RX62N Group

uIP TCP/IP Protocol Stack Demonstration

R01AN0169EU0101
Rev.1.01
July 8, 2011

Introduction

This application note demonstrates the features and capabilities of the following Renesas Ethernet connectivity target devices with open source uIP TCP/IP protocol stack. This application note assumes some experience with Ethernet, TCP/IP, and HTML. For more introductory material on these subjects please see the references.

Target Devices

RX62N Group.

Contents

1. Overview

The uIP TCP/IP stack demonstration project provides an example of Ethernet connectivity and a sample Web server application that controls LEDs on the Renesas Starter Kit (RSK) boards.

2. How to Run the Demo

The procedure below provides step by step instructions for how to setup the demonstration project and run it.

2.1 Setup

Set up a demonstration environment as shown in Figure 1. In this setup router is used as a DHCP server and PC as a Web client. Several RSK boards can be connected to multiport router. Use straight RJ-45 Ethernet cables to make connections. Depending on RSK board used there may be other settings to be configured before running the demonstration project. These may include instructions for connecting a debugger to RSK board. Please refer to Quick Start Guide (QSG) of the RSK board for more information.

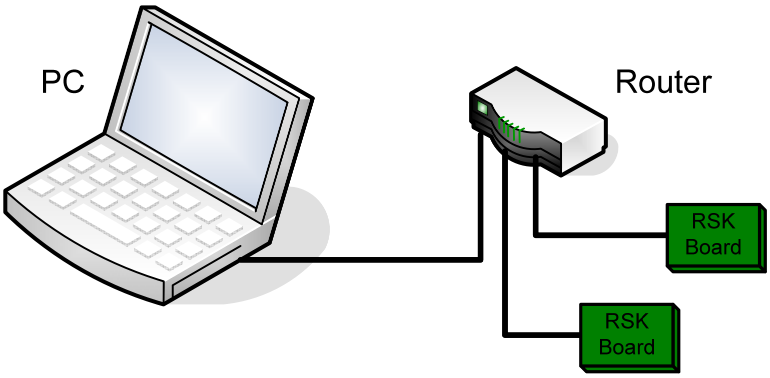


Figure 1 Demonstration Setup

The demonstration project by default is configured to run in little endian mode. Please make sure RSK board is configured for the same endian mode. Please refer to QSG how to change endian mode of operation.

For a simpler setup, an RSK board can be directly connected to PC. In this setup, router is not used and the demonstration project is designed to fall back to a static IP address if a DHCP server is not found in about 10 seconds. In this case RSK board assumes the default IP address of 192.168.1.10.

Figure 2 shows a more detailed test setup environment as an alternative. In this setup all devices are on the same collision domain and all network activity can be monitored and analyzed on PC. If this setup is used, make sure the center connectivity device is a true hub rather than a switch. Router can be disconnected and connected back to the network independent of the connections between PC and RSK boards. This allows monitoring RSK board behavior with or without a DHCP server on the network.

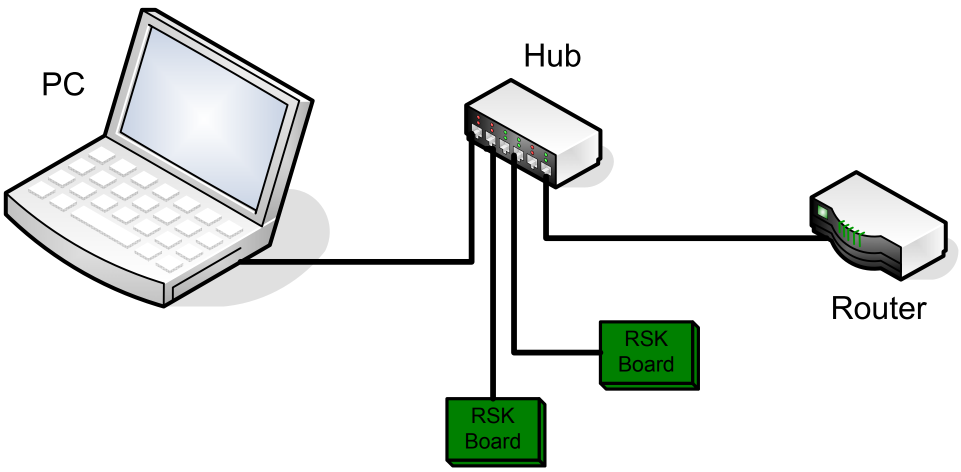


Figure 2 Test Setup

Configure IP address of router to 192.168.1.1. This is usually the default IP address for most home or office routers. Configure DHCP start IP address to 192.168.1.100 with two maximum DHCP users. A snapshot of the router configuration is shown in Figure 3. Section 5 has more information on DHCP in general and how to debug a system with DHCP server.

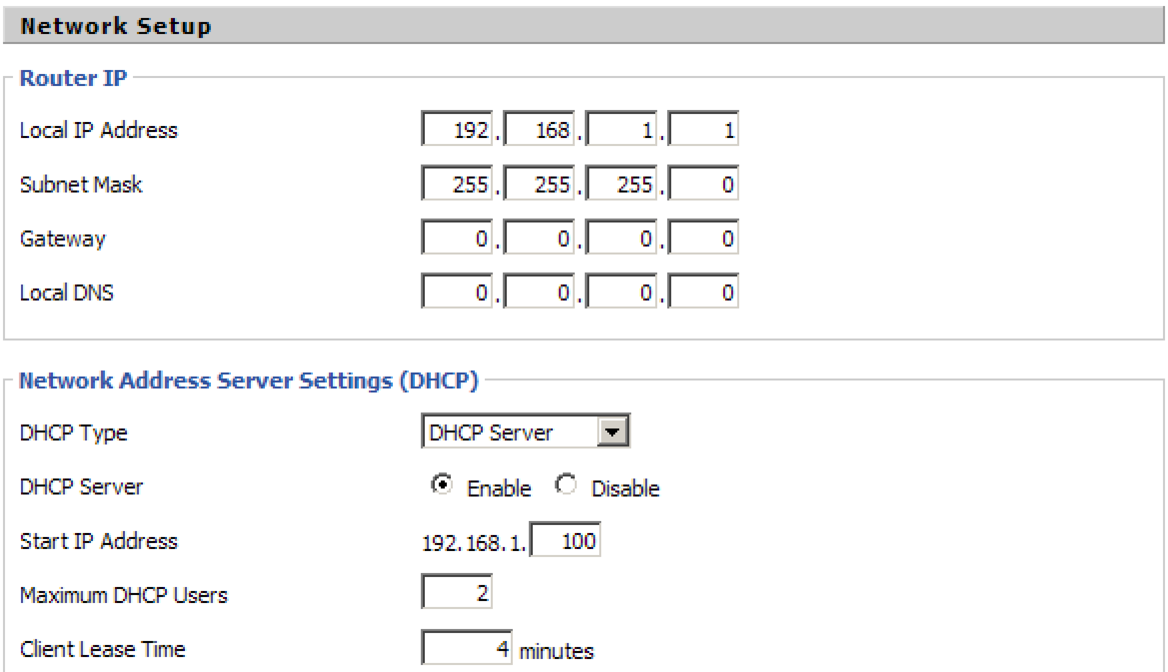


Figure 3 Router Network Configuration

Configure Ethernet port of PC with a static IP address of 192.168.1.2. Internet Protocol (TCP/IP) properties of the PC is shown in Figure 4. Make sure that this Ethernet port is not used to access to a corporate network or a workgroup. If there is not a spare Ethernet port available, it is recommended to use an USB to Ethernet adaptor. Please follow the adaptor manufacturer manual for installation instructions.

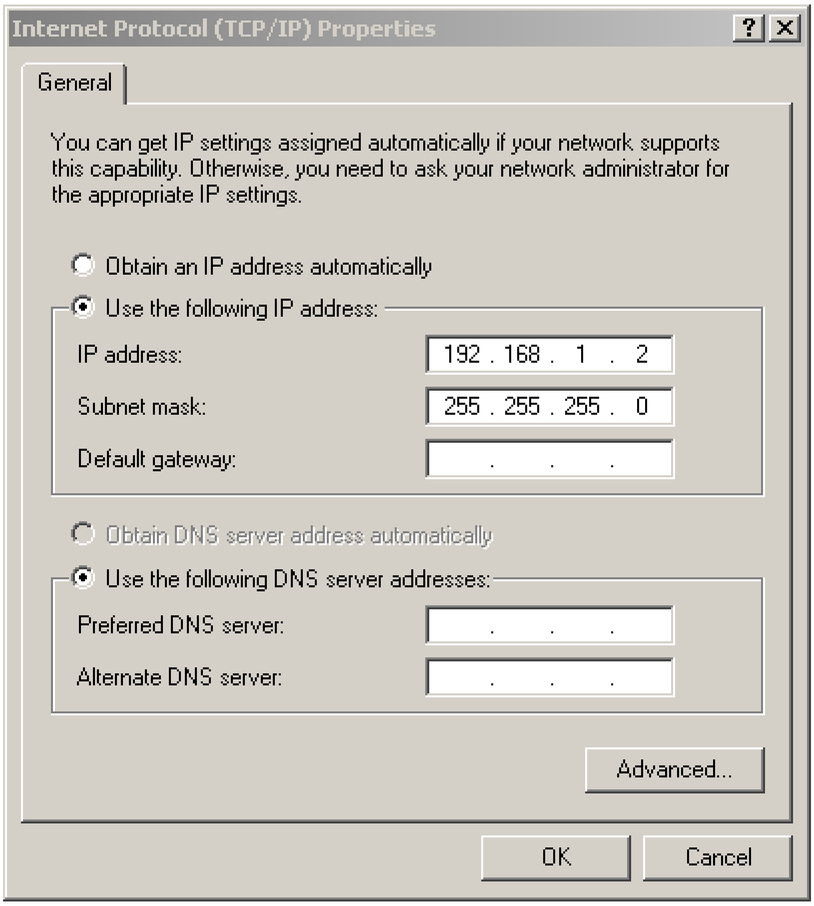


Figure 4 PC Network Configuration

2.2 Sample Project Directory Structure

Figure 5 shows the demonstration project directory structure. The \src folder contains the source code and has 4 subfolders: bsp, driver uip, and user-app. The \src\bsp and \src\driver folders have Renesas board specific source code and the Ethernet drivers for the RSK board. The uIP stack is in the \src\uip folder.

The open source uIP TCP\IP stack comes with its own documentation and it is in the \src\uip\doc folder. The main.c file is in \src\uip\uip and the example Web pages are in src\uip\apps\webserver\httpd-fs folders.

The demonstration project includes a simple user application that controls the LEDs on the RSK board. This application is in the \src\user-app folder.

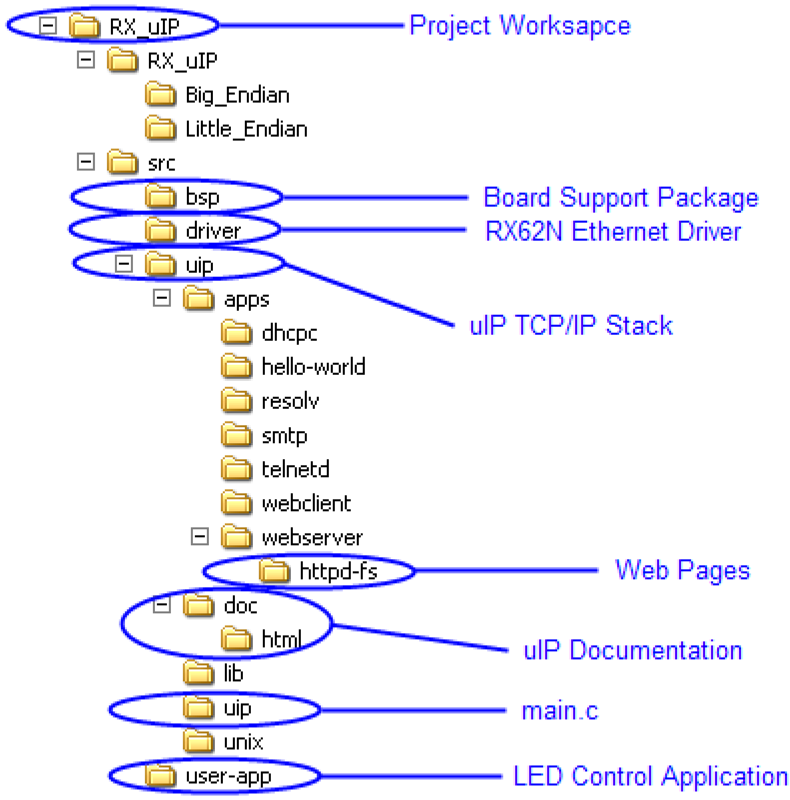
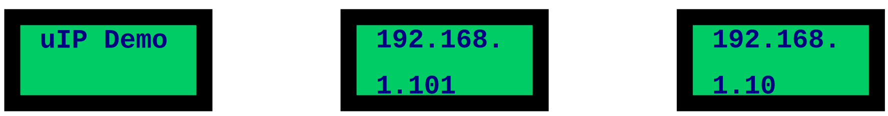


Figure 5 Directory Structure of uIP Demonstration Project

2.3 What to do with the demo

The demonstration project displays “uIP Demo” on RSK board LCD on power up. The IP address used is displayed on the LCD. RSK board either receives its IP address from a DHCP server or uses its default setting of 192.168.1.10. For the test setup in this application note, DHCP server can assign IP addresses of 192.168.1.100 and 192.168.1.101. Make sure Ethernet cables are connected and devices are powered up if no IP address is displayed after 10 seconds. Some of the possible LCD settings are shown below.



Router's status page can also be used to find which IP address is assigned to the RSK board as shown in Figure 6. Another way is simply to ping IP addresses and use the IP address with the reply. Ping messages can be generated by the following DOS Shell commands. :

```
C:\>ping 192.168.1.10
C:\>ping 192.168.1.100
C:\>ping 192.168.1.101
```

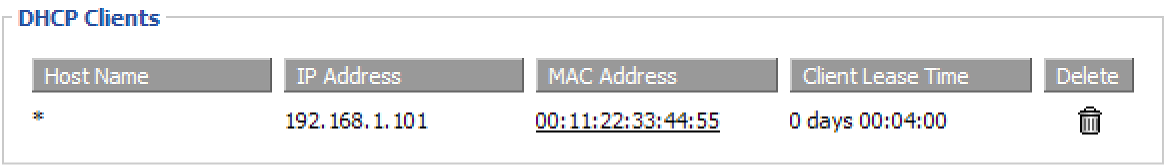


Figure 6 DHCP Client Information

Lunch a Web browser and use the IP address of RSK board in URL field. After a successful connection, user sees the welcome page shown in Figure 7. This is also the Front page. Note that the Renesas logo is linked to http://www.renesas.com/. By activating this link the user can easily access to the Renesas Electronics internet site.

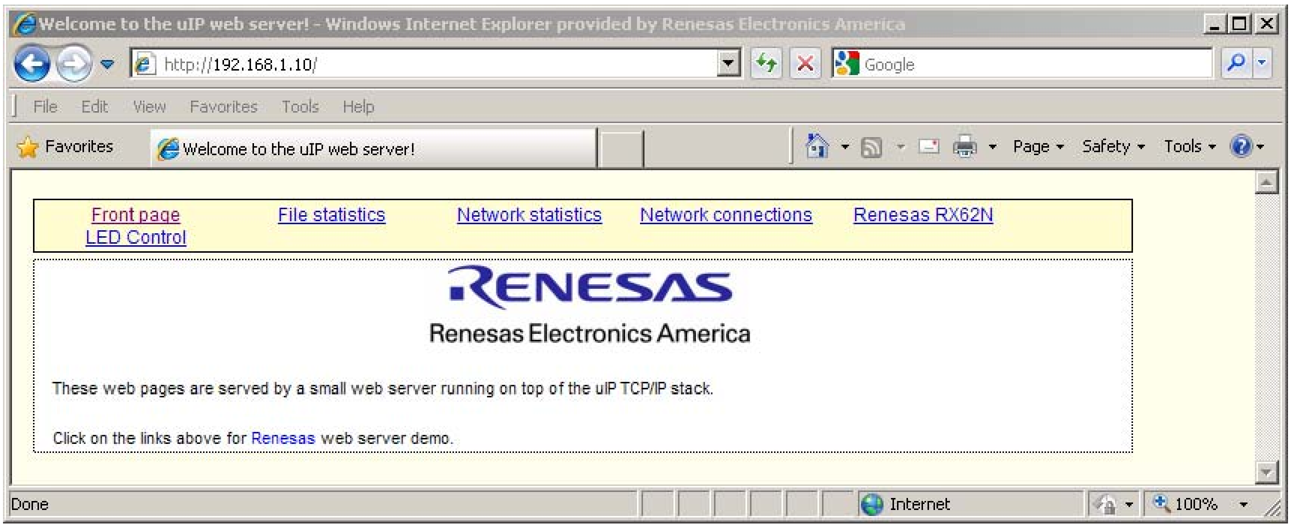


Figure 7 uIP Welcome Page

All other pages can be accessed by links provided in the top banner. The file statistics page shows the number of times a specific page is accessed. The network statistics page displays the number of IP, ICMP, and TCP packet reception and transmission information. The network connections page shows the current status of established TCP connections within the uIP stack. These pages are dynamic and recreated every time when they are accessed.

Two custom pages are created and included within the demonstration project. First is the RSK board specific page. An example of this page is shown in Figure 8. In this case it is an RX62N custom page. The demonstration project is personalized for different target devices and target specific images are shown on this page. This is to show that creating custom Web pages can be easily achieved and integrated with the uIP TCP/IP stack. Section 3 describes step by step instructions for creating a new Web page.

Figure 8 RSK Custom Page

Second custom page is the simple user application that controls the LEDs on the RSK board. One of the LEDs on the RSK board is used to indicate system timer activity. The other three is used by the LED Control Web page as shown in Figure 9. From this Web page, user can turn on and off the LEDs on the RSK board. The reset button selects the off setting for all LEDs. Note that the LED names in Figure 9 are generic and labeled as A, B, and C to allow portability between different RSK boards. They are usually grouped together on the target board.

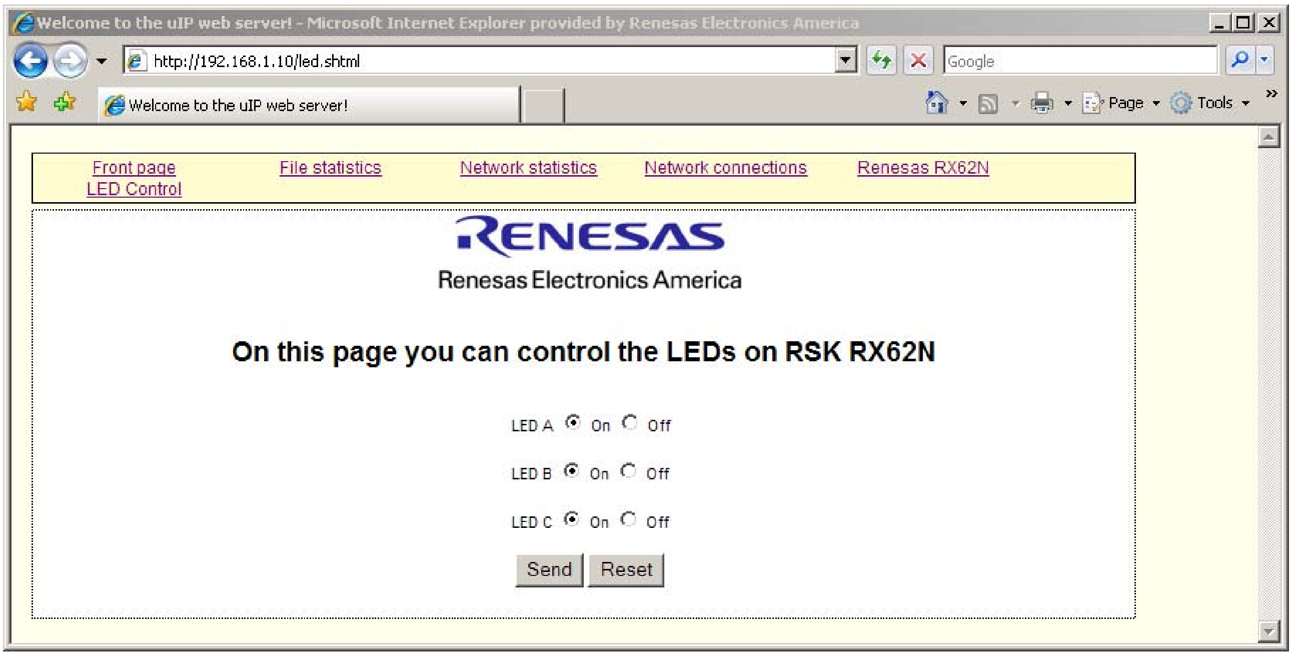


Figure 9 LED Control Web Application Page

3. Steps to Create a New Web Page

A new web page can be easily created and added to the demonstration project by following the steps below. With a little HTML language knowledge user can create custom web page applications. All the tools and the information needed are provided with the demo project and in this application note.

1. Write a new Web page application. See led.shtml example in src\uip\apps\webserver\httpd-fs.
2. Copy it to src\uip\apps\webserver\httpd-fs directory.
3. Run makefsdata.exe from src\uip\apps\webserver directory to generate the new fttpd-fsdat.c file.
4. Re-build the project.

The makefsdata.exe is a Renesas add-on program created from makefsdata Perl script. This executable is included into the demonstration project in apps\webserver\ directory to make it easier for the user to generate the httpd-fsdata.c file without need to find and install a Perl interpreter.

4. More on uIP TCP/IP Stack

uIP TCP/IP stack was originally developed by Adam Dunkels of the Networked Embedded Systems Group at the Swedish Institute of Computer Science. uIP TCP/IP stack includes some higher layer example applications such as web server, web client, Trivial File Transfer Protocol (TFTP), and DNS hostname server.

uIP TCP/IP stack does not require a real time operating system. However, there are versions of it ported to an open source FreeRTOS operating system and they are available on the Internet. It is also ported to several other Renesas MCU devices and sample code can be downloaded from the Renesas sample code site.

4.1 Usage Considerations with uIP TCP/IP Stack

One consideration when using uIP TCP/IP stack is that it supports only one TCP segment in transit. If uIP TCP/IP stack is used with a TCP receiver using delayed acknowledgment algorithm, throughput performance can be poor. You can modify TCP acknowledgment behavior of your PC if you experience this condition with your default PC setup.

More information can be found at http://support.microsoft.com/kb/328890. This situation is also discussed in uIP TCP/IP reference manual.

Another consideration is that uIP TCP/IP stack supports one TCP and one UDP applications at a given time. In this demonstration project, HTTP Web server application uses TCP and DHCP client runs on UDP. An application multiplexer layer based on connection port number can be added to the uIP TCP/IP stack to support multiple TCP or UDP applications.

5. More on DHCP

Dynamic Host Configuration Protocol (DHCP) is a protocol used by networked devices to obtain IP addresses and other parameters such as default gateway, subnet mask, and IP address of Domain Name Server (DNS) from a DHCP server. The protocol is defined by RFC 2131. DHCP eases the management of above tasks and ensures that all IP addresses on the network are unique and unused IP addresses are returned back to the IP address poll for reassignment for other devices joining the network.

The demonstration project makes use of the dynamic mode of DHCP. In dynamic mode, a client is provided with an IP address and time duration in which this IP address is valid. This time duration is called lease time.

5.1 DHCP Operation

There are 4 main messages exchanged between a DHCP client and a DHCP server during dynamic IP address assignment. They are shown in Table 1.

Table 1 DHCP Messages

Message	From	To	Details
DHCP Discovery	Client	Server	Client discovers DHCP servers and asks for an IP address
DHCP Offer	Server	Client	Server reserves an IP address and offers it to the client
DHCP Request	Client	Server	Client tells all DHCP servers that it accepted the offer
DHCP Acknowledge	Server	Client	Server initiates final phase of the configuration. This message includes the lease time and other option parameters.

5.2 How to Use DHCP Client

There are certain things to consider while using a DHCP client. The most important consideration is to ensure that each device on the network has a unique MAC address. DHCP servers assign IP addresses based on client MAC addresses. For end customer production devices the MAC address can be purchased from IEEE.

Another consideration is how to know what IP address is assigned to a device. One way to find this information is to query the DHCP server through its management interface. Figure 10 shows that the RSK board with MAC address 00:11:22:33:44:55 is assigned to an IP address of 192.168.1.101. On the other hand, the RSK board with MAC address 00:11:22:33:44:56 is given an IP address of 192.168.1.100.

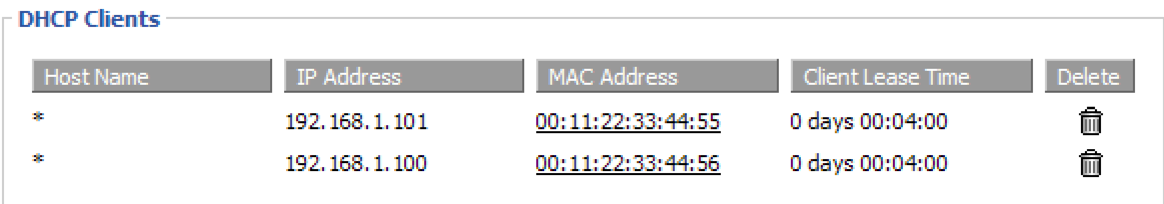


Figure 10 DHCP Client Status Information on the DHCP Server

Debugging a system with a DHCP server can be tricky. Here are some recommendations. First, use of a network analyzer is a great help. Wireshark has been used throughout the development of this project. It is PC based network analyzer software. For more information on Wireshark see http://www.wireshark.org/.

Second, the IP address of the PC Ethernet port used by the network analyzer must be on the same network and subnet with the DHCP server and its clients e.g. the Renesas target board(s). This can be achieved by assigning a static IP address to the PC Ethernet port that is outside the IP addresses can be given out by the DHCP server but still making sure that network and subnet requirements are met.

For example in Figure 3 the DHCP sever is configured with a start IP address of 192.168.1.100. Figure 4 shows that the PC Ethernet port is configured to use the192.168.1.2 and it is outside the range of the IP addresses that can be given to its clients by the server.

6. Renesas Ethernet Drivers

The RSK Ethernet drivers include circular buffers to transmit and receive Ethernet frames. The operation of Ethernet Controller Direct Memory Access Controller (E-DMAC) is controlled by descriptor lists and data buffers. This is explained in detailed in E-DMAC section of the Hardware Manual. In summary, E-DMAC provides hardware support for transferring data between the Ethernet Controller (EtherC) and the transmit and receive buffers in RAM.

The structure of transmit and receive descriptors are very similar. They differ in which members of the structure defines the actual number of bytes transmitted or received. In Figure 11 the TBL (transmit buffer length) field identifies the actual number of bytes to transmit. A pointer to the transmit buffer is stored in the TBA (transmit buffer address) field. Data to be transmitted is written into a memory location addressed by this pointer before transmit operation is started.

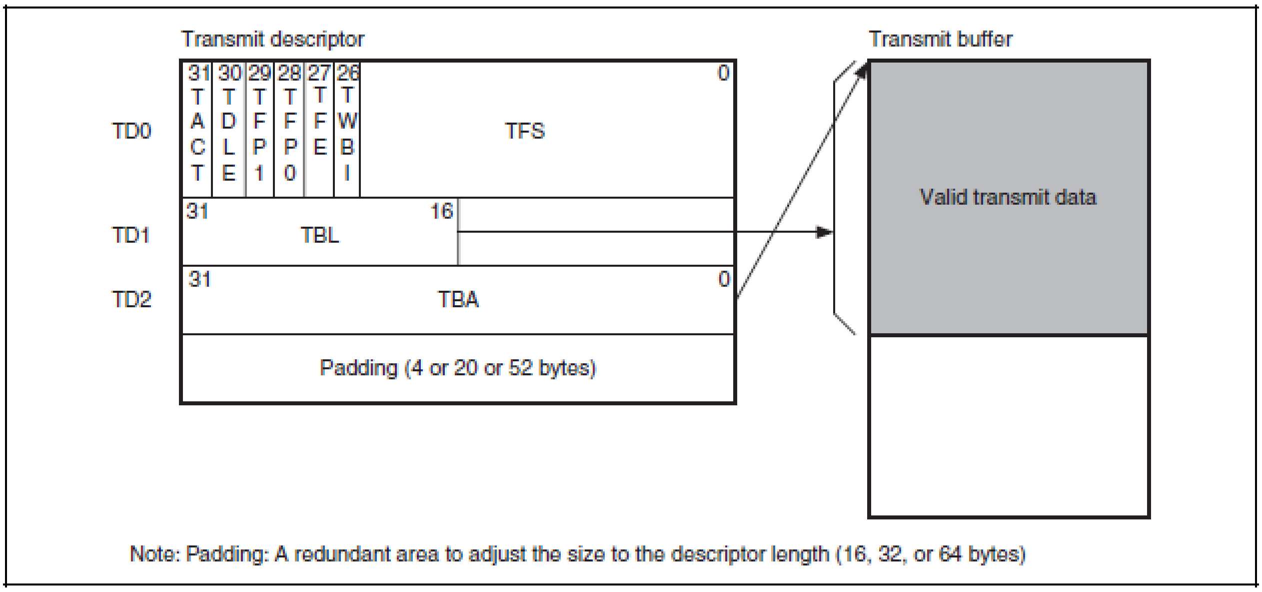


Figure 11 Transmit Descriptor

For a receive descriptor, shown in Figure 12, the RBL (receive buffer length) defines the receive buffer size but not the actual received byte count. That information is in the RFL (receive frame length) field. Depended on the target device the receive buffers must be aligned with 16-byte or 32-byte boundaries. The drivers in this demonstration project uses 32-byte receive buffer boundaries. E-DMAC stores the received data by EtherC in a memory location addressed by a pointer in the RBA (receive buffer address).

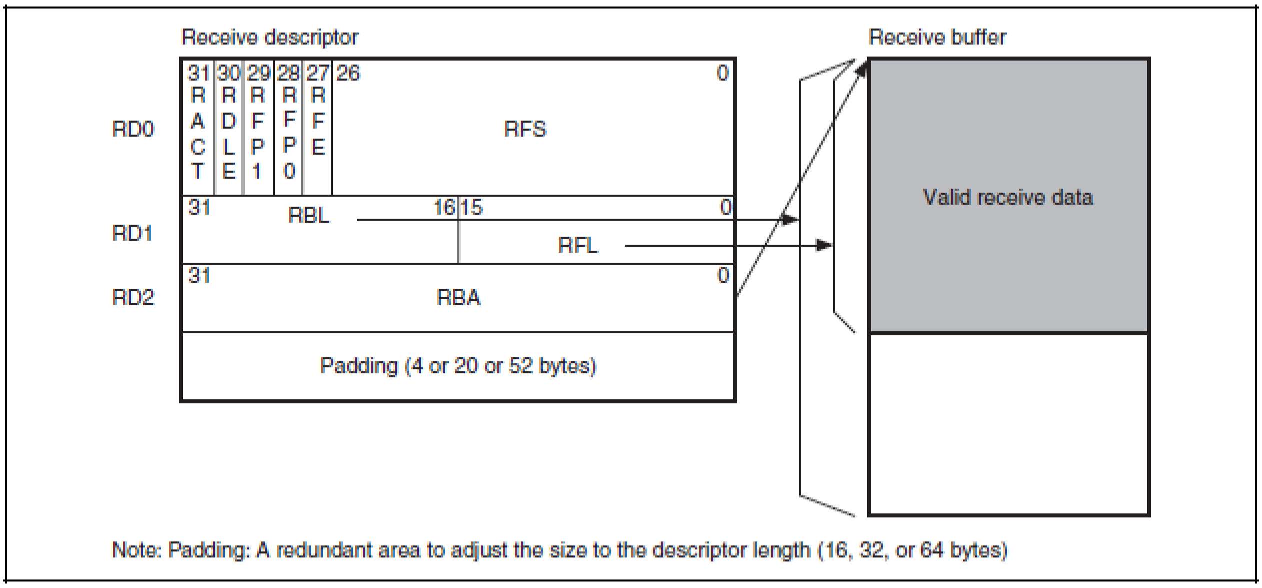


Figure 12 Receive Descriptor

The demonstration project uses 8 transmit and receive descriptors. Each descriptor points to the next descriptor and the last descriptor points back to the first descriptor creating a circular buffer. Each descriptor also points to a 256-byte transmit or receive buffer. This is shown in Figure 13.

The number of descriptors and the size of transmit and receive buffers are depended on the end application. The average Internet frame size is between 230 to 250 bytes. However, applications that may need bigger size buffers and more of them can be easily configured by the following definitions in src\driver\eth.h file.

```
#define BUFSIZE      (256)
#define ENTRY          (8)
```

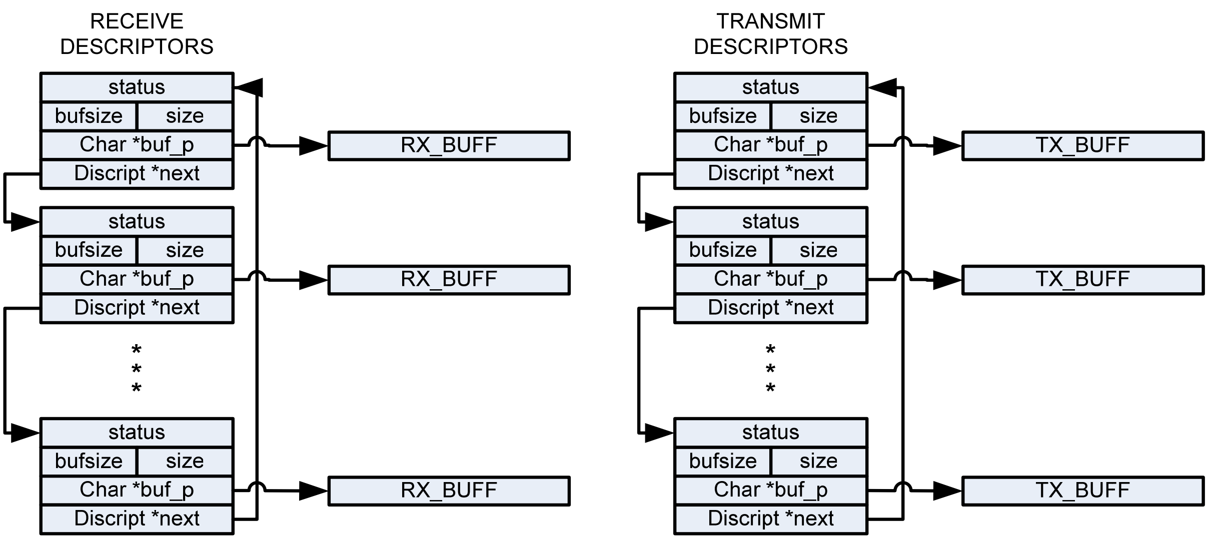


Figure 13 Transmit and Receive Descriptors and Buffers

7. Renesas Ethernet Driver APIs

7.1 Renesas Unique Functions

The following functions serve as the driver interface to the uIP TCP/IP stack. These are the standard Renesas APIs (RAPI) for the Renesas Ethernet devices.

R_Ether_Open

R_Ether_Close

R_Ether_Read

R_Ether_Write

R_Ether_Write_Sync

R_Ether_Open

The R_Ether_Open function performs EtherC and E-DMAC peripheral, physical device, and transmit and receive data buffer initializations. The EtherC and E-DMAC are powered up separately as part of power on reset initialization.

Format

```
int32_t  R_Ether_Open(uint32_t ch, uint8_t mac_addr[]);
```

Parameters

ch

Specifies the EtherC channel number.

mac_addr

Specifies the MAC address of EtherC.

Return Values

R_ETHER_OK(0)
R_ETHER_ERROR(-1)

Properties

Prototyped in file eth.h
Implemented in file eth.c

Description

The R_Ether_Open function initializes the EtherC and E-DMAC subsystems. E-DMAC descriptors and buffers are setup for the initial use. The MAC address is used to initialize MAC address registers in EtherC.

By default, physical device is configured to auto negotiate mode.

If there is only one Ethernet channel, then channel number is set to zero. If there are two Ethernet channels, then 0 and 1 are used for channel number. If MAC address is 0, it should be acquired from the system depending on user implementation. (e.g.: EEPROM)

R_Ether_Close

The R_Ether_Close function disables transmit and receive functionality of EtherC peripheral. This function does not power down EtherC and E-DMAC peripherals.

Format

```
int32_t  R_Ether_Close(uint32_t ch);
```

Parameters

ch

Specifies the EtherC channel number.

Return Values

R_ETHER_OK(0)
R_ETHER_ERROR(-1)

Properties

Prototyped in file eth.h
Implemented in file eth.c

Description

The R_Ether_Close function disables transmit and receive functionality of the EtherC peripheral

If there is only one Ethernet channel, then channel number is set to zero. If there are two Ethernet channels, then 0 and 1 are used for channel number.

R_Ether_Read

The R_Ether_Read function receives data into application receive buffer.

Format

```
int32_t  R_Ether_Read(uint32_t ch, void *buf);
```

Parameters

ch

Specifies the EtherC channel number.

**buf*

Pointer to receive data buffer.

Return Values

Returns the number of bytes received. A zero value indicates no data is received.
R_ETHER_ERROR(-1)

Properties

Prototyped in file eth.h
Implemented in file eth.c

Description

The R-Ether_read function reads data from a buffer pointed by the receive E-DMAC descriptor. It updates the status of the receive descriptor as new data is processed. Read data is copied into the receive data buffer.

If there is only one Ethernet channel, then channel number is set to zero. If there are two Ethernet channels, then use 0 and 1 are used for channel number.

R_Ether_Write

The R_Ether_Write function transmits data from application transmit buffer.

Format

```
int32_t  R_Ether_Write(uint32_t ch, void *buf, uint32_t len);
```

Parameters

ch

Specifies the EtherC channel number.

**buf*

Pointer to Ethernet data to be sent

len

Ethernet frame length.

Return Values

R_ETHER_OK(0)
R_ETHER_ERROR(-1)

Properties

Prototyped in file eth.h
Implemented in file eth.c

Description

The R_Ether_Write function writes transmit data to a buffer pointed by the transmit E-DMAC descriptor. It updates the status of the transmit descriptor as new data is processed. .Data written is transmitted by EtherC.

If there is only one Ethernet channel, then channel number is set to zero. If there are two Ethernet channels, then 0 and 1 are used for channel number.

R_Ether_Write_Sync

The Ether_Write_Sync function writes transmit data to transmit buffer and waits until the data is transmission is complete.

Format

```
int32_t  R_Ether_Write_Sync(uint32_t ch, void *buf, uint32_t len);
```

Parameters

ch

Specifies the EtherC channel number.

**buf*

Pointer to Ethernet data to be sent

len

Ethernet frame length.

Return Values

R_ETHER_OK(0)
R_ETHER_ERROR(-1)

Properties

Prototyped in file eth.h
Implemented in file eth.c

Description

The R_Ether_Write function writes transmit data to a buffer pointed by the transmit E-DMAC descriptor. It updates the status of the transmit descriptor as new data is processed and waits until the transmission is complete.

If there is only one Ethernet channel, then channel number is set to zero. If there are two Ethernet channels, then 0 and 1 are used for channel number.

This API is not used and therefore not implemented in uIP TCP/IP stack demonstration project.

References

1. Group Hardware Manuals for the Renesas device on the RSK board.
2. The uIP Embedded TCP/IP Stack, The uIP 1.0 Reference Manual, June 2006, Adam Dunkels, Swedish Institute of Computer Science
3. IEEE 802.3 Ethernet, IEEE Standards Association, http://standards.ieee.org/getieee802/802.3.html
4. HTTP – Hypertext Transfer Protocol. World Wide Web Consortium, http://www.w3.org/Protocols/
5. RFC 2131 Dynamic Host Configuration Protocol, IETF, http://www.ietf.org/rfc/rfc2131.txt
6. RFC 2132 DHCP Options and BOOTP Vendor Extensions, IETF, http://www.ietf.org/rfc/rfc2132.txt

Website and Support

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

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

All trademarks and registered trademarks are the property of their respective owners.

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Aug.23.10	—	First edition issued
1.01	July 8, 2011	6,8	Fixed URL reference links

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 manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

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

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. 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.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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.
5. When exporting the 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. You should not use Renesas Electronics products or the 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. 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.
6. 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.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. 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 categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. 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 an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
   "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.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. 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.
9. 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 system manufactured by you.
10. 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.
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.

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.
2880 Scott Boulevard Santa Clara, CA 95050-2554, 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.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

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: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan, R.O.C.
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

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.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

© 2010 Renesas Electronics Corporation. All rights reserved.

Colophon 1.0