Renesas Synergy™ Platform

NetX™, Netx Duo™ DHCP Server Module Guide

Introduction

This module guide will enable you to effectively use a module in your own design. Upon completion of this guide you will be able to add this module to your own design, configure it correctly for the target application, and write code using the included application project code as a reference and efficient starting point. References to more detailed API descriptions and suggestions of other application projects that illustrate more advanced uses of the module are available in the Renesas Synergy™ Knowledge Base (as described in the Reference section at the end of this document) and will be valuable resources for creating more complex designs.

The Dynamic Host Configuration Protocol (DHCP) is designed to completely automate both DHCP Server allocation and dynamic IP address allocation by leasing an IP address to a client for a specified time.

In IPv6 networks, DHCP protocol serves no use because it is limited to IPv4. Therefore, DHCPv6 is the protocol to use for dynamic global IPv6 address assignment from a DHCPv6 Server. **This guide covers only the IPv4 version of DHCP, but applies to NetX™ and NetX™ Duo**. Any differences in use between NetX and NetX Duo will be clearly identified by user notes. To simplify wording in this document, NetX DHCPv4 will be used to stand for NetX and NetX Duo DHCP for IPv4.

An overview of the key elements related to the NetX DHCPv4 Server module implementation on the Renesas Synergy™ Platform are provided, with adding and configuring the NetX DHCPv4 Server module onto a Renesas Synergy™ Platform project as the primary focus. For other details on the operation of this module, see the *NetX ™ Dynamic Host Configuration Protocol (DHCP) Server User Guide* for the Renesas Synergy Platform document. This user guide is included in the X-Ware™ and NetX™ Component Documents for Renesas Synergy™ zip file. The zip file is available as a download from the Synergy Software Package site (www.renesas.com/synergy/ssp) on the Renesas Synergy Gallery.

Contents

1. NetX DHCP Server Module Features

The NetX DHCP is compliant with RFC2132, RFC2131, and related RFCs.

- Provides high-level APIs to:
  - Create and delete a DHCPv4 Server instance
  - Set network parameters for DHCPv4 Server messages to the client
  - Create a pool of assignable IP addresses
  - Start and stop the DHCP Server task thread

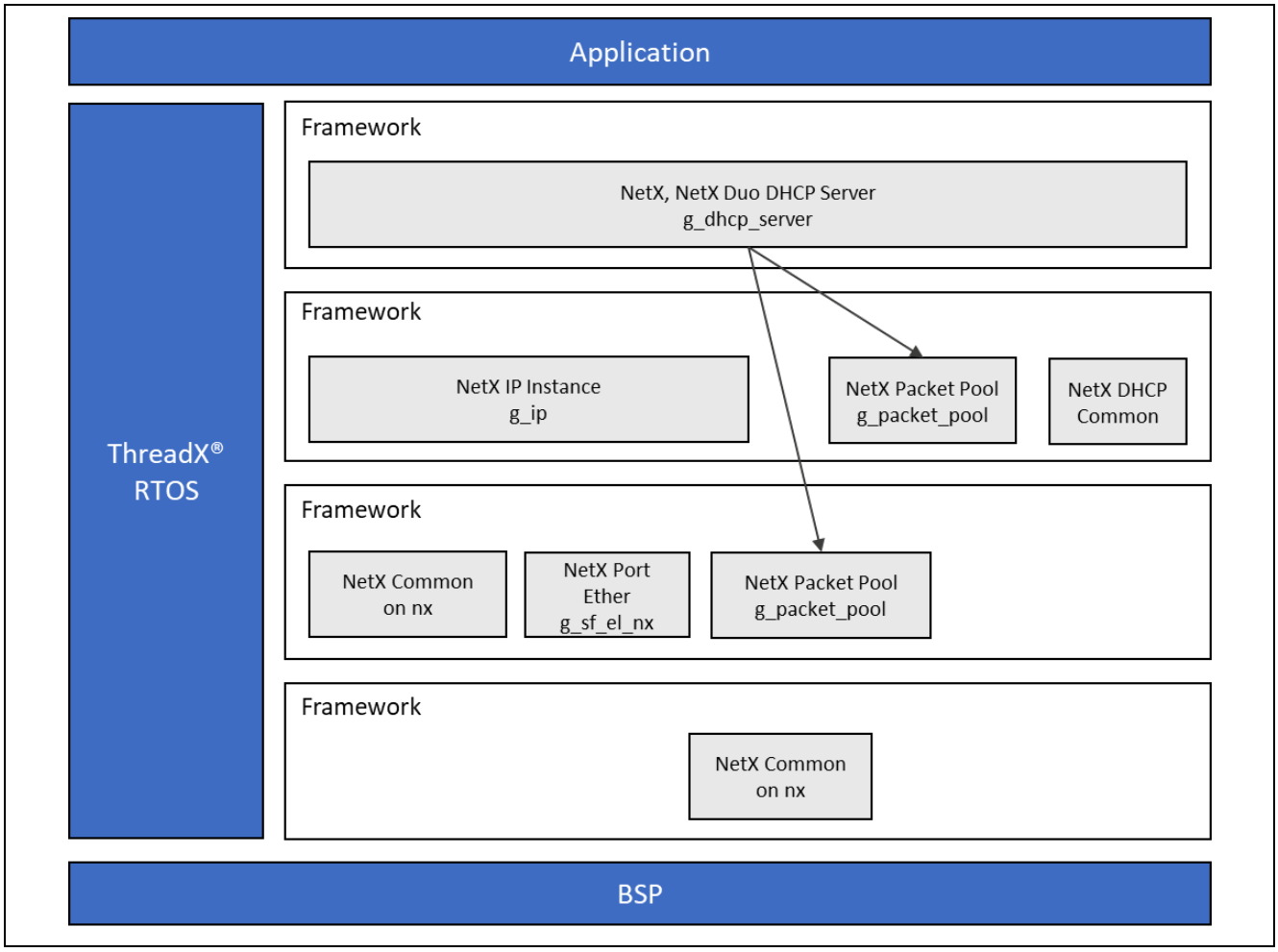

Figure 1. NetX DHCP Server Module Block Diagram

2. NetX DHCP Server Module APIs Overview

The NetX DHCPv4 Server defines APIs to create, delete, remove, start, and stop the server, as well as create the pool of assignable IP addresses and set up the network information for the client. A complete list of the available APIs, an example API call, and a short description of each API can be found in the following table. A table of status return values follows the API summary table.

Table 1. NetX DHCP Server Module API Summary

Function Name	Example API Call and Description
`nx_dhcp_server_create`	`nx_dhcp_server_create(&dhcp_server, &server_ip, pointer, DEMO_SERVER_STACK_SIZE, SERVER_IP_ADDRESS_LIST, "DHCP server", &server_pool);` Create a DHCP Server instance.

Function Name	Example API Call and Description
nx_dhcp_create_server_ip_address_list	nx_dhcp_create_server_ip_list (&dhcp_server, iface_index, START_IP_ADDRESS_LIST, END_IP_ADDRESS_LIST, &addresses_added); Create pool of available IP addresses to assign to DHCP Clients on the specified network index.
nx_dhcp_clear_client_record	nx_dhcp_clear_client_record (&dhcp_server, &dhcp_client_ptr); Remove Client record in the Server database.
nx_dhcp_set_interface_network_parameters	nx_dhcp_set_interface_network_parameters(&dhcp_server, iface_index, NX_DHCP_SUBNET_MASK, NX_DHCP_DEFAULT_GATEWAY, NX_DHCP_DNS_SERVER); Set DHCP options for adding critical network parameters on specified interface in messages to Clients.
nx_dhcp_server_delete	nx_dhcp_server_delete(&dhcp_server); Delete a DHCP Server instance.
nx_dhcp_server_start	nx_dhcp_server_start(&dhcp_server); Start or resume DHCP Server processing.
nx_dhcp_server_stop	nx_dhcp_server_stop(&dhcp_server); Stop DHCP server processing.

Note: For details on operation and definitions for the function data structures, typedefs, defines, API data, API structures and function variables, review the associated *Express Logic User's Manual* accessible as described in the Reference section later in this document.

Table 2. Status Return Values

Name	Description
NX_SUCCESS	Successful DHCP call
NX_PTR_ERROR*	Invalid pointer input
NX_DHCP_PARAMETER_ERROR	Invalid non-pointer input
NX_DHCP_INADEQUATE_PACKET_POOL_PAYLOAD	Packet payload too small error
NX_DHCP_NO_SERVER_OPTION_LIST	Missing option list; cannot create Server
NX_DHCP_SERVER_BAD_INTERFACE_INDEX	Index does not match addresses
NX_DHCP_INVALID_IP_ADDRESS	Invalid IP address or network interface for creating Server address list
NX_DHCP_INVALID_IP_ADDRESS_LIST	Illogical start/end IP addresses for Server list
NX_DHCP_INVALID_NETWORK_PARAMETERS	Invalid network parameters for DHCP messages to Client
NX_DHCP_SERVER_ALREADY_STARTED	The DHCP instance has already been started
NX_DHCP_SERVER_NOT_STARTED	DHCP Server not started
NX_CALLER_ERROR*	Invalid caller of service

Note: Lower-level drivers may return common error codes. Refer to the *SSP User's Manual* API References for the associated module for a definition of all relevant status return values.

*-These error codes are returned only when error-checking is enabled. For other details on error-checking services, see *NetX™ User Guide* for the Renesas Synergy Platform.

3. NetX DHCP Server Module Operational Overview

The DHCPv4 server utilizes the UDP protocol to receive DHCP Client requests and transmit responses. It handles all the details in creating an IP instance, such as initializing the driver, creating the UDP socket, and binding to DHCP port 67 to receive client requests.

When it is created the DHCPv4 Server is assigned a packet pool, it can share the packet pool used by the IP instance (the IP default packet pool), or the module can create a separate one for the server. The packet payload must be large enough to include DHCP data, IP and UDP headers, and the physical frame header. DHCP data size is set by the **Size of the BOOT Buffer (bytes)** property, which defaults to 548 bytes.

Before starting the DHCPv4 Server, the application must create a pool of assignable IP addresses; it does so by calling the `nx_dhcp_create_server_ip_address_list` service. This service takes as input a starting IP address and an ending IP address. The server verifies the addresses are local network addresses*. It fills a table of IP addresses sequentially beginning with the starting IP address. The `addresses_added` pointer input returns the number of addresses added, which is equal to or less than the size of this table. The IP address table size is defined by the **Maximum size of an IP addresses list** property, which defaults to 20. There is one such table for each network interface on which the DHCP Server is receiving DHCP Client requests.

The DHCPv4 Server keeps a record of each client (or rather the client's DISCOVER request) in its client record table. The record lives for as long as the client keeps the assigned IP address. If the client fails to renew or fails to respond to the DHCP protocol before reaching the bound (IP address assigned) state, the record is deleted. One table holds all client records from all network interfaces on which the server receives DHCP requests. The size of the table is set by the **Size of client record table (units)** property, which defaults to 50.

Once the DHCPv4 Server is running and has created client records and assigned IP addresses, it periodically checks the time remaining on each of the client IP leases. The length of the IP lease is set in the **Client IP address lease time (seconds)** property. The default value of 0xFFFFFFFF is essentially a permanent lease. To assign leases of finite length, set the lease to a standard time. An example of a lease time might be 10 days (0x0d5930 or 874,800 seconds). The interval on which the DHCPv4 Server checks the time remaining on assigned IP leases is set to 1000 seconds. If a lease expires, the server simply removes the client record from the client record table, and returns that IP address back to the pool of assignable IP addresses. No message is sent to the client. The client should have initiated renew or rebind requests before its lease expired; or possibly when the client has left the network.

The DHCPv4 Server also keeps an inactivity timeout on each client session. When a client sends a packet, the inactivity timeout for that client is reset. The interval on which the DHCPv4 Server checks the time remaining is the **Fast-periodic timer interval to check valid sessions (ticks),** which defaults to 10 ticks. This session timeout is this interval value multiplied by the ratio of ticks per second to produce a session timeout of 10 seconds. If a client record session time out expires, that client's IP address is returned to the pool of assignable IP addresses and the client record is cleared. No message is sent to the client.

3.1 NetX DHCP Server Module Operational Notes and Limitations

3.1.1 NetX DHCP Server Module Operational Notes

The DHCPv4 Server defines in the **Server option list** property options to the client for critical network parameters. Its default value is set to **1 3 6**, the option codes for the Subnet Mask, Router/Gateway address and DNS Server IP address, respectively. The **Server option list size** property defines the number of options to set, which defaults to **3**.

DHCP Type (Option 53) and DHCP Server Identifier (Option 54) are the DHCP parameters the server must supply to the DHCP Client.

* The DHCPv4 Server services are interface-specific, including creating the IP address list and setting network parameters. The assumed network interface the DHCPv4 Server is running on is the primary interface (index is zero).

3.1.2 NetX DHCP Server Module Limitations

- The choice of options the DHCPv4 Server provides are limited to some or all the following:
  - Subnet Mask (Option 1)
  - Router/Gateway address (Option 3)
  - DNS Server IP address (Option 6).

  Therefore, setting the **Server option list size** to greater than three has no effect. Setting the list of options to an option other than 1,3, or 6 has no effect.
- The NetX DHCPv4 Server does not verify that its assignable IP addresses are not in use elsewhere in the network. It is expected that the client will check the uniqueness of the IP address assigned.
- The NetX DHCPv4 Server does not support the `FORCE RENEW` message.
- The **Relay agent** field of the DHCP header is left null because the NetX DHCPv4 Server does not support out-of-network DHCP requests.
- The DHCP Server does not correctly update the time remaining on the assigned IP it leased. The slow periodic timer interval is set to 1000 ticks. Internally, this value is converted to seconds, so the actual interval on which the server checks the IP lease timeout is about 1000 * 100, assuming there are 100 ticks per second on the NetX device. If the client lease time is left at the default value of 0xFFFFFFFF, it's a permanent lease until the client decides to release it and it should not be affected by this bug.

For any additional operational limitations for this module, see the latest *SSP Release Notes*.

4. Including the NetX DHCP Server Module in an Application

This section describes how to include the NetX DHCPv4 Server module in an application using the SSP configurator.

Note: It's assumed you are familiar with creating a project, adding threads, adding a stack to a thread, and configuring a block within the stack. If you are unfamiliar with any of these items, refer to the first few chapters of the *SSP User's Manual* to learn how to manage each of these important steps in creating SSP-based applications.

To add the NetX DHCP Server module to an application, simply add it to a thread using the stacks selection sequence given in the following table. (The default name for the NetX DHCP Server module is `g_dhcp_server0`. This name can be changed in the associated **Properties** window.)

Table 3. NetX DHCP Server Module Selection Sequence

Resource	ISDE Tab	Stacks Selection Sequence
g_dhcp_server0 NetX DHCPv4 Server	Threads	New Stack> X-Ware> NetX> Protocols> NetX DHCP Server
g_dhcp_server0 NetX Duo DHCPv4 Server	Threads	New Stack> X-Ware> NetX Duo> Protocols> NetX Duo DHCP IPv4 Server

When the NetX DHCPv4 Server is added to the thread stack as shown in the following figure, the configurator automatically adds any needed lower-level modules. Any modules that need additional configuration information are box text highlighted in Red. Modules with a Gray band are individual modules that stand alone. Modules with a Blue band are shared or common. They need only be added once and can be used by multiple stacks. Modules with a Pink band can require the selection of lower-level modules; these are either optional or recommended. (This is indicated in the block with the inclusion of this text.) If adding lower-level modules is required, the module description has **Add** in the text. Clicking on any Pink banded modules brings up the **New** icon and displays possible choices.

The DHCPv4 Server has its own packet pool, `g_packet_pool1`; it can alternatively share the packet pool (`g_packet_pool0`) used by the IP instance.

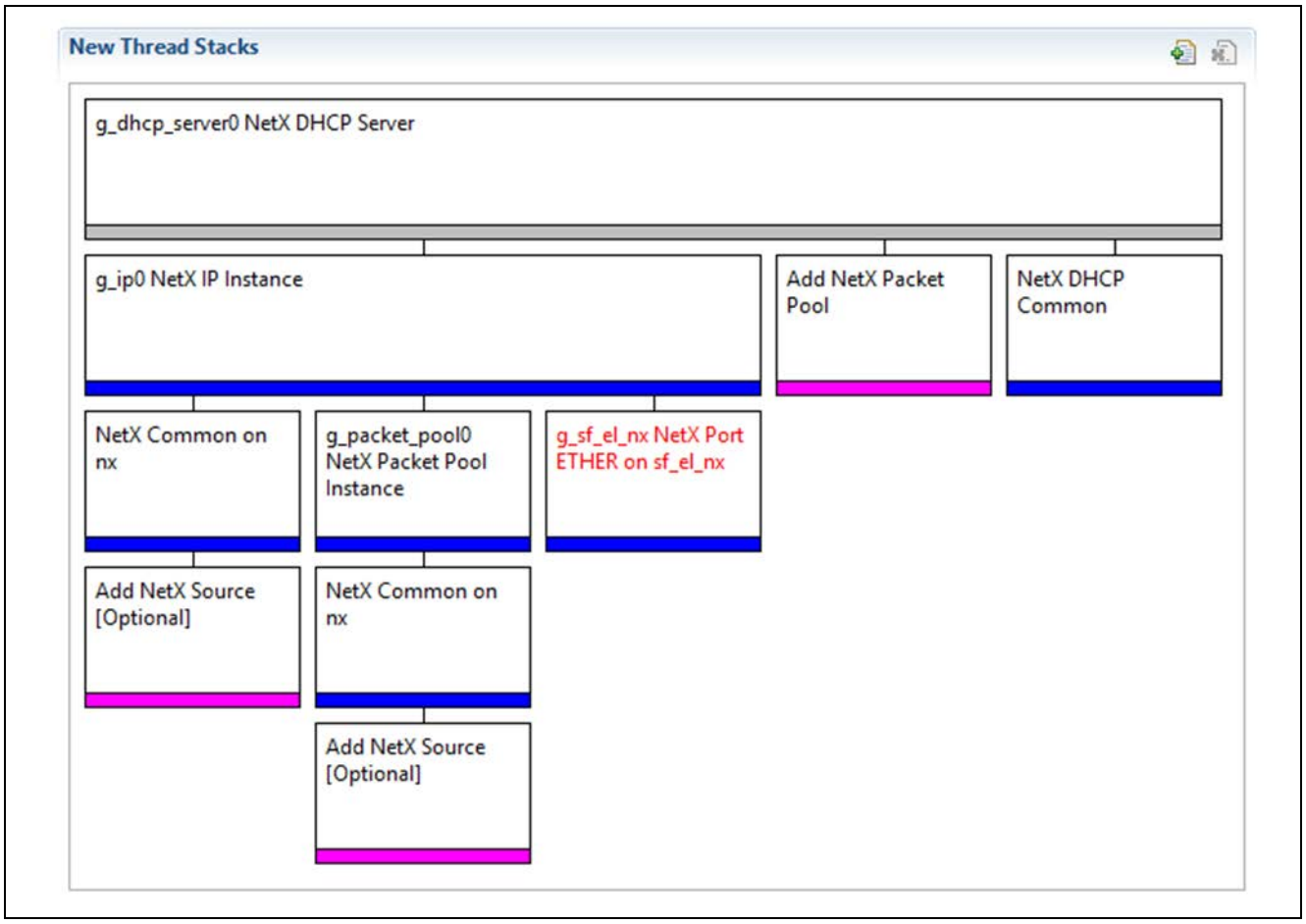

Figure 2. NetX DHCP Server Module Stack

5. Configuring the NetX DHCP Server Module

The NetX DHCP Server module must be configured by the user for the desired operation. The SSP configuration window automatically identifies (by highlighting the block in red) any required configuration selections, such as interrupts or operating modes, necessary for lower-level modules to enable successful operation. Only properties that can be changed without causing conflicts are available for modification. Other properties are **locked** with a lock icon in the **Properties** window in the ISDE and unavailable for change. This approach simplifies the configuration process and makes it much less error-prone than previous manual approaches to configuration. The available configuration settings and defaults for user-accessible properties are given in the properties tab within the SSP configurator and are shown in the following tables for easy reference.

One of the properties most often identified as requiring a change is the interrupt priority, and this configuration setting is available within the **Properties** window of the associated module. Simply select the indicated module and then view it in the **Properties** window. Interrupt settings are often toward the bottom of the properties list, so scroll down until they become available. Note that the interrupt priorities listed in the **Properties** window in the ISDE indicate the validity of the setting based on the targeted MCU (CM4 or CM0+). This level of detail is not included in the following configuration properties tables, but is easily visible within the ISDE when configuring interrupt-priority levels.

Note: You may want to open your ISDE, create the module, and explore the property settings in parallel while looking over the following configuration table values. This helps to orient you and can be a useful hands-on approach to learning the ins and outs of developing with SSP.

Table 4. Configuration Settings for the NetX DHCPv4 Server Module

Parameter	Value	Description
Internal thread priority	1	Internal thread priority selection
Packet allocate timeout (seconds)	2	Packet allocate timeout selection
Server IPv4 address (use commas for separation)	192,168,0,1	Server IPv4 address selection
Subnet Mask for clients (use commas for separation)	255,255,255,0	Subnet Mask for clients' selection
Maximum client identifier length (bytes)	8	Maximum client identifier length selection. Note: This is no longer in use (deprecated).
Fast periodic timer interval to check valid sessions (ticks)	10	Fast periodic timer interval to check valid sessions selection
Next Client message wait (ticks)	10	Next Client message wait selection
Client IP address lease time (seconds)	0xFFFFFFFF	Client IP address lease time selection
Size of array for holding available IP addresses	50	Size of array for holding available IP address selection
Size of the array containing current requested options (units)	12	Size of the array containing current requested options selection
Server option list (optional - use space for separation	1 3 6	Module server option list (optional - use space for separation
Server option list size (optional)	3	Server option list size selection
Server option list (required - use space for separation)	53 54	Server option list selection
Size of the current client hostname buffer (byte)	30	Size of the current client hostname buffer selection
Maximum size of an IP addresses list (units)	20	Maximum size of an IP addresses list (unit selection
Size of the client record table (units)	50	Size of the client record table selection
Size of the BOOT buffer (bytes)	548	Size of the BOOT buffer selection
Size of the DHCP header (bytes)	236	Size of the DHCP header selection
Name	g_dhcp_server0	Module name
Internal thread stack size (bytes)	4096	Internal thread stack size selection

Note: The example settings and defaults are for a project using the Synergy S7G2 MCU Group. Other MCUs may have different default values and available configuration settings.

In some cases, settings other than the defaults for lower-level modules can be desirable. For example, it might be useful to select different ethernet interface pins and resets. The configurable properties for the lower-level stack modules are given in the following sections for completeness and as a reference.

Note: Most of the property settings for lower-level modules are intuitive and usually can be determined by inspection of the associated properties window from the SSP configurator.

5.1 Configuration Settings for the NetX DHCP Client Lower-Level Modules

Only a small number of settings must be modified from the default for lower-level modules, as indicated via the red text in the thread stack block. Notice that some of the configuration properties must be set to a certain value for proper framework operation and are locked to prevent user modification. The following table identifies all the settings within the **Properties** section for the module.

Table 5. Configuration Settings for the NetX IP Instance

ISDE Property	Value	Description
Name	g_ip0	Module name
IPv4 Address (use commas for separation)	192,168,0,2	IPv4 Address selection
Subnet Mask (use commas for separation)	255,255,255,0	Subnet Mask selection
**IPv6 Global Address (use commas for separation)	0x2001, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x1	IPv6 global address selection
**IPv6 Link Local Address (use commas for separation; All zeros means use the MAC address)	0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0	IPv6 link local address selection
IP Helper Thread Stack Size (bytes)	2048	IP Helper Thread Stack Size (bytes) selection
IP Helper Thread Priority	3	IP Helper Thread Priority selection
ARP	Enable	ARP selection
ARP Cache Size in Bytes	520	ARP Cache Size in Bytes selection
Reverse ARP	Disable	Reverse ARP selection
TCP	Enable, Disable Default: Enable	TCP selection
UDP	Enable	UDP selection
ICMP	Enable, Disable Default: Enable	ICMP selection
IGMP	Enable, Disable Default: Enable	IGMP selection
IP fragmentation	Enable, Disable Default: Disable	IP fragmentation selection

Note: The example settings and defaults are for a project using the Synergy S7G2 MCU Group. Other MCUs may have different default values and available configuration settings.

Table 6. Configuration Settings for the NetX Duo DHCP Common Instance

ISDE Property	Value	Description
Type of Service for UDP requests	Normal, Minimum delay, Maximum data, Maximum reliability, Minimum cost Default: Normal	Type of service UDP requests selection
Fragmentation option	Don't fragment, Fragment okay Default: Don't fragment	Fragment option selection
Time to live	128	Time to live selection
Packet Queue depth	5	Packet queue depth selection

Note: The example settings and defaults are for a project using the Synergy S7G2 MCU Group. Other MCUs may have different default values and available configuration settings.

Table 7. Configuration Settings for the NetX Packet Pool Instance

ISDE Property	Value	Description
Name	g_packet_pool0	Module name
Packet Size in Bytes	640	Packet size selection
Number of Packets in Pool	16	Number of packets in pool selection

Note: The example settings and defaults are for a project using the Synergy S7G2 MCU Group. Other MCUs may have different default values and available configuration settings.

Table 8. Configuration Settings for the NetX Port ETHER

ISDE Property	Value	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enable or disable the parameter checking
Channel 0 Phy Reset Pin	IOPORT_PORT_09_PIN_03	Channel 0 Phy reset pin selection
Channel 0 MAC Address High Bits	0x00002E09	Channel 0 MAC address high bits selection
Channel 0 MAC Address Low Bits	0x0A0076C7	Channel 0 MAC address low bits selection
Channel 1 Phy Reset Pin	IOPORT_PORT_07_PIN_06	Channel 1 Phy reset pin selection
Channel 1 MAC Address High Bits	0x00002E09	Channel 1 MAC address high bits selection
Channel 1 MAC Address Low Bits	0x0A0076C8	Channel 1 MAC address low bits selection
Number of Receive Buffer Descriptors	8	Number of receive buffer descriptors selection
Number of Transmit Buffer Descriptors	32	Number of transmit buffer descriptors selection
Ethernet Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX®), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	Ethernet interrupt priority selection
Name	g_sf_el_nx	Module name
Channel	0	Channel selection
Callback	NULL	Callback selection

Note: The example settings and defaults are for a project using the Synergy S7G2 MCU Group. Other MCUs may have different default values and available configuration settings.

Table 9. Configuration Settings for the NetX Packet Pool Instance (when added as new)

ISDE Property	Value	Description
Name	g_packet_pool1	Module name
Packet Size in Bytes	640	Packet size selection
Number of Packets in Pool	16	Number of packets in pool selection

Note: The example settings and defaults are for a project using the Synergy S7G2 MCU Group. Other MCUs may have different default values and available configuration settings.

Table 10. Configuration Settings for the NetX Common Instance

ISDE Property	Value	Description
No configurable settings		

Note: The example settings and defaults are for a project using the Synergy S7G2 MCU Group. Other MCUs may have different default values and available configuration settings.

5.2 NetX DHCP Server Module Clock Configuration

The ETHERC peripheral module uses PCLKA as its clock source. The PCLKA frequency is set by using the SSP configurator clock tab prior to a build, or by using the CGC Interface at run-time.

5.3 NetX DHCP Server Module Pin Configuration

The ETHERC peripheral module uses pins on the MCU to communicate to external devices. I/O pins must be selected and configured as required by the external device. The following table indicates the method to select pins within the SSP configuration window. The subsequent table has an example selection for I^2C pins.

Note: The operation mode selected determines the peripheral signals available and which MCU pins are required.

Table 11. Pin Selection for the ETHERC Module

Resource	ISDE Tab	Pin selection Sequence
ETHERC	Pins	Select Peripherals > Connectivity:ETHERC > ETHERC1.RMII

Note: The selection sequence assumes ETHERC1 is the desired hardware target for the driver.

Table 12. Pin Configuration Settings for the ETHERC1

Property	Value	Description
Operation Mode	Disabled, Custom, RMII (Default: Disabled)	Select RMII as the Operation Mode for ETHERC1
Pin Group Selection	Mixed, _A only (Default: _A only)	Pin group selection
REF50CK	P701	REF50CK Pin
TXD0	P700	TXD0 Pin
TXD1	P406	TXD1 Pin
TXD_EN	P405	TXD_EN Pin
RXD0	P702	RXD0 Pin
RXD1	P703	RXD1 Pin
RX_ER	P704	RX_ER Pin
CRS_DV	P705	CRS_DV Pin
MDC	P403	MDC Pin
MDIO	P404	MDIO Pin

Note: The example settings are for a project using the Synergy S7G2 MCU Group and the SK-S7G2 Kit. Other Synergy MCUs and Synergy Kits may have different available pin configuration settings.

6. Using the NetX DHCP Server Module in an Application

The following example assumes the NetX system is already initialized. The typical steps to using the NetX DHCPv4 server in an application are:

1. Create a pool of assignable IP addresses using the `nx_dhcp_create_server_ip_address_list` API.
2. Set network parameters that will be returned by the server using the `nx_dhcp_set_interface_network_parameters` API.

3. Start the DHCPv4 server with the `nx_dhcp_server_start` API.

The server is now running.

The following diagram shows common steps in a typical operational flow:

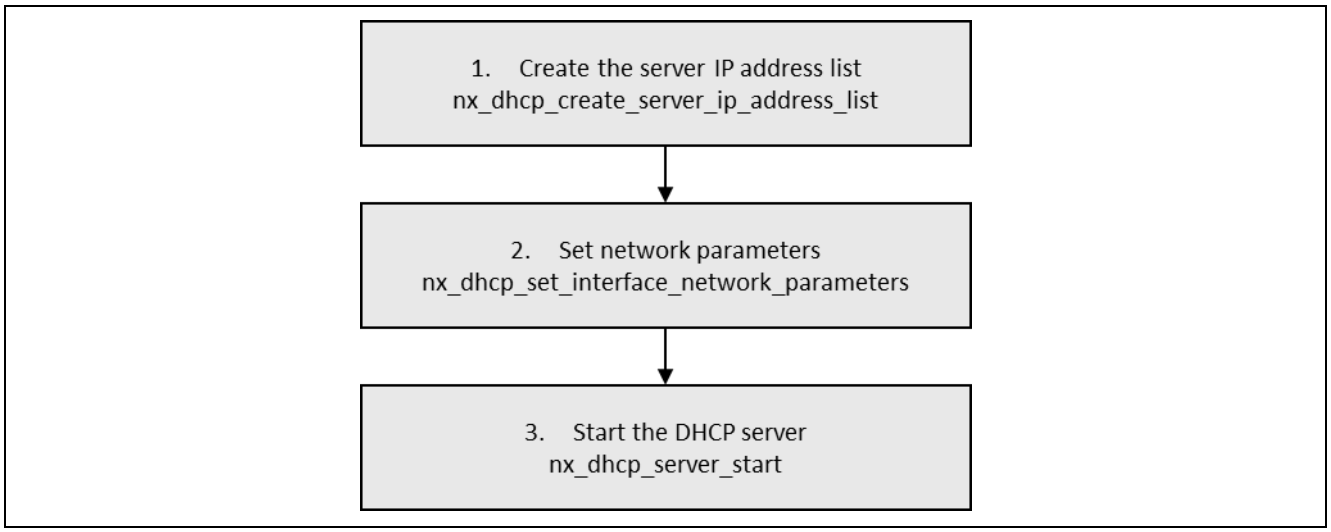

Figure 3. Flow Diagram of a Typical NetX DHCP Server Module Application

7. The NetX DHCP Server Module Application Project

The application project associated with this module guide demonstrates the steps in a full design. The project can be found using the link provided in the References section at the end of this document. You may want to import and open the application project within the ISDE and view the configuration settings for the NetX DHCPv4 Server module. You can also read over the code (in `dhcp_thread_entry.c`) that depicts the NetX DHCPv4 Server module APIs in a complete design.

The application project demonstrates the typical use of the NetX DHCPv4 Server module APIs. The application project main thread entry configures the list of IP addresses and network parameters for the NetX DHCPv4 Server module. After configuration, the DHCP Server is started and ready to handle requests. The following table identifies the target versions for the associated software and hardware used by the application project:

Table 13. Software and Hardware Resources Used by the Application Project

Resource	Revision	Description
e² studio	5.4.0.023 or later	Integrated Solution Development Environment
SSP	1.3.0 or later	Synergy Software Platform
IAR EW for Synergy	7.71.2 or later	IAR Embedded Workbench® for Renesas Synergy™
SSC	5.4.0.023 or later	Synergy Standalone Configurator
SK-S7G2	v3.0 to v3.1 or later	Starter Kit

The following diagram shows a simple flow of the application project.

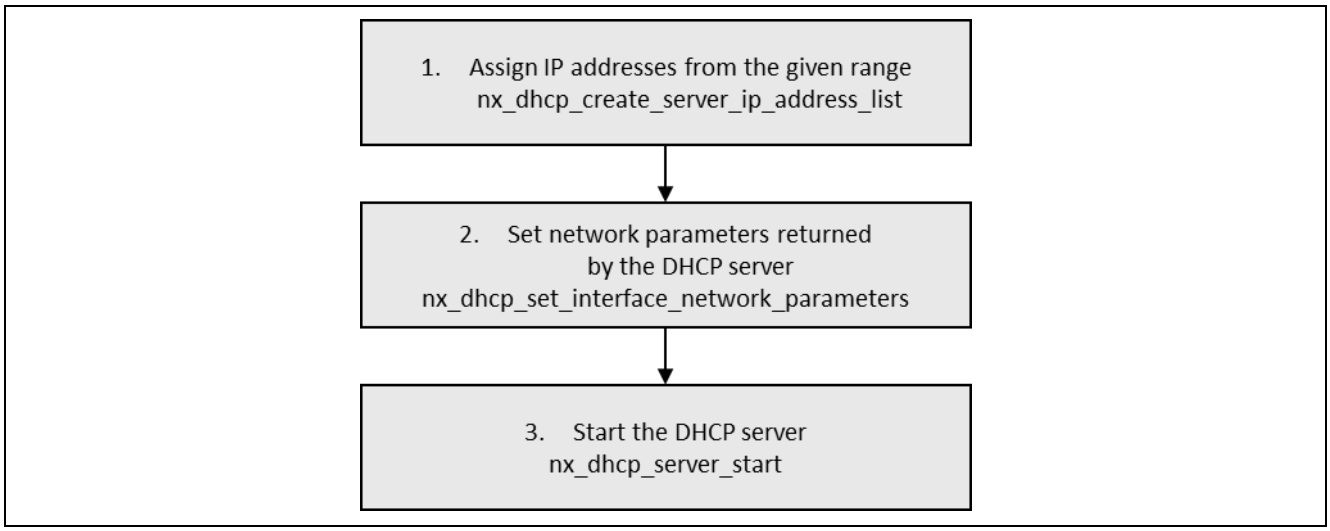

Figure 4. NetX DHCP Server Module Application Project Flow Diagram

The `dhcp_thread_entry.c` file is located in the project once it has been imported into the ISDE. You can open this file within the ISDE and follow along with the description provided to help identify key uses of APIs.

The first section of `dhcp_thread_entry.c` has the header files that reference the NetX DHCPv4 Server instance; the first and the last IP address that can be assigned by the DHCP Server, and the NetX APIs. The next section is an interface index definition as a macro constant. The last section is the entry function for the main program-control section. The NetX DHCPv4 Server is configured using the `nx_dhcp_create_server_ip_address_list` API and the `nx_dhcp_set_interface_network_parameters` API. The first one creates an IP address list using the first and the last IP address, the second sets network parameters like the subnet mask, gateway IP address, and DNS IP address. These parameters get sent by the server in response to client requests. Inside the infinite while loop a thread sleep function pauses execution for 10 ThreadX-timer ticks. Upon the successful start of the DHCP server, the onboard LED1 glows. Upon lease out of the new IP, or at renewal after power up, the onboard LED2 glows.

A few key properties are configured in this application project to support the required operations and the physical properties of the target board and MCU. The properties with the values set for this specific project are listed in the following tables. You can also open the application project and view these settings in the **Properties** window as a hands on exercise.

Table 14. NetX DHCPv4 Server Module Configuration Settings for the Application Project

ISDE Property	Value Set
Internal thread priority	1
Packet allocate timeout (seconds)	2
Server IPv4 address (use commas for separation)	192,168,0,1
Subnet Mask for clients (use commas for separation)	255,255,255,0
Subnet router IPv4 address (use for commas separation)	192,168,0,1
Subnet DNS IPv4 address (use commas for separation)	192,168,0,1
Maximum client identifier length (bytes)	8 (Note: This is no longer in use (deprecated).
Fast periodic timer interval to check valid sessions (ticks)	10
Next Client message wait (ticks)	10
Client IP address lease time (seconds)	0xFFFFFFFF
Size of array for holding available IP addresses Size of client record table	50 This is incorrect for SSP 1.2.0; it is the size of client records table (e.g., stores IP addresses assigned to clients and client data). In SSP

	1.3.0 and later, it is correctly defined as size of client record table (default still 50).
ISDE Property	**Value Set**
Size of the array containing current requested options (units)	12
Server option list (optional - use space for separation	1 3 6
Server option list size (optional)	3
Server option list (required - use space for separation)	53 54
Size of the current client hostname buffer (byte)	30
Maximum size of an IP addresses list (units)	20
Size of the client record table (units)	50
Size of the BOOT buffer (bytes)	548
Size of the DHCP header (bytes)	236
Name	`g_dhcp_server0`
Internal thread stack size (bytes)	2048

Note: Entries in blue are no longer available after SSP 1.2.0.
Entries in gray are not defined incorrectly in SSP 1.2.0

Table 15. NetX IP Instance Configuration Settings for the Application Project

ISDE Property	**Value Set**
Name	`g_ip0`
IPv4 Address (use commas for separation)	192,168,0,1
Subnet Mask (use commas for separation)	255,255,255,0
IP Helper Thread Stack Size (bytes)	2048
IP Helper Thread Priority	3
ARP	Enable
ARP Cache Size in Bytes	520
Reverse ARP	Disable
TCP	Enable
UDP	Enable
ICMP	Enable
IGMP	Enable
IP fragmentation	Disable

Note: Entries in yellow need to be adjusted to the developer's local network.

Table 16. NetX Packet Pool Instance Configuration Settings for the Application Project

ISDE Property	**Value Set**
Name	`g_packet_pool1`
Packet Size in Bytes	640 (recommend 1568)
Number of Packets in Pool	16

Note: Entries in green have a recommended value different than the default value.

Table 17. NetX Packet Pool Instance Configuration Settings for the Application Project

ISDE Property	Value Set
Name	g_packet_pool0
Packet Size in Bytes	640
Number of Packets in Pool	16

Table 18. NetX DHCP Common Configuration Settings for the Application Project

ISDE Property	Value Set
Type of Service for UDP requests	Normal
Fragmentation option	Don't fragment
Time to live	128
Packet Queue depth	5

Table 19. NetX Port ETHER Configuration Settings for the Application Project

ISDE Property	Value Set
Parameter Checking	BSP, Enabled, Disabled (Default: BSP)
Channel 0 Phy Reset Pin	IOPORT_PORT_09_PIN_03
Channel 0 MAC Address High Bits	0x00002E09
Channel 0 MAC Address Low Bits	0x0A0076C7
Channel 1 Phy Reset Pin	IOPORT_PORT_08_PIN_06
Channel 1 MAC Address High Bits	0x00002E09
Channel 1 MAC Address Low Bits	0x0A0076AB
Number of Receive Buffer Descriptors	8
Number of Transmit Buffer Descriptors	32
Ethernet Interrupt Priority	Priority 2
Name	g_sf_el_nx
Channel	1
Callback	NULL

Note: Entries in green have a recommended value different than the default value.

8. Customizing the NetX DHCP Server Module for a Target Application

Some configuration settings are normally changed by the developer from those shown in the application project. For example, the user can easily change the IP address of the DHCP Server. The IP address should be set using the **Threads** tab and the **Properties** editor of the NetX DHCPv4 Server and NetX IP Instance modules. The user can also change the range of the IP addresses assigned by the server. The macro constant definitions for the start and the end IP addresses are defined in the `dhcp_ip_range.h` file.

9. Running the NetX DHCP Server Module Application Project

To run the NetX DHCPv4 Server Module application project and to see it executed on a target kit, you can simply import it into your ISDE, compile, and run debug.

Note: The following steps are in sufficient detail for someone experienced with the basic flow through the Synergy development process. If these steps are unfamiliar, refer to the first few chapters of the *SSP User's Manual* to learn how to accomplish these steps.

1. Refer to the *Renesas Synergy™ Project Import Guide* (11an0023eu0121-synergy-ssp-import-guide.pdf, included in this package) for instructions on importing the project into e^2 studio ISDE or IAR Embedded Workbench® for Renesas Synergy™ and building/running the application.
2. Connect to the host PC via a micro USB cable to J19 on SK-S7G2 Synergy MCU.
3. Start to debug the application.

4. Connect a DHCP client via Ethernet cable to J11.
5. The application's status can be viewed on the Renesas Debug Virtual Console (image on the right). A user can cross check the settings received from the DHCP server by verifying the network parameters of the connected DHCP client (**ipconfig/all** command for a windows PC gives the network details (left side of image)).

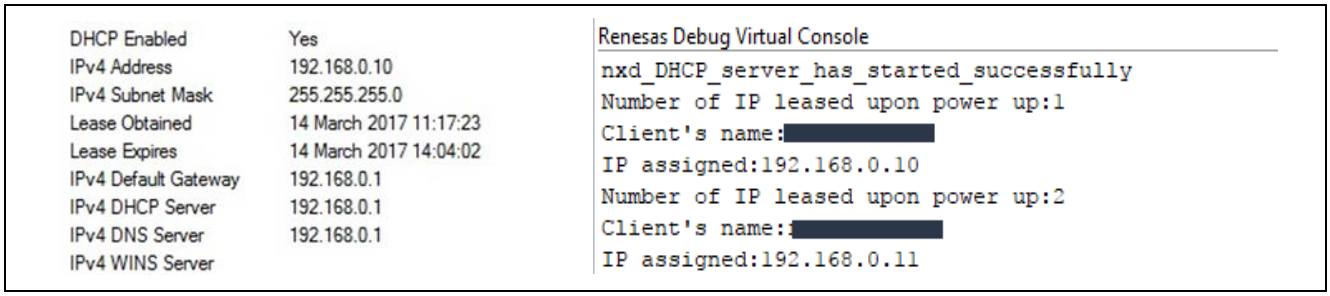

Figure 5. Example Output from NetX DHCP Server Module Application Project

10. NetX DHCP Server Module Conclusion

This module guide has provided all the background information needed to select, add, configure, and use the module in an example project. Many of these steps were time consuming and error-prone activities in previous generations of embedded systems. The Renesas Synergy Platform makes these steps a lot less time consuming and removes the common errors, like conflicting configuration settings or incorrect selection of lower-level drivers. The use of high-level APIs (as demonstrated in the application project) illustrate additional development time savings by allowing work to begin at a high level and avoiding the time required in older development environments to use or, in some cases, create, lower-level drivers.

11. NetX DHCP Server Module Next Steps

After you have mastered a simple NetX DHCPv4 Server project, you may want to review a more complex example. You may find that a different protocol should be handled in your application. NetX and NetX Duo for the Synergy Platform provides many modules, including DHCP, DNS, HTTP, and more. Module guides for these applications can be found using instructions in the References section at the end of this document.

12. Reference Information

The SSP User Manual is available in HTML in the SSP distribution package, and at the Synergy Gallery (www.renesas.com/synergy/software) as a pdf.

Links to all the most up-to-date NetX DHCP Server module reference materials and resources are available on the Synergy Knowledge Base: https://en-support.renesas.com/knowledgeBase/16977455.

Website and Support

Visit the following vanity URLs to learn about key elements of the Synergy Platform, download components and related documentation, and get support.

Synergy Software	www.renesas.com/synergy/software
Synergy Software Package	www.renesas.com/synergy/ssp
Software add-ons	www.renesas.com/synergy/addons
Software glossary	www.renesas.com/synergy/softwareglossary
Development tools	www.renesas.com/synergy/tools
Synergy Hardware	www.renesas.com/synergy/hardware
Microcontrollers	www.renesas.com/synergy/mcus
MCU glossary	www.renesas.com/synergy/mcuglossary
Parametric search	www.renesas.com/synergy/parametric
Kits	www.renesas.com/synergy/kits
Synergy Solutions Gallery	www.renesas.com/synergy/solutionsgallery
Partner projects	www.renesas.com/synergy/partnerprojects
Application projects	www.renesas.com/synergy/applicationprojects
Self-service support resources:	
Documentation	www.renesas.com/synergy/docs
Knowledgebase	www.renesas.com/synergy/knowledgebase
Forums	www.renesas.com/synergy/forum
Training	www.renesas.com/synergy/training
Videos	www.renesas.com/synergy/videos
Chat and web ticket	www.renesas.com/synergy/resourcelibrary

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jun.21.17	–	Initial Release
1.01	Dec.06.17	–	Added NetX Duo and updated revisions for releases
1.02	Jan.08.19	–	Updates for SSP v1.5.0
1.03	May.02.19	–	Updates for SSP v1.6.0

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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. 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 shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended 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; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or 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 damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, 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 and impractical, you are responsible for evaluating 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. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall 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 shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, 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.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.