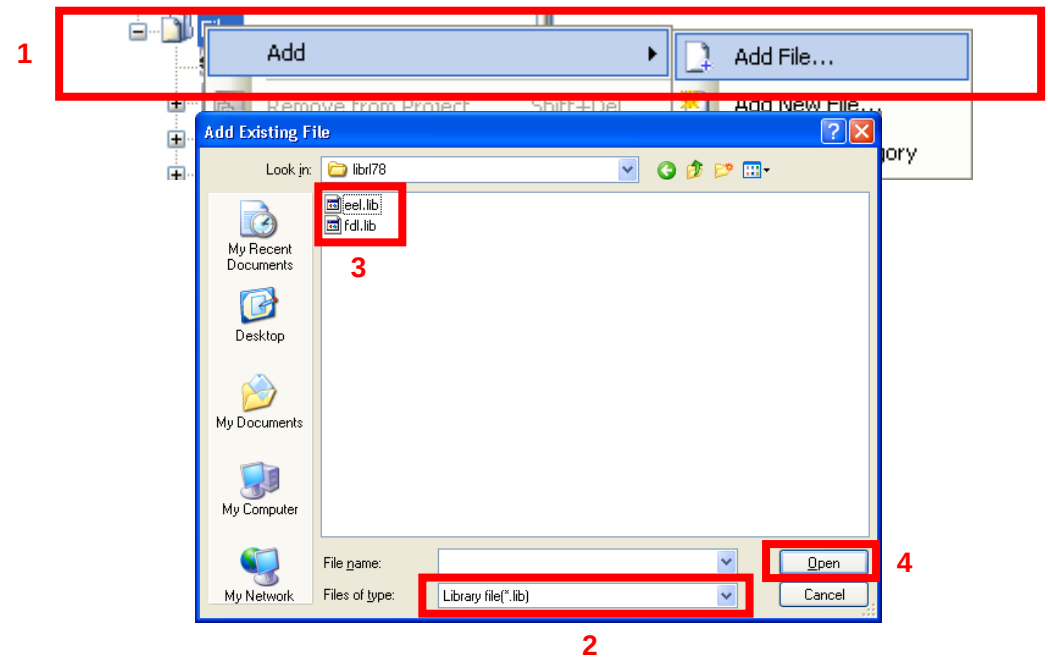


(3) Specifying the library file

In the CubeSuite+ **Project Tree** window, right-click the **File** node, click **Add**, and then click **Add File**. The **Add Existing File** dialog box is displayed (as shown in Figure 6-7).

Click the **Files of type** drop-down list, select **Library file (*.lib)**, and then register the EEPROM emulation library and data flash library files (eel.lib, fdl.lib).

Figure 6-7. Specifying the library File

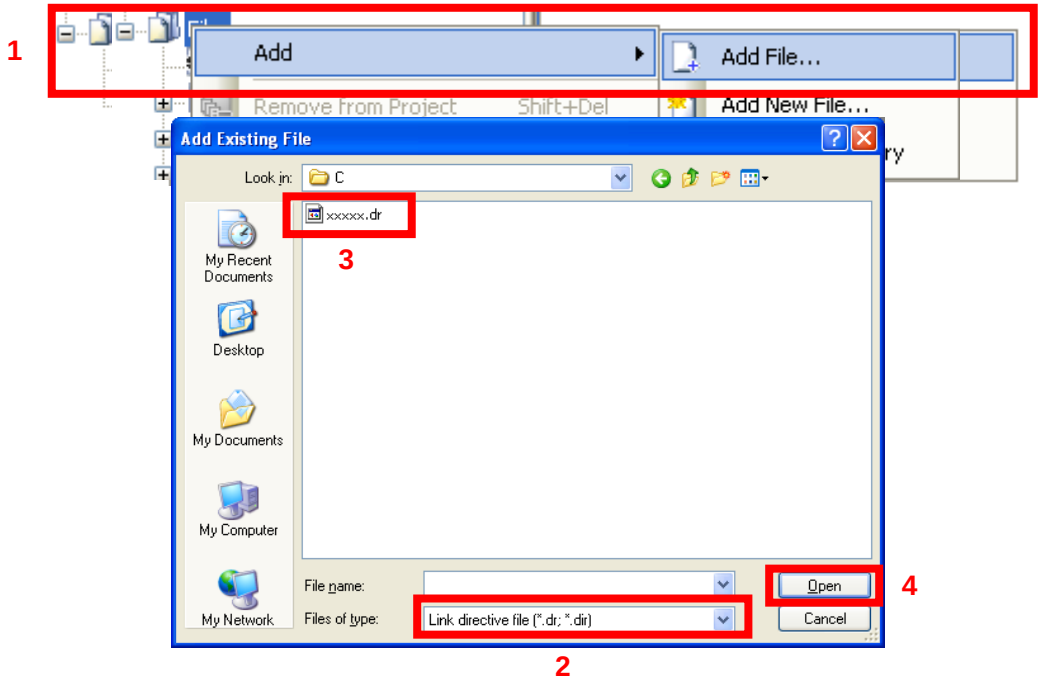


(4) Specifying the link directive file

In the CubeSuite+ **Project Tree** window, right-click the **File** node, click **Add**, and then click **Add File**. The **Add Existing File** dialog box is displayed (as shown in Figure 6-8).

Click the **Files of type** drop-down list, select **Link Directive File (*.dr;*.dir)**, and then register the link directive file that has the same name as the user-created program.

Figure 6-8. Specifying the Link Directive File



(5) Building

On the CubeSuite **Build** menu, click **Build Project** to build the project.

6.3 Notes at Build

(1) When the on-chip debugging function is in use

After the on-chip debugging function is enabled in the CubeSuite+, building a program generates the following type of error.

```
RA78K0R error E3212: Default segment can't allocate to memory - ignored
                     Segment '??OCDROM' at xxxxxH-200H
```

This error occurs when the segment for the monitor area (OCDROM) used by the on-chip debugging function cannot be allocated. Therefore, to avoid this error, add the following code to the link directive file (*.dr) embedded in the project and prepare a separate area for allocating the segment.

```
MEMORY OCD_ROM : ( 0xxxxxH, 00200H )
```

Note: 1. xxxxx: Start address at where the error has occurred

2. The area name "OCD_ROM" is a reference example.

Chapter 7 How To Debug a Program

For details about how to perform debugging by using IECUBE or the on-chip debug emulator E1 or E20, see the following document. Download the following document from the “Integrated Development Environment CubeSuite+” page on the Renesas Electronics Web site.

Title
CubeSuite+ Integrated Development Environment User's Manual: RL78 Debug

7.1 Notes at Debug

The following describes notes apply when using the EEPROM Emulation Library Pack02 with the E1 or E20 on-chip debugging emulator.

(1) When a command of the EEPROM Emulation Library Pack02 Ver. 1.01 is executed in a version older than CubeSuite+ Ver. 1.01 and the E1 or E20 on-chip debugging emulator is in use, do not execute a break until you have confirmed completion of the command by the sequencer. The sequencer will malfunction if a break occurs before the sequencer has completed the command.

(2) The simulator cannot be used to debug the flash library.

Chapter 8 Sample Program

The attached sample program (r_eel_sample_c.c) is provided to enable the usage method of the EEPROM Emulation Library Pack02 to be easily confirmed on the QB-R5F100LE-TB boards with R5F100LEA (RL78/G13) as the target microcontrollers. The sample program is just a reference example and the user program does not have to be created to match the sample program. The sample program should be used as a simple program to confirm operation.

The link directive file (r_eel_sample_c.dr) for the sample program has a purpose to specify that a stack or data buffer used by the sample program is not allocated to an area where allocation is prohibited [Note1]. When using the sample program, this file should also be embedded with the sample program.[Note2]

Notes: 1. For details, refer to chapter 6.2 "Software Resource" in the user's manual.

2. The data in usage may be placed at an unintended area depending on how the environment in use or the program is changed. After an execution module is generated, the map file and allocation state of programs or data must be confirmed. For the definition method and allocation conditions of each code or data, refer to the user's manual of the CubeSuite+.

8.1 Initial Settings of the Sample Program

The sample program operates with the following initial settings. When these settings need to be changed, modify the sample program.

- CPU operating frequency: High-speed on-chip oscillator 32 MHz
- Voltage mode: Full-speed mode

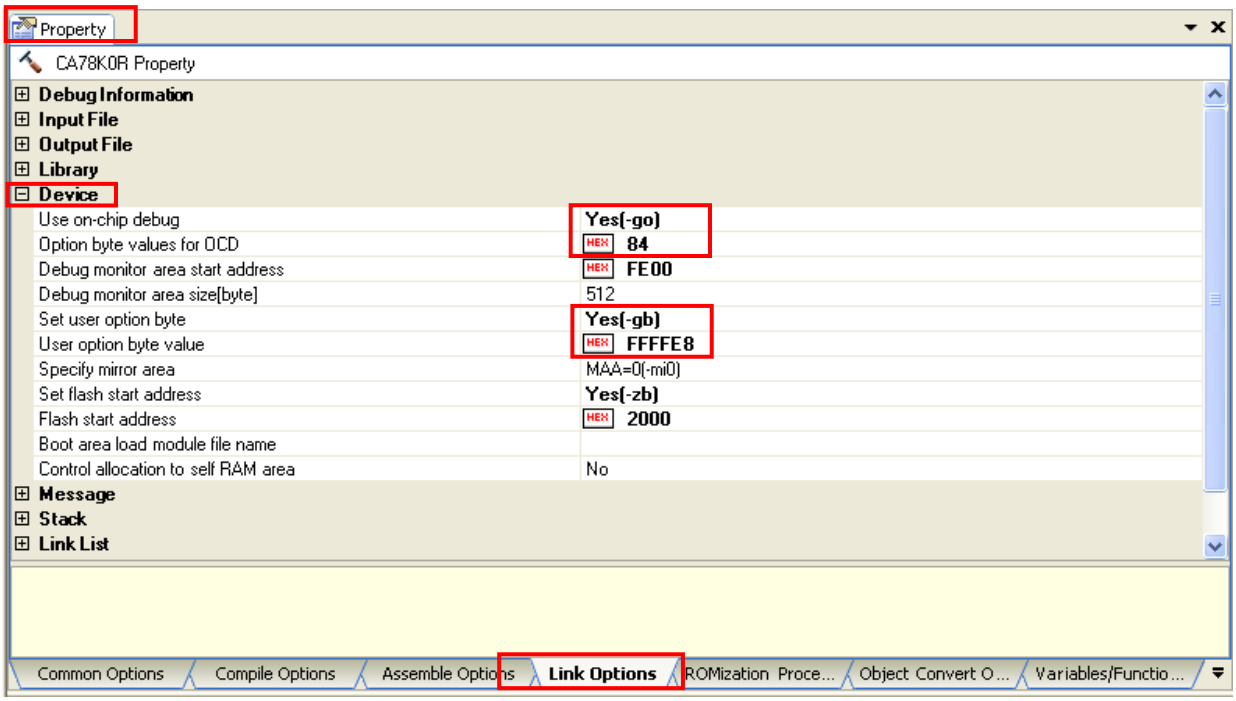
8.2 Settings of Option Byte and On-Chip Debugging

The sample program normally operates by setting the high-speed on-chip oscillator at 32 MHz.

After setting “Set user option byte” to “Yes” from the Link options of CA78K0R, specify “xxxxE8” for the value of the user option byte and set the high-speed on-chip oscillator at “32 MHz”.

When performing on-chip debug, set “Ues on-chip debug” to “Yes” and specify “84” for the controlling value of the on-chip debugging option byte.

Figure 8-1 Setting of Option Byte



8.3 Definition of the On-Chip RAM Area

This section describes how the on-chip RAM area is defined in the link directive file.

Normally, the entire on-chip RAM area is automatically defined as "RAM" unless otherwise stated in the link directive file. Stacks are to be allocated to "RAM" except as otherwise noted[Note]. However, in this case, stacks or data buffers are allocated to an area (self-RAM or FFE20H to FFEFFH) that is prohibited to be used by the EEPROM Emulation Library Pack02 and the program may not operate correctly.

In the attached link directive file for the sample program, as a solution, "RAM" is re-defined not to include the area which the EEPROM Emulation Library Pack02 is prohibited to use so that stacks or data buffers are not allocated to that area.

```
MEMORY RAM          :(0FF300H, 000B20H)
```

In the above code, "RAM" is re-defined to be an area (FF300H to FFE1FH)[Note] with a size of B20H bytes and starting from address FF300H. This prevents the area which the EEPROM Emulation Library Pack02 is prohibited to use from being included in "RAM".

However, if only this setting is made, the area of "FFE20H to FFEFFH" cannot be used for any other purposes either. Therefore, the following definition must be added separately. There are no particular restrictions regarding the name of this area.

```
MEMORY SADDR_RAM:(0FFE20H, 0001E0H)
```

If there is a self-RAM area, by defining its range with the name of "SELFRAM", automatic allocation of variables to this area can be restricted.

```
MEMORY SELFRAM    :(0FEF00H, 000400H)
```

A setting example for the RL78/G13 (product with 4-Kbyte RAM and 64-Kbyte ROM) is given below.

```
; ----------------------------------------------------------
; Define new memory entry for Self-RAM
; ----------------------------------------------------------
MEMORY SELFRAM     : ( 0FEF00H, 000400H )   <-- Definition of the self-RAM area

; ----------------------------------------------------------
; Redefined default data segment RAM
; ----------------------------------------------------------
MEMORY RAM         : ( 0FF300H, 000B20H )   <-- Definition of the RAM area to be used normally

; ----------------------------------------------------------
; Define new memory entry for saddr area
; ----------------------------------------------------------
MEMORY RAM_SADDR : ( 0FFE20H, 0001E0H )     <-- Definition of the area from FFE20H to FFFFFH
```

Note: The CA78K0R linker allocates data whose allocation destination is not specified (segment type of DSEG or BSEG) to the on-chip RAM area according to the re-allocation attribute of the data. Accordingly, specific data may not be allocated to the area with the name of "RAM" in some situations.
For details on the definition or allocation method of each data, refer to the user's manual of the CubeSuite+.
The map file (*.map) generated at build must be referenced to confirm the allocation state.

Website and Support

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

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

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Mar 28, 2014	All	First edition issued

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 accordance 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 part 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

Colophon 4.0