APPLICATION NOTE

RX64M, RX71M Group

R01AN2608EJ0103
Rev.1.03
Feb 29, 2016

TCP/IP Protocol Stack Based Network Solution for Industrial Applications

RX Driver Package Application

Introduction

This document is an application note of the industrial network solution using TCP/IP protocol stack (referred to as “M3S-T4-Tiny”).

This application note includes sample code for a main program that uses Network Middleware Modules (DHCP client, DNS client, HTTP server, FTP server), when used in combination with the RX64M, RX71M Group RX Driver Package, allows the construction of various network systems.

Target Device

RX64M Group (Renesas Starter Kit+ RX64M)

RX71M Group (Renesas Starter Kit+ RX71M)

RX Driver Package Used

RX64M, RX71M Group RX Driver Package Ver.1.02 (R01AN2606EJ)

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.

Contents

1. Overview

1.1 This Application Note

This application note describes the procedure for main program evaluation by combining the Board Support Package (referred to as “BSP”), USB driver (Host Mass Storage Class Driver “USB HMSC” and “Basic Firmware”), M3S-TFAT-Tiny FAT file system (referred to as “TFAT”), Ethernet driver, TCP/IP protocol stack (M3S-T4-Tiny), and Network Middleware of FIT modules included in the RX64M, RX71M Group RX Driver Package.

This application note operates on the Renesas Starter Kit+ for RX64M, Renesas Starter Kit+ for RX71M (referred to as “RSK” in the remainder of this document).

1.2 Operating Environment

This application note operates in the following environment.

Table 1-1 Operating Environment

Microcontroller	RX64M Group
Evaluation board	Renesas Starter Kit+ RX64M http://www.renesas.com/products/tools/introductory_evaluation_tools/renesas_starter_kits/rsk_rx64m/index.jsp
Integrated development environment (IDE)	e^2 studio, V4.1.0 or later
Cross tools	RX Family C/C++ Compiler Package V2.03.00 or later
Emulator	E1 (included in the Renesas Starter Kit+ for RX64M), E20
RX Driver Package	RX64M, RX71M Group RX Driver Package Ver1.02 (R01AN2606EJ)*

Table 1-2 Operating Environment

Microcontroller	RX71M Group
Evaluation board	Renesas Starter Kit+ RX71M http://wwwrenesas.com/products/tools/introductory_evaluation_tools/renesas_starter_kits/rsk_rx71m/index.jsp
Integrated development environment (IDE)	e^2 studio, V4.1.0 or later
Cross tools	RX Family C/C++ Compiler Package V2.03.00 or later
Emulator	E1 (included in the Renesas Starter Kit+ for RX71M), E20
RX Driver Package	RX64M, RX71M Group RX Driver Package Ver1.02 (R01AN2607EJ)*

Note: * Operation of this application note has been verified when the modules in the RX Driver Package mentioned above are incorporated. If any of the modules used in this application note are replaced with a different module, the user must verify the operation.

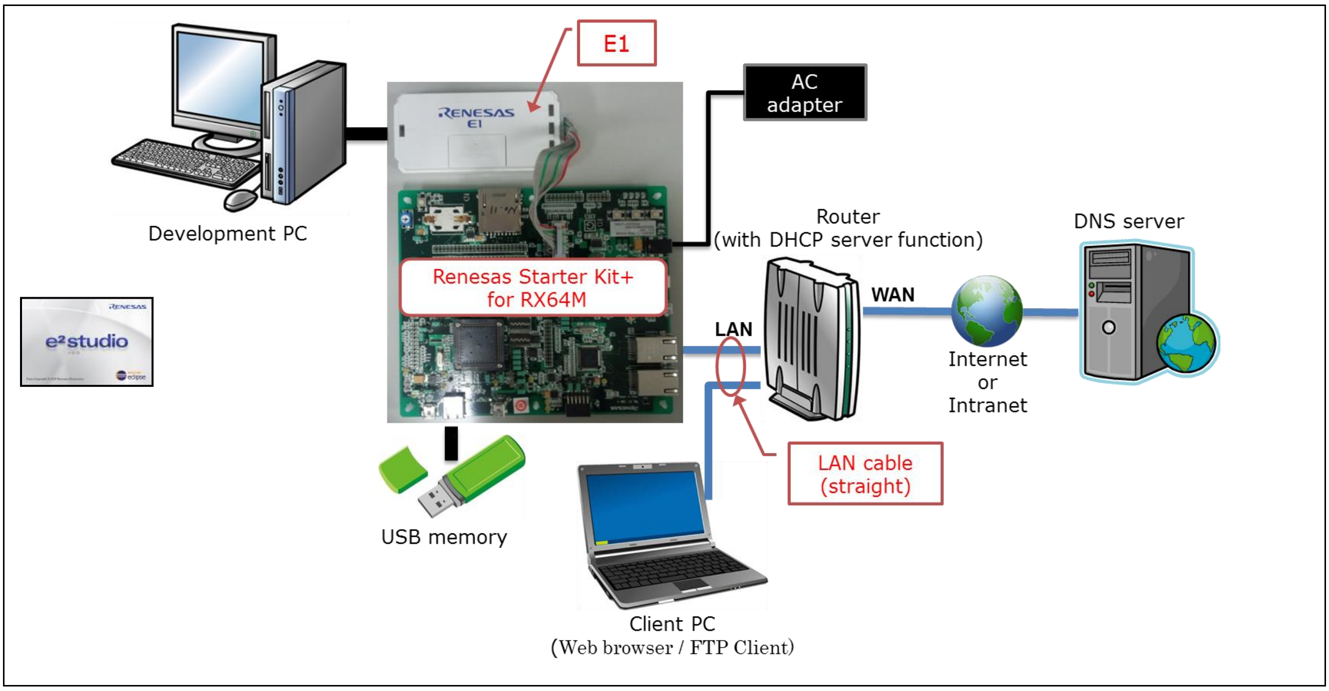

E1
AC adapter
Development PC
Renesas Starter Kit+ for RX64M
Router (with DHCP server function)
DNS server
WAN
LAN
Internet or Intranet
LAN cable (straight)
USB memory
Client PC (Web browser / FTP Client)
e²studio

Figure 1-1 Sample Operating Environment

1.3 Module Structure

This section shows the structure of the modules used by this application note and a list of those modules.

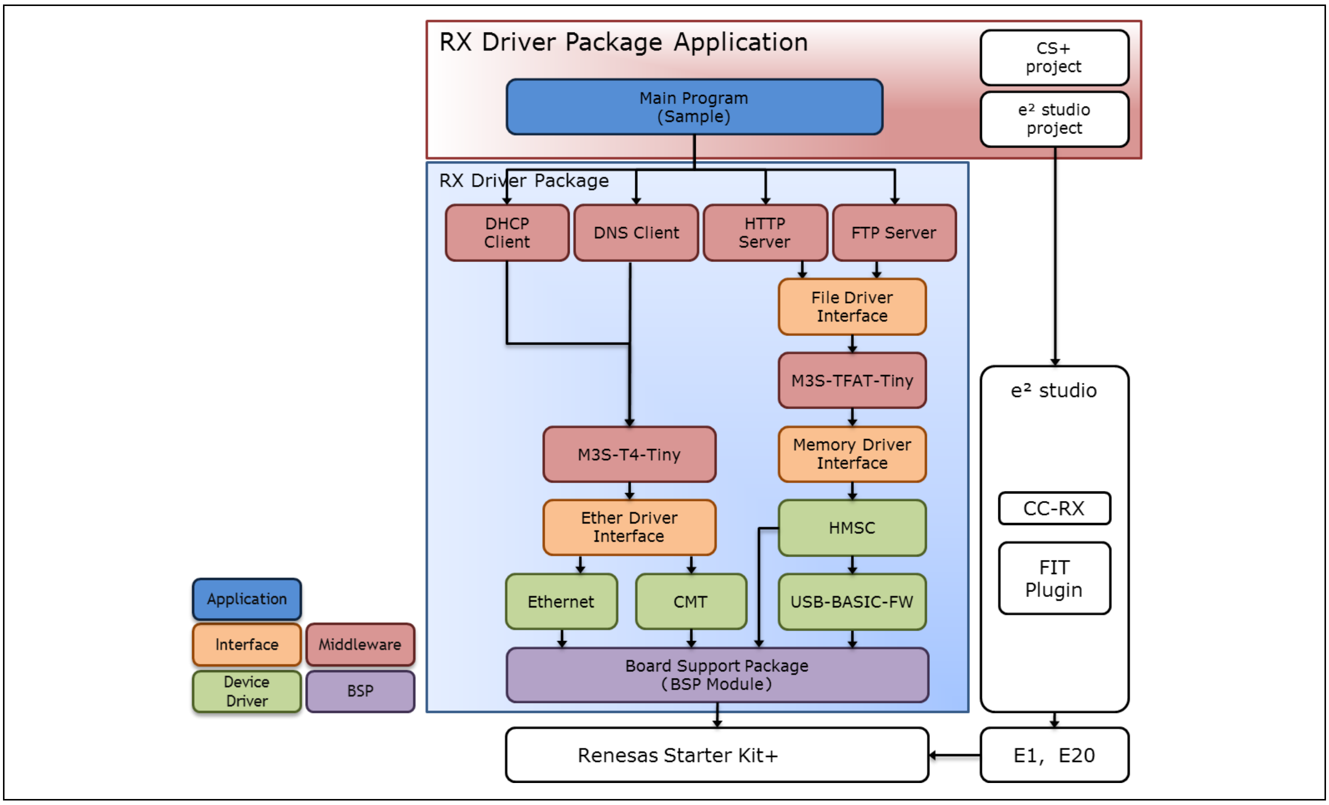


Figure 1-2 Module Structure

Table 1-3 Modules

Type	Module	FIT Module Name	Rev.
Board Support Package	Board Support Package (BSP)	r_bsp	3.00
Device Driver	Compare Match Timer (CMT)	r_cmt_rx	2.60
Device Driver	USB Basic Firmware	r_usb_basic	1.10
Device Driver	USB Host Mass Storage Class	r_usb_hmsc	1.10
Device Driver	Ethernet controller (ETHERC)	r_ether_rx	1.02
Middleware	TCP/IP M3S-T4-Tiny for Embedding (Note 1)	r_t4_rx	2.02
Interface	Interface conversion module for Ethernet Driver and Embedded System M3S-T4-Tiny	r_t4_driver_rx	1.02
Middleware	DHCP client using the embedded TCP/IP M3S-T4-Tiny Module	r_t4_dhcp_client_rx	1.03
Middleware	DNS client using the embedded TCP/IP M3S-T4-Tiny Module	r_t4_dns_client_rx	1.02
Middleware	FTP server using the embedded TCP/IP M3S-T4-Tiny Module	r_t4_ftp_server_rx	1.03
Middleware	Web server using the embedded TCP/IP M3S-T4-Tiny Module	r_t4_http_server_rx	1.04
Interface	File driver for FTP server and Web server Module	r_t4_file_driver_rx	1.01
Middleware	M3S-TFAT-Tiny (FAT file system)	r_tfat_rx	3.02
Interface	M3S-TFAT-Tiny Memory Driver Interface Module	r_tfat_driver_rx	1.02
Application	Main program	src	1.01

Note 1: This package contains the TCP/IP M3S-T4-Tiny evaluation version. For commercial version, refer to the following URL.
http://www.renesas.com/mw/t4

1.4 File Structure

This section describes the file structure used in this application note.

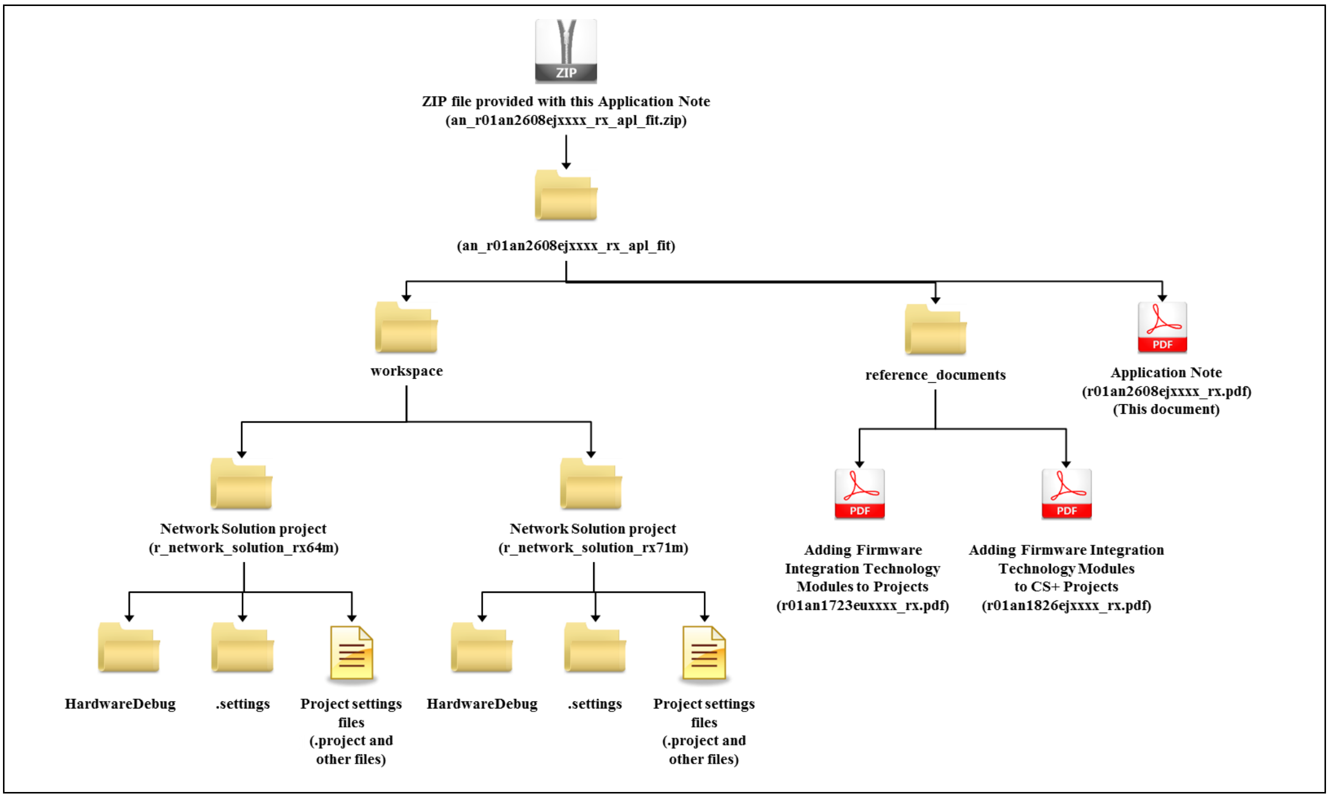

ZIP
ZIP file provided with this Application Note
(an_r01an2608ejxxxx_rx_apl_fit.zip)
(an_r01an2608ejxxxx_rx_apl_fit)
workspace
reference_documents
PDF
Application Note
(r01an2608ejxxxx_rx.pdf)
(This document)
Network Solution project
(r_network_solution_rx64m)
Network Solution project
(r_network_solution_rx71m)
Adding Firmware Integration Technology Modules to Projects
(r01an1723euxxxx_rx.pdf)
Adding Firmware Integration Technology Modules to CS+ Projects
(r01an1826ejxxxx_rx.pdf)
HardwareDebug
.settings
Project settings files
(.project and other files)
HardwareDebug
.settings
Project settings files
(.project and other files)

Figure 1-3 File Structure

When the ZIP file provided with this application note is decompressed, a folder with the same name is created and the various folders and files are created within that folder.

”Network Solution project (r_network_solution_rx64m)” or ”Network Solution project (r_network_solution_rx71m)” under the “workspace” folder is the project that built this application. Operation of the application can be verified by importing the project to the e² studio workspace.

Documents that describe using the FIT modules in various development environments are included in the **reference_documents** folder. The document “Adding Firmware Integration Technology Modules to Projects” (R01AN1723EU) describes the method for including the FIT modules, as a FIT plugin, in an e^2 studio project. The document “Adding Firmware Integration Technology Modules to CS+ Projects” (R01AN1826EJ) describes the method for including the FIT modules in a CS+ project.

1.5 Projects

This application note includes an e^2 studio project for building and evaluating this application. These projects register both a build structure that stores the build settings and a debug structure that stores debug settings.

The table below lists the build structure and debug structure registered in these projects.

Table 1-4 Project Settings

	Structure	Description
Build structure	HardwareDebug (Debug on hardware)	This structure is used to generate a load module with debugging information included. Main settings
Debug structure	HardwareDebug (E1)	Used for hardware debugging over an E1 emulator using a load module generated by HardwareDebug (Debug on hardware).

The table below lists the target-specific setting.

Table 1-5 Target-specific setting

Items	Setting
Tool chain version	v2.03.00
Debug Hardware	E1 (RX)
Data endian	Little-endian data
Select Target	R5F564MLCxFC (RX64M LQFP 176pin) or R5F571MLCxFC (RX71M LQFP 176pin)
Renesas RTOS Support (Note 1)	None

Note 1: For use, OS environment download is required.

2. Acquiring a Development Environment

2.1 Acquire e^2 studio

Access the following URL and download the e^2 studio.

http://www.renesas.com/e2studio_download

This document requires you to use e^2 studio V4.1.0 or later. If the version older than V4.1.0 is used, some functions of the e^2 studio may not be available. For download, obtain the latest version of the e^2 studio on the website.

2.2 Acquire a Compiler Package

Access the following URL and download the RX Family C/C++ Compiler Package.

http://www.renesas.com/e2studio_download

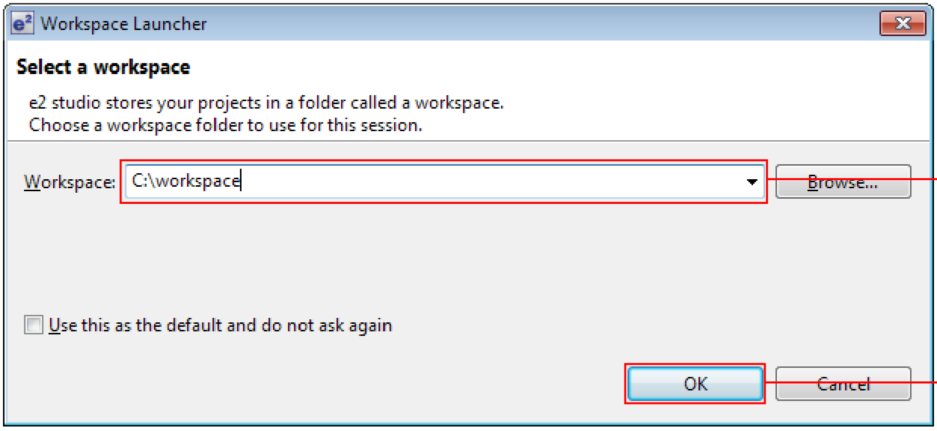
3. Building a Project

This application note includes environment-built project. The procedure to import a Project using e^2 studio Smart browser is described below.

3.1 Create a Workspace

1. Start e^2 studio.

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



Workspace Launcher

Select a workspace

e2 studio stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

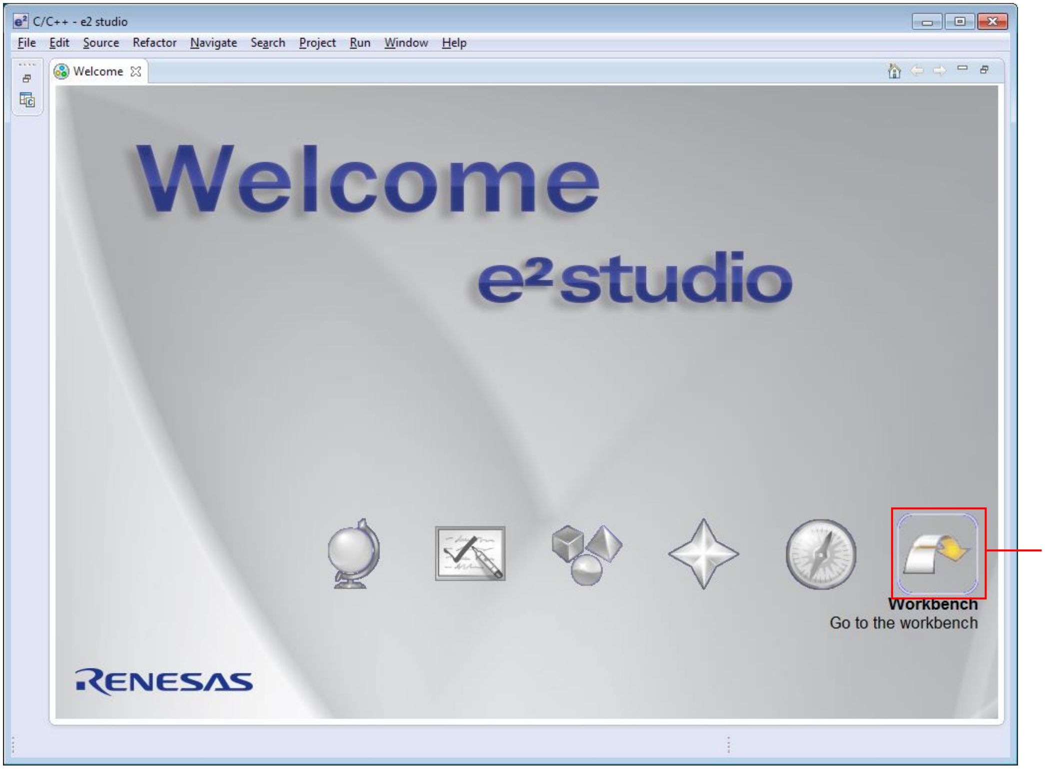
Workspace: C:\workspace

□ Use this as the default and do not ask again

Enter a workspace folder.

Click **[OK]**.

3. When the following window is displayed, click [Workbench].



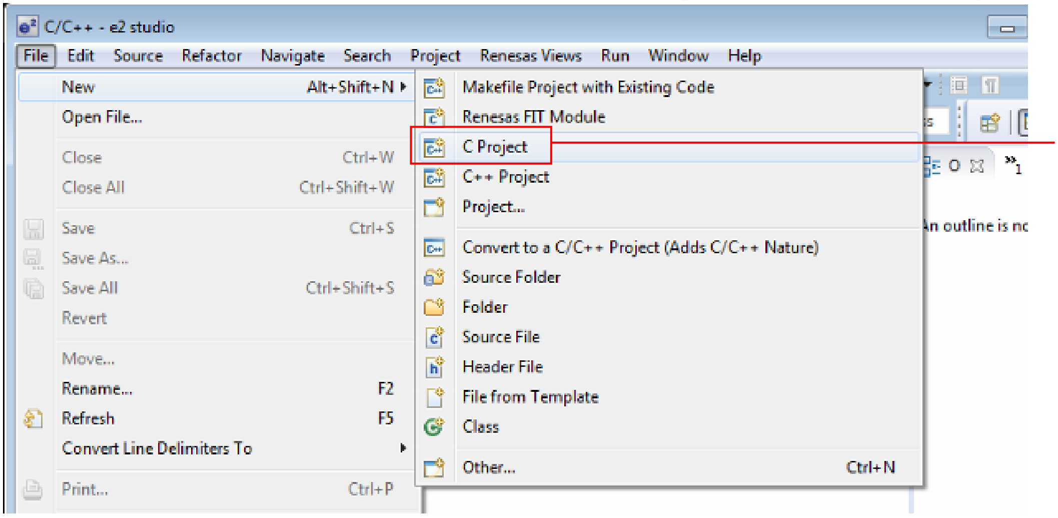
Click **[Workbench]**.

3.2 Create a Project

When using the Smart browser function, the target project or file needs to be selected. To use this function, create the project that specified the target device (Note 1).

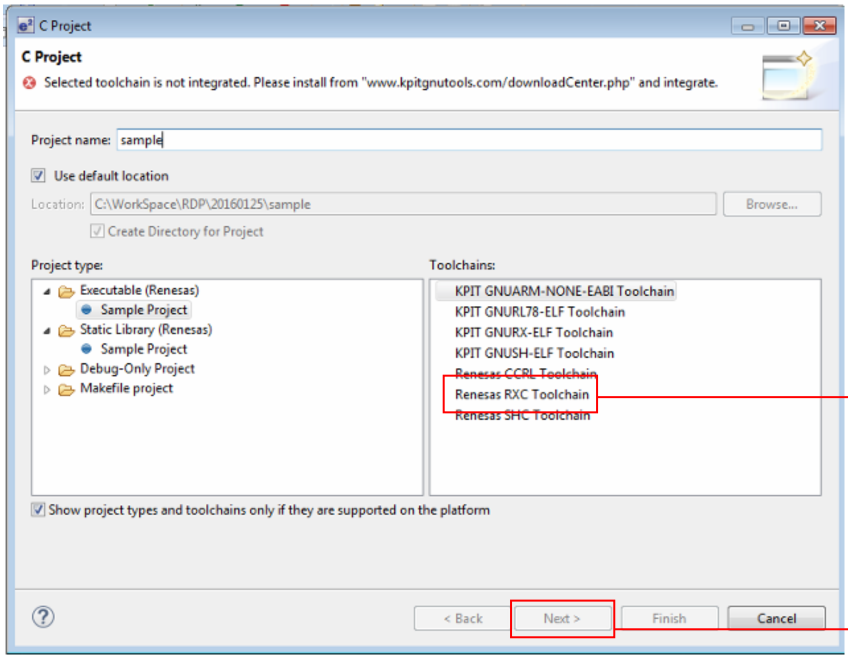
Note 1 : The project to be created here is a dummy to use the Smart browser. For the setting of a Project to be imported, refer to Table 1-5 Target-specific setting.

1. Click [File (F)], [New (N)], then [C Project] to create new C project. Start [Create New project wizard].



Click here.

2. Input any project name and select [Renesas RXC Toolchain]. Click [Next (N)].


C Project

C Project

Selected toolchain is not integrated. Please install from "www.kpitgnutools.com/downloadCenter.php" and integrate.

Project name: sample

☑ Use default location

Location: C:\WorkSpace\RDP\20160125\sample

☑ Create Directory for Project

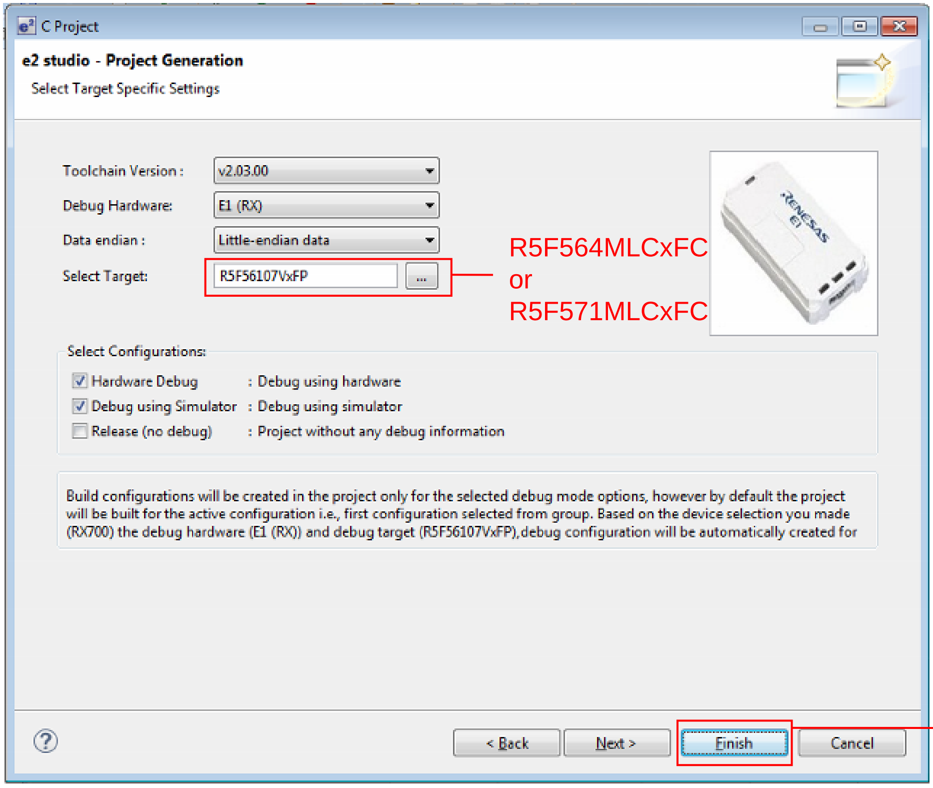
☑ Show project types and toolchains only if they are supported on the platform

Select “Renesas RXC Toolchain”.

Click here.

3. Set Select Target ; for RX64M :R5F564Mxxxxx, for RX71M :R5F571Mxxxxx. For other items, any setting is OK (Note 1). When the setting is completed, click [Finish(F)].

Note 1 : Setting for dummy project. For the setting of the project environment to be imported using the smart browser, refer to Table 1-5 Target-specific setting.



e2 studio - Project Generation

Select Target Specific Settings

Toolchain Version : v2.03.00

Debug Hardware: E1 (RX)

Data endian : Little-endian data

Select Target: R5F56107VxFP ...

R5F564MLCxFC
or
R5F571MLCxFC

Select Configurations:

☑ Hardware Debug : Debug using hardware

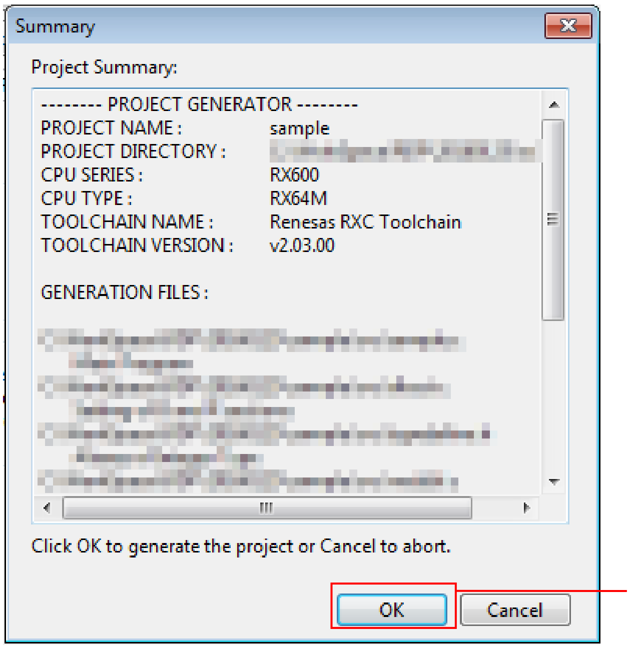
☑ Debug using Simulator : Debug using simulator

□ Release (no debug) : Project without any debug information

Build configurations will be created in the project only for the selected debug mode options, however by default the project will be built for the active configuration i.e., first configuration selected from group. Based on the device selection you made (RX700) the debug hardware (E1 (RX)) and debug target (R5F56107VxFP),debug configuration will be automatically created for

Click here.

4. Click [OK].



Project Summary:

```
-------- PROJECT GENERATOR --------
PROJECT NAME :        sample
PROJECT DIRECTORY :   [redacted]
CPU SERIES :          RX600
CPU TYPE :            RX64M
TOOLCHAIN NAME :      Renesas RXC Toolchain
TOOLCHAIN VERSION :   v2.03.00

GENERATION FILES :

[redacted]
```

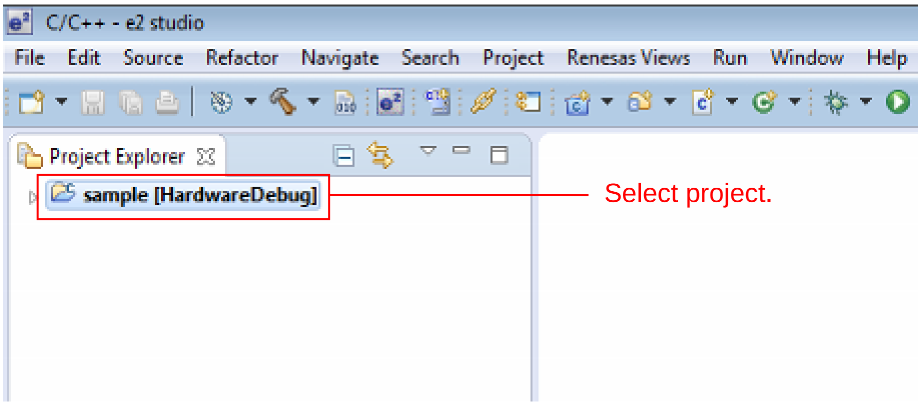
Click OK to generate the project or Cancel to abort.

Click here.

3.3 Import a Project

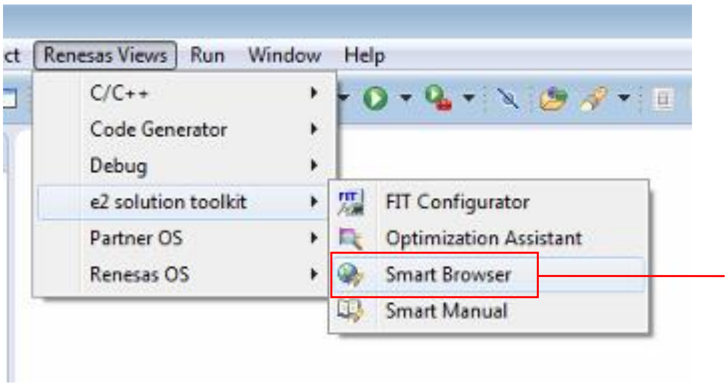
Import the project of Main program in the workspace created.

1. Select the project created in ”3.2 Create a Project” from Project explorer.


C/C++ - e2 studio
File Edit Source Refactor Navigate Search Project Renesas Views Run Window Help
Project Explorer
sample [HardwareDebug]

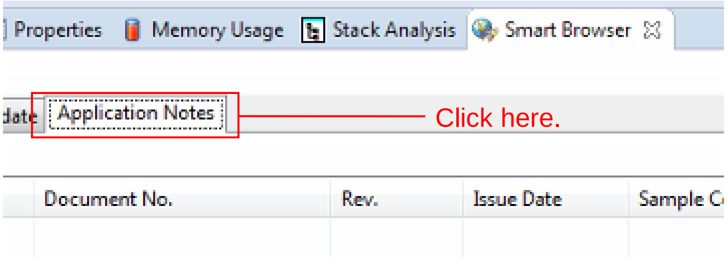
Select project.

2. Click on [Renesas Views] → [e2 Solution Tool kit] → [Smart browser] to start the Smart browser.



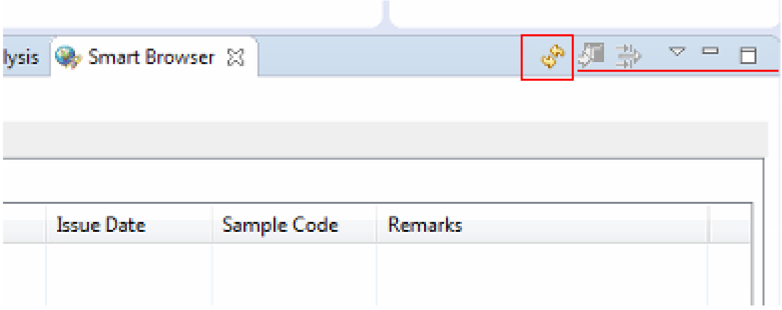
Click here.

3. Click on the [Application Note] on the [Smart browser] tabbed page.


Properties Memory Usage Stack Analysis Smart Browser
date Application Notes
Document No. Rev. Issue Date Sample C

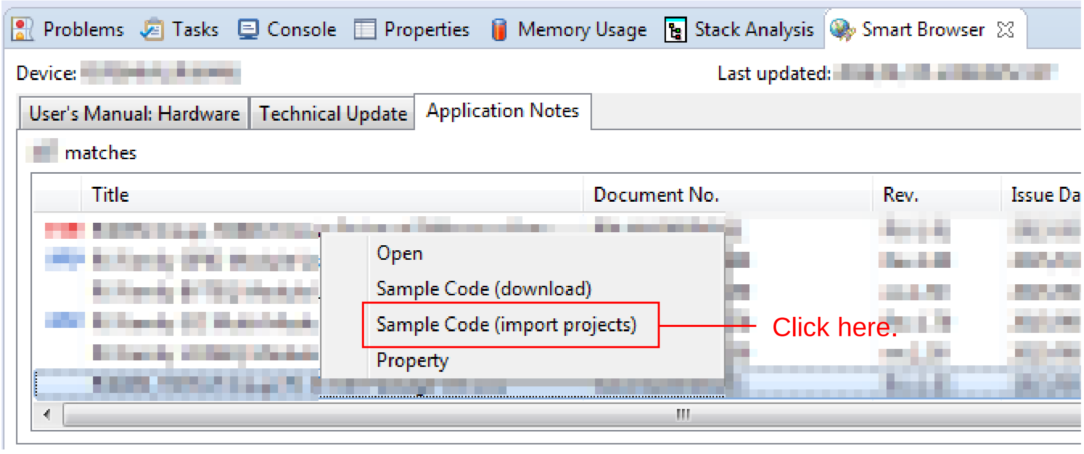
Click here.

4. Click [Update].

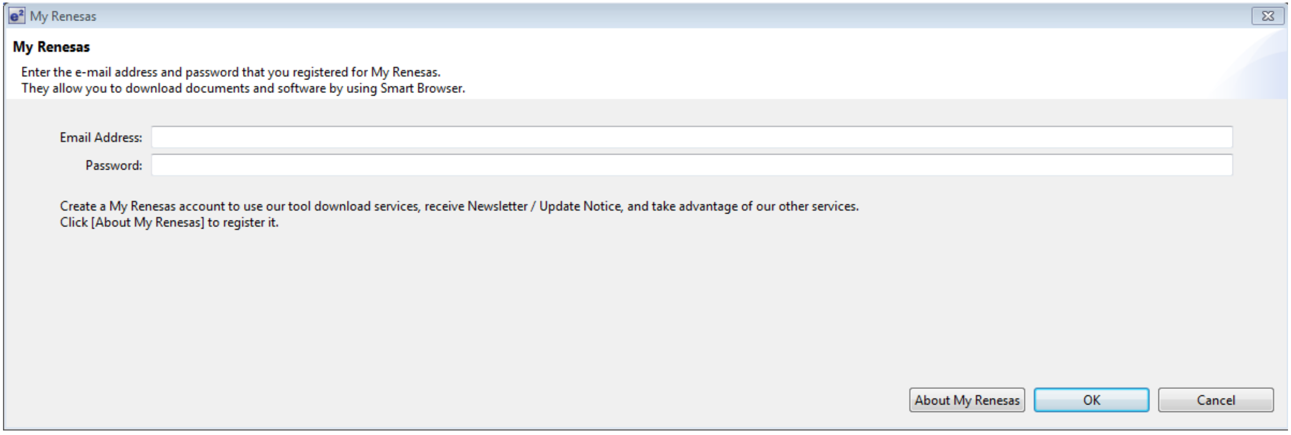


Click here.

5. Select the application note and right-click. Then, click on [Sample code (Project import)] in the context menu (Note 1).



Note 1: If authentication by My Renesas has never been performed, "My Renesas" dialog opens when downloading the file. Enter your mail address and password registered in the Renesas website.



My Renesas

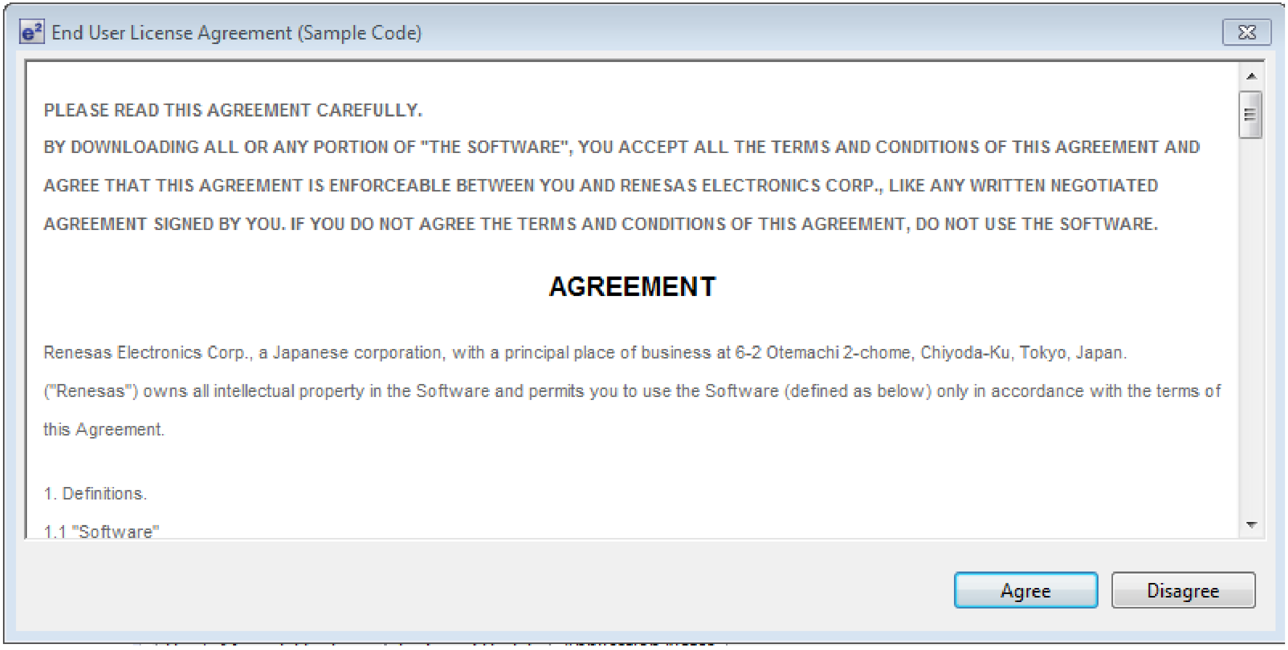
Enter the e-mail address and password that you registered for My Renesas.
They allow you to download documents and software by using Smart Browser.

Email Address:

Password:

Create a My Renesas account to use our tool download services, receive Newsletter / Update Notice, and take advantage of our other services.
Click [About My Renesas] to register it.

6. Click [Agree].



End User License Agreement (Sample Code)

PLEASE READ THIS AGREEMENT CAREFULLY.

BY DOWNLOADING ALL OR ANY PORTION OF "THE SOFTWARE", YOU ACCEPT ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT AND AGREE THAT THIS AGREEMENT IS ENFORCEABLE BETWEEN YOU AND RENESAS ELECTRONICS CORP., LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU. IF YOU DO NOT AGREE THE TERMS AND CONDITIONS OF THIS AGREEMENT, DO NOT USE THE SOFTWARE.

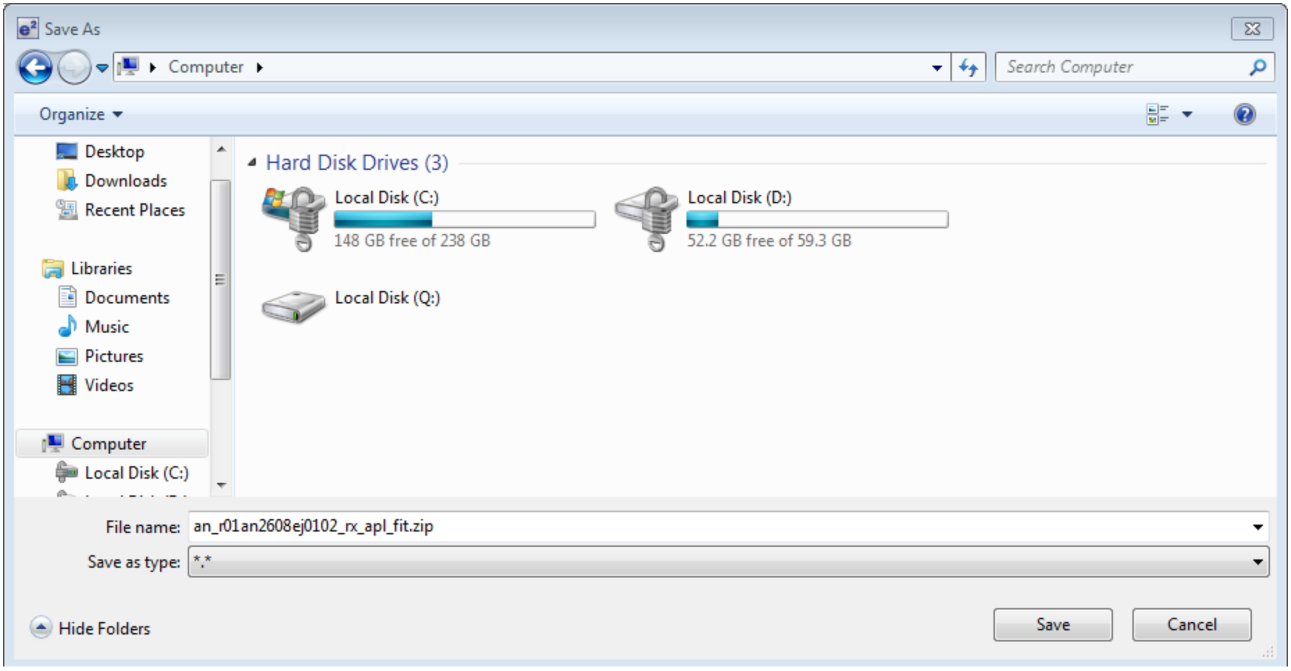
AGREEMENT

Renesas Electronics Corp., a Japanese corporation, with a principal place of business at 6-2 Otemachi 2-chome, Chiyoda-Ku, Tokyo, Japan. ("Renesas") owns all intellectual property in the Software and permits you to use the Software (defined as below) only in accordance with the terms of this Agreement.

1. Definitions.

1.1 "Software"

7. Save the application note.

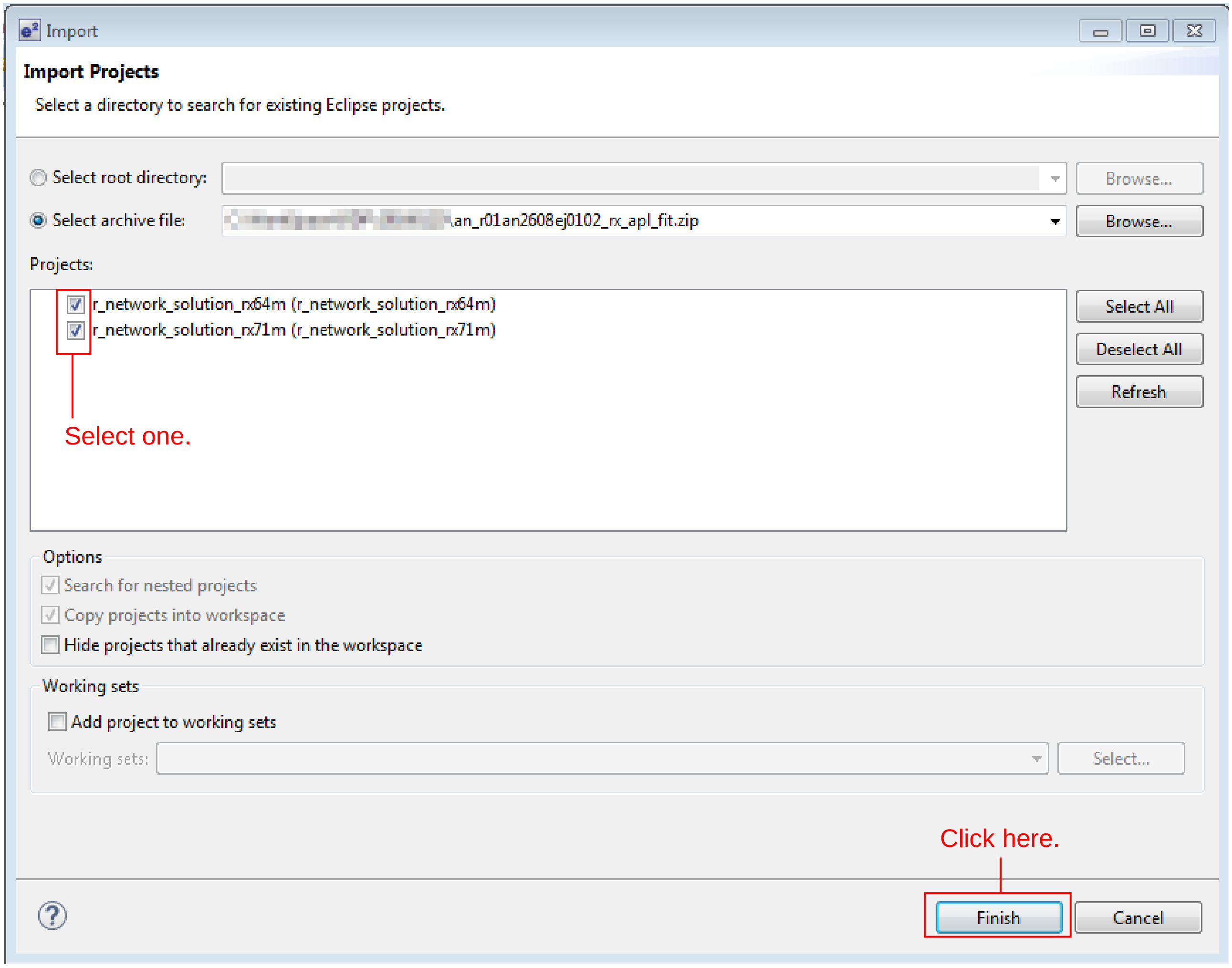


Save As

File name: an_r01an2608ej0102_rx_apl_fit.zip

Save as type: *.*

8. Select either [r_network_solution_rx64m] or [r_network_solution_rx71m] displayed in [Project(P)], and click [Finish(F)].



Import

Import Projects

Select a directory to search for existing Eclipse projects.

○ Select root directory:

● Select archive file: [redacted]\an_r01an2608ej0102_rx_apl_fit.zip

Projects:

☑ r_network_solution_rx64m (r_network_solution_rx64m)
☑ r_network_solution_rx71m (r_network_solution_rx71m)

Select one.

Options

☑ Search for nested projects
☑ Copy projects into workspace
□ Hide projects that already exist in the workspace

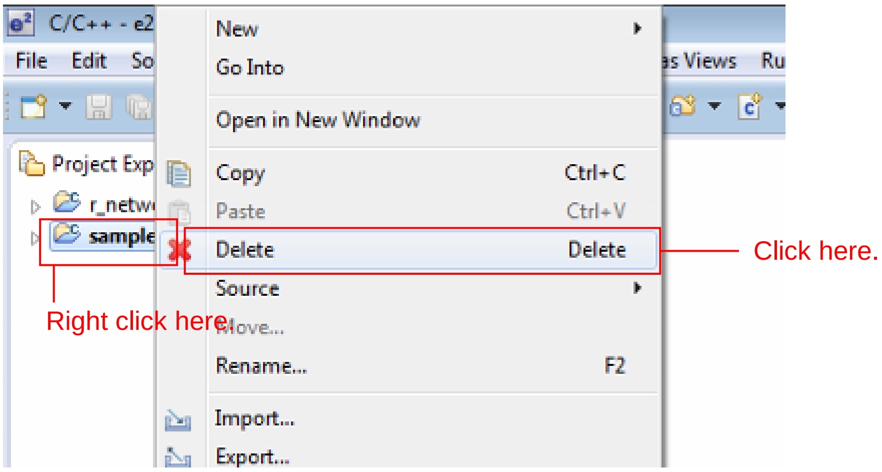
Working sets

□ Add project to working sets

Working sets:

Click here.

9. The figure below shows when [r_network_solution_rx64m] is selected. Delete the project (shown as “sample” here) created to use the Smart browser as this is not required.



Right click here.

Click here.

3.4 Modify Configuration

In this project, the configuration file setting and project setting for each FIT module are changed to configure the application. The detail is shown as follows.

Refer to this information when building new project. To use the project imported, go to “4. Verify Operation”.

3.4.1 Change Configuration

The configuration files for each FIT module configuring this application require modification.

Refer to the manuals and other files in the doc folder for each FIT module for details on the items and the settings in the configuration files.

The places to be changed in the configuration files are shown below.

(1) Change Interrupt Stack Size

In this application, the TCP/IP processing is performed from the Ethernet controller's interrupt handler. This requires about 2.5 KB of interrupt stack.

Modify the interrupt stack size defined in the r_bsp configuration file as shown below.

r_config/r_bsp_config.h

```
/* Interrupt Stack size in bytes. The Renesas RX toolchain sets the stack size using
the #pragma stacksize directive.
 * If the interrupt stack is the only stack being used then the user will likely want
to increase the default size
 * below.
 */
#pragma stacksize si=0x1000
```

(2) Change Compare Match Timer Driver Settings

Set interrupt priority of compare match timer lower than interrupt priority of the USB driver (IPR=3).

r_config/r_cmt_rx_config.h

```
/* The interrupt priority level to be used for CMT interrupts. */
#define CMT_RX_CFG_IPR          (2)
```

(3) Change USB Driver Settings

Set channel 0 to Unused (USB_NOUSE_PP) and set channel 1 to Host (USB_HOST_PP).

r_config/r_usb_basic_config.h

```
/*
    Select USB mode to USB IP(USBb)

    USB_HOST_PP  : USB Host Mode
    USB_PERI_PP  : USB Peripheral Mode
    USB_NOUSE_PP : Not Used (USBb)
*/
    #define USB_FUNCSEL_USBIP0_PP   USB_NOUSE_PP

/*
    Select USB mode to USB IP(USBAa/USBA)

    USB_HOST_PP  : USB Host Mode
    USB_PERI_PP  : USB Peripheral Mode
    USB_NOUSE_PP : Not Used (USBAa/USBA)
*/
    #define USB_FUNCSEL_USBIP1_PP   USB_HOST_PP
```

r_config/r_usb_hmsc_config.h

```
#define USB_MEDIA_INITIALIZE(data1)  R_tfat_disk_initialize((uint8_t)data1)
```

(4) Change T4 Settings

Change the T4 settings as shown below.

Comment out the t4_callback function external reference declaration and add a new external reference declaration for the http_callback, ftp_callback, ftp_data_callback and dns_callback function.

r_t4_rx/src/config_tcpudp.c

```
#include "r_t4_itcpip.h"
//extern ER t4_callback(ID cepid, FN fncd , VP p_parblk);
extern ER http_callback(ID cepid, FN fncd , VP p_parblk);
extern ER ftp_callback(ID cepid, FN fncd , VP p_parblk);
extern ER ftp_data_callback(ID cepid, FN fncd , VP p_parblk);
extern ER dns_callback(ID cepid, FN fncd , VP p_parblk);
```

Change the TCP reception point setting as shown below.

r_t4_rx/src/config_tcpudp.c

```
/***  Definition of TCP reception point (only port number needs to be set) ***/
T_TCP_CREP tcp_crep[18] =
{
    /* HTTP server use setting below. */
    { 0x0000, { 0, 80 }},
    { 0x0000, { 0, 80 }},
    { 0x0000, { 0, 80 }},
    { 0x0000, { 0, 80 }},
    { 0x0000, { 0, 80 }},
    { 0x0000, { 0, 80 }},
/* FTP server use setting below. */
    { 0x0000, { 0, 21 }},
    { 0x0000, { 0, 20 }},     /* The port number in the active mode is 20, and the
port number in the passive mode is 1024. */
    { 0x0000, { 0, 21 }},
    { 0x0000, { 0, 20 }},     /* The port number in the active mode is 20, and the
port number in the passive mode is 1025. */
    { 0x0000, { 0, 21 }},
    { 0x0000, { 0, 20 }},     /* The port number in the active mode is 20, and the
port number in the passive mode is 1026. */
    { 0x0000, { 0, 21 }},
    { 0x0000, { 0, 20 }},     /* The port number in the active mode is 20, and the
port number in the passive mode is 1024. */
    { 0x0000, { 0, 21 }},
    { 0x0000, { 0, 20 }},     /* The port number in the active mode is 20, and the
port number in the passive mode is 1025. */
    { 0x0000, { 0, 21 }},
    { 0x0000, { 0, 20 }},     /* The port number in the active mode is 20, and the
port number in the passive mode is 1026. */
};
```

Change the TCP communication end point setting as shown below.

r_t4_rx/src/config_tcpudp.c

```
/***  Definition of TCP communication end point
      (only receive window size needs to be set) ***/
T_TCP_CCEP tcp_ccep[18] =
{
    /* { attribute of TCP communication end point,
         top address of transmit window buffer, size of transmit window buffer,
         top address of receive window buffer, size of receive window buffer,
         address of callback routine }
    */
/* HTTP server use setting below. */
    { 0, 0, 0, 0, 1460, http_callback },
    { 0, 0, 0, 0, 1460, http_callback },
    { 0, 0, 0, 0, 1460, http_callback },
    { 1, 0, 0, 0, 1460, http_callback },
    { 1, 0, 0, 0, 1460, http_callback },
    { 1, 0, 0, 0, 1460, http_callback },
/* FTP server use setting below. */
{ 0, 0, 0, 0, 128, ftp_callback },              /* FTP control connection    */
{ 0, 0, 0, 0, 1460, ftp_data_callback },        /* FTP data connection       */
{ 0, 0, 0, 0, 128, ftp_callback },              /* FTP control connection    */
{ 0, 0, 0, 0, 1460, ftp_data_callback },        /* FTP data connection       */
{ 0, 0, 0, 0, 128, ftp_callback },              /* FTP control connection    */
{ 0, 0, 0, 0, 1460, ftp_data_callback },        /* FTP data connection       */
{ 1, 0, 0, 0, 128, ftp_callback },              /* FTP control connection    */
{ 1, 0, 0, 0, 1460, ftp_data_callback },        /* FTP data connection       */
{ 1, 0, 0, 0, 128, ftp_callback },              /* FTP control connection    */
{ 1, 0, 0, 0, 1460, ftp_data_callback },        /* FTP data connection       */
{ 1, 0, 0, 0, 128, ftp_callback },              /* FTP control connection    */
{ 1, 0, 0, 0, 1460, ftp_data_callback },        /* FTP data connection       */
};
```

Change the 2MSL Wait Time from 1 min. to 10 ms.

r_t4_rx/src/config_tcpudp.c

```
/***  2MSL wait time (unit:10ms)  ***/
const UH    _tcp_2msl[] =
{
    (1),      /* 10 ms */
    (1),      /* 10 ms */
};
```

Change the UDP communication end point setting as shown below.

r_t4_rx/src/config_tcpudp.c

```
/***  Definition of UDP communication end point  ***/
T_UDP_CCEP udp_ccep[1] =
{
    /* only setting port number */
    { 0x0000, { 0, 1365 }, dns_callback },
};
```

(5) Change HTTP Server Settings

Set the MAX_CGI_FILE to 2.

r_config/r_t4_http_server_rx_config.h

```
// If user uses CGI function, please specify cgi file name, and number of cgi files.
#define MAX_CGI_FILE        2
```

Change the CGI_FILE_NAME_TABLE_LIST as shown below.

r_config/r_t4_http_server_rx_config.h

```
/*#define CGI_FILE_NAME_TABLE_LIST */
/*  {"cgi_smpl.cgi", NULL},   */
extern ER cgi_sample_function(ID cepid, void *res_info);
extern ER cgi_dns_demo_function(ID cepid, void *res_info);
extern ER cgi_dns_demo_pending_release(ID cepid, void *res_info);
#define CGI_FILE_NAME_TABLE_LIST
    {"cgi_smpl.cgi", cgi_sample_function, NULL},
{"dns_demo.cgi", cgi_dns_demo_function, cgi_dns_demo_pending_release},
```

Change the number of communication endpoints used in the HTTP server to 6 to match tcp_ccep table of r_t4_rx/src/config_tcpudp.c.

r_config/r_t4_http_server_rx_config.h

```
// set same value number of CEPID in config_tcpudp.c
#define   HTTP_TCP_CEP_NUM 6
```

(6) Change DNS client Settings

Change IP address of the DNS server used to verify the operation (Following is example).

r_config/r_t4_dns_client_rx_config.h

```
/**************************************************************************
Configuration Options
**************************************************************************/
//#define DNS_IP_ADDR_1   172,30,11,5      /* DNS server IP address (primary) */
//#define DNS_IP_ADDR_2   10,29,72,33      /* DNS server IP address (secondary) */
#define DNS_IP_ADDR_1     10,164,247,5     /* DNS server IP address (primary) */
#define DNS_IP_ADDR_2     10,29,72,33      /* DNS server IP address (secondary) */
```

(7) Change FTP server Settings

Change the position of the TCP communication end point used in FTP server to match TCP communication end point setting of the T4.

r_config/r_t4_ftp_server_rx_config.h

```
#include "r_t4_http_server_rx_config.h"
/**************************************************************************
Configuration Options
**************************************************************************/
#define   FTP_TCP_CEP_NUM     12     // Adapt myself to the communication endpoint
number of config_tcpudp.c

#define   FTP_START_TCP_CEP   HTTP_TCP_CEP_NUM    // starting position of the
communication endpoint in config_tcpudp.c
```

(8) Change File driver for FTP server and Web server Settings

Change the setting to use the USB.

r_config/r_t4_file_driver_rx_config.h

```
/* Only one is effective.
   (priority is given to SERVER_FILE_DRV_USE_EXTERNAL)

   Setting to 1 : used
              0 : unused
*/
#define SERVER_FILE_DRV_USE_EXTERNAL    (1)    // USB, SD card etc
#define SERVER_FILE_DRV_USE_INTERNAL    (0)    // internal-RAM / external-SDRAM
```

(9) Change M3S-TFAT-Tiny Memory Driver Settings

Change the setting to use the USB.

r_config/r_tfat_driver_rx_config.h

```
/* Number of logical drives to be used.
   Setting to 0     : unused memory
              other : number of logical drives
   (USB and SDHI can be used together.)
*/
#define TFAT_USB_DRIVE_NUM          (1)
#define TFAT_SDHI_DRIVE_NUM         (0)
#define TFAT_USB_MINI_DRIVE_NUM    (0)

/* alocate a drive number
  <valid define>
  TFAT_CTRL_USB  : for USB
  TFAT_CTRL_SDHI : for SDHI
  NULL           : unallocated drive

  MAX 10 drives(TFAT module spec)

  ex.)
      #define TFAT_DRIVE_ALLOC_NUM_0    TFAT_CTRL_USB
      #define TFAT_DRIVE_ALLOC_NUM_1    TFAT_CTRL_SDHI
      #define TFAT_DRIVE_ALLOC_NUM_2    TFAT_CTRL_SDHI
      #define TFAT_DRIVE_ALLOC_NUM_3    TFAT_CTRL_USB
*/
#define TFAT_DRIVE_ALLOC_NUM_0    TFAT_CTRL_USB
#define TFAT_DRIVE_ALLOC_NUM_1    NULL
```

(10) Change Ethernet Settings

Change the PHY IC channel used to control the PHY IC form Ethernet controller to ch0.

r_config/r_ether_rx_config.h

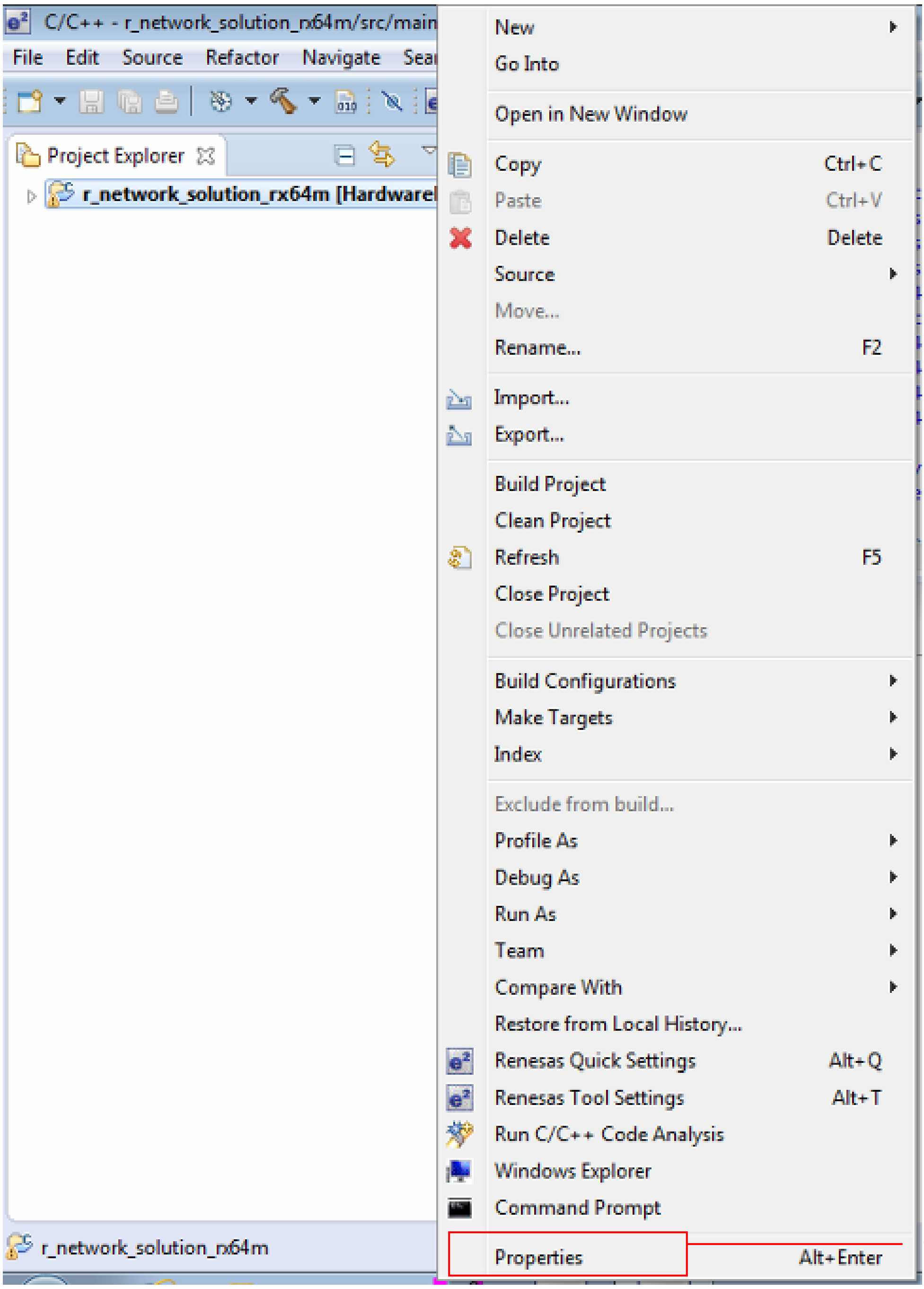
```
/* The register bus of PHY0/1 for ETHER0/1 select
   0 = The access of the register of PHY uses ETHER0.
   1 = The access of the register of PHY uses ETHER1.
*/
#define ETHER_CFG_CH0_PHY_ACCESS        (0)
#define ETHER_CFG_CH1_PHY_ACCESS        (0)
```

3.4.2 Change Project Setting

The setting when building is changed from default setting to the content shown in Table 3-1 Changed build setting.

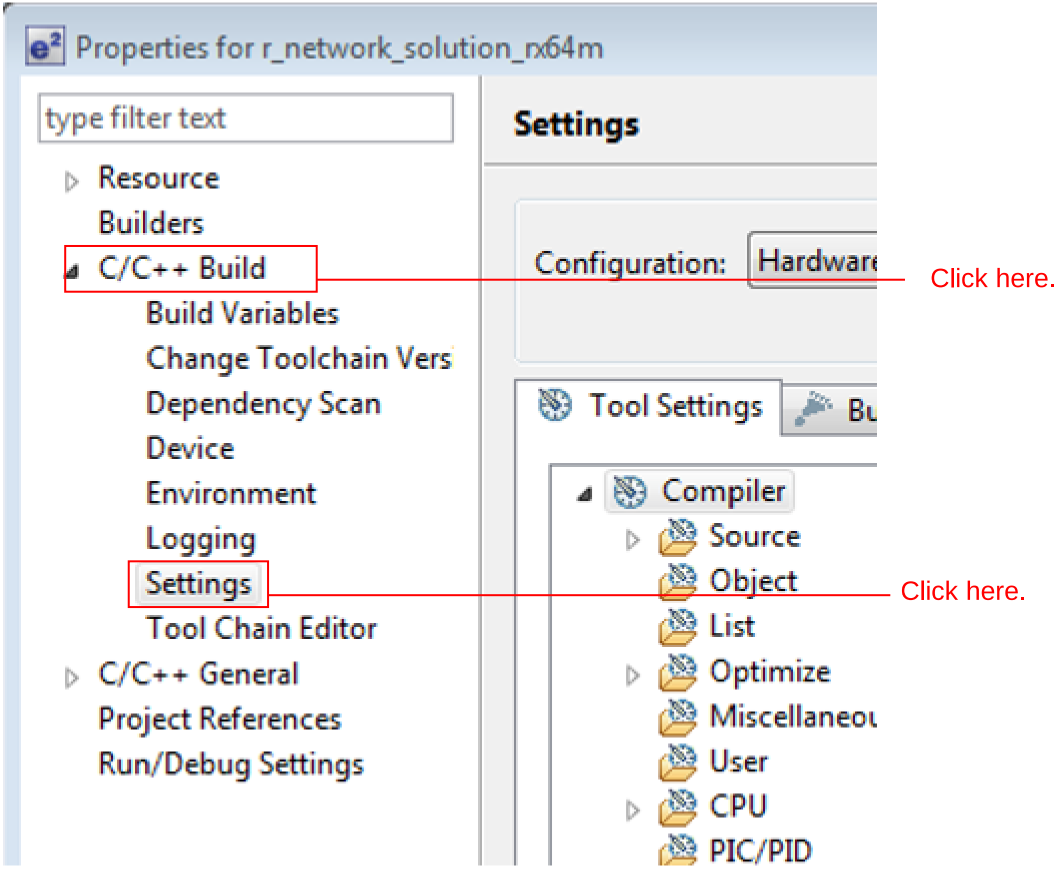
To check the project setting, use the following procedure.

1. Select the project for the e² studio ([r_network_solution_rx64m] or [r_network_solution_rx71m]) and right-click. Then, click [Property (R)].



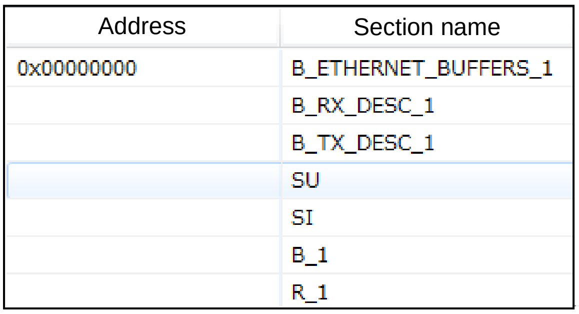
Click here.

2. Click on [C/C++ Build], and [Setting].



3. Refer to the contents of Table 3-1 Changed build setting on the [Tool Setting] tab.

Table 3-1 Changed build setting

Items	Changed contents	Description
Compiler - Source	Add include path to "Include file directory"	Set an include path that requires the setting for each FIT module. It is automatically set when including each FIT module using "FIT Configurator".
Linker - Input	For Little endian : Add "${workspace_loc:/${ProjName}/r_tfa t_rx/lib/tfat_rx600_little.lib}" (Note 1) For Big endian: Add "${workspace_loc:/${ProjName}/r_tfa t_rx/lib/tfat_rx600_big.lib}" (Note 1)	Setting is required when using TFAT. (required when using FAT file system driver) It's not automatically set even when using "FIT Configurator". Setting by the user is required when creating new project. (Note 2)
	For Little endian: Add "${workspace_loc:/${ProjName}/r_t4_rx/lib/T4_Library_rxv2_ether_little.lib}" (Note 1) For Big endian: Add "${workspace_loc:/${ProjName}/r_t4_rx/lib/T4_Library_rxv2_ether_big.lib}" (Note 1)	Setting is required when using Ethernet. It's not automatically set even when using "FIT Configurator". Setting by the user is required when creating new project. (Note 2)
Linker - Section	Add B_ETHERNET_BUFFERS_1 section, B_RX_DESC_1 section, and B_TX_DESC_1 section to RAM capacity	Setting is required when using Ethernet. It's not automatically set even when using "FIT Configurator". Setting by the user is required when creating new project. <Setting examples> 

Address	Section name
0x00000000	B_ETHERNET_BUFFERS_1
	B_RX_DESC_1
	B_TX_DESC_1
	SU
	SI
	B_1
	R_1

(Note 1) The setting change is required when creating the project that includes each FIT module. For the setting, refer to the manuals, etc. in the doc folder of each FIT module.

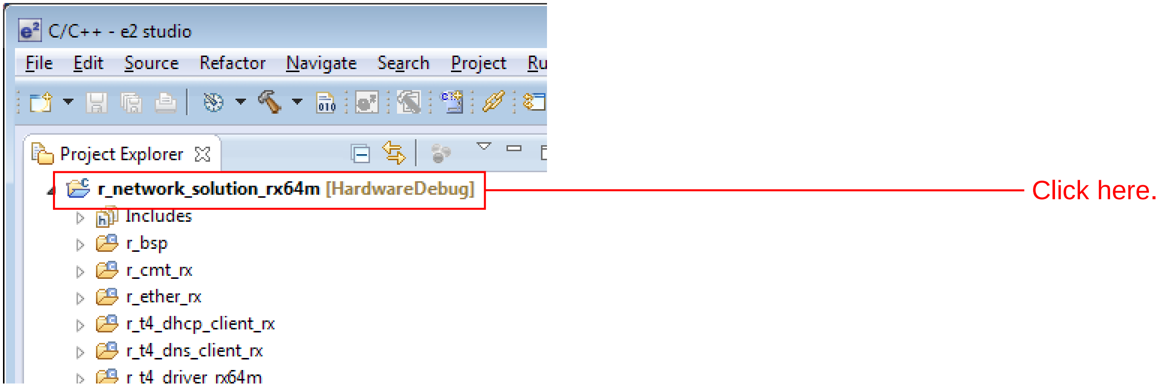
(Note 2) Automatic setting is available in e^2 studio V4.1 or later. However, as the FIT module used this time does not meet the specification, the automatic setting function is not available. This will be supported in the latest FIT module.

4. Verify Operation

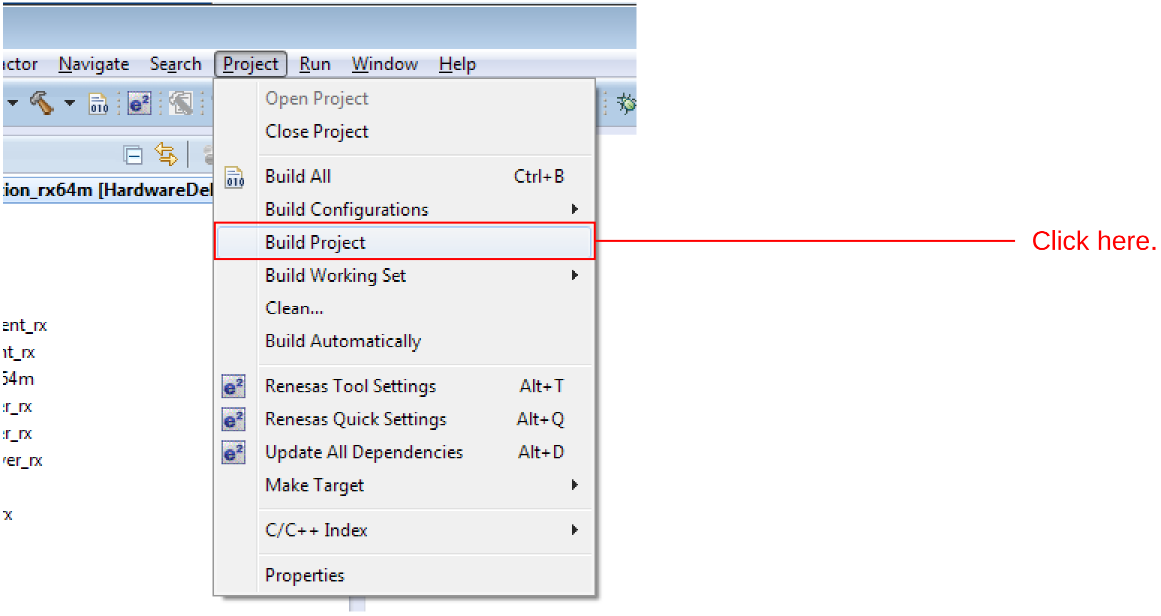
4.1 Build the Project

Use the following procedure to build the project and generate a load module.

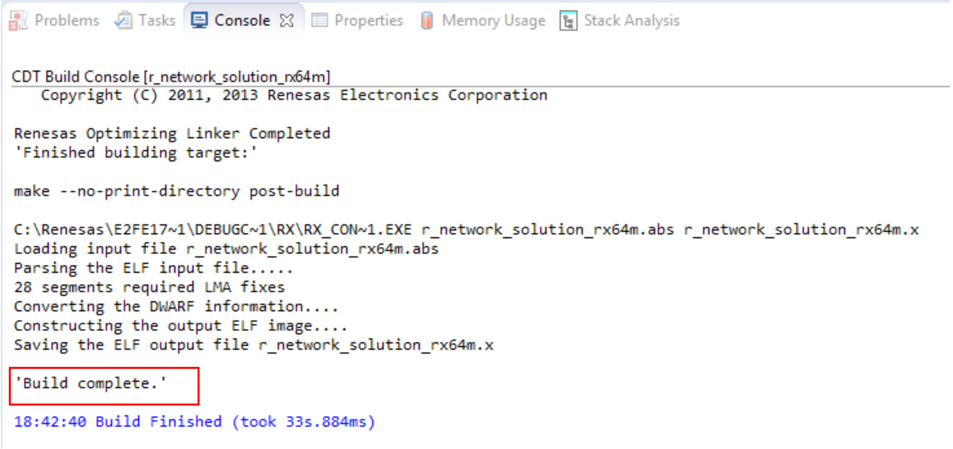
1. Click the project to build from the **Project Explorer**.



2. Click [Build project] from the **Project** menu.



3. When “Build complete” is displayed on the **Console panel**, the build will have completed.



CDT Build Console [r_network_solution_rx64m]

```
   Copyright (C) 2011, 2013 Renesas Electronics Corporation

Renesas Optimizing Linker Completed
'Finished building target:'

make --no-print-directory post-build

C:\Renesas\E2FE17~1\DEBUGC~1\RX\RX_CON~1.EXE r_network_solution_rx64m.abs r_network_solution_rx64m.x
Loading input file r_network_solution_rx64m.abs
Parsing the ELF input file.....
28 segments required LMA fixes
Converting the DWARF information....
Constructing the output ELF image....
Saving the ELF output file r_network_solution_rx64m.x

'Build complete.'

18:42:40 Build Finished (took 33s.884ms)
```

4.2 Prepare for Debugging

4.2.1 Configure Hardware

The evaluation board must be configured before starting debugging.

A table of the required equipment and its configuration are shown below.

Table 4-1 Hardware Configuration

No.	Device	Supplementary Information
1	Development PC	PC used for development
2	Evaluation board Renesas Starter Kit+ for RX64M or Renesas Starter Kit+ for RX71M	
3	USB memory	Memory that is formatted as either FAT or FAT32.
4	Client PC (Web browser / FTP Client)	The development PC can be used for this function.
5	DHCP server, DNS server	When verify operation of the DHCP client and DNS client.
6	One of the following must be provided as a network environment. • Router × 1 • Switching hub × 1 • LAN cable (straight or cross) × 1 to 3	For example of the network environment, Please refer to the following.

If you have a DNS server and DHCP server

Router Configuration (with DHCP server function)

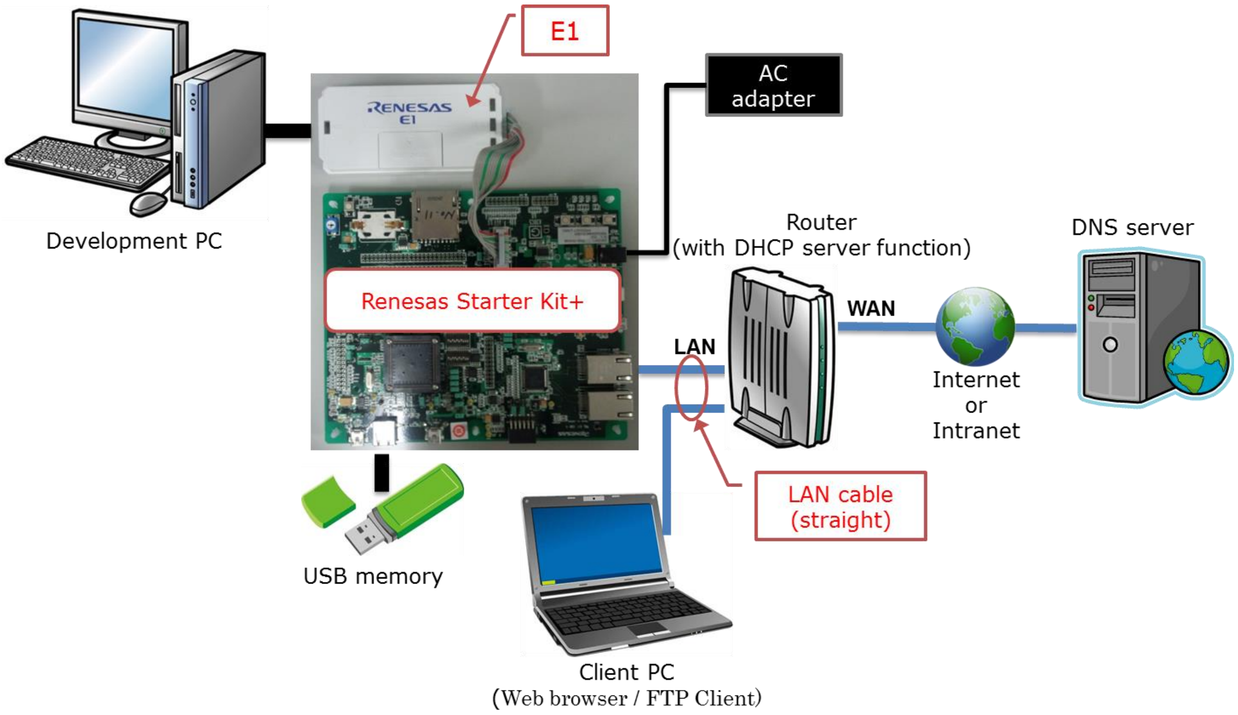


Figure 4-1 DNS server and DHCP server - Router Configuration (with DHCP server function)

If you have a DNS server and DHCP server

Switching Hub Configuration

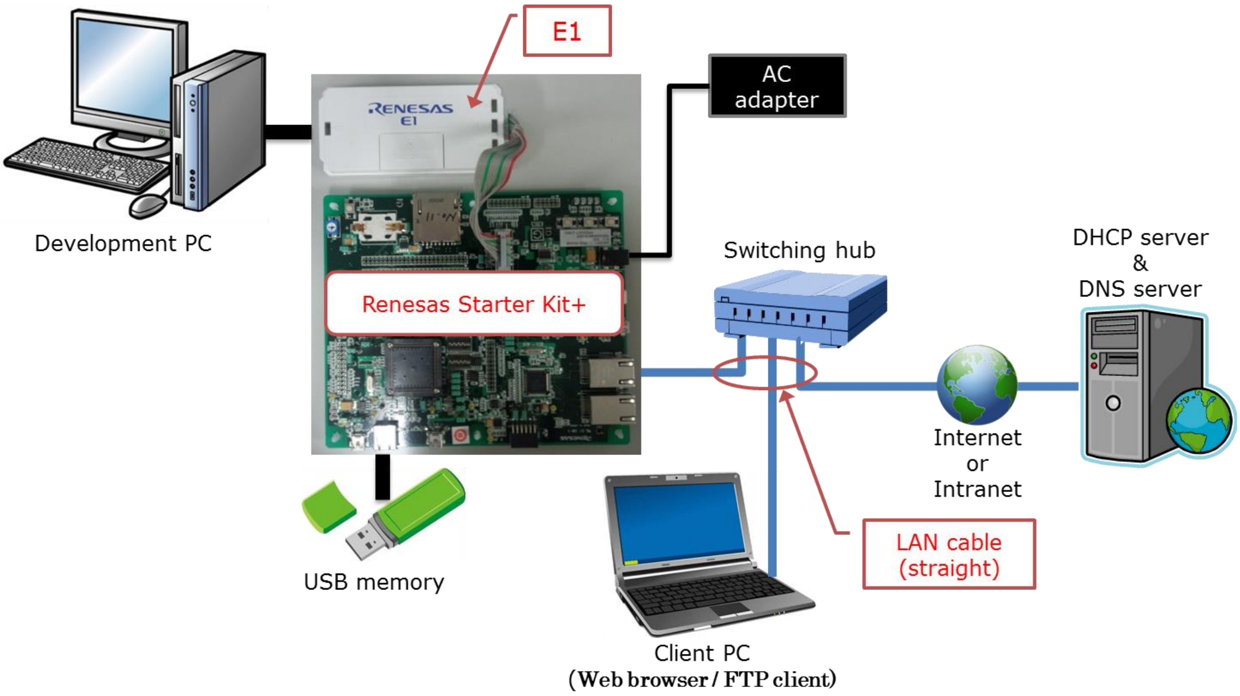


Figure 4-2 DNS server and DHCP server – Switching Hub Configuration

If you don’t have a DNS server and DHCP server

Switching Hub Configuration

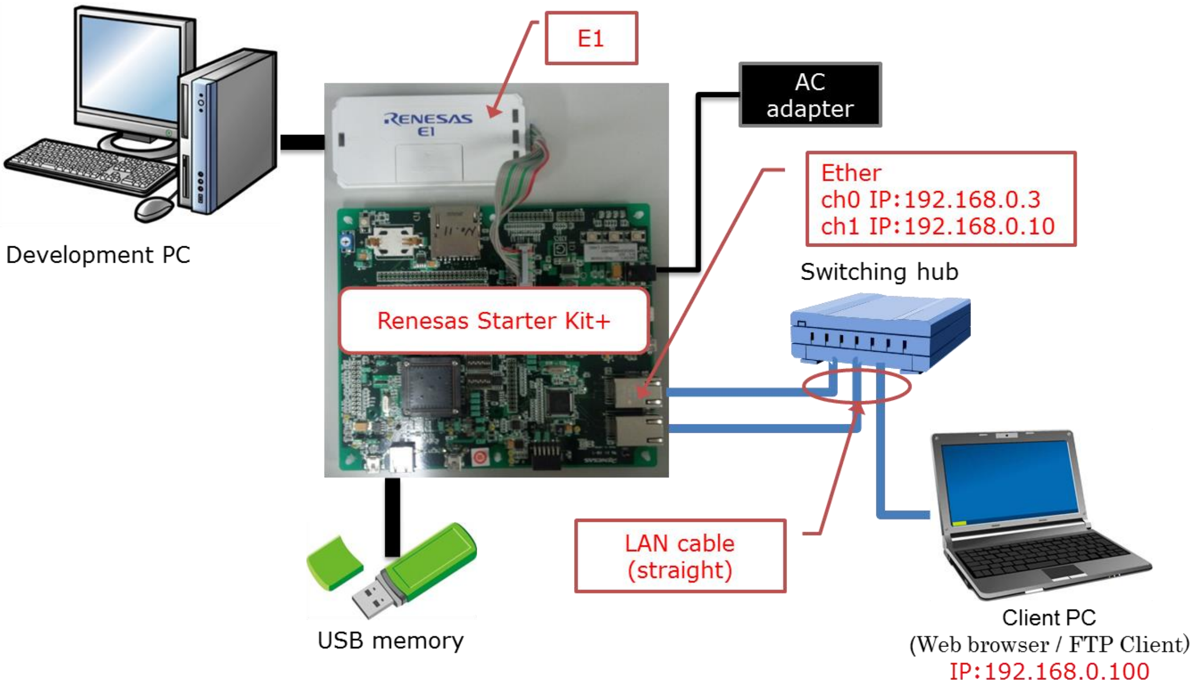


Figure 4-3 No DNS server and No DHCP server – Switching Hub Configuration

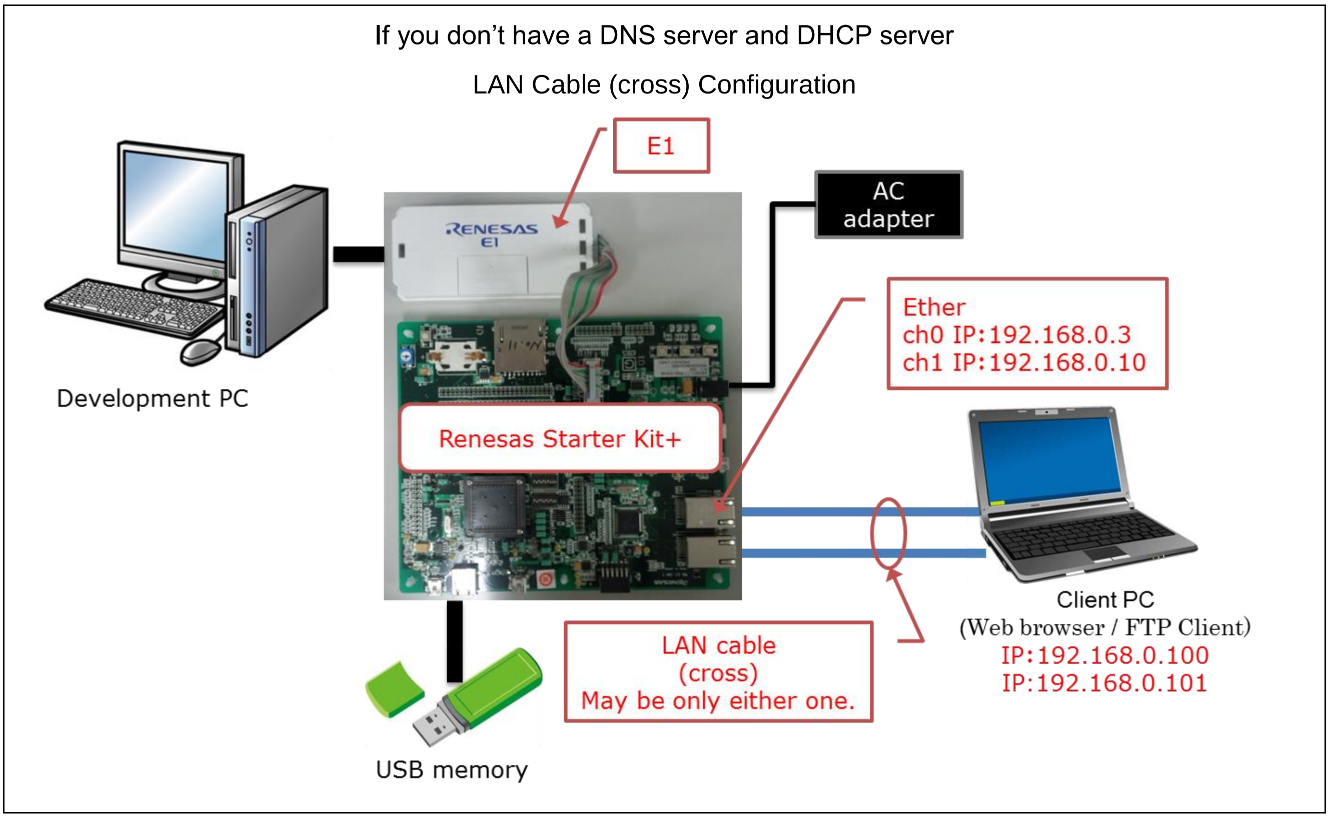


Figure 4-4 No DNS server and No DHCP server – LAN Cable (cross) Configuration

4.2.2 Set Up the Evaluation Board

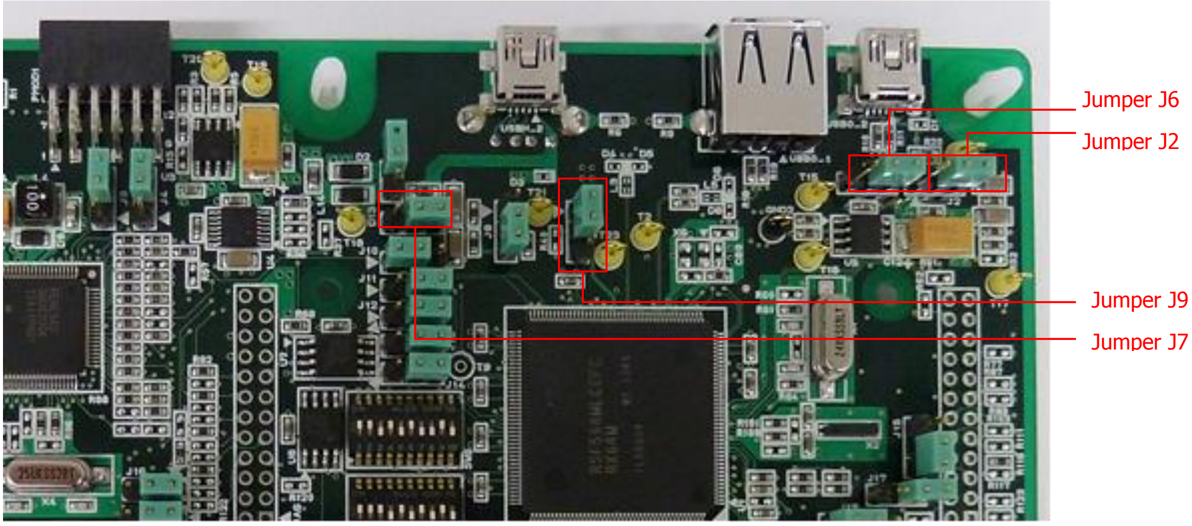
The evaluation board settings required to operate this application are shown below.

(1) Set Up USB Configuration

1. Set the USB ch0 mode (host/peripheral). Set jumpers J2 and J6 to match the setting of USB_FUNCSEL_USBIP0_PP in r_usb_config.h.
2. Set the USB ch1 mode (host/peripheral). Set jumpers J7 and J9 to match the setting of USB_FUNCSEL_USBIP1_PP in r_usb_config.h.

Table 4-2 Jumper Settings

No.	Setting	Jumper	Setting contents	The setting used to verify the operation
1	When use USB0 in host mode.	J2	Short 1 to 2.	Unused
	(USB_FUNCSEL_USBIP0_PP = USB_HOST_PP)	J6	Short 2 to 3.	Unused
	When use USB0 in peripheral mode.	J2	Short 2 to 3.	Unused
	(USB_FUNCSEL_USBIP0_PP = USB_PERI_PP)	J6	Short 1 to 2.	Unused
2	When use USB1 in host mode.	J7	Short 1 to 2.	Used
	(USB_FUNCSEL_USBIP1_PP = USB_HOST_PP)	J9	Short 2 to 3.	Used
	When use USB1 in peripheral mode.	J7	Short 2 to 3.	Unused
	(USB_FUNCSEL_USBIP1_PP = USB_PERI_PP)	J9	Short 1 to 2.	Unused



Note: The image and the actual settings differ.

Figure 4-5 Renesas Starter Kit+ for RX64M Jumper Locations [USB]

(2) Set Up Ethernet Configuration

1. Specify the PHY IC channel used to control the PHY IC from the Ethernet controller. Set jumpers J3 and J4 to match the settings of ETHER_CFG_CH0_PHY_ACCESS and ETHER_CFG_CH1_PHY_ACCESS in r_ether_rx_config.h.
2. If Ethernet controller (ET0) is used, set the switches and jumpers.
3. If Ethernet controller (ET1) is used, set the jumpers.

Table 4-3 Ethernet Setting list

No.	Setting contents	Jumper & Switch		The setting used to verify the operation
		No.	Setting contents	
1	Control PHY IC with ch0	J3	Short 1 to 2	Used
		J4	Short 1 to 2	
	Control PHY IC with ch1	J3	Short 2 to 3	Unused
		J4	Short 2 to 3	
2	When use Ethernet Controller (ET0)	J10	Short 1 to 2	Used
		J11	Short 2 to 3	
		J12	Short 2 to 3	
		J13	Short 2 to 3	
		J14	Short 2 to 3	
		SW5	1 On	
			2 On	
			3 Off	
			4 On	
			5 Off	
			6 On	
			7 Off	
			8 Off	
			9 On	
			10 Off	
		SW6	1 On	
			2 On	
			3 On	
			4 Off	
			5 On	
			6 Off	
			7 On	
			8 Off	
			9 On	
			10 Off	
		SW7	1 On	
			2 Off	
			3 Off	
			4 On	
			5 Off	
			6 Off	
			7 On	
			8 Off	
			9 On	
			10 Off	
3	When use Ethernet Controller (ET1)	J16	Short 1 to 2	Unused
		J18	Short 1 to 2	
		J20	Short 1 to 2	

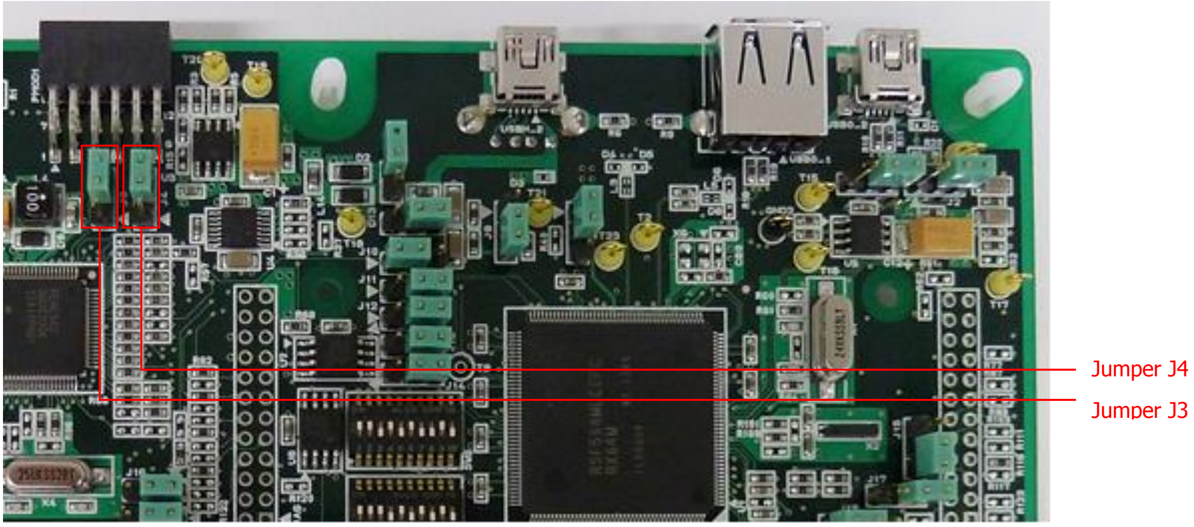
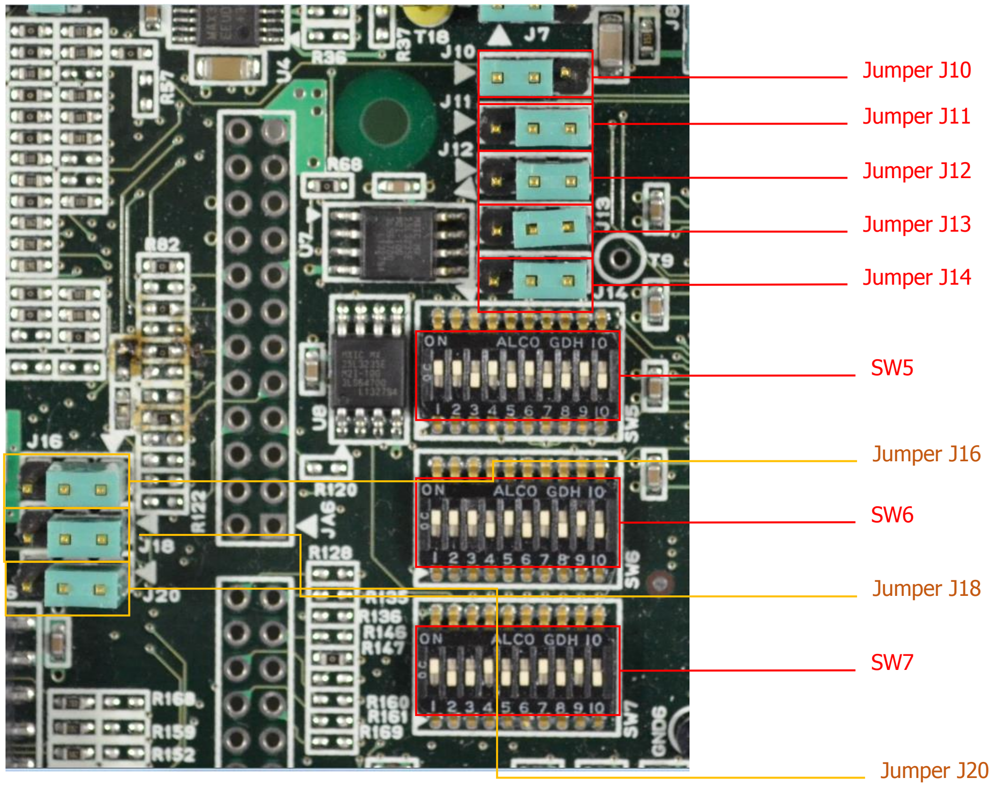


Figure 4-6 Renesas Starter Kit+ for RX64M/RX71M Jumper locations [Ethernet Controller PHY IC]



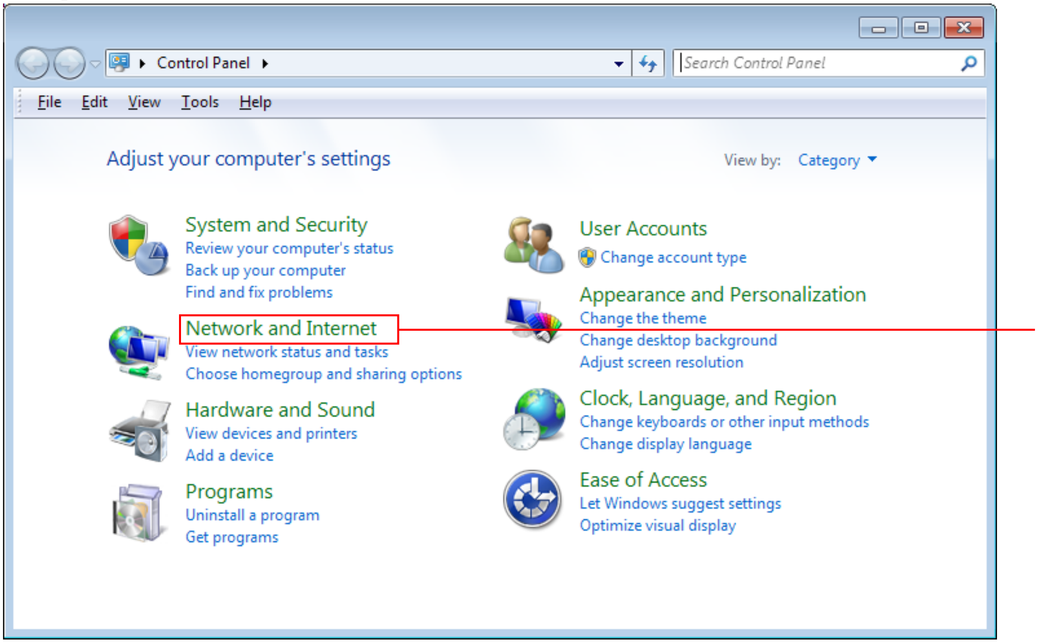
(Note) ET0 : Red、ET1 : Orange

Figure 4-7 Renesas Starter Kit+ for RX64M/RX71M Jumper locations [Ethernet Controller ET0/ET1]

4.2.3 Set Up Client PC

Set up the network on the client PC. This section shows the procedure when using Windows 7 as an example.

1. Open the Control Panel on the client PC and click [Network and Internet].



Control Panel ▸ | Search Control Panel

File Edit View Tools Help

Adjust your computer's settings — View by: Category

System and Security
Review your computer's status
Back up your computer
Find and fix problems

Network and Internet
View network status and tasks
Choose homegroup and sharing options

Hardware and Sound
View devices and printers
Add a device

Programs
Uninstall a program
Get programs

User Accounts
Change account type

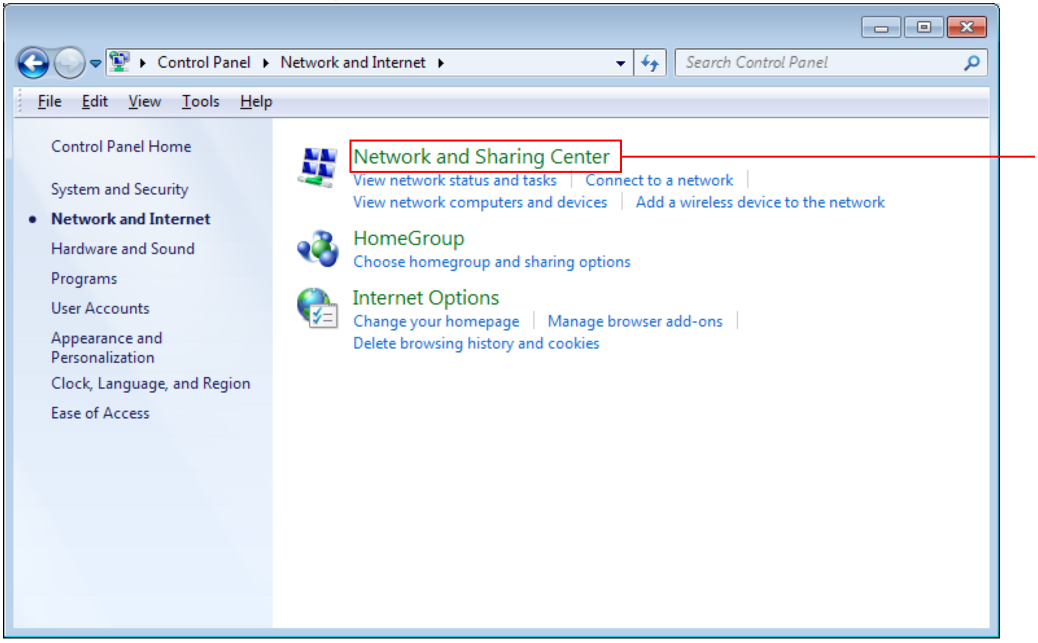
Appearance and Personalization
Change the theme
Change desktop background
Adjust screen resolution

Clock, Language, and Region
Change keyboards or other input methods
Change display language

Ease of Access
Let Windows suggest settings
Optimize visual display

Click here.

2. Click [Network and Sharing Center].



Control Panel ▸ Network and Internet ▸ | Search Control Panel

File Edit View Tools Help

Control Panel Home
System and Security
● Network and Internet
Hardware and Sound
Programs
User Accounts
Appearance and Personalization
Clock, Language, and Region
Ease of Access

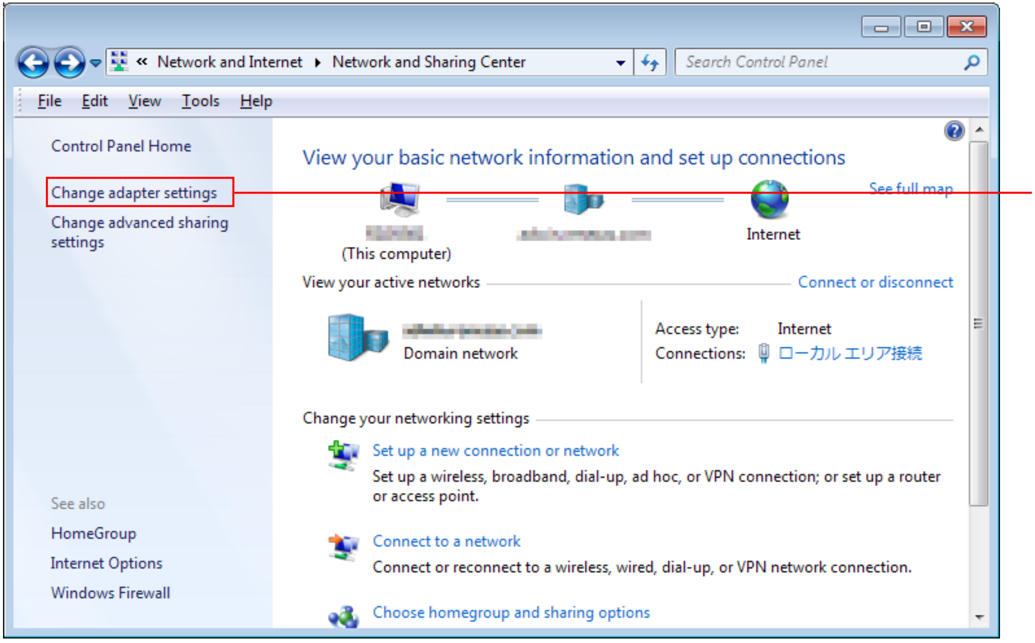
Network and Sharing Center
View network status and tasks | Connect to a network | View network computers and devices | Add a wireless device to the network

HomeGroup
Choose homegroup and sharing options

Internet Options
Change your homepage | Manage browser add-ons | Delete browsing history and cookies

Click here.

3. Click [Change adapter settings].


« Network and Internet ▸ Network and Sharing Center
Search Control Panel
File Edit View Tools Help
Control Panel Home
Change adapter settings
Change advanced sharing settings
See also
HomeGroup
Internet Options
Windows Firewall
See full map
[redacted]
(This computer)
[redacted]
Internet
View your active networks
Connect or disconnect
[redacted]
Domain network
Access type: Internet
Connections: ローカル エリア接続
Change your networking settings
Set up a new connection or network
Connect to a network
Choose homegroup and sharing options

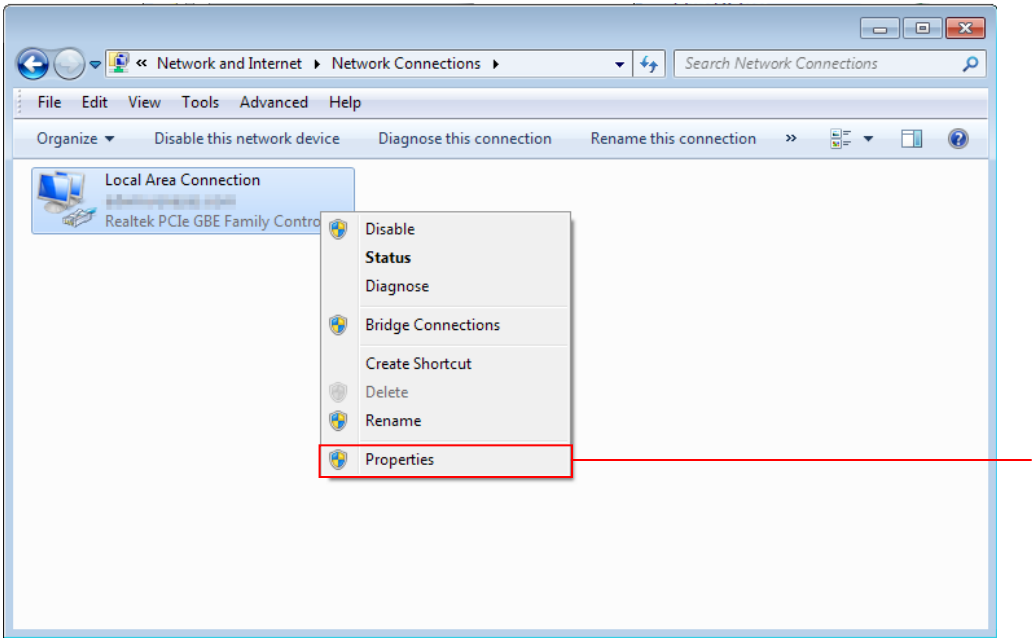
View your basic network information and set up connections

Set up a wireless, broadband, dial-up, ad hoc, or VPN connection; or set up a router or access point.

Connect or reconnect to a wireless, wired, dial-up, or VPN network connection.

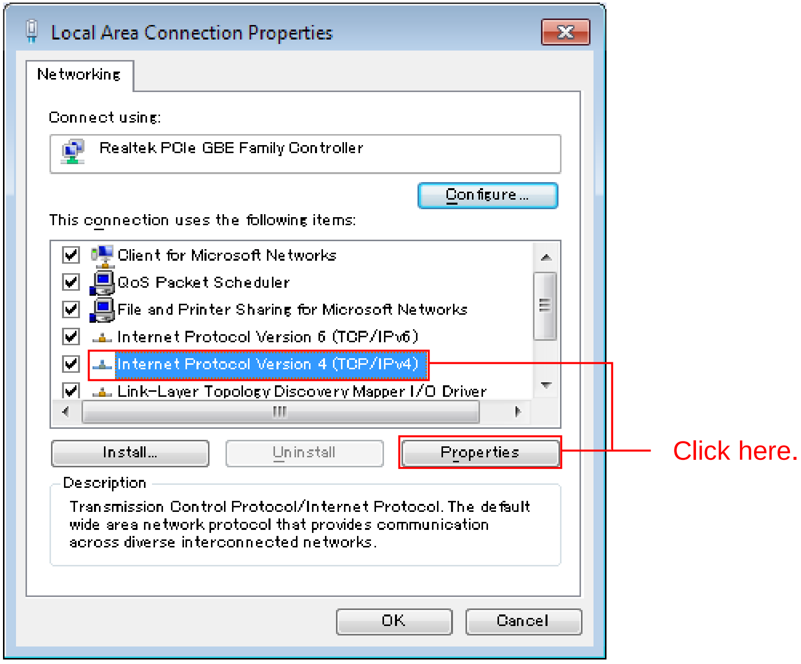
Click here.

4. Right click [Local Area Connection] and select [Properties].



Click here.

5. Select [Internet Protocol Version 4 (TCP/IPv4)] and click [Properties].



Local Area Connection Properties

Networking

Connect using:

Realtek PCIe GBE Family Controller

This connection uses the following items:

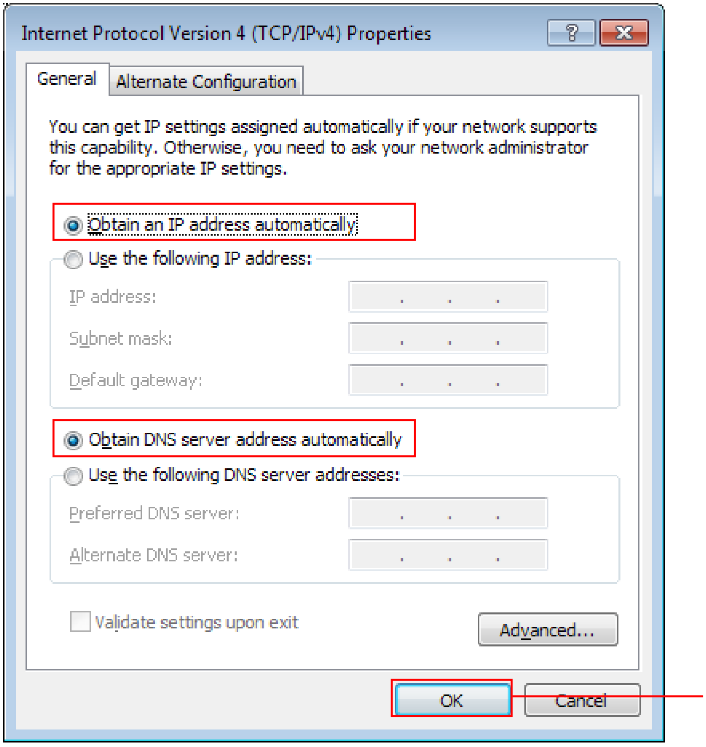
- ☑ Client for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ Internet Protocol Version 6 (TCP/IPv6)
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- ☑ Link-Layer Topology Discovery Mapper I/O Driver

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

6. The IP address and other settings will be displayed. Set these as shown below and click [OK] depending on the existence of a DHCP server.

(a) If exist the DHCP server (Obtain an IP address automatically).



Internet Protocol Version 4 (TCP/IPv4) Properties

General | Alternate Configuration

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

● Obtain an IP address automatically

○ Use the following IP address:

IP address: . . .

Subnet mask: . . .

Default gateway: . . .

● Obtain DNS server address automatically

○ Use the following DNS server addresses:

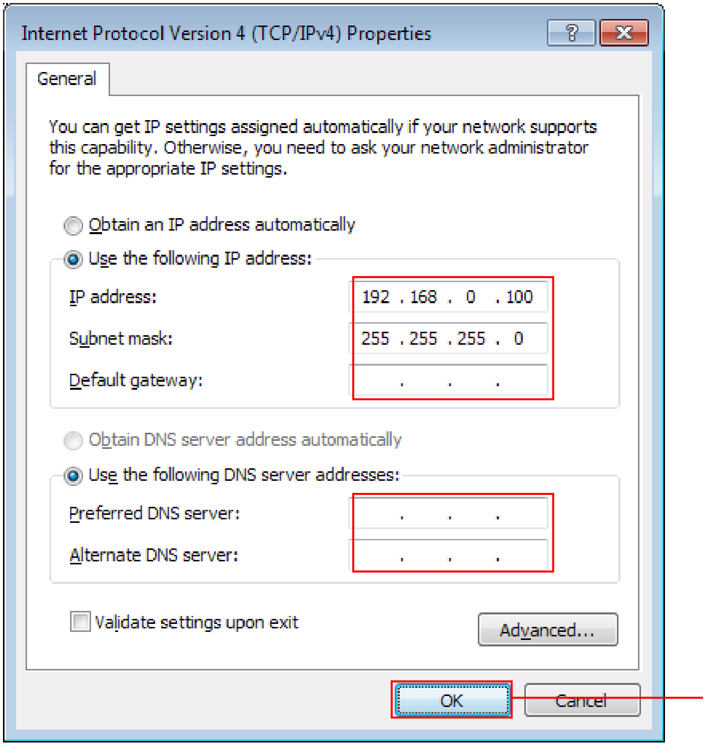
Preferred DNS server: . . .

Alternate DNS server: . . .

□ Validate settings upon exit

Click here.

(b) If the DHCP server does not exist (Fixed IP address).



Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 0 . 100

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

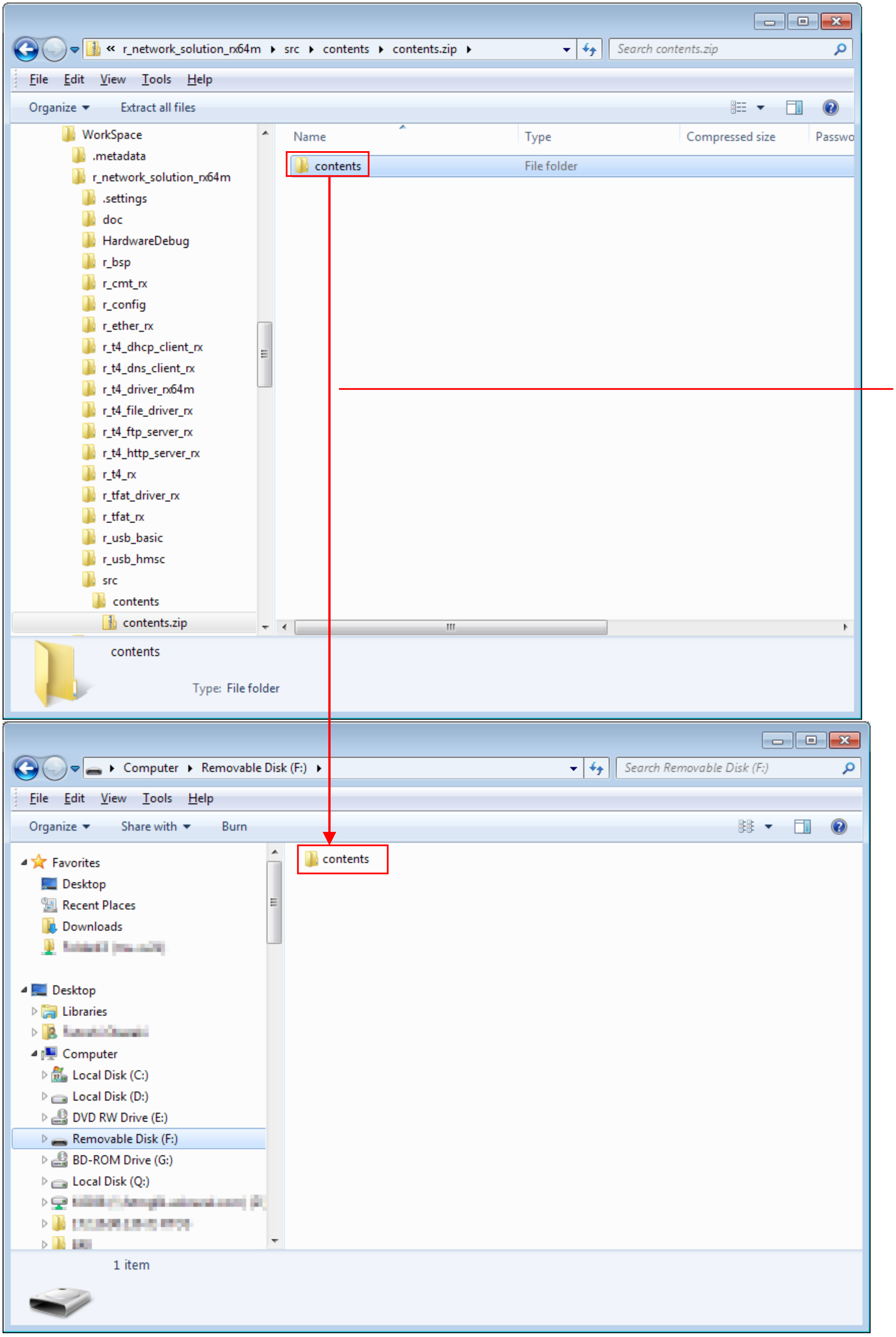
□ Validate settings upon exit

Click here.

4.2.4 Prepare USB Memory

Store the HTML content on the USB memory.

1. Open the **src** folder in the project, then open the **contents** folder in the folder. Open the **contents.zip** file in the **contents** folder. Copy the **contents** folder to the USB memory.



Copy the contents folder.

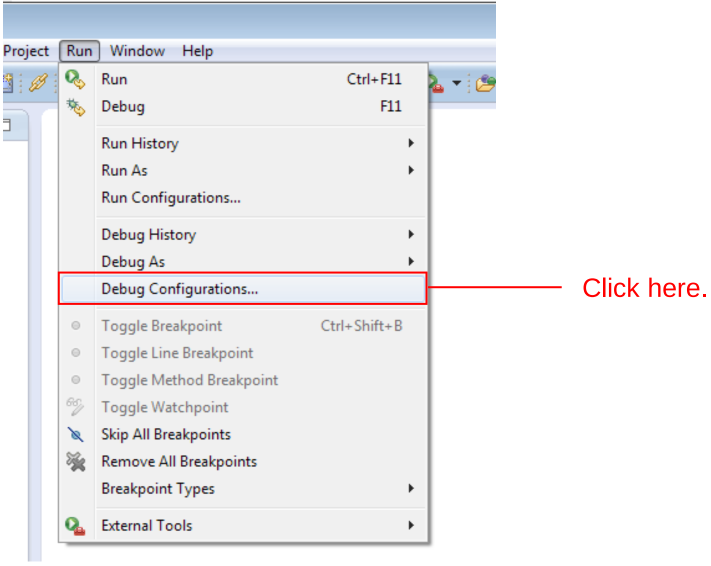
4.3 Debug the Project

Use the following procedure to start debugging the project.

1. Connect the development PC to the E1 emulator with a USB cable.

2. Connect the evaluation board (Renesas Starter Kit+ for RX64M or Renesas Starter Kit+ for RX71M) to the adapter and turn on the power.

3. Click [Debug Configurations] in the e^2 studio **Run** menu.



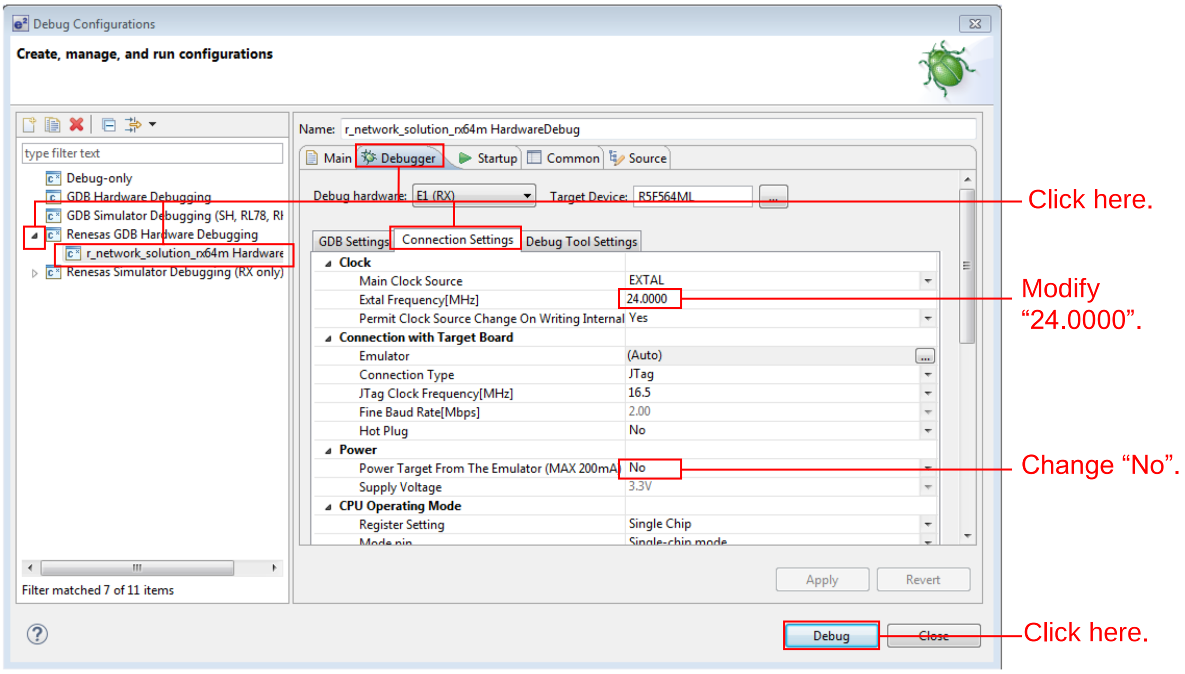
4. Click on [r_network_solution_rx64m HardwareDebug] or [r_network_solution_rx71m HardwareDebug] under [Renesas GDB Hardware Debugging] and click on [Debug].

Click on the [Debugger] tab, and the [Connection Setting] tab.

Change [EXTAL frequency] to 24.0000, then, [Power target from the emulator] to [No (Note 1)].

When completed, click on [Debug (D)].

Note 1 : The setting is used when using an external power supply. Select [Yes] for Power target from the emulator.



Debug Configurations

Create, manage, and run configurations

Name: r_network_solution_rx64m HardwareDebug

Debug hardware: E1 (RX) Target Device: R5F564ML

Clock	
Main Clock Source	EXTAL
Extal Frequency[MHz]	24.0000
Permit Clock Source Change On Writing Internal	Yes
Connection with Target Board	
Emulator	(Auto)
Connection Type	JTag
JTag Clock Frequency[MHz]	16.5
Fine Baud Rate[Mbps]	2.00
Hot Plug	No
Power	
Power Target From The Emulator (MAX 200mA)	No
Supply Voltage	3.3V
CPU Operating Mode	
Register Setting	Single Chip
Mode pin	Single-chip mode

Filter matched 7 of 11 items

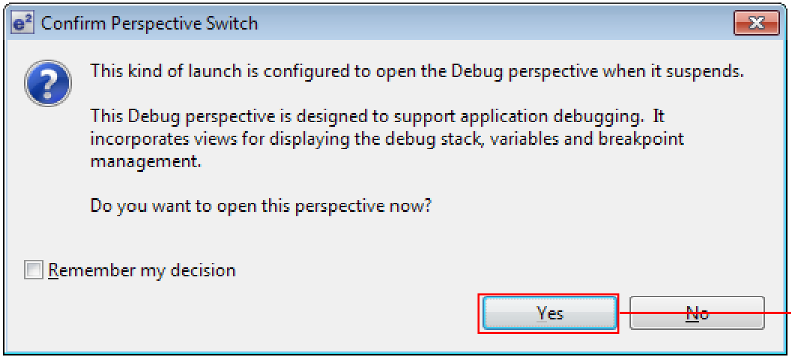
Click here.

Modify "24.0000".

Change "No".

Click here.

When the following message is displayed, click [Yes].



Confirm Perspective Switch

This kind of launch is configured to open the Debug perspective when it suspends.

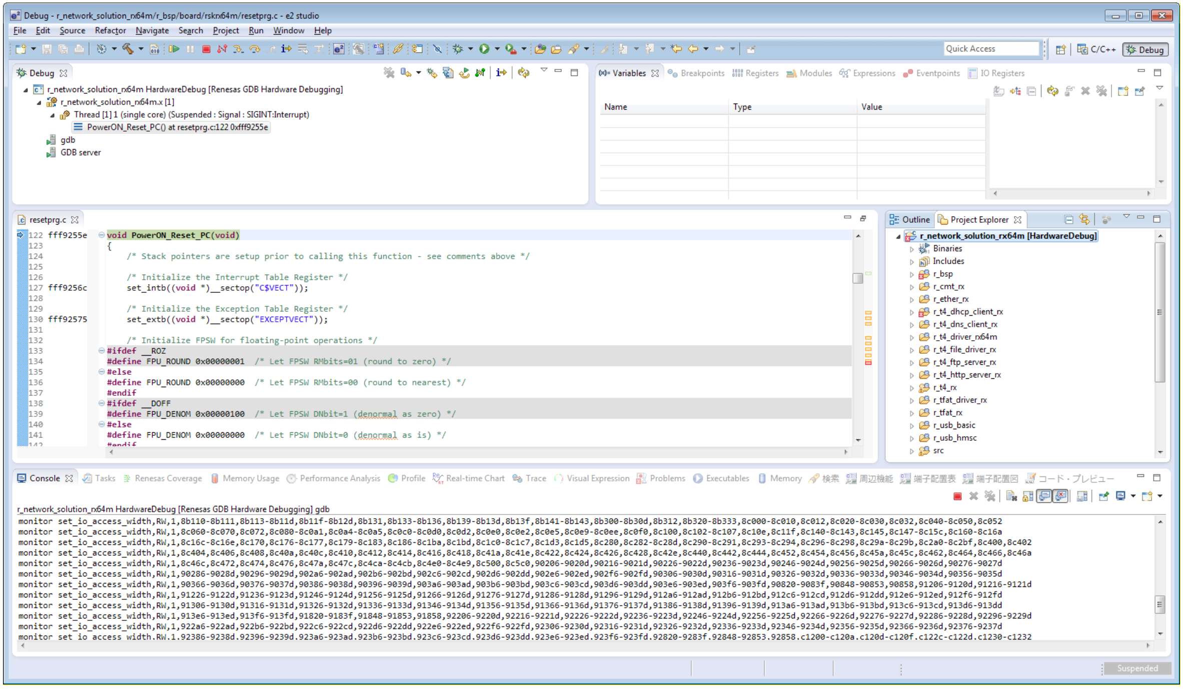
This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.

Do you want to open this perspective now?

□ Remember my decision

Click here.

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



Debug - r_network_solution_rx64m/r_bsp/board/rskrx64m/resetprg.c - e2 studio

r_network_solution_rx64m HardwareDebug [Renesas GDB Hardware Debugging]
r_network_solution_rx64m.x [1]
Thread [1] 1 (single core) (Suspended : Signal : SIGINT:Interrupt)
PowerON_Reset_PC() at resetprg.c:122 0xfff9255e
gdb
GDB server

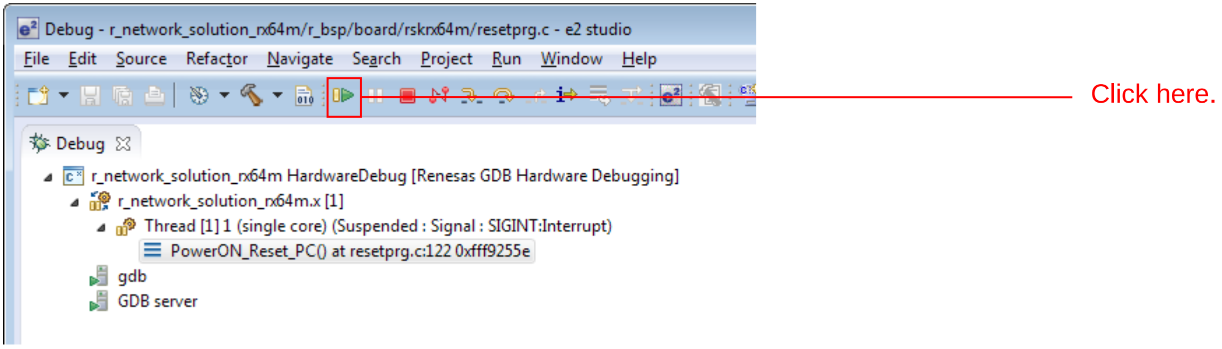
resetprg.c

```
122 fff9255e  void PowerON_Reset_PC(void)
123           {
124               /* Stack pointers are setup prior to calling this function - see comments above */
125
126               /* Initialize the Interrupt Table Register */
127 fff9256c      set_intb((void *)__sectop("C$VECT"));
128
129               /* Initialize the Exception Table Register */
130 fff92575      set_extb((void *)__sectop("EXCEPTVECT"));
131
132               /* Initialize FPSW for floating-point operations */
133           #ifdef __ROZ
134           #define FPU_ROUND 0x00000001  /* Let FPSW RMbits=01 (round to zero) */
135           #else
136           #define FPU_ROUND 0x00000000  /* Let FPSW RMbits=00 (round to nearest) */
137           #endif
138           #ifdef __DOFF
139           #define FPU_DENOM 0x00000100  /* Let FPSW DNbit=1 (denormal as zero) */
140           #else
141           #define FPU_DENOM 0x00000000  /* Let FPSW DNbit=0 (denormal as is) */
142           #endif
```

r_network_solution_rx64m HardwareDebug [Renesas GDB Hardware Debugging] gdb

```
monitor set_io_access_width,RW,1,8b110-8b111,8b113-8b11d,8b11f-8b12d,8b131,8b133-8b136,8b139-8b13d,8b13f,8b141-8b143,8b300-8b30d,8b312,8b320-8b333,8c000-8c010,8c012,8c020-8c030,8c032,8c040-8c050,8c052
monitor set_io_access_width,RW,1,8c060-8c070,8c072,8c080-8c0a1,8c0a4-8c0a5,8c0c0-8c0d0,8c0d2,8c0e0,8c0e2,8c0e5,8c0e9-8c0ee,8c0f0,8c100,8c102-8c107,8c10e,8c11f,8c140-8c143,8c145,8c147-8c15c,8c160-8c16a
monitor set_io_access_width,RW,1,8c16c-8c16e,8c170,8c176-8c177,8c179-8c183,8c186-8c1ba,8c1bd,8c1c0-8c1c7,8c1d3,8c1d5,8c280,8c282-8c28d,8c290-8c291,8c293-8c294,8c296-8c298,8c29a-8c29b,8c2a0-8c2bf,8c400,8c402
monitor set_io_access_width,RW,1,8c404,8c406,8c408,8c40a,8c40c,8c410,8c412,8c414,8c416,8c418,8c41a,8c41e,8c422,8c424,8c426,8c428,8c42e,8c440,8c442,8c444,8c452,8c454,8c456,8c45a,8c45c,8c462,8c464,8c466,8c46a
monitor set_io_access_width,RW,1,8c46c,8c472,8c474,8c476,8c47a,8c47c,8c4ca-8c4cb,8c4e0-8c4e9,8c500,8c5c0,90206-9020d,90216-9021d,90226-9022d,90236-9023d,90246-9024d,90256-9025d,90266-9026d,90276-9027d
monitor set_io_access_width,RW,1,90286-9028d,90296-9029d,902a6-902ad,902b6-902bd,902c6-902cd,902d6-902dd,902e6-902ed,902f6-902fd,90306-9030d,90316-9031d,90326-9032d,90336-9033d,90346-9034d,90356-9035d
monitor set_io_access_width,RW,1,90366-9036d,90376-9037d,90386-9038d,90396-9039d,903a6-903ad,903b6-903bd,903c6-903cd,903d6-903dd,903e6-903ed,903f6-903fd,90820-9083f,90848-90853,90858,91206-9120d,91216-9121d
monitor set_io_access_width,RW,1,91226-9122d,91236-9123d,91246-9124d,91256-9125d,91266-9126d,91276-9127d,91286-9128d,91296-9129d,912a6-912ad,912b6-912bd,912c6-912cd,912d6-912dd,912e6-912ed,912f6-912fd
monitor set_io_access_width,RW,1,91306-9130d,91316-9131d,91326-9132d,91336-9133d,91346-9134d,91356-9135d,91366-9136d,91376-9137d,91386-9138d,91396-9139d,913a6-913ad,913b6-913bd,913c6-913cd,913d6-913dd
monitor set_io_access_width,RW,1,913e6-913ed,913f6-913fd,91820-9183f,91848-91853,91858,92206-9220d,92216-9221d,92226-9222d,92236-9223d,92246-9224d,92256-9225d,92266-9226d,92276-9227d,92286-9228d,92296-9229d
monitor set_io_access_width,RW,1,922a6-922ad,922b6-922bd,922c6-922cd,922d6-922dd,922e6-922ed,922f6-922fd,92306-9230d,92316-9231d,92326-9232d,92336-9233d,92346-9234d,92356-9235d,92366-9236d,92376-9237d
monitor set_io_access_width,RW,1,92386-9238d,92396-9239d,923a6-923ad,923b6-923bd,923c6-923cd,923d6-923dd,923e6-923ed,923f6-923fd,92820-9283f,92848-92853,92858,c1200-c120a,c120d-c120f,c122c-c122d,c1230-c1232
```

5. Click [Resume] on the toolbar. The program will be executed and a break will occur at the start of the main function.



Debug - r_network_solution_rx64m/r_bsp/board/rskrx64m/resetprg.c - e2 studio

r_network_solution_rx64m HardwareDebug [Renesas GDB Hardware Debugging]
r_network_solution_rx64m.x [1]
Thread [1] 1 (single core) (Suspended : Signal : SIGINT:Interrupt)
PowerON_Reset_PC() at resetprg.c:122 0xfff9255e
gdb
GDB server

File Edit Source Refactor Navigate Search Project Run Window Help
Debug
Click here.

6. After the break at the start of the main function, click [Resume] on the tool bar again.

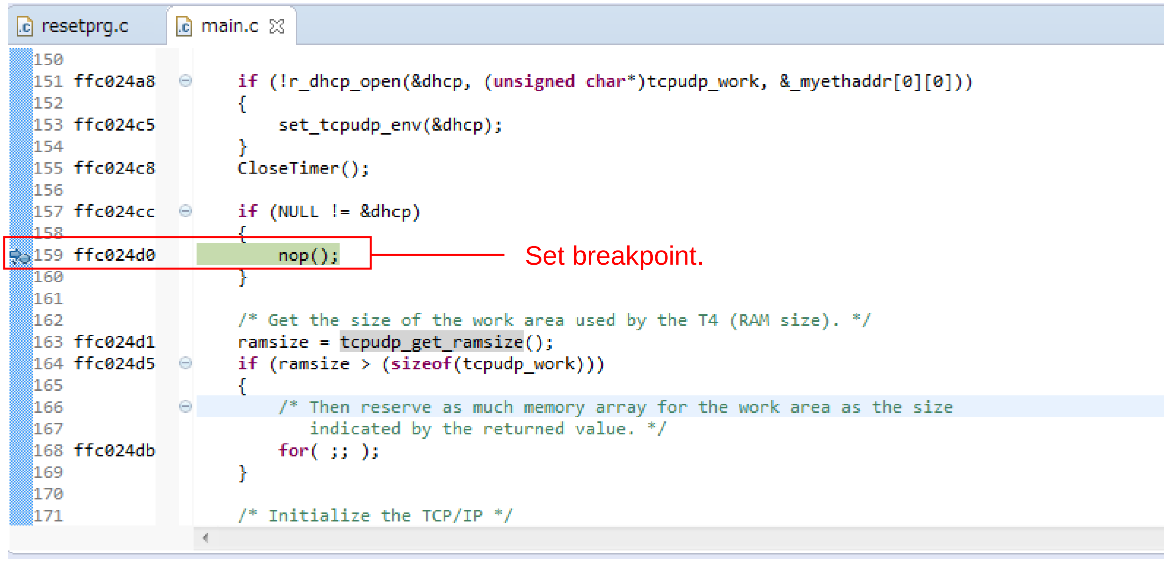
4.3.1 Verify operation of the HTTP server

1. Start a web browser on the client PC and enter the following address according to which port the LAN cable is connected.

Ethernet Port Number	DHCP server	Web Server Address
0	Not existing	http://192.168.0.3 (can be changed in the configuration file.)
	Existing	Obtain from DHCP server
1	Not support	http://192.168.0.10

Note: Note that the http server address can be changed in the configuration file.

The procedure for confirmation about IP address obtained from the DHCP server is shown below.
When complete the connection, open the **src/main.c**.
Set the breakpoint to the **nop()** function in main().

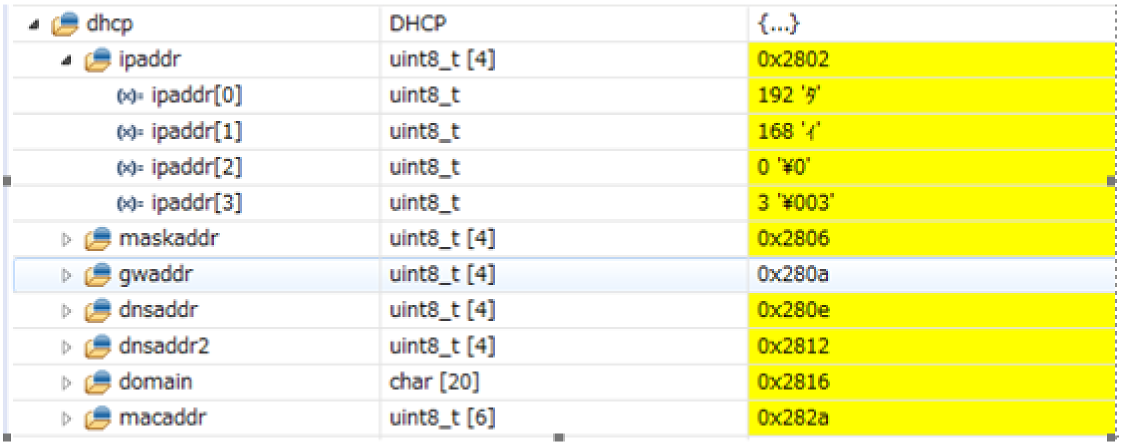


```
ffc024a8      if (!r_dhcp_open(&dhcp, (unsigned char*)tcpudp_work, &_myethaddr[0][0]))
              {
ffc024c5          set_tcpudp_env(&dhcp);
              }
ffc024c8      CloseTimer();

ffc024cc      if (NULL != &dhcp)
              {
ffc024d0          nop();
              }

              /* Get the size of the work area used by the T4 (RAM size). */
ffc024d1      ramsize = tcpudp_get_ramsize();
ffc024d5      if (ramsize > (sizeof(tcpudp_work)))
              {
                  /* Then reserve as much memory array for the work area as the size
                     indicated by the returned value. */
ffc024db          for( ;; );
              }

              /* Initialize the TCP/IP */
```

Set breakpoint.

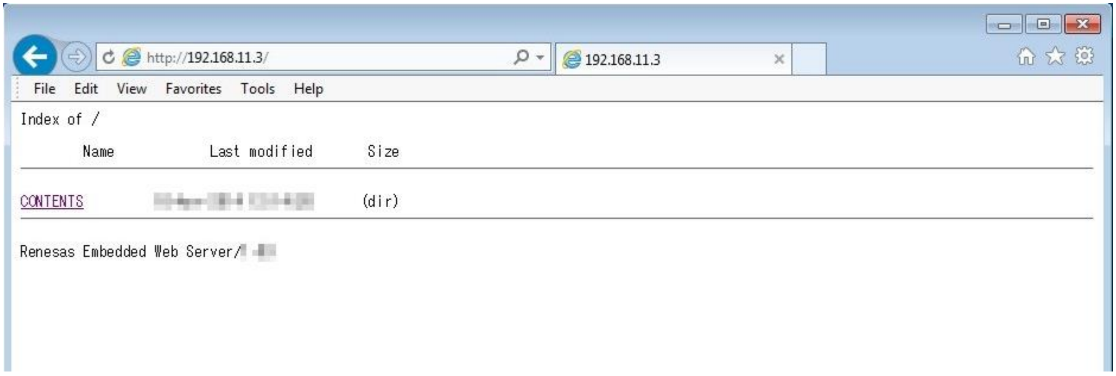
Execute a Program to confirm the program stopped at the set breakpoint.
Then, confirm the variables "dhcp" in the watch window.

The information set to each member of the variables “dhcp” :
— ipaddr : IP address
— maskaddr : Subnetmask
— gwaddr : Gateway address
— dnsaddr : Primary DNS server IP address
— dnsaddr2 : Secondary DNS server address



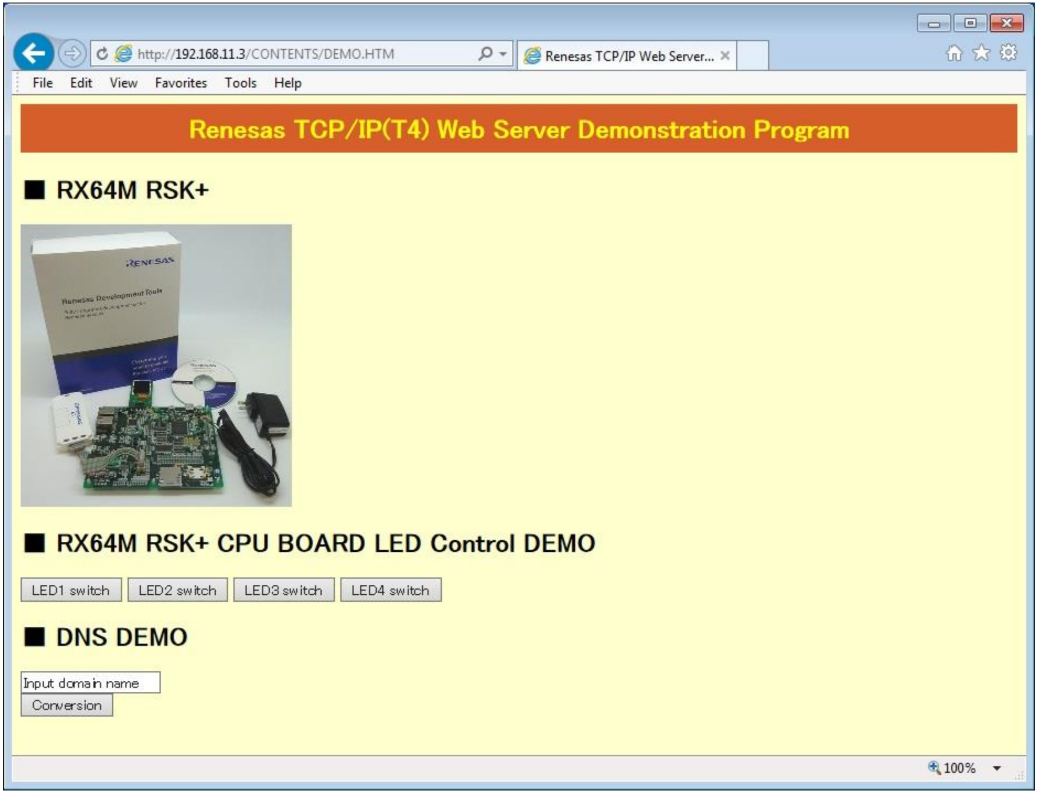
dhcp	DHCP	{...}
ipaddr	uint8_t [4]	0x2802
ipaddr[0]	uint8_t	192 'ﾀ'
ipaddr[1]	uint8_t	168 'ｨ'
ipaddr[2]	uint8_t	0 '¥0'
ipaddr[3]	uint8_t	3 '¥003'
maskaddr	uint8_t [4]	0x2806
gwaddr	uint8_t [4]	0x280a
dnsaddr	uint8_t [4]	0x280e
dnsaddr2	uint8_t [4]	0x2812
domain	char [20]	0x2816
macaddr	uint8_t [6]	0x282a

A list of files in the root directory on the USB memory is displayed.
The file name is listed in the Name field, the last update date is listed in the Last modified field. For size, if it is directories, (dir) is listed and the size is shown in bytes. Click on [Parent Directory] to move to the next higher directory.



```
Index of /
         Name                Last modified        Size

CONTENTS           [redacted]                    (dir)

Renesas Embedded Web Server/[redacted]
```

2. Click [CONTENTS] and then click the file [DEMO.HTM]. This will display a page like the one shown below.



http://192.168.11.3/CONTENTS/DEMO.HTM

Renesas TCP/IP Web Server...

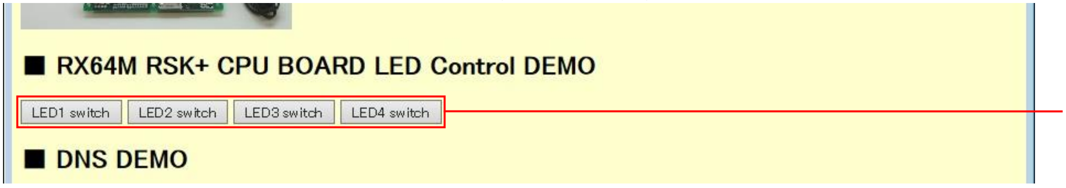
Renesas TCP/IP(T4) Web Server Demonstration Program

■ RX64M RSK+

■ RX64M RSK+ CPU BOARD LED Control DEMO

■ DNS DEMO

3. Confirm LED control using CGI function
Press the [LEDx switch] button.to control (turn on or off) LEDs on the board.

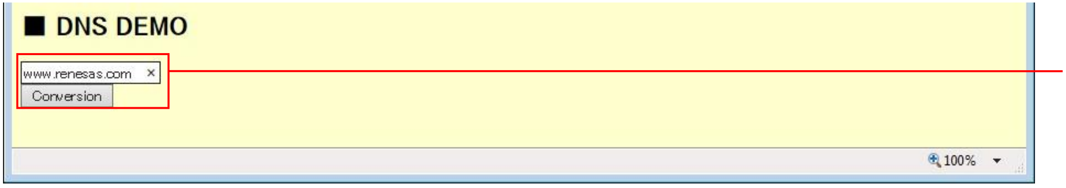


■ RX64M RSK+ CPU BOARD LED Control DEMO

■ DNS DEMO

Click here.

4. Confirm DNS client function using CGI function
Input any domain name into the "Input domain name" text box and push "Conversion" button.

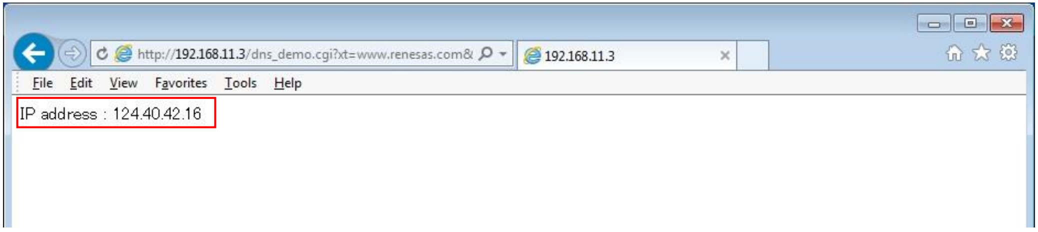


■ DNS DEMO

www.renesas.com

Input domain name and Click here.

Web browser shows the IP address confirmation page as follows.



http://192.168.11.3/dns_demo.cgi?xt=www.renesas.com&

192.168.11.3

IP address : 124.40.42.16

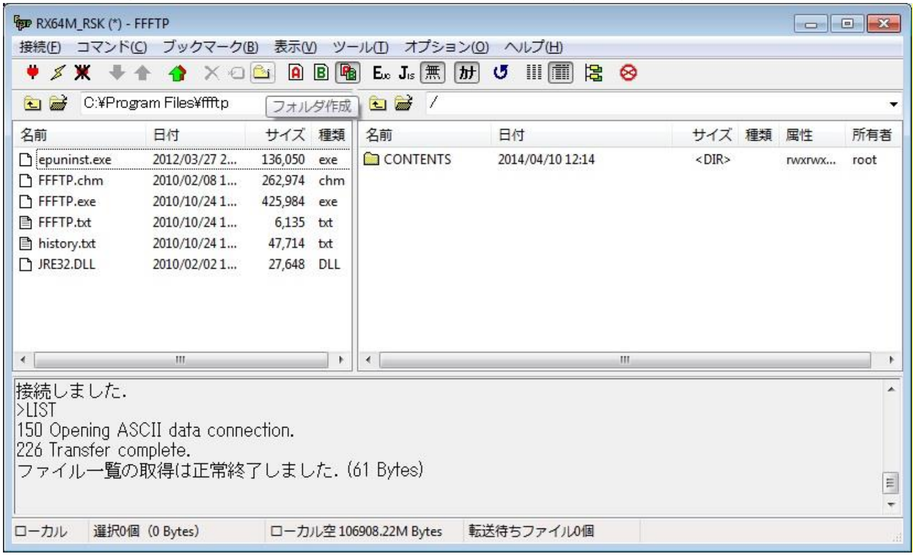
4.3.2 Verify operation of the FTP server

1. Start FTP client software (e.g. FFFTP).

2. Use the user name and password that is defined to "r_t4_ftp_server_rx_config.h", connect to IP address below.

Ethernet Port Number	DHCP server	FTP Server Address
0	Not existing	http://192.168.0.3 (can be changed in the configuration file.)
	Existing	Obtain from DHCP server
1	Not support	Not support

A file listing on the root directory in the USB memory is displayed as follows.


RX64M_RSK (*) - FFFTP
接続(F) コマンド(C) ブックマーク(B) 表示(V) ツール(T) オプション(O) ヘルプ(H)
C:¥Program Files¥ffftp
フォルダ作成
/

名前	日付	サイズ	種類
epuninst.exe	2012/03/27 2...	136,050	exe
FFFTP.chm	2010/02/08 1...	262,974	chm
FFFTP.exe	2010/10/24 1...	425,984	exe
FFFTP.txt	2010/10/24 1...	6,135	txt
history.txt	2010/10/24 1...	47,714	txt
JRE32.DLL	2010/02/02 1...	27,648	DLL

名前	日付	サイズ	種類	属性	所有者
CONTENTS	2014/04/10 12:14	<DIR>		rwxrwx...	root

```
接続しました.
>LIST
150 Opening ASCII data connection.
226 Transfer complete.
ファイル一覧の取得は正常終了しました. (61 Bytes)
```

ローカル 選択0個 (0 Bytes) ローカル空 106908.22M Bytes 転送待ちファイル0個

Note: This shows the example using the FTP Client software (Japanese version).

5. Network Middleware Specifications

For the specifications of the DHCP client, DNS client, HTTP server and FTP server, refer to the document attached to each FIT module.

6. Main Program Specifications

6.1 Files

The main program is included in the project. The source files of the main program is included in the **src** folder. The files of the main program are listed below.

Table 6-1 Main Program Files

Folder Name	File Name	Description
src	main.c	Main source file
	led.c	LED initialization processing source file
	led.h	LED initialization processing header file
	r_http_server_cgi_sample.c	CGI sample source file
	r_sys_time.c	HTTP server system timer source file
	r_sys_time.h	HTTP server system timer header file
	r_usb_hmsc_api.c	USB driver call processing source file
	r_usb_hmsc_api.h	USB driver call processing header file
src/contents	contens.zip	ZIP file of HTML content to be stored in the USB memory

6.2 Modules

The following table lists the modules in the main program.

Table 6-2 Main Program Modules

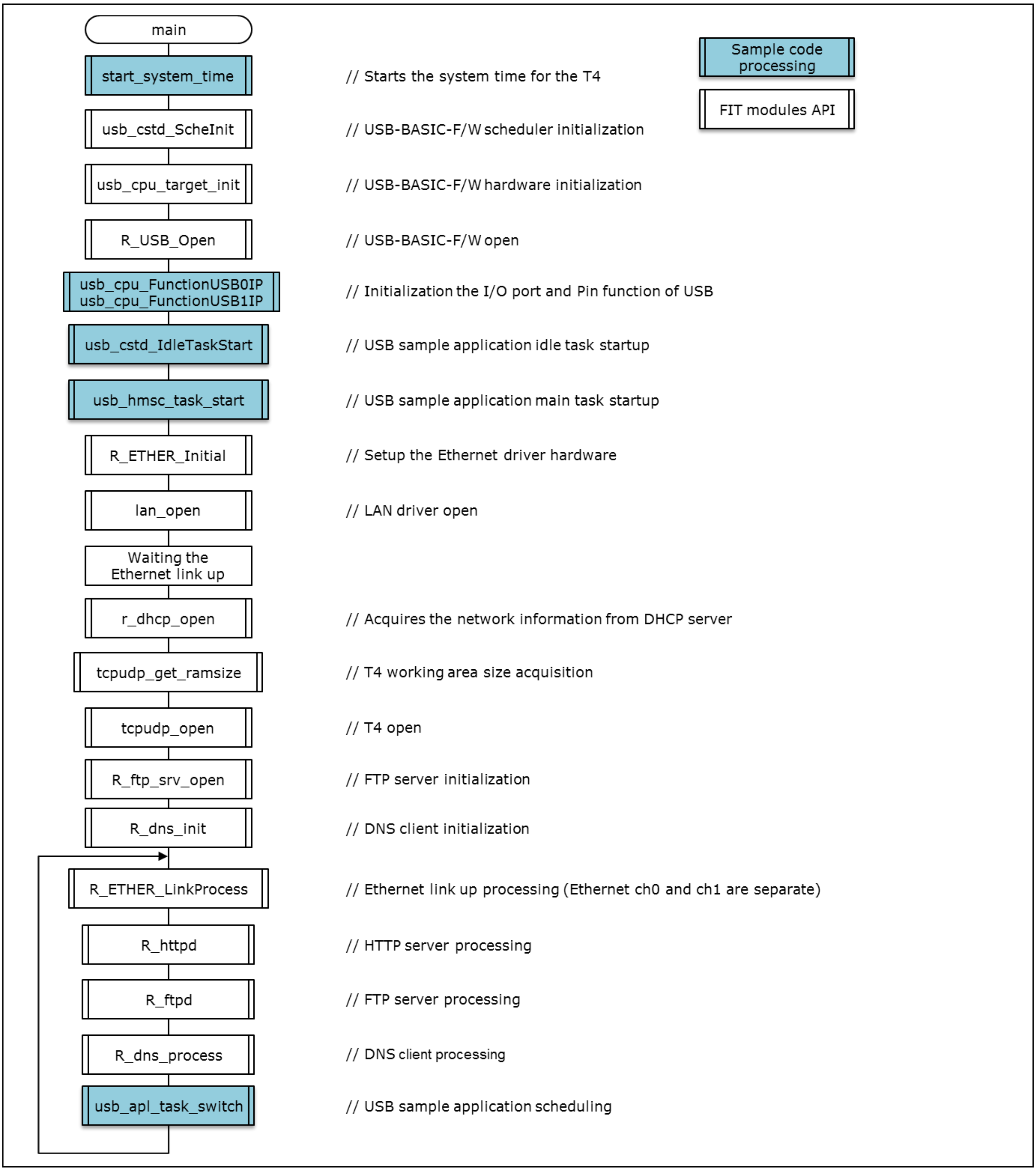
File Name	Module Name	Description
main.c	main	Main processing for the main program Calls initialization processing for each of the FIT modules and drives the main processing for the DNS client, HTTP server, FTP server, USB driver, and Ethernet driver (uses an infinite loop to implement periodic activation).
	set_tcpudp_env	Setting the network information which obtained from DHCP server to the environment variable of T4.
	usb_cpu_FunctionUSB0IP	Port initialization and Pin setting of USB ch0.
	usb_cpu_FunctionUSB1IP	Port initialization and Pin setting of USB ch1.
r_usb_hmsc_api.c	usb_cstd_IdleTaskStart	Starts the idle task used in low-power mode.
	usb_cstd_IdleTask	Idle task used in low-power mode. Performs no processing in host operation.
	usb_hmsc_task_start	HMSC driver activation processing. Performs USB IP initialization and class driver registration.
	usb_apl_task_switch	Performs task scheduling for the USB drivers in non-OS environments.
	usb_hapl_task_start	Starts the HMSC driver application task.
	usb_hmsc_DummyFunction	HMSC driver dummy function
	usb_hmsc_DriveOpen	HMSC driver open processing
	usb_hapl_registration	Registers HMSC drivers.
	usb_hmsc_apl_init	Initializes HMSC driver application task internal variables.
	usb_hmsc_StrgCommandResult	R_usb_hmsc_StrgDriveSearch() callback processing
	usb_hmsc_SampleAplTask	HMSC driver application task processing. Detects USB memory and mounts the file system.
led.c	led_init	Initialization of LEDs
r_http_server_cgi_sample.c	-	Please refer to the attached document to HTTP server module (RX Family Web server using the embedded TCP/IP M3S-T4-Tiny Module Firmware Integration Technology / R20AN0075EJ).
r_sys_time.c	get_sys_time	Please refer to the attached document to HTTP server module (RX Family Web server using the embedded TCP/IP M3S-T4-Tiny Module Firmware Integration Technology / R20AN0075EJ).
	start_system_time	Start the timer for HTTP server.
	update_sys_time	Interrupt handler of the timer for HTTP server.
	stop_system_time	Stop the timer for HTTP server.

6.3 Flowcharts

This section shows the flowcharts for the modules in the main program.

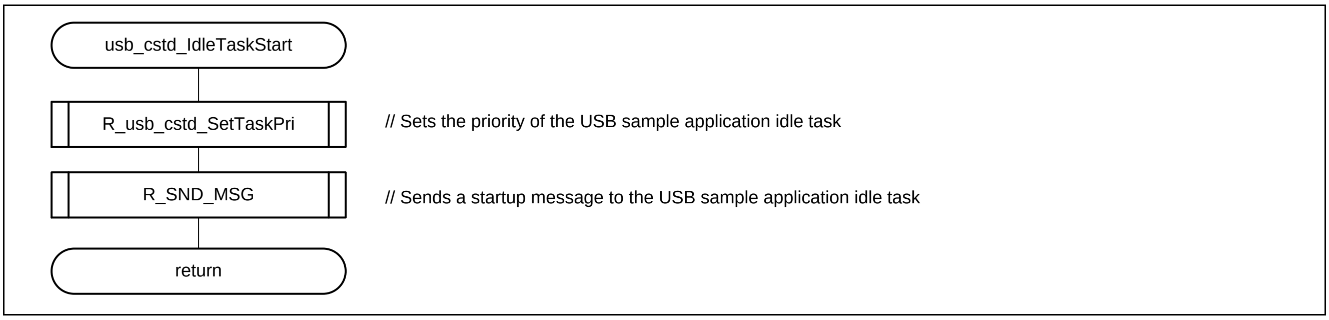
(1) main()

This is the main() function and is first called from the startup routine for the board support package (BSP module). It initializes the drivers and T4 and then periodically calls Ethernet driver link up processing, DNS client, HTTP server and FTP server main processing, and USB driver scheduling from an infinite loop.


main
start_system_time
usb_cstd_ScheInit
usb_cpu_target_init
R_USB_Open
usb_cpu_FunctionUSB0IP
usb_cpu_FunctionUSB1IP
usb_cstd_IdleTaskStart
usb_hmsc_task_start
R_ETHER_Initial
lan_open
Waiting the Ethernet link up
r_dhcp_open
tcpudp_get_ramsize
tcpudp_open
R_ftp_srv_open
R_dns_init
R_ETHER_LinkProcess
R_httpd
R_ftpd
R_dns_process
usb_apl_task_switch
// Starts the system time for the T4
// USB-BASIC-F/W scheduler initialization
// USB-BASIC-F/W hardware initialization
// USB-BASIC-F/W open
// Initialization the I/O port and Pin function of USB
// USB sample application idle task startup
// USB sample application main task startup
// Setup the Ethernet driver hardware
// LAN driver open
// Acquires the network information from DHCP server
// T4 working area size acquisition
// T4 open
// FTP server initialization
// DNS client initialization
// Ethernet link up processing (Ethernet ch0 and ch1 are separate)
// HTTP server processing
// FTP server processing
// DNS client processing
// USB sample application scheduling
Sample code processing
FIT modules API

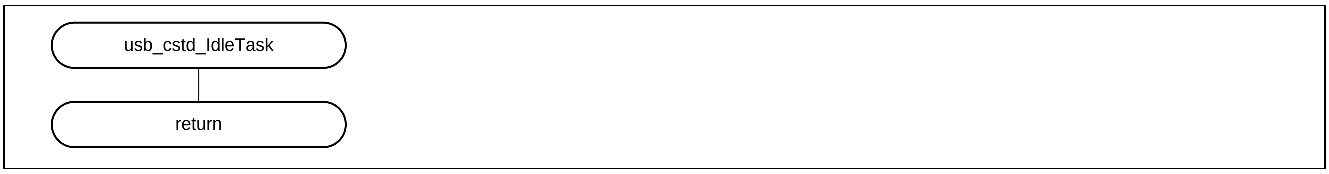
(2) usb_cstd_IdleTaskStart

Starts the USB driver processing idle task.


usb_cstd_IdleTaskStart
R_usb_cstd_SetTaskPri
// Sets the priority of the USB sample application idle task
R_SND_MSG
// Sends a startup message to the USB sample application idle task
return

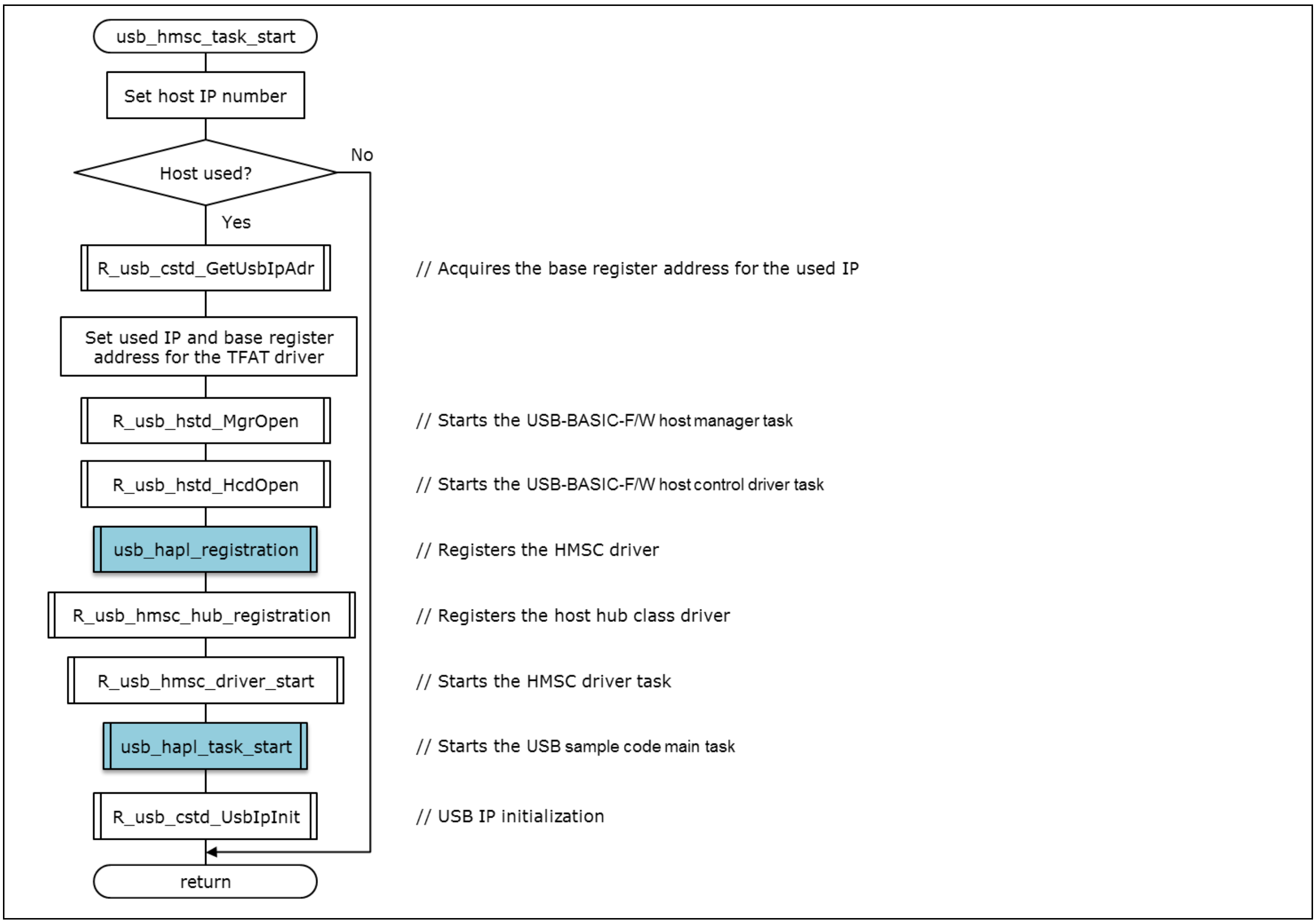
(3) usb_cstd_IdleTask

This is the USB driver processing idle task.



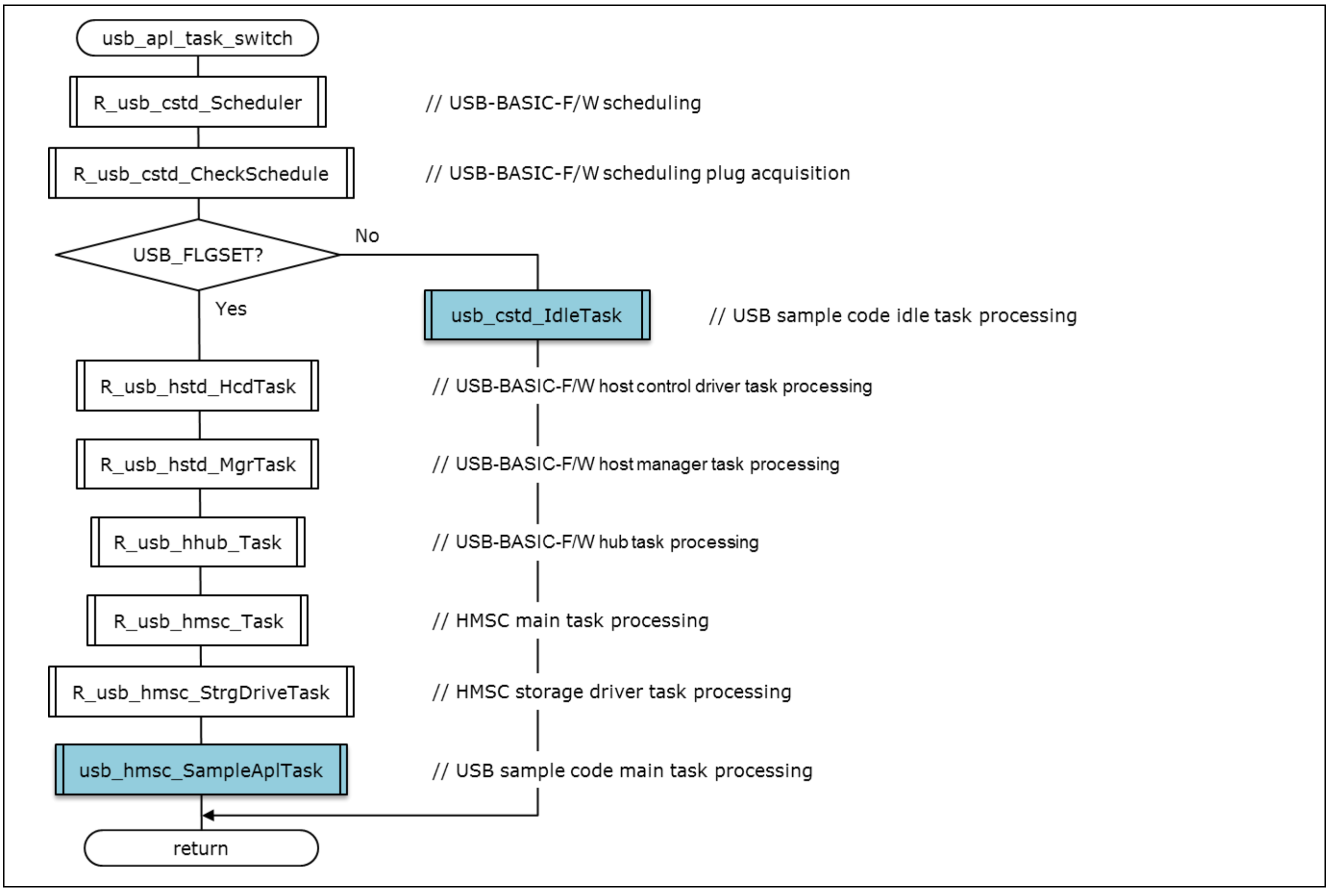
(4) usb_hmsc_task_start

Starts the various tasks within the USB driver, registers class drivers, and starts the USB memory mount processing task.


usb_hmsc_task_start
Set host IP number
Host used?
No
Yes
R_usb_cstd_GetUsbIpAdr
// Acquires the base register address for the used IP
Set used IP and base register address for the TFAT driver
R_usb_hstd_MgrOpen
// Starts the USB-BASIC-F/W host manager task
R_usb_hstd_HcdOpen
// Starts the USB-BASIC-F/W host control driver task
usb_hapl_registration
// Registers the HMSC driver
R_usb_hmsc_hub_registration
// Registers the host hub class driver
R_usb_hmsc_driver_start
// Starts the HMSC driver task
usb_hapl_task_start
// Starts the USB sample code main task
R_usb_cstd_UsbIpInit
// USB IP initialization
return

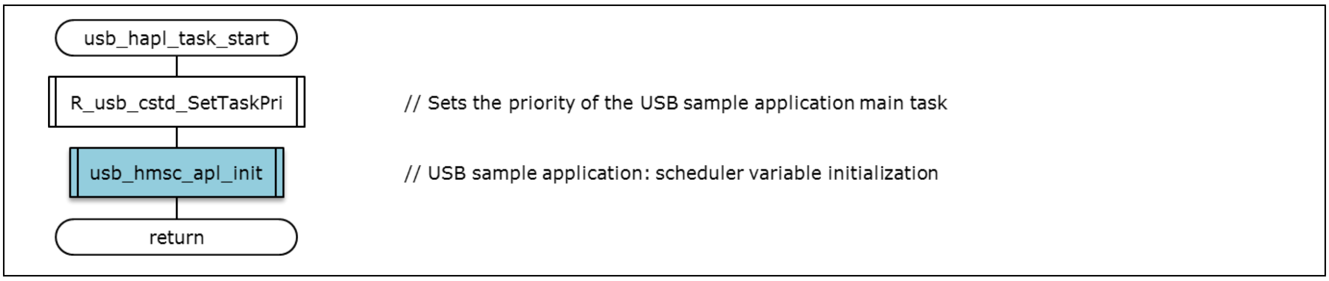
(5) usb_apl_task_switch

Performs the USB driver scheduling.


usb_apl_task_switch
R_usb_cstd_Scheduler
// USB-BASIC-F/W scheduling
R_usb_cstd_CheckSchedule
// USB-BASIC-F/W scheduling plug acquisition
USB_FLGSET?
No
Yes
usb_cstd_IdleTask
// USB sample code idle task processing
R_usb_hstd_HcdTask
// USB-BASIC-F/W host control driver task processing
R_usb_hstd_MgrTask
// USB-BASIC-F/W host manager task processing
R_usb_hhub_Task
// USB-BASIC-F/W hub task processing
R_usb_hmsc_Task
// HMSC main task processing
R_usb_hmsc_StrgDriveTask
// HMSC storage driver task processing
usb_hmsc_SampleAplTask
// USB sample code main task processing
return

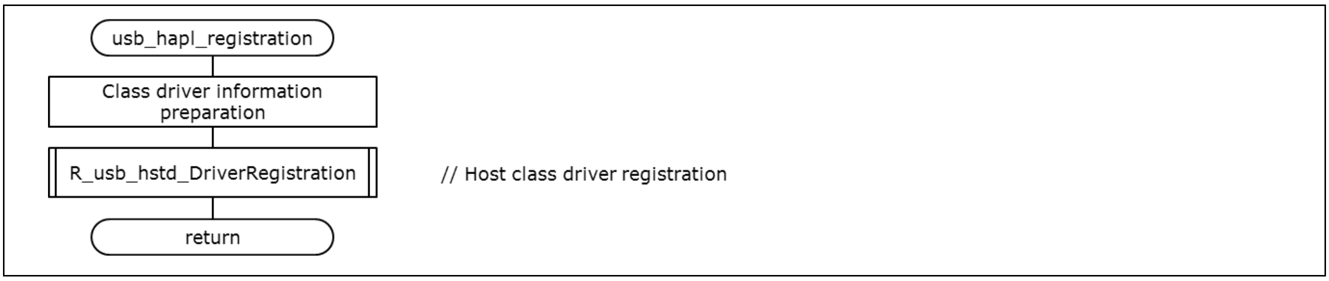
(6) usb_hapl_task_start

Initializes the USB memory mount processing task.


usb_hapl_task_start
R_usb_cstd_SetTaskPri
// Sets the priority of the USB sample application main task
usb_hmsc_apl_init
// USB sample application: scheduler variable initialization
return

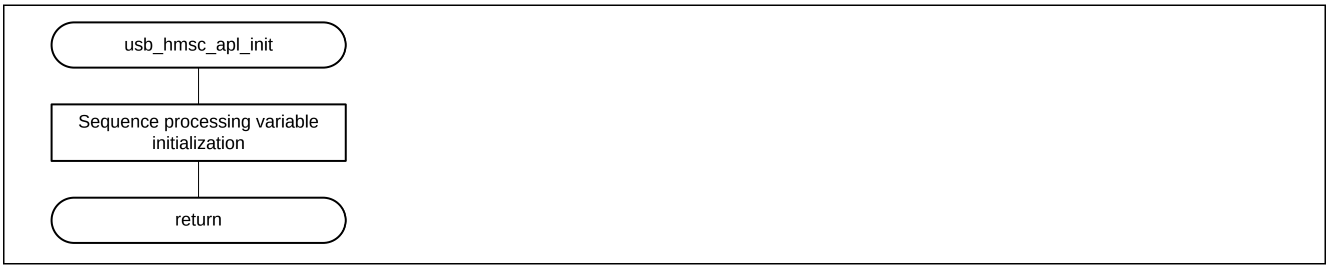
(7) usb_hapl_registration

Performs the class driver registration processing.


usb_hapl_registration
Class driver information preparation
R_usb_hstd_DriverRegistration
// Host class driver registration
return

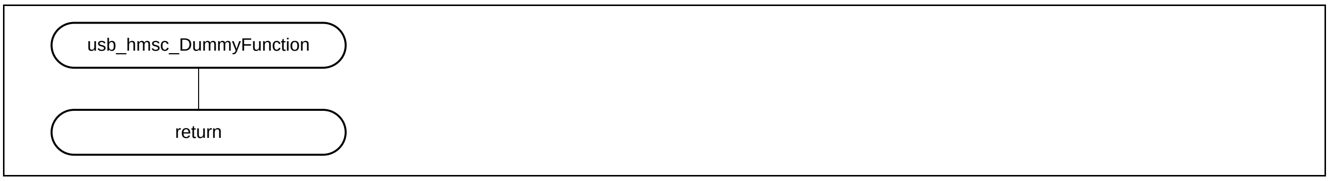
(8) usb_hmsc_apl_init

Initializes the sequence processing variables for the USB memory mount processing task.



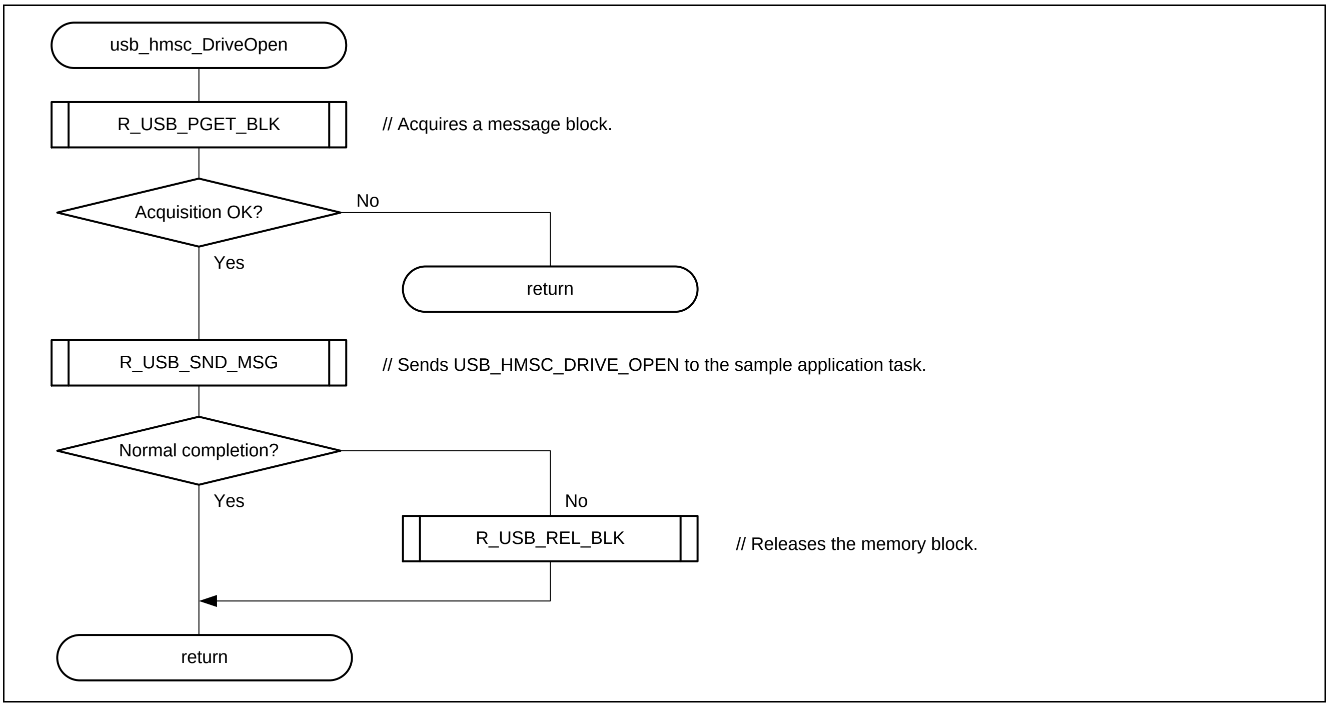
(9) usb_hmsc_DummyFunction

Dummy function for suspend and resume specified at class driver registration.



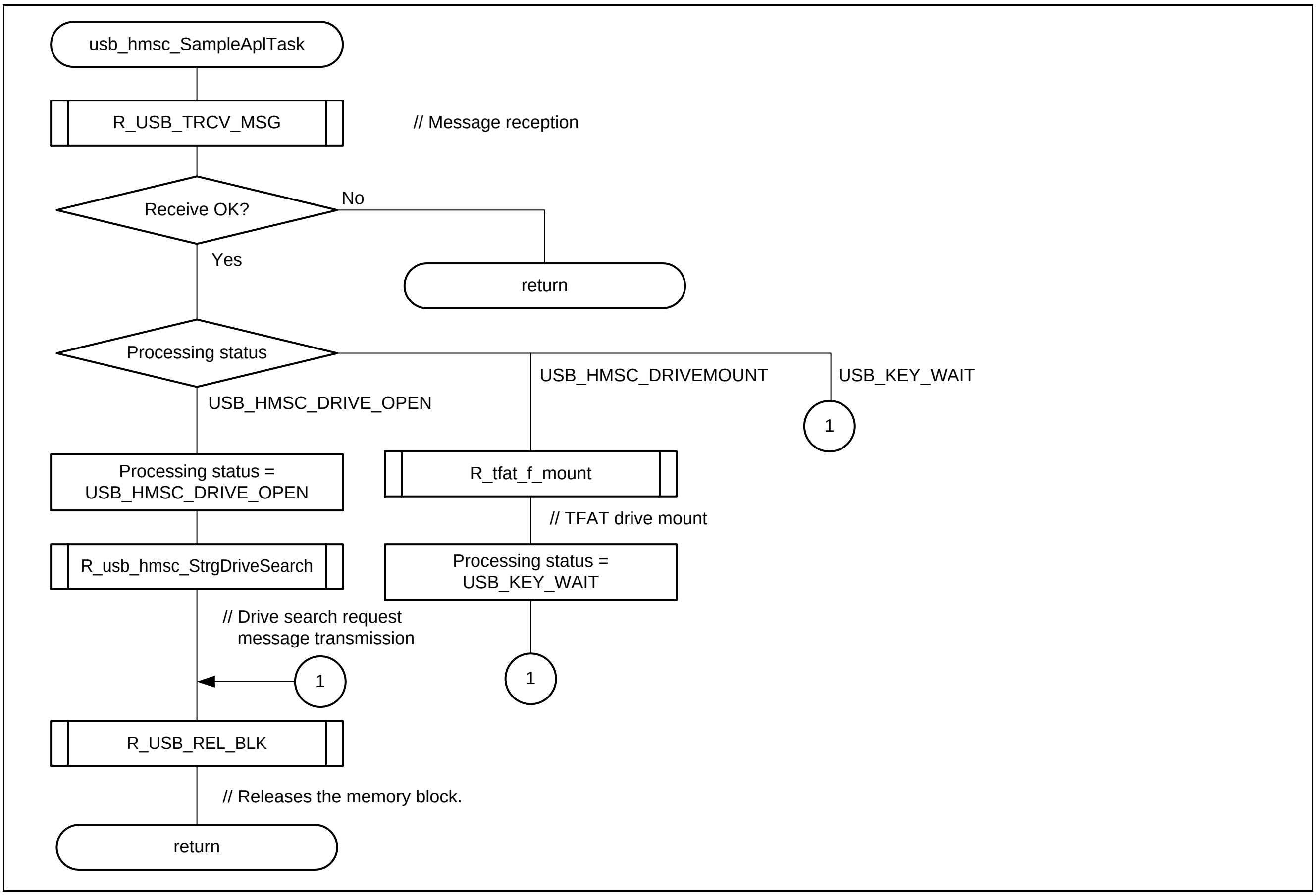
(10) usb_hmsc_DriveOpen

This is the callback function called from the USB driver when USB memory is inserted. It sends a USB_HMSC_DRIVE_OPEN message for the sample application task.


usb_hmsc_DriveOpen
R_USB_PGET_BLK
// Acquires a message block.
Acquisition OK?
No
Yes
return
R_USB_SND_MSG
// Sends USB_HMSC_DRIVE_OPEN to the sample application task.
Normal completion?
Yes
No
R_USB_REL_BLK
// Releases the memory block.
return

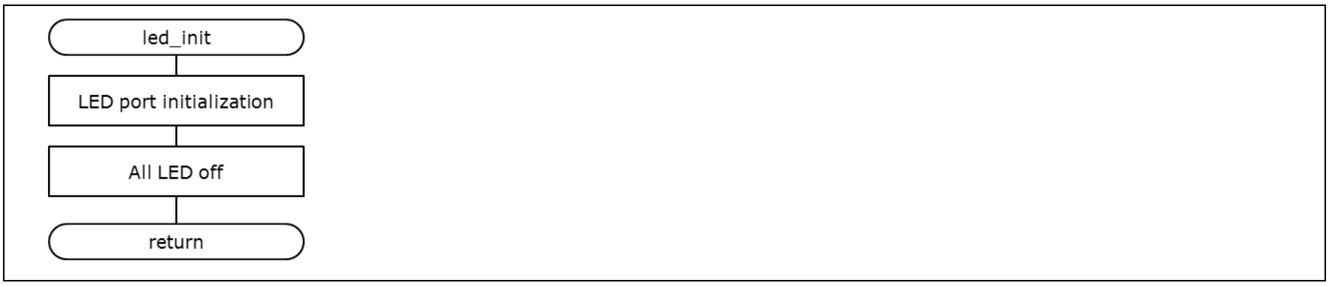
(11) usb_hmsc_SampleAplTask

This function performs the sample application task processing. It receives the USB_HMSC_DRIVE_OPEN message issued from the usb_hmsc_DriveOpen function and detects a mountable drive.
Also, it receives the USB_HMSC_DRIVEMOUNT message issued from the usb_hmsc_StrgCommandResult function and performs a mount for the file system.



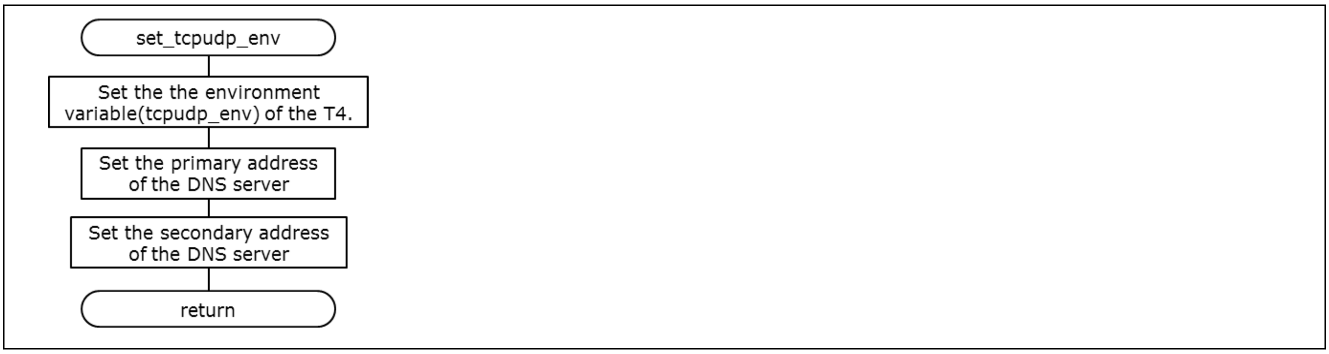
(12) led_init

This function performs the initialization process for using the LED on the Renesas Starter Kit + for RX64M or the Renesas Starter Kit + for RX71M.



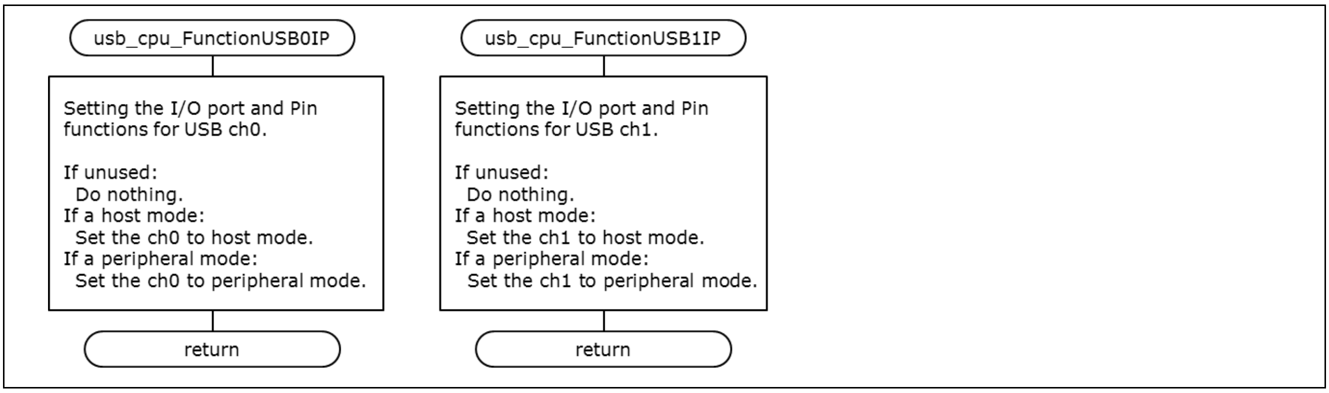
(13) set_tcpudp_env

Setting the network information which obtained from DHCP server to the environment variable of T4.


set_tcpudp_env
Set the the environment variable(tcpudp_env) of the T4.
Set the primary address of the DNS server
Set the secondary address of the DNS server
return

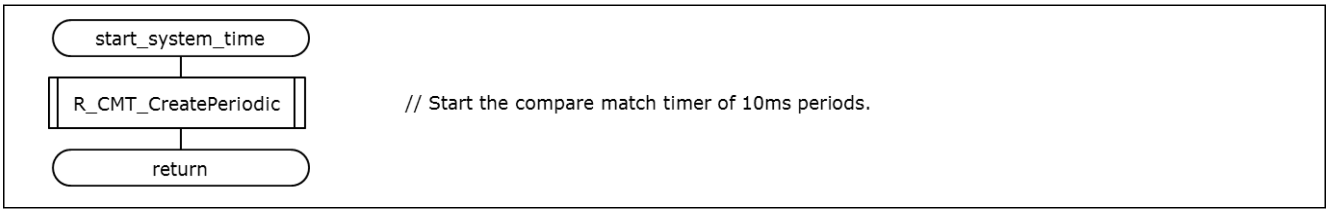
(14) usb_cpu_FunctionUSB0IP, usb_cpu_FunctionUSB1IP

Performs I/O port and Pin functions initialization for using the USB ch0 and ch1.


usb_cpu_FunctionUSB0IP
Setting the I/O port and Pin functions for USB ch0.
If unused:
Do nothing.
If a host mode:
Set the ch0 to host mode.
If a peripheral mode:
Set the ch0 to peripheral mode.
return
usb_cpu_FunctionUSB1IP
Setting the I/O port and Pin functions for USB ch1.
If unused:
Do nothing.
If a host mode:
Set the ch1 to host mode.
If a peripheral mode:
Set the ch1 to peripheral mode.
return

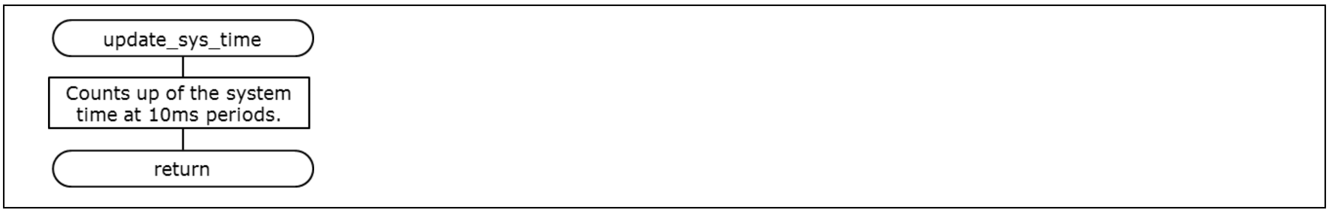
(15) start_system_time

Start the compare match timer (10ms interval) for the system time of HTTP server.



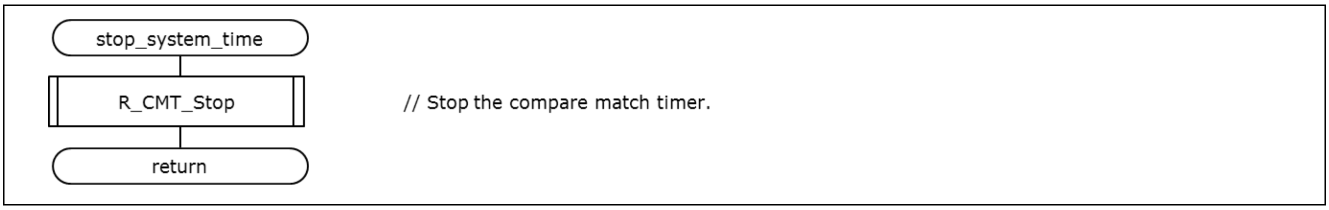
(16) update_sys_time

This function is an interrupt handler that counts up of the system time of HTTP server.



(17) stop_system_time

Stop the compare match timer for the system time of HTTP server.



7. User-Defined Functions

The user defined functions must be coded by the user to match the user system environment. Some of the user-defined functions are required by the FIT modules.

This package includes the following user-defined function samples. See the corresponding FIT module manual or other documentation for specifications of these user-defined functions.

Table 7-1 User-Defined Functions

User-Defined Function	File Name	FIT Module Name	Document Name/Catalog Number
System timer interface	r_sys_time.c	r_t4_http_server_rx	RX Family Web server using the embedded TCP/IP M3S-T4-Tiny Module Firmware Integration Technology / R20AN0075EJ (4.2 System Timer)

8. Supplement

8.1 USB Driver Limitations

When both USB channels ch0 and ch1 are set to host mode, only ch0 can recognize USB memory. To use ch1 in host mode, set ch0 to either unused or peripheral mode. For the setting examples, refer to “3.4.1(3) Change USB Driver Settings.

8.2 This Application Limitations

After program operation, if the USB memory is removed it will not be recognized if it is reinserted. The program should be restarted.

8.3 FTP Server and DHCP Client Limitations

FTP server and DHCP client can be operated at Ethernet ch0 only.

8.4 Notes on Using the Free Evaluation Version of the RX Family C/C++ Compiler Package

There is a usage period limitation and certain usage limitations on the free evaluation version of the RX Family C/C++ Compiler Package. If the usage period exceeds, load modules may not be generated correctly due to the usage limitations.

See the page on evaluation software on the Renesas web site at the link below.

http://www.renesas.com/products/tools/evaluation_software/index.jsp

Website and Support

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

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Sep 1, 2014	—	First edition issued
1.03	Feb 29, 2016	—	Added DHCP client, DNS client and FTP server. Added RX71M. Changed an application note to R01AN2608EJ

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
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

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 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

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 1207, 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 India Pvt. Ltd.
No.777C, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 5.0